

Numerical Methods for Partial Differential Equations

Prof. R. Hiptmair, Prof. Ch. Schwab,

Prof. H. Harbrecht, Dr. V. Gradinaru

Dr. A. Chernov, Prof. P. Grohs

Draft version March 22, 2012, SVN rev.

(C) Seminar für Angewandte Mathematik, ETH Zürich

URL: <http://www.sam.math.ethz.ch/~hiptmair/tmp/NPDE12.pdf>

Contents

1 Case Study: A Two-point Boundary Value Problem	21	R. Hiptmair C. Schwab, H. Harbrecht V. Gradinaru A. Chernov P. Grohs
1.1 Introduction	22	
1.2 A model(ing) problem	25	
1.2.1 Linear elastic string	25	
1.2.2 Mass-spring model	32	
1.2.3 Continuum limit	37	
1.3 Variational approach	46	
1.3.1 Virtual work equation	46	
1.3.2 Regularity (smoothness) requirements	56	
1.3.3 Differential equation	59	
1.4 Simplified model	65	0.0
1.5 Discretization	74	p. 2

1.5.1	Ritz-Galerkin discretization	78	Numerical Methods for PDEs
1.5.1.1	Spectral Galerkin scheme	87	
1.5.1.2	Linear finite elements	109	
1.5.2	Collocation	127	
1.5.2.1	Spectral collocation	131	
1.5.2.2	Spline collocation	137	
1.5.3	Finite differences	139	
1.6	Convergence	143	
1.6.1	Norms on function spaces	145	
1.6.2	Algebraic and exponential convergence	153	
2	Second-order Scalar Elliptic Boundary Value Problems	169	R. Hiptmair C. Schwab, H. Harbrecht V. Gradinaru A. Chernov P. Grohs SAM, ETHZ
2.1	Equilibrium models	173	
2.1.1	Taut membrane	174	
2.1.2	Electrostatic fields	181	
2.1.3	Quadratic minimization problems	186	
2.2	Sobolev spaces	197	
2.3	Variational formulations	217	
2.3.1	Linear variational problems	217	
2.3.2	Stability	223	
2.4	Equilibrium models: Boundary value problems	231	
2.5	Diffusion models (Stationary heat conduction)	243	
2.6	Boundary conditions	249	
2.7	Characteristics of elliptic boundary value problems	254	
2.8	Second-order elliptic variational problems	258	0.0
2.9	Essential and natural boundary conditions	265	p. 3

3	Finite Element Methods (FEM)	276	
3.1	Galerkin discretization	279	Numerical Methods for PDEs
3.2	Case study: Triangular linear FEM in two dimensions	289	
3.2.1	Triangulations	291	
3.2.2	Linear finite element space	293	
3.2.3	Nodal basis functions	296	
3.2.4	Sparse Galerkin matrix	301	
3.2.5	Computation of Galerkin matrix	307	
3.2.6	Computation of right hand side vector	318	
3.3	Building blocks of general FEM	323	
3.3.1	Meshes	324	
3.3.2	Polynomials	328	R. Hiptmair C. Schwab, H. Harbrecht V. Gradinaru A. Chernov P. Grohs
3.3.3	Basis functions	332	
3.4	Lagrangian FEM	337	
3.4.1	Simplicial Lagrangian FEM	338	
3.4.2	Tensor-product Lagrangian FEM	345	SAM, ETHZ
3.5	Implementation of FEM	355	
3.5.1	Mesh file format	356	
3.5.2	Mesh data structures [7, Sect. 1.1]	362	
3.5.3	Assembly [7, Sect. 5]	369	
3.5.4	Local computations and quadrature	382	
3.5.5	Incorporation of essential boundary conditions	397	
3.6	Parametric finite elements	406	
3.6.1	Affine equivalence	406	
3.6.2	Example: Quadrilateral Lagrangian finite elements	415	
3.6.3	Transformation techniques	422	
3.6.4	Boundary approximation	428	0.0
3.7	Linearization	430	p. 4

4	Finite Differences (FD) and Finite Volume Methods (FV)	441	Numerical Methods for PDEs
4.1	Finite differences	443	
4.2	Finite volume methods (FVM)	455	
4.2.1	Discrete balance laws	455	
4.2.2	Dual meshes	458	
4.2.3	Relationship of finite elements and finite volume methods	462	
5	Convergence and Accuracy	471	
5.1	Galerkin error estimates	472	
5.2	Empirical (asymptotic) convergence of FEM	481	
5.3	A priori finite element error estimates	497	R. Hiptmair C. Schwab, H. Harbrecht V. Gradinaru A. Chernov P. Grohs
5.3.1	Estimates for linear interpolation in 1D	499	
5.3.2	Error estimates for linear interpolation in 2D	507	
5.3.3	The Sobolev scales	519	
5.3.4	Anisotropic interpolation error estimates	524	
5.3.5	General approximation error estimates	533	SAM, ETHZ
5.4	Elliptic regularity theory	548	
5.5	Variational crimes	559	
5.5.1	Impact of numerical quadrature	561	
5.5.2	Approximation of boundary	564	
5.6	Duality techniques	568	
5.6.1	Linear output functionals	568	
5.6.2	Case study: Boundary flux computation	578	
5.6.3	L^2 -estimates	588	0.0
5.7	Discrete maximum principle	598	p. 5

6	2nd-Order Linear Evolution Problems	614	
6.1	Parabolic initial-boundary value problems	617	Numerical Methods for PDEs
6.1.1	Heat equation	617	
6.1.2	Spatial variational formulation	621	
6.1.3	Method of lines	629	
6.1.4	Timestepping	633	
6.1.4.1	Single step methods	634	
6.1.4.2	Stability	638	
6.1.5	Convergence	661	
6.2	Wave equations	673	R. Hiptmair C. Schwab, H. Harbrecht V. Gradinaru A. Chernov P. Grohs
6.2.1	Vibrating membrane	674	
6.2.2	Wave propagation	682	
6.2.3	Method of lines	689	SAM, ETHZ
6.2.4	Timestepping	692	
6.2.5	CFL-condition	704	
7	Convection-Diffusion Problems	716	
7.1	Heat conduction in a fluid	716	
7.1.1	Modelling fluid flow	718	
7.1.2	Heat convection and diffusion	722	
7.1.3	Incompressible fluids	726	0.0
7.1.4	Transient heat conduction	731	p. 6

7.2	Stationary convection-diffusion problems	732
7.2.1	Singular perturbation	736
7.2.2	Upwinding	742
7.2.2.1	Upwind quadrature	753
7.2.2.2	Streamline diffusion	762
7.3	Transient convection-diffusion BVP	778
7.3.1	Method of lines	779
7.3.2	Transport equation	785
7.3.3	Lagrangian split-step method	789
7.3.3.1	Split-step timestepping	789
7.3.3.2	Particle method for advection	795
7.3.3.3	Particle mesh method	804
7.3.4	Semi-Lagrangian method	818
8	Numerical Methods for Conservation Laws	830
8.1	Conservation laws: Examples	831
8.1.1	Linear advection	833
8.1.2	Traffic modeling [2]	837
8.1.2.1	Particle model	838
8.1.2.2	Continuum traffic model	848
8.1.3	Inviscid gas flow	853
8.2	Scalar conservation laws in 1D	858

8.2.1	Integral and differential form	858	Numerical Methods for PDEs
8.2.2	Characteristics	864	
8.2.3	Weak solutions	874	
8.2.4	Jump conditions	878	
8.2.5	Riemann problem	881	
8.2.6	Entropy condition	897	
8.2.7	Properties of entropy solutions	904	
8.3	Conservative finite volume discretization	908	
8.3.1	Semi-discrete conservation form	911	
8.3.2	Discrete conservation property	916	
8.3.3	Numerical flux functions	920	R. Hiptmair C. Schwab, H. Harbrecht V. Gradinaru A. Chernov P. Grohs
8.3.3.1	Central flux	920	
8.3.3.2	Lax-Friedrichs flux	927	
8.3.3.3	Upwind flux	931	
8.3.3.4	Godunov flux	942	
8.3.4	Montone schemes	953	SAM, ETHZ
8.4	Timestepping	963	
8.4.1	CFL-condition	967	
8.4.2	Linear stability	973	
8.4.3	Convergence	987	
8.5	Higher-order conservative schemes	998	
8.5.1	Piecewise linear reconstruction	1000	
8.5.2	Slope limiting	1016	
8.5.3	MUSCL scheme	1028	0.0
8.6	Outlook: systems of conservation laws	1034	p. 8

9	Finite Elements for the Stokes Equations	1035	
9.1	Viscous fluid flow	1036	Numerical Methods for PDEs
9.2	The Stokes equations	1044	
9.2.1	Constrained variational formulation	1044	
9.2.2	Saddle point problem	1050	
9.2.3	Stokes system	1061	
9.3	Saddle point problems: Galerkin discretization	1064	
9.3.1	Pressure instability	1069	
9.3.2	Stable Galerkin discretization	1081	
9.3.3	Convergence	1095	
9.4	The Taylor-Hood element	1101	
10	Adaptive Finite Element Discretization	1107	
10.1	Concept of adaptivity	1108	SAM, ETHZ
10.2	A priori hp-adaptivity	1108	
10.2.1	Graded meshes in 1D	1108	
10.2.2	Triangular graded meshes	1108	
10.2.3	hp-approximation in 1D	1108	
10.2.4	hp-adapted Lagrangian finite elements	1108	
10.3	A posteriori mesh adaptation	1108	
10.3.1	Equidistribution principle	1108	
10.3.2	Local mesh refinement	1108	
10.3.3	Refinement control	1108	
10.4	A posteriori error estimation	1108	
10.4.1	Hierarchical error estimator	1108	0.0
10.4.2	Goal oriented error estimation	1108	p. 9

11 Multilevel iterative solvers	1109	
11.1 Solving finite element linear systems	1110	Numerical Methods for PDEs
11.2 Subspace correction	1110	
11.2.1 Successive subspace correction algorithm (SSC)	1110	
11.2.2 Gauss-Seidel iteration	1110	
11.2.3 Hierarchical basis multigrid	1110	
11.2.3.1 Hierarchical basis in 1D	1110	
11.2.3.2 Hierarchical transformations	1110	
11.2.3.3 Hierarchical basis in 2D	1110	
11.2.3.4 Convergence	1110	
11.3 Multigrid method	1110	
11.3.1 Multilevel subspace corrections	1110	
11.3.2 Multigrid algorithm	1110	
11.3.2.1 Transfer operators	1110	
11.3.2.2 Local relaxations	1110	
11.3.3 Nested iteration	1110	
11.4 Algebraic multigrid	1110	
11.5 Multilevel preconditioning	1110	
12 Sparse Grids Galerkin Methods	1111	
12.1 The curse of dimension	1112	
12.2 Hierarchical basis	1112	
12.3 Sparse grids	1112	
12.4 Approximation on sparse grids	1112	0.0
12.5 Sparse grids algorithms	1112	p. 10

Index	1112	
Keywords	1112	Numerical Methods for PDEs
Examples	1124	
Definitions	1130	
MATLAB-CODE	1132	
Symbols	1133	

Course history

- Summer semester 04, R. Hiptmair (for RW/CSE undergraduates)
- Winter semester 04/05, C. Schwab (for RW/CSE undergraduates)
- Winter semester 05/06, H. Harbrecht (for RW/CSE undergraduates)
- Winter semester 06/07, C. Schwab (for BSc RW/CSE)
- Autumn semester 07, A. Chernov (for BSc RW/CSE)
- Autumn semester 08, C. Schwab (for BSc RW/CSE)
- Autumn semester 09, V. Gradinaru (for BSc RW/CSE, Subversion Revision: 22844)
- Spring semester 10, R. Hiptmair (for BSc Computer Science)
- Autumn semester 10, R. Hiptmair (for BSc RW/CSE, Subversion Revision: 30025)
- Autumn semester 11, P. Grohs (for for BSc RW/CSE, Subversion Revision: 39100)
- Spring semester 12, R. Hiptmair (for RW/CSE + D-INFK undergraduates)

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

This lecture is a core course for

- BSc in Computational Science and Engineering (RW/CSE),
- BSC in Computer Science with focus Computational Science.

Main *skills* to be acquired in this course:

- Ability to *implement* advanced numerical methods for the solution of partial differential equations in MATLAB efficiently
- Ability to *modify* and *adapt* numerical algorithms guided by awareness of their mathematical foundations
- Ability to *select* and *assess* numerical methods in light of the predictions of theory
- Ability to *identify features* of a PDE (= partial differential equation) based model that are relevant for the selection and performance of a numerical algorithm

- Ability to *understand research publications* on theoretical and practical aspects of numerical methods for partial differential equations.

This course \neq Numerical analysis of PDE (\rightarrow mathematics curriculum)
(401-3651-00V *Numerical methods for elliptic and parabolic partial differential equations,*)
Instruction on how to apply software packages

Reading instructions

This course materials are neither a textbook nor lecture notes.
They are meant to be supplemented by explanations given in class.

Some pieces of advice:

- these lecture slides are not designed to be self-contained and can be understood only when actively participating in class and tutorials.

- this document is not meant for mere reading, but for working with,
- turn pages all the time and follow the numerous cross-references,
- study the relevant section of the course material when doing homework problems.

What to expect

- The course is difficult and demanding (*ie.* ETH level)
- Do **not** expect to understand everything in class. The average student will
 - understand about one third of the material when attending the lectures,
 - understand another third when making a *serious effort* to solve the homework problems,
 - hopefully understand the remaining third when studying for the examination after the end of the course.

Perseverance will be rewarded!

Practical information

Course: 401-3663-00L Numerical Methods for Partial Differential Equations

Lectures: Mo 15-17 HG E 7

Fr 08-10 HG E 3

Tutorials: We 08-10 ML F 40 CSE

We 13-15 HG F 26.5 INFK

Th 13-15 HG E 41

D-INFK students, please sign into an exercise class at http://www.math.ethz.ch/bholger/n_dgl2012/
(will be online in the first week of the semester)

Lecturer: **Prof. Ralf Hiptmair**, office: HG G 58.2, e-mail: hiptmair@sam.math.ethz.ch

Assistants: **Eivind Fonn**, office: HG J 59, e-mail: eivind.fonn@sam.math.ethz.ch
(3rd year PhD student at Seminar of Applied Mathematics)

Holger Brandsmeier, office: HG J 46, e-mail: bholger@ethz.ch
(3rd year PhD student at Seminar of Applied Mathematics)

Axel Obermeier, office: HG J 45, e-mail: axel.obermeier@sam.math.ethz.ch
(1st year PhD student at Seminar of Applied Mathematics)

Laura Scarabosio, office: HG J 46, e-mail: laura.scarabosio@sam.math.ethz.ch
(1st year PhD student at Seminar of Applied Mathematics)

Office hours:

- Prof. Ralf Hiptmair, Monday, 17:15-17:45, HG G 58.2
- Eivind Fonn, TBA (day, time, place)
- Holger Brandsmeier, TBA (day, time, place)
- Axel Obermeier, TBA (day, time, place)
- Laura Scarabosio, TBA (day, time, place)

- Assignments:
- 13 weekly assignment sheets, handed out on Monday. Due to be handed in, in the following week during the exercise class or (beforehand) in the labeled box at HG G 53.x.
 - “Testat” requirement:
 - Exercises are marked either as core or non-core problems. Core problems (about 2 per sheet) are supposed to be essential for following the course.
 - Every exercise (core- or non-core-) gives a certain number of points.
 - A testat is given for achieving at least an equivalent to 90% of all points awarded for all core problems.
 - The testat is **not** required to participate in the exam. The testat will be taken into account through a 10% **point bonus** in the written examination.
 - Submit your MATLAB solutions via the online submission interface <http://www.math.ethz.ch/grsam/submit/?VL=04>.

Examination: “Sessionsprüfung”: computer based examination, programming and theoretical tasks

3 hour examination on ??, August ??, 2012

Reporting errors

Please report errors in the electronic lecture notes via a [wiki page](#) !

```
http://elbanet.ethz.ch/wikifarm/rhiptmair/index.php?n=Main.NPDECourse
```

(Password: NPDE, please choose [EDIT](#) menu to enter information)

Please supply the following information:

- (sub)section where the error has been found,
- precise location (e.g, after Equation (4), Thm. 2.3.3, etc.). Refrain from giving page numbers,
- brief description of the error.

Contribute to the forum

Numerical Methods for Partial Differential Equations

to be found at the URL

`http://forum.vis.ethz.ch/forumdisplay.php?f=79`

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

This forum has been set up so that you can post questions on the programming exercises that accompany the course. One of the assistants will look at the entries in this forum and

- write an answer in this forum or
- discuss the question in a consulting session and post an answer later.

A second purpose of this forum is that the assistants can collect FAQs and post answers here.

This course is designed to be rather self-contained and additional study of literature should not be crucial to follow the lecture. These references point to further sources of information, mainly devoted mathematical theory.

- D. Braess. *Finite Elements*. Cambridge University Press, 2nd edition, 2001.
- S. Brenner and R. Scott. *Mathematical theory of finite element methods*. Texts in Applied Mathematics. Springer–Verlag, New York, 1994.
- A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*, volume 159 of *Applied Mathematical Sciences*. Springer, New York, 2004.
- Ch. Großmann and H.-G. Roos. *Numerik partieller Differentialgleichungen*. Teubner Studienbücher Mathematik. Teubner, Stuttgart, 1992.
- W. Hackbusch. *Elliptic Differential Equations. Theory and Numerical Treatment*, volume 18 of *Springer Series in Computational Mathematics*. Springer, Berlin, 1992.
- P. Knabner and L. Angermann. *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*, volume 44 of *Texts in Applied Mathematics*. Springer, Heidelberg, 2003.
- S. Larsson and V. Thomée. *Partial Differential Equations with Numerical Methods*, volume 45 of *Texts in Applied Mathematics*. Springer, Heidelberg, 2003.
- R. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, UK, 2002.

1 Case Study: A Two-point Boundary Value Problem

This chapter offers a brief tour of

- **mathematical modelling** of a physical system based on **variational principles**,
- the derivation of *differential equations* from these variational principles,
- the **discretization** of the variational problems and/or of the differential equations using various approaches.

1.1 Introduction

The term “partial differential equation” (PDE) usually conjures up formulas like

$$\operatorname{div}(\sqrt{1 + \|\mathbf{grad} u(\mathbf{x})\|^2} \mathbf{grad} u(\mathbf{x})) + \mathbf{v} \cdot \mathbf{grad} u = f(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d.$$

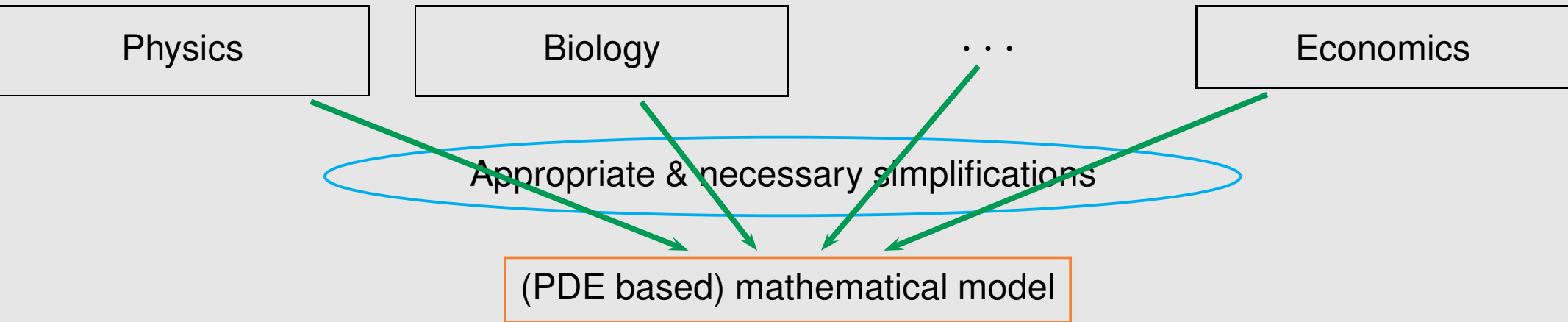
This chapter aims to wean you off this impulse and instil an appreciation that

a meaningful PDE encodes structural principles
(like equilibrium, conservation, etc.)

! The design and selection of numerical methods has to take into account these governing principles.

Remark 1.1.1 (Mathematical modelling).

Prerequisite for numerical simulation: **Mathematical modelling**



Necessary simplification:

$\left\{ \begin{array}{l} \text{system} \\ \text{phenomenon} \end{array} \right\}$ described by *a few* variables/functions in **configuration space**

The art of modelling: devise “faithful model”

Essential/relevant **traits** of $\left\{ \begin{array}{l} \text{system} \\ \text{phenomenon} \end{array} \right\} \longrightarrow$ **structural properties** of model

This chapter will offer a glimpse of considerations typical of mathematical modeling approaches that aim at differential equations.



Remark 1.1.2 (“PDEs” for univariate functions).

The classical concept of a PDE inherently involves functions of several independent variables. However, when one embraces the concept of a PDE as encoding fundamental structural properties of a model, then simple representatives in a univariate setting can be discussed.

☞ ordinary differential equations (ODEs) offer simple specimens of important classes of PDEs!

Thus, in this chapter we examine ODEs that are related to the important class of **elliptic PDEs**.



Note: The developments below cannot live up to standards of mathematical rigor, because what has deliberately omitted is the discussion of the **functional analytic framework** (function space theory) required for a complete statement of, for instance, minimization problems and variational problems.

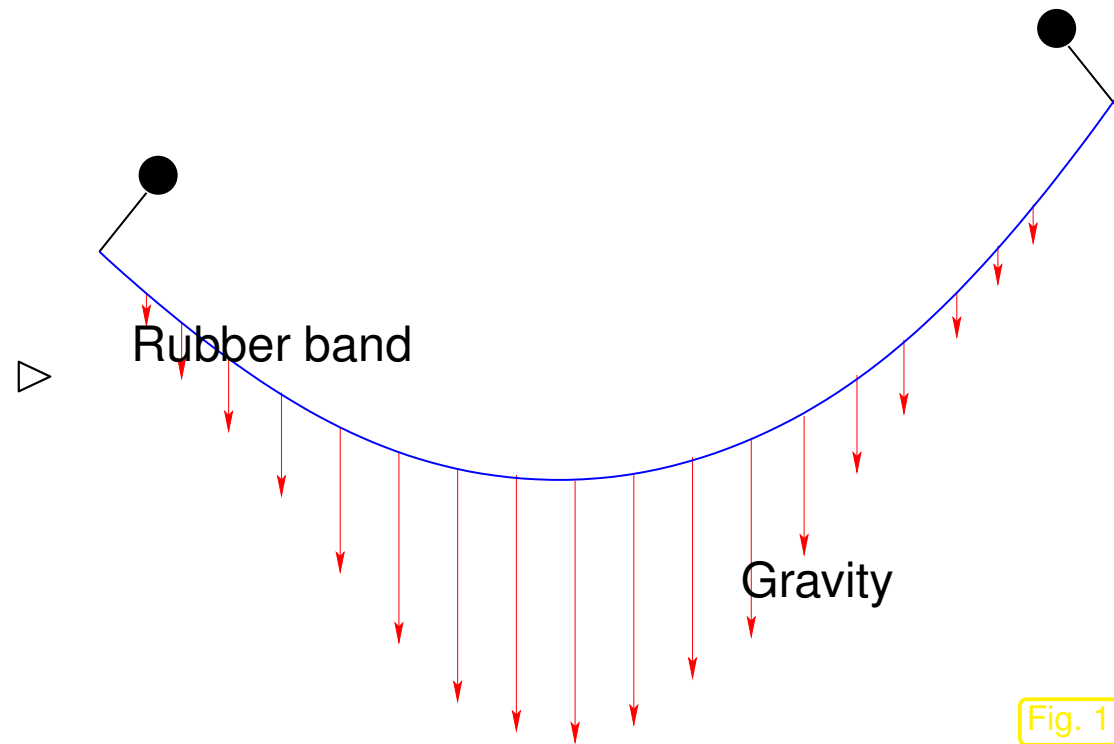
1.2 A model(ing) problem

1.2.1 Linear elastic string

Static mechanical problem:

Deformation of elastic “1D” string (rubber band) under its own weight

Constraint: string pinned at endpoints



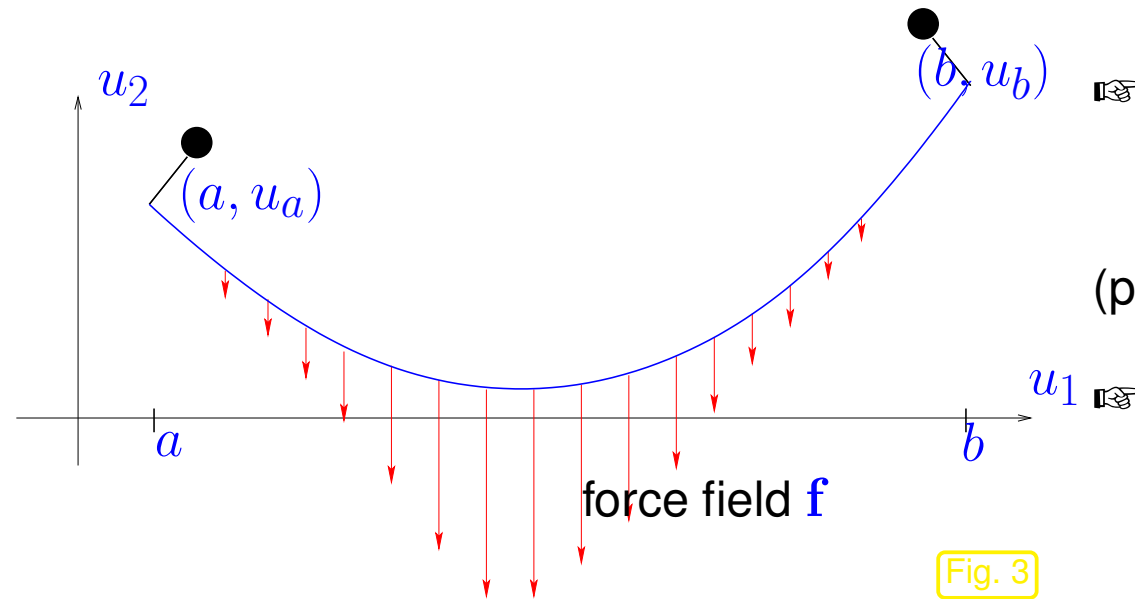
R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Sought: (Approximation of) “shape” of elastic string

Configuration space



= space of curves

shape of string



curve $\mathbf{u} : [0, 1] \mapsto \mathbb{R}^2$, $\mathbf{u} = \mathbf{u}(\xi)$
(physical units $[\mathbf{u}] = 1\text{m}$)

Pinning conditions (**boundary conditions**):

$$\mathbf{u}(0) = \begin{pmatrix} a \\ u_a \end{pmatrix} \in \mathbb{R}^2, \quad \mathbf{u}(1) = \begin{pmatrix} b \\ u_b \end{pmatrix} \in \mathbb{R}^2. \quad (1.2.1)$$

Note: description of a curve in the plane by a mapping $[0, 1] \mapsto \mathbb{R}^2$ requires **coordinate system**. Of course, the choice of the coordinate system must not affect the shape obtained from the mathematical model, a property called **frame indifference**.

Current concept of force field \mathbf{f} : force “pulls” at elastic string. Alternative: force due to a **potential**. Gravitational force (*i.e.*, a constant force field) allows both descriptions.

Terminology: $[0, 1] \hat{=}$ parameter domain,  notation Ω

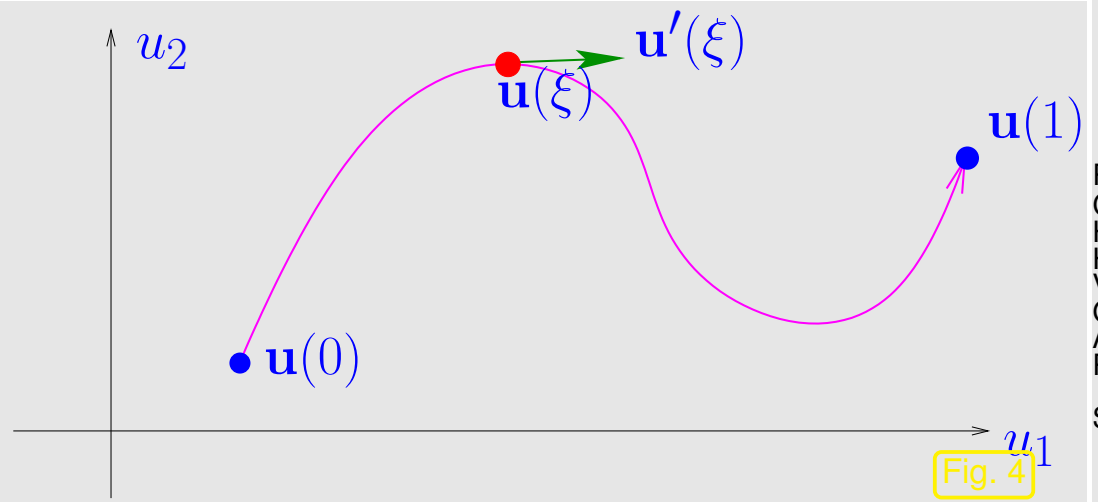
Remark 1.2.2 (Parameterization of a curve). \rightarrow [32, Sect. 7.4]

We consider a curve in \mathbb{R}^2 $\mathbf{u} : [0, 1] \mapsto \mathbb{R}^2$

$$\mathbf{u} \in (C^0([0, 1]))^2$$


$$\Updownarrow$$

connected curve



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

 notation: $C^k([a, b]) \hat{=}$ k -times continuously differentiable functions on $[a, b] \subset \mathbb{R}$, see [32, Sect. 5.4]

$(C^k([a, b]))^2 \hat{=}$ k -times continuously differentiable curves $\mathbf{u} : [a, b] \mapsto \mathbb{R}^2$, that is, if $\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$, then $u_1, u_2 \in C^k([a, b])$.

Geometric intuition: $\mathbf{u}(\xi)$ moves along the curve as ξ increases from 0 to 1.

Interpretation of curve parameter ξ : “virtual time”

➤ $\|\mathbf{u}'\| \hat{=}$ “speed” with which curve is traversed

➤ $\int_0^1 \|\mathbf{u}'(\xi)\| \, d\xi \hat{=}$ length of curve

✎ notation: $\|\cdot\| \hat{=}$ Euclidean norm of a vector $\in \mathbb{R}^n$

➤ parameterization is supposed to be *locally injective*:

$$\forall \xi \in]0, 1[: \exists \epsilon > 0: \forall \eta, |\eta - \xi| < \epsilon: \mathbf{u}(\eta) \neq \mathbf{u}(\xi) .$$

▶ For $\mathbf{u} \in (C^1([0, 1]))^2$ we expect $\mathbf{u}'(\xi) \neq 0$ for all $0 \leq \xi \leq 1$

✎ notation: $' \hat{=}$ derivative w.r.t. curve parameter, here ξ

Remark 1.2.4 (Material coordinate).

Interpretation of curve parameter ξ :

ξ : unique identifier for each infinitesimal section of the string,
a *label* for each “material point” on the string

► $\xi \hat{=}$ material coordinate, unrelated to “position in space” (= physical coordinate),
 ξ has no physical dimension ► $'$ does not affect dimension.



Remark 1.2.6 (Non-dimensional equations).

By fixing reference values for the basic physical units occurring in a model (“**scaling**”), one can switch to a **non-dimensional** form of the model equations.

In the case of the elastic string model the basic units are

- unit of length 1m ,
- unit of force 1N .

Thus, non-dimensional equations arise from fixing a reference length ℓ_0 and a reference force f_0 .

Below, following a (bad) habit of mathematicians, physical units will be routinely dropped, which tacitly assumes a priori scaling.



Quantities that have to be specified to allow the unique determination of a configuration in a mathematical model are called **problem data/parameters**. In the elastic string model the problem parameters are

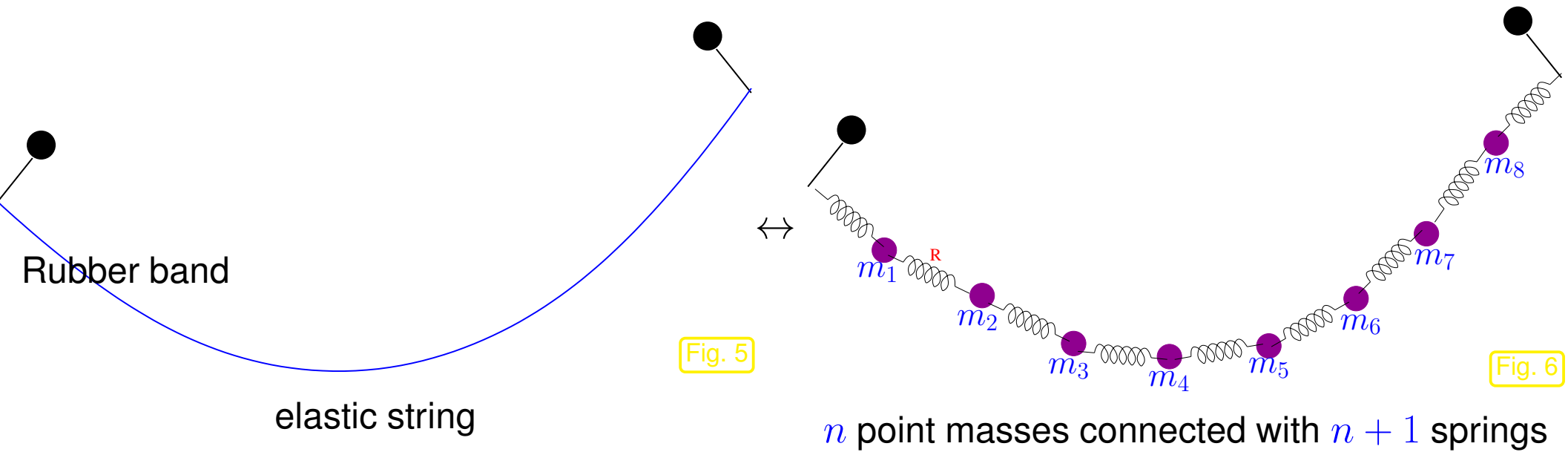
- the boundary conditions (1.2.1),
- the force field $\mathbf{f} : [0, 1] \mapsto \mathbb{R}^2$, $[\mathbf{f}] = 1\text{N}$, $\mathbf{f}(\xi) \hat{=}$ force “pulling at” a material point ξ .

Special case: gravitational force $\mathbf{f}(\xi) := -g\rho(\xi) \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $0 \leq \xi \leq 1$, $g = 9.81\text{ms}^{-2}$
with density $\rho : [0, 1] \mapsto \mathbb{R}^+$, $[\rho] = \text{kg}$,

local elastic material properties, see Sect. 1.2.3.

1.2.2 Mass-spring model

Idea: model string as a system of many simple components that interact in simple ways



Assumption:

linear springs ↔ Hooke's law

Force $F(l) = \kappa \left(\frac{l}{l_0} - 1 \right)$ (relative elongation) . (1.2.9)

$\kappa \hat{=}$ spring constant (stiffness), $[\kappa] = 1\text{N}$, $\kappa > 0$,

$l_0 \hat{=}$ **equilibrium length** of (relaxed) spring.

► **elastic energy** stored in linear spring at length $l > 0$

$$E_{\text{el}} = \int_{l_0}^l F(\tau) \, d\tau = \frac{1}{2} \frac{\kappa}{l_0} (l - l_0)^2, \quad [E_{\text{el}}] = 1\text{J} . \quad (1.2.10)$$

Configuration space for mass-spring model:

$\mathbf{u}^i \in \mathbb{R}^2 \hat{=}$ position of i -th mass point, $i = 1, \dots, n$

➤ **finite-dimensional** configuration space = $(\mathbb{R}^2)^n$

Models, for which configurations can be described by means of finitely many real numbers are called **discrete**. Hence, the mass-spring model is a **discrete model**, see Sect. 1.5.

► Total elastic energy of mass-spring model in configuration $(\mathbf{u}^1, \dots, \mathbf{u}^n) \in (\mathbb{R}^2)^n$:

$$(1.2.10) \quad \Rightarrow \quad J_{\text{el}}^{(n)} = J_{\text{el}}^{(n)}(\mathbf{u}^1, \dots, \mathbf{u}^n) := \frac{1}{2} \sum_{i=0}^n \underbrace{\frac{\kappa_i}{l_i} (\|\mathbf{u}^{i+1} - \mathbf{u}^i\| - l_i)^2}_{\text{elastic energy of } i\text{-th spring}}, \quad (1.2.11)$$

where $\mathbf{u}^0 := \begin{pmatrix} a \\ u_a \end{pmatrix}$, $\mathbf{u}^{n+1} := \begin{pmatrix} b \\ u_b \end{pmatrix}$ (pinning positions (1.2.1)),
 $\kappa_i \hat{=}$ spring constant of i -th spring, $i = 0, \dots, n$,
 $l_i > 0 \hat{=}$ equilibrium length of i -th spring.

► Total “gravitational” energy of mass-spring model in configuration $(\mathbf{u}^1, \dots, \mathbf{u}^n)$ due to external force field:

$$J_{\text{f}}^{(n)} = J_{\text{f}}^{(n)}(\mathbf{u}^1, \dots, \mathbf{u}^n) := - \sum_{i=1}^n \mathbf{f}^i \cdot \mathbf{u}^i, \quad (1.2.12)$$

where $\mathbf{f}^i \hat{=}$ force acting on i -th mass, $i = 1, \dots, n$.

notation: $\mathbf{u} \cdot \mathbf{v} := \mathbf{u}^T \mathbf{v} = \sum_{j=1}^n u_j v_j \hat{=} \text{inner product of vectors in } \mathbb{R}^n.$

Known from classical mechanics, static case: **equilibrium principle**

systems attains configuration(s) of minimal (potential) energy

$$J^{(n)} := J_{\text{el}}^{(n)} + J_{\text{f}}^{(n)}$$

► equilibrium configuration $\mathbf{u}_*^1, \dots, \mathbf{u}_*^n$ of mass-spring system solves

$$(\mathbf{u}_*^1, \dots, \mathbf{u}_*^n) = \underset{(\mathbf{u}^1, \dots, \mathbf{u}^n) \in \mathbb{R}^{2n}}{\operatorname{argmin}} J^{(n)}(\mathbf{u}^1, \dots, \mathbf{u}^n). \quad (1.2.13)$$

Plot of $J^{(1)}(\mathbf{u}^1)$



Mass-spring system with only one point mass
(non-dimensional $l_1 = l_2 = 1$, $\kappa_1 = \kappa_2 = 1$,
 $\mathbf{u}^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\mathbf{u}^2 = \begin{pmatrix} 1 \\ 0.2 \end{pmatrix}$, $\mathbf{f}^1 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$)

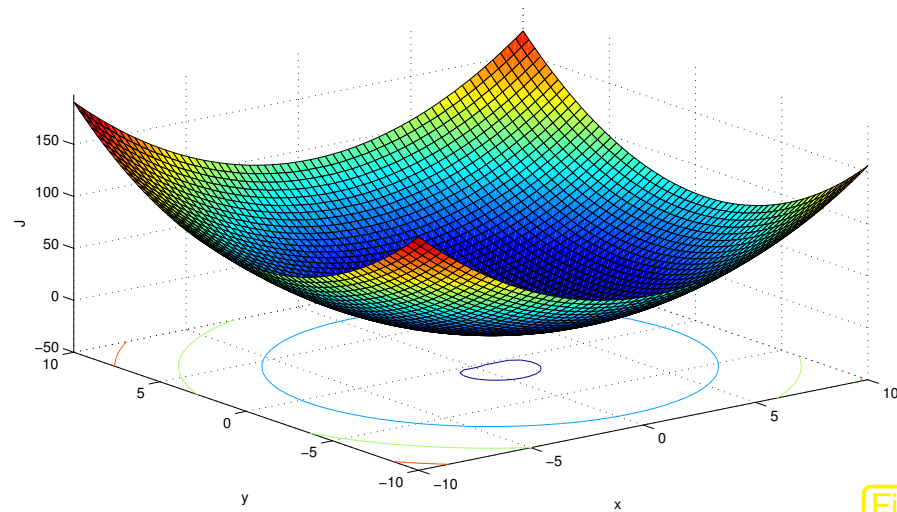


Fig. 7

Note: solutions of (1.2.13) need not be unique !

To see this, consider the case $L := \sum_{i=0}^n l_i > \|\mathbf{u}^{n+1} - \mathbf{u}^0\|$ and $\mathbf{f} \equiv 0$ (slack ensemble of springs without external forcing). In this situation many crooked arrangements of the masses will have zero total potential energy.

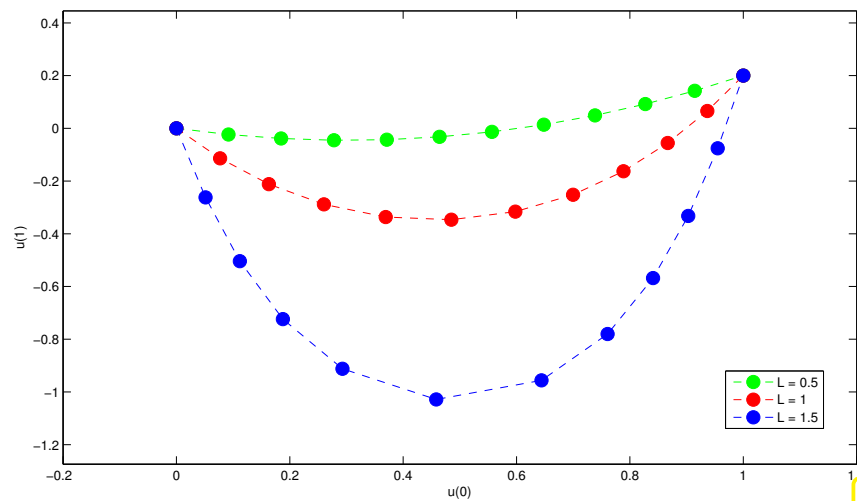


Fig. 8

◁ minimal energy configuration of a mass spring system for variable L .

($n = 10$, non-dimensional $\kappa_i = 1$, $l_i = L/n$, $i = 0, \dots, 10$)

1.2.3 Continuum limit

Heuristics: elastic string = spring-mass system with “infinitely many infinitesimal masses” and “infinitesimally short” springs.

- Policy:
- consider sequence $(SMS_n)_{n \in \mathbb{N}}$ of spring-mass systems with n masses,
 - identify material coordinate (\rightarrow Rem. 1.2.4) of point masses,
 - choose system parameters with meaningful limits,
 - derive expressions for energies as $n \rightarrow \infty$,
 - use them to define the “continuous elastic string model”.

Assumption: equal equilibrium lengths of all springs

$$l_i = \frac{L}{n+1}, L > 0,$$

➤ $L \hat{=}$ equilibrium length of elastic string: $L = \sum_i l_i, [L] = 1\text{m}.$

Equilibrium configuration of mass-spring system ➤

(non-dimensional $l_i = \frac{L}{n+1}, \kappa_i = 1, m_i = \frac{1}{n}, \mathbf{f}_i = \frac{1}{n} \begin{pmatrix} 0 \\ -1 \end{pmatrix}, L = 1, n$ varying)

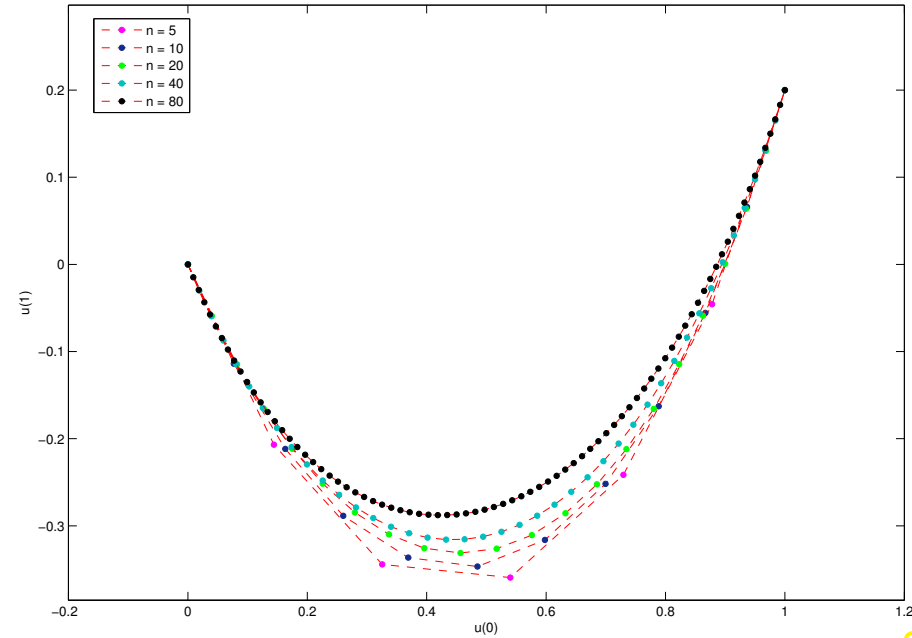
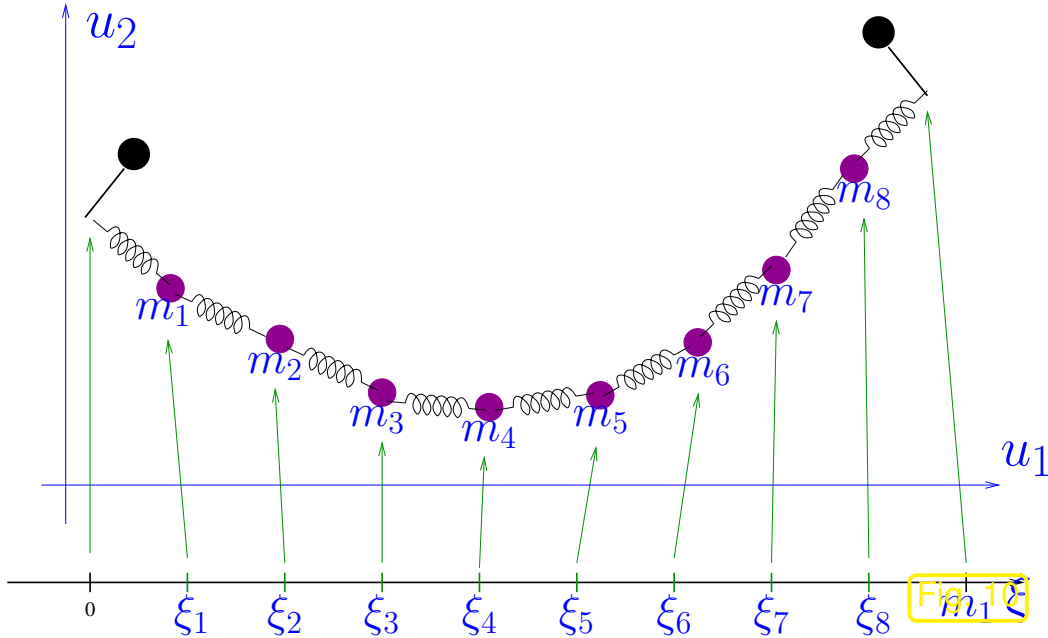


Fig. 9

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



- ▶ masses are uniformly spaced **on string** \mathbf{u} :
 $[0, 1] \mapsto \mathbb{R}^2$
- material coordinate of i -th mass in \mathcal{SMS}_n :

$$\xi_i^{(n)} := \frac{i}{n+1}: \quad \mathbf{u}^i := \mathbf{u}(\xi_i^{(n)})$$

In the spring-mass model each spring has its own stiffness κ_i and every mass point its own force \mathbf{f}^i acting on it. When considering the “limit” of a sequence of spring-mass models, we have to detach stiffness and force from springs and masses and attach them to material points, *cf.* Rem. 1.2.4. In other words stiffness κ_i and force \mathbf{f}^i have to be induced by a stiffness function $\kappa(\xi)$ and force function $\mathbf{f}(\xi)$. This linkage has to be done in a way to allow for a meaningful limit $n \rightarrow \infty$ for the potential energy.

“Limit-compatible” system parameters: $(\xi_{i+1/2}^{(n)} := \frac{1}{2}(\xi_{i+1}^{(n)} + \xi_i^{(n)}))$

• $\kappa_i = \kappa(\xi_{i+1/2}^{(n)})$ with *integrable* stiffness function $\kappa : [0, 1] \mapsto \mathbb{R}^+$,

• $\mathbf{f}^i = \int_{\xi_{i-1/2}^{(n)}}^{\xi_{i+1/2}^{(n)}} \mathbf{f}(\xi) d\xi$ “lumped force”, integrable force field $\mathbf{f} : [0, 1] \mapsto \mathbb{R}^2$

► energies, see (1.2.11), (1.2.12)

$$J_{\text{el}}^{(n)}(\mathbf{u}) = \frac{1}{2} \sum_{i=0}^n \frac{n+1}{L} \kappa(\xi_{i+1/2}^{(n)}) \left(\left\| \mathbf{u}(\xi_{i+1}^{(n)}) - \mathbf{u}(\xi_i^{(n)}) \right\| - \frac{L}{n+1} \right)^2, \quad (1.2.16)$$

$$J_{\text{f}}^{(n)}(\mathbf{u}) = - \sum_{i=1}^n \int_{\xi_{i-1/2}^{(n)}}^{\xi_{i+1/2}^{(n)}} \mathbf{f}(\xi) d\xi \cdot \mathbf{u}(\xi_i^{(n)}). \quad (1.2.17)$$

Assumption: $\mathbf{u} \in (C^2([0, 1]))^2$ (twice continuously differentiable)

① Simple limit for potential energy due to external force:

$$J_f(\mathbf{u}) = \lim_{n \rightarrow \infty} J_f^{(n)}(\mathbf{u}) = \lim_{n \rightarrow \infty} - \sum_{i=1}^n \int_{\xi_{i-1/2}^{(n)}}^{\xi_{i+1/2}^{(n)}} \mathbf{f}(\xi) \, d\xi \cdot \mathbf{u}(\xi_i^n) = - \int_0^1 \mathbf{f}(\xi) \cdot \mathbf{u}(\xi) \, d\xi . \quad (1.2.18)$$

② Limit of elastic energy:

Tool: Taylor expansion: for $\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \in C^2$ with derivative \mathbf{u}' , $1 \gg \eta \rightarrow 0$

$$\begin{aligned} \|\mathbf{u}(\xi + \eta) - \mathbf{u}(\xi - \eta)\| &= \sqrt{(u_1(\xi + \eta) - u_1(\xi - \eta))^2 + (u_2(\xi + \eta) - u_2(\xi - \eta))^2} \\ &= \sqrt{(2u'_1(\xi)\eta + O(\eta^3))^2 + (2u'_2(\xi)\eta + O(\eta^3))^2} \\ &= 2\eta \|\mathbf{u}'(\xi)\| \sqrt{1 + O(\eta^2)} = 2\eta \|\mathbf{u}'(\xi)\| + O(\eta^3). \end{aligned} \quad (1.2.22)$$

Apply this to (1.2.16) with $\eta = \frac{1}{2n+1}$ for $n \rightarrow \infty$, “ O -terms” vanish in the limit

$$\begin{aligned} J_{\text{el}}^{(n)}(\mathbf{u}) &= \frac{1}{2} \sum_{i=0}^n \frac{n+1}{L} \kappa(\xi_{i+1/2}^{(n)}) \left(\frac{1}{n+1} \|\mathbf{u}'(\xi_{i+1/2}^{(n)})\| + O\left(\frac{1}{(n+1)^2}\right) - \frac{L}{n+1} \right)^2 \\ &= \frac{1}{2L} \frac{1}{n+1} \sum_{i=0}^n \kappa(\xi_{i+1/2}^{(n)}) \left(\|\mathbf{u}'(\xi_{i+1/2}^{(n)})\| + O\left(\frac{1}{n+1}\right) - L \right)^2 \end{aligned} \quad (1.2.23)$$

Consideration: integral as limit of Riemann sums, see [32, Sect. 6.2]:

$$q \in C^0([0, 1]): \quad \lim_{n \rightarrow \infty} \frac{1}{n+1} \sum_{j=0}^n q\left(\frac{j+1/2}{n+1}\right) = \int_0^1 q(\xi) \, d\xi. \quad (1.2.24)$$

$$\Rightarrow J_{\text{el}}(\mathbf{u}) = \lim_{n \rightarrow \infty} J_{\text{el}}^{(n)}(\mathbf{u}) = \frac{1}{2L} \int_0^1 \kappa(\xi) (\|\mathbf{u}'(\xi)\| - L)^2 \, d\xi. \quad (1.2.25)$$

► Equilibrium condition for limit model (minimal total potential energy):

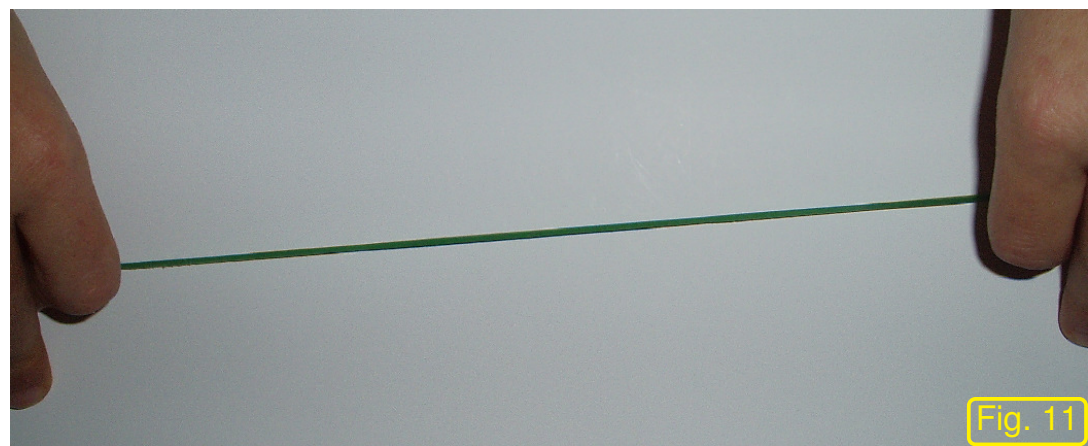
$$\mathbf{u}_* = \operatorname{argmin}_{\mathbf{u} \in (C^1([0,1]))^2 \text{ \& (1.2.1)}} \underbrace{\int_0^1 \frac{\kappa(\xi)}{2L} (\|\mathbf{u}'(\xi)\| - L)^2 - \mathbf{f}(\xi) \cdot \mathbf{u}(\xi) \, d\xi}_{=: J(\mathbf{u})}. \quad (1.2.26)$$

total potential energy functional, $[J] = 1\text{J}$
 = a minimization problem in a **function space** !

Example 1.2.27 (Tense string without external forcing).

Setting:

- no external force: $\mathbf{f} \equiv 0$
- homogeneous string: $\kappa = \kappa_0 = \text{const}$
- tense string: $L < \|\mathbf{u}(0) - \mathbf{u}(1)\|$
(\triangleright positive elastic energy)



$$\blacktriangleright (1.2.26) \Leftrightarrow \mathbf{u}_* = \underset{\mathbf{u} \in (C^1([0,1]))^2 \& (1.2.1)}{\text{argmin}} \frac{\kappa_0}{2L} \int_0^1 (\|\mathbf{u}'(\xi)\| - L)^2 d\xi. \quad (1.2.31)$$

Note: in (1.2.31) \mathbf{u} enters J only through \mathbf{u}' !

Constraint on \mathbf{u}' : by triangle inequality for integrals, see [32, Sect. 6.3]

$$\ell := \|\mathbf{u}(1) - \mathbf{u}(0)\| = \left\| \int_0^1 \mathbf{u}'(\xi) d\xi \right\| \leq \int_0^1 \|\mathbf{u}'(\xi)\| d\xi. \quad (1.2.32)$$

➔ Consider related minimization problem

$$w_* = \operatorname{argmin}_w \left\{ \frac{\kappa_0}{2L} \int_0^1 (w - L)^2 d\xi : \begin{array}{l} w \in (C^0([0, 1]))^2, \\ \int_0^1 w(\xi) d\xi \geq \ell \end{array} \right\}. \quad (1.2.33)$$

⇒ unique solution $w_*(\xi) = \ell$ (constant solution)

$\|\mathbf{u}'(\xi)\| = \ell$ and the boundary conditions (1.2.1) are satisfied for the **straight line solution** of (1.2.31)

$$\mathbf{u}_*(\xi) = (1 - \xi)\mathbf{u}(0) + \xi\mathbf{u}(1).$$

It is exactly the “straight string” solution that physical intuition suggests.



1.3 Variational approach

We face the task of minimizing a functional over an ∞ -dimensional function space. In this section necessary conditions for the minimizer will formally be derived in the form of variational equations. This idea is one of the cornerstone of a branch of analysis called **calculus of variations**.

We will not dip into this theory, but perform manipulations at a formal level. Yet, all considerations below can be justified rigorously.

1.3.1 Virtual work equation

notation: $C_0^k([0, 1]) := \{v \in C^k([0, 1]): v(0) = v(1) = 0\}, k \in \mathbb{N}_0$

Main “idea of calculus of variations”:

$$\mathbf{u}_* \text{ solves (1.2.26)} \Rightarrow J(\mathbf{u}_*) \leq J(\mathbf{u}_* + t\mathbf{v}) \quad \forall t \in \mathbb{R}, \mathbf{v} \in (C_0^2([0, 1]))^2. \quad (1.3.1)$$

▶ $\varphi(t) := J(\mathbf{u}_* + t\mathbf{v})$ has global minimum for $t = 0$

▶ If φ differentiable, then $\frac{d\varphi}{dt}(0) = 0$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

(1.3.1) expresses the fact that \mathbf{u}_* can only be a minimal energy configuration, if no admissible perturbation leads to a decrease of the total energy.

Note: $\mathbf{v}(0) = \mathbf{v}(1) = 0$, because we must not tamper with the pinning conditions (1.2.1).

Rule: Variation \mathbf{v} must vanish where argument function \mathbf{u} is fixed.

Computation of $\frac{d\varphi}{dt}(0)$ for J from (1.2.26) amounts to computing a “**configurational derivative**” in direction \mathbf{v} .

We pursue a separate treatment of energy contributions (This also demonstrates a simple formal approach to computing configurational derivatives.):

❶ Potential energy (1.2.18) due to external force:

$$\lim_{t \rightarrow 0} \frac{J_f(\mathbf{u}_* + t\mathbf{v}) - J_f(\mathbf{u}_*)}{t} = - \lim_{t \rightarrow 0} \frac{1}{t} \int_0^1 \mathbf{f}(\xi) \cdot t\mathbf{v}(\xi) \, d\xi = - \int_0^1 \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi . \quad (1.3.7)$$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

❷ Elastic energy (1.2.25): more difficult, tool: Taylor expansion [32, Sect. 5.5]

Analogous to (1.2.22), $\mathbf{x} \in \mathbb{R}^2 \setminus \{0\}$, $\mathbf{h} \in \mathbb{R}^2$, for $\mathbb{R} \ni t \rightarrow 0$

$$\begin{aligned} \|\mathbf{x} + t\mathbf{h}\| &= \sqrt{(x_1 + th_1)^2 + (x_2 + th_2)^2} = \sqrt{\|\mathbf{x}\|^2 + 2t\mathbf{x} \cdot \mathbf{h} + t^2 \|\mathbf{h}\|^2} \\ &= \|\mathbf{x}\| \sqrt{1 + 2t \frac{\mathbf{x} \cdot \mathbf{h}}{\|\mathbf{x}\|^2} + t^2 \frac{\|\mathbf{h}\|^2}{\|\mathbf{x}\|^2}} = \|\mathbf{x}\| + t \frac{\mathbf{x} \cdot \mathbf{h}}{\|\mathbf{x}\|} + O(t^2) , \end{aligned} \quad (1.3.8)$$

where we used the truncated Taylor series for $\sqrt{1+x}$

$$\sqrt{1+\delta} = 1 + \frac{1}{2}\delta + O(\delta^2) \quad \text{for } \delta \rightarrow 0. \quad (1.3.9)$$

Use (1.3.8) in the perturbation analysis for the elastic energy:

$$\begin{aligned} \blacktriangleright \quad (\|\mathbf{u}'(\xi) + t\mathbf{v}'(\xi)\| - L)^2 &= \left(\|\mathbf{u}'(\xi)\| + t \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} + O(t^2) - L \right)^2 \\ &= (\|\mathbf{u}'(\xi)\| - L)^2 + 2t (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} + O(t^2). \end{aligned}$$

$$\blacktriangleright \quad J_{\text{el}}(\mathbf{u} + t\mathbf{v}) - J_{\text{el}}(\mathbf{u}) = \frac{t}{L} \int_0^1 \kappa(\xi) (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} + O(t^2) \, d\xi. \quad (1.3.10)$$

$$\blacktriangleright \quad \lim_{t \rightarrow 0} \frac{J_{\text{el}}(\mathbf{u}_* + t\mathbf{v}) - J_{\text{el}}(\mathbf{u}_*)}{t} = \int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} \, d\xi. \quad (1.3.11)$$

Here we take for granted $\|\mathbf{u}'(\xi)\| \neq 0$, which is an essential property of a meaningful parameterization of the elastic string, see Rem. 1.2.2.



Necessary condition for \mathbf{u}_* solving (1.2.26)

$$\int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'_*(\xi)\| - L) \frac{\mathbf{u}'_*(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'_*(\xi)\|} - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) d\xi = 0 \quad \forall \mathbf{v} \in (C_0^2([0, 1]))^2. \quad (1.3.12)$$

This is a **non-linear variational equation** on domain $\Omega = [0, 1]$

Remark 1.3.16 (Differentiating a functional on a space of curves).

For a C^2 -function $F : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$, $d \in \mathbb{N}$, consider the functional

$$J : (C^1([0, 1]))^d \mapsto \mathbb{R} \quad , \quad J(\mathbf{u}) := \int_0^1 F(\mathbf{u}'(\xi), \mathbf{u}(\xi)) d\xi .$$

Use the multidimensional Taylor's formula [32, Satz 7.5.2]

$$F(\mathbf{u} + \delta\mathbf{u}, \mathbf{v} + \delta\mathbf{v}) = F(\mathbf{u}, \mathbf{v}) + D_1F(\mathbf{u}, \mathbf{v})\delta\mathbf{u} + D_2F(\mathbf{u}, \mathbf{v})\delta\mathbf{v} + O(\|\delta\mathbf{u}\|^2 + \|\delta\mathbf{v}\|^2) . \quad (1.3.17)$$

Here, D_1F and D_2F are the **partial derivatives** of F w.r.t the first and second vector argument, respectively. These are *row vectors*.

$$J(\mathbf{u} + t\mathbf{v}) = J(\mathbf{u}) + t \underbrace{\int_0^1 D_1F(\mathbf{u}'(\xi), \mathbf{u}(\xi))\mathbf{v}'(\xi) + D_2F(\mathbf{u}'(\xi), \mathbf{u}(\xi))\mathbf{v}(\xi) d\xi}_{\text{"directional derivative"} (D_{\mathbf{v}}J)(\mathbf{u})(\mathbf{v})} + O(t^2) . \quad (1.3.18)$$

The derivatives \mathbf{u}' , \mathbf{v}' are just regular 1D derivatives w.r.t. the parameter ξ . They yield column vectors. Hence, we deal with a scalar integrand.

Remark 1.3.20 (Virtual work principle).

In statics, the derivation of variational equations from energy minimization (equilibrium principle, see (1.2.13)) is known as the method of **virtual work**: Small admissible changes of the equilibrium configuration of the system invariably entail active work.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 1.3.21 (Non-linear variational equation).

Now we unravel the structure behind the non-linear variational problem (1.3.12).

Recall essential terminology from linear algebra:

Definition 1.3.23 ((Bi-)linear forms).

Given an \mathbb{R} -vector space V , a **linear form (linear functional)** l is a mapping $f : V \mapsto \mathbb{R}$ that satisfies

$$l(\alpha u + \beta v) = \alpha l(u) + \beta l(v) \quad \forall u, v \in V, \forall \alpha, \beta \in \mathbb{R}.$$

A **bilinear form** a on V is a mapping $a : V \times V \mapsto \mathbb{R}$, for which

$$\begin{aligned} a(\alpha_1 v_1 + \beta_1 u_1, \alpha_2 v_2 + \beta_2 u_2) &= \\ &= \alpha_1 \alpha_2 a(v_1, v_2) + \alpha_1 \beta_2 a(v_1, u_2) + \beta_1 \alpha_2 a(u_1, v_2) + \beta_1 \beta_2 a(u_1, u_2) \end{aligned}$$

for all $u_i, v_i \in V$, $\alpha_i, \beta_i \in \mathbb{R}$, $i = 1, 2$.

 notation: $a, b, \dots \hat{=} \text{bilinear forms}$

In the case of (1.3.12) we make a very important observation, namely that, keeping \mathbf{u}_* fixed, the left hand side is a **linear** functional (linear form) in the **test function** \mathbf{v} :

Structure of

$$\int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'_*(\xi)\| - L) \frac{\mathbf{u}'_*(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'_*(\xi)\|} - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = 0 \quad \forall \mathbf{v} \in (C_0^2([0, 1]))^2 : \quad (1.3.12)$$

▷ abstract non-linear variational equation

$$u \in V: \quad \mathbf{a}(u; v) = \ell(v) \quad \forall v \in V_0, \quad (1.3.24)$$

- $V_0 \hat{=}$ (real) *vector space* of functions,
- $V \hat{=}$ *affine space* of functions: $V = u_0 + V_0$, with **offset function** $u_0 \in V$,
- $\ell \hat{=}$ a linear mapping $V_0 \mapsto \mathbb{R}$, a **linear form**,
- $\mathbf{a} \hat{=}$ a mapping $V \times V_0 \mapsto \mathbb{R}$, *linear in the second argument*, that is

$$\mathbf{a}(u; \alpha v + \beta w) = \alpha \mathbf{a}(u; v) + \beta \mathbf{a}(u; w) \quad \forall u \in V, v, w \in V_0, \alpha, \beta \in \mathbb{R}. \quad (1.3.25)$$

Terminology related to variational problem (1.3.24):

V is called **trial space**

V_0 is called **test space**

Explanation of terminology:

- **trial space** $\hat{=}$ the function space in which we seek the solution
- **test space** $\hat{=}$ the space of eligible **test functions** v in a variational problem like (1.3.24) = space of admissible variations (shape perturbations) in (1.3.1).

The two spaces need not be the same: $V \neq V_0$ is common and already indicated by the notation. For many variational problem, which are not examined in this course, they may even comprise functions with different smoothness properties.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

In concrete terms (for elastic string continuum model):

$$\begin{aligned}
 \bullet \quad V_0 &:= (C_0^2([0, 1]))^2, \\
 \bullet \quad V &:= \left\{ \mathbf{u} \in (C^2([0, 1]))^2 : \mathbf{u}(0) = \begin{pmatrix} a \\ u_a \end{pmatrix}, \mathbf{u}(1) = \begin{pmatrix} b \\ u_b \end{pmatrix} \right\} \\
 &= \underbrace{[\xi \mapsto (1 - \xi)\mathbf{u}(0) + \xi\mathbf{u}(1)]}_{=:\mathbf{u}_0} + V_0, \qquad (1.3.29)
 \end{aligned}$$

$$\bullet \quad \ell(\mathbf{v}) := \int_0^1 \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi, \quad (1.3.30)$$

$$\bullet \quad a(\mathbf{u}; \mathbf{v}) := \int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} \, d\xi. \quad (1.3.31)$$

Thus, for variational problem (1.3.12) arising from the elastic string model we find the common pattern $V = V_0 + u_0$, that is, the trial space is an *affine space*, arising from the test space V_0 by adding an offset function u_0 .

If $V = V_0 + u_0$, then there is a way to recast (1.3.24) as a variational problem with the same trial and test space V_0 :

Rewriting (1.3.24) using the offset function $u_0 \in V$:

$$(1.3.24) \quad \Rightarrow \quad w \in V_0: \quad a(u_0 + w; v) = \ell(v) \quad \forall v \in V_0 \quad \text{and} \quad u = u_0 + w. \quad (1.3.32)$$

△

1.3.2 Regularity (smoothness) requirements

Issue: The derivation of the continuum models (1.2.26) (\rightarrow Sect. 1.2.3) and (1.3.12) was based on the assumption $\mathbf{u} \in (C^2([0, 1]))^2$.

Is $\mathbf{u} \in (C^2([0, 1]))^2$ required to render the minimization problem (1.2.26)/variational problem 1.2.3) meaningful?

We will find that curves with less smoothness can still yield relevant solutions of (1.2.26)/(1.3.12).

Obvious (\rightarrow Rem. 1.2.2):

$\mathbf{u} \in (C^0([0, 1]))^2$
(string must not be torn)

$$\mathbf{u}_* = \operatorname{argmin}_{\mathbf{u} \in (C^1([0, 1]))^2 \& (1.2.1)} \int_0^1 \frac{\kappa(\xi)}{2L} (\|\mathbf{u}'(\xi)\| - L)^2 - \mathbf{f}(\xi) \cdot \mathbf{u}(\xi) \, d\xi . \quad (1.2.26)$$

Observation: $J(\mathbf{u})$ from (1.2.26), \mathbf{a} , ℓ from (1.3.31) well defined for

merely *continuous, piecewise continuously differentiable* functions $\mathbf{u}, \mathbf{v} : [0, 1] \mapsto \mathbb{R}^2$,

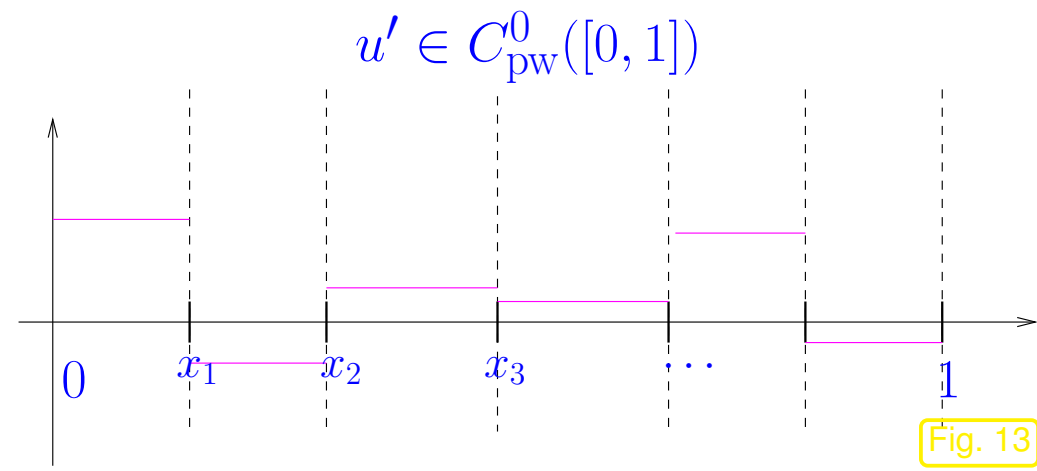
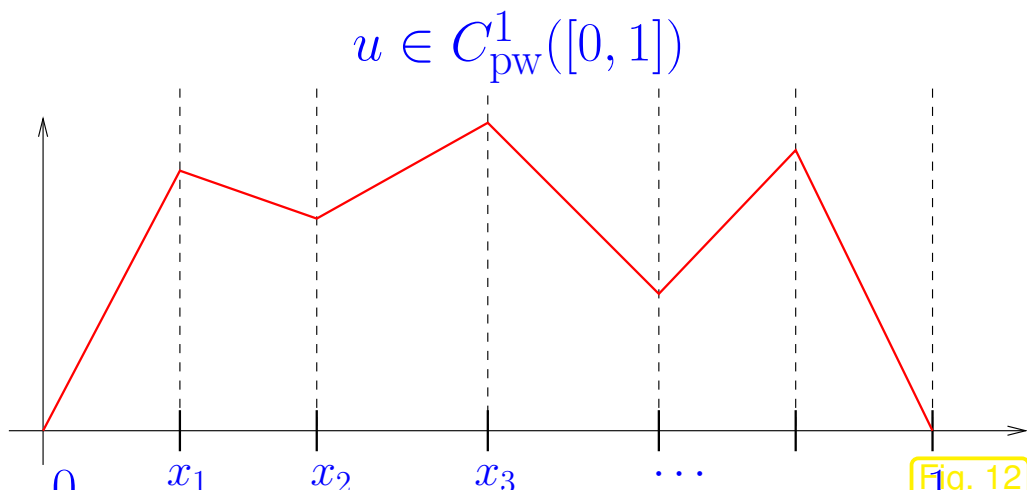
➤ \mathbf{u}' will be piecewise continuous and can be integrated.

• mere integrability of κ , \mathbf{f} sufficient.

notation: $C_{pw}^k([a, b]) \hat{=}$ globally C^{k-1} and piecewise k -times continuously differentiable functions on $[a, b] \subset \mathbb{R}$: for each $v \in C_{pw}^k([a, b])$ there is a finite partition $\{a = \tau_0 < \tau_1 < \dots < \tau_m = b\}$ such that $v|_{[\tau_{i-1}, \tau_i]}$ can be extended to a function $\in C^k([\tau_{i-1}, \tau_i])$. $C_{pw}^0([a, b]) \hat{=}$ piecewise continuous functions with only a finite number of discontinuities.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Example 1.3.33 (Non-smooth external forcing).

Setting: $\kappa = \text{const}$ (homogeneous string)

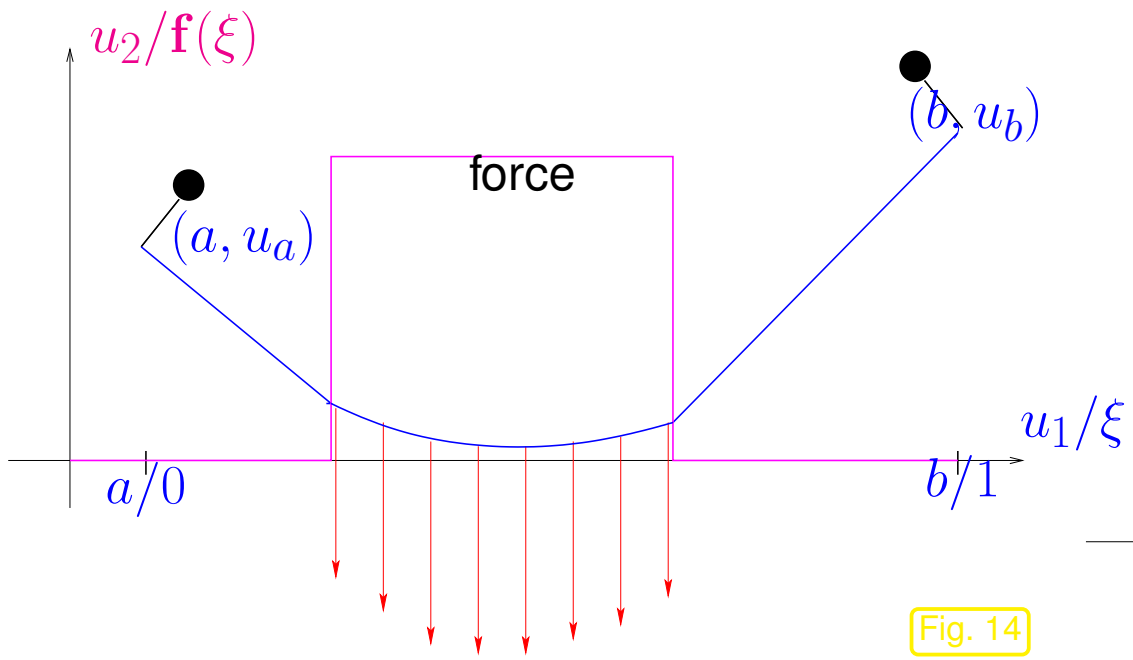


Fig. 14

✓ discontinuous \mathbf{f}

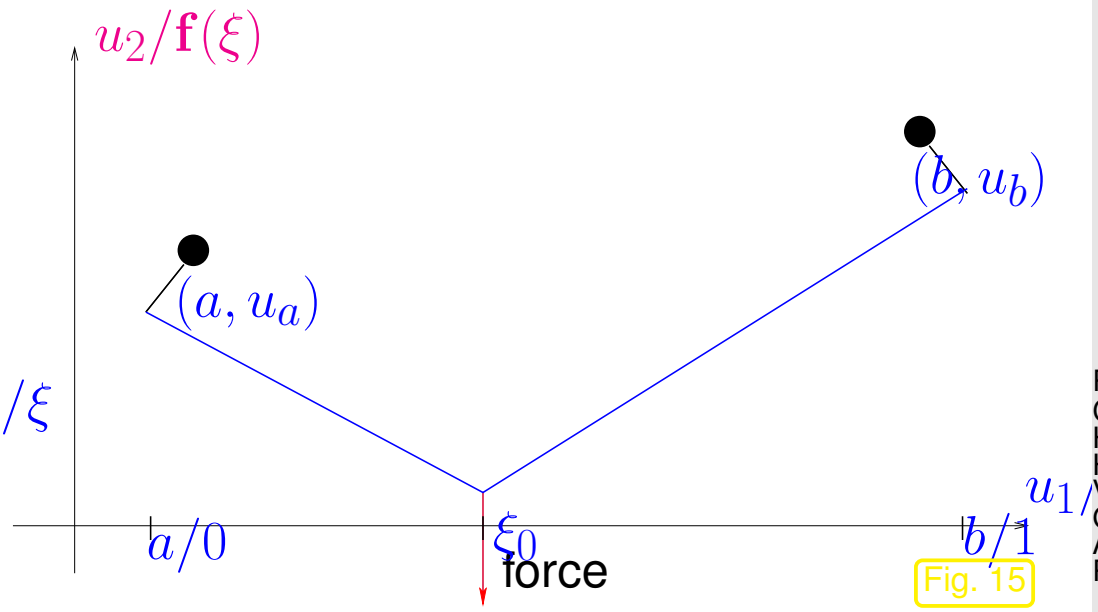


Fig. 15

✓ point force $\mathbf{f}(x) = \delta(\xi - \xi_0) \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

- ⇒ $\mathbf{u}_* \notin (C^2([0, 1]))^2$ physically meaningful:
- $\mathbf{u}_* \in (C^1([0, 1]))^2$ for discontinuous \mathbf{f}
 - merely $\mathbf{u}_* \in (C^0([0, 1]))^2$ for point force concentrated in ξ_0 : kink at ξ_0 !



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

1.3.3 Differential equation

Consider non-linear variational equation (1.3.12):

$$\int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) d\xi = 0 \quad \forall \mathbf{v} \in (C_{0,\text{pw}}^1([0, 1]))^2. \quad (1.3.12)$$

Assumption:

$$\mathbf{u} \in (C^2([0, 1]))^2 \ \& \ \kappa \in C^1([0, 1]) \ \& \ \mathbf{f} \in (C^0([0, 1]))^2 \quad (1.3.34)$$

Recall: **integration by parts** formula [32, Satz 6.1.2]:

$$\int_0^1 u(\xi)v'(\xi) \, d\xi = - \int_0^1 u'(\xi)v(\xi) \, d\xi + \underbrace{(u(1)v(1) - u(0)v(0))}_{\text{boundary terms}} \quad \forall u, v \in C_{\text{pw}}^1([0, 1]) . \quad (1.3.36)$$

Apply to elastic energy contribution in (1.3.12):

$$\begin{aligned} \int_0^1 \left(\frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|} \right) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi \\ = \int_0^1 \left\{ -\frac{d}{d\xi} \left(\frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|} \right) - \mathbf{f}(\xi) \right\} \cdot \mathbf{v}(\xi) \, d\xi . \end{aligned}$$

Note: $\mathbf{v}(0) = \mathbf{v}(1) = 0 \Rightarrow$ boundary terms vanish !

$$(1.3.12) \Rightarrow \int_0^1 \underbrace{\left\{ -\frac{d}{d\xi} \left(\frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|} \right) - \mathbf{f}(\xi) \right\}}_{\in C_{\text{pw}}^0([0,1])} \cdot \mathbf{v}(\xi) \, d\xi = 0$$

$$\forall \mathbf{v} \in (C_0^1([0, 1]))^2$$

Lemma 1.3.37 (fundamental lemma of the calculus of variations).

Let $f \in C_{\text{pw}}^0([a, b])$, $-\infty < a < b < \infty$, satisfy

$$\int_a^b f(\xi)v(\xi) \, d\xi = 0 \quad \forall v \in C^k([a, b]), v(a) = v(b) = 0 .$$

for some $k \in \mathbb{N}_0$. This implies $f \equiv 0$.

$$\text{Ass. (1.3.34) \& (1.3.12)} \xrightarrow{\text{Lemma 1.3.37}} -\frac{d}{d\xi} \left(\frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|} \right) = \mathbf{f}(\xi) \quad 0 \leq \xi \leq 1 .$$

If $\kappa \in C^1$, $\mathbf{f} \in C^0$, then a C^2 -minimizer of J /a C^2 -solution of (1.3.12) solve the 2nd-order ODE

$$-\frac{d}{d\xi} \left(\frac{\kappa(\xi)}{L} (\|\mathbf{u}'\| - L) \frac{\mathbf{u}'}{\|\mathbf{u}'\|} \right) = \mathbf{f} \quad \text{on } [0; 1] . \quad (1.3.38)$$

Summary: policy for obtaining a differential equation from a variational equation

- Use **integration by parts** to remove all derivatives from test functions and shift them onto expressions containing only the trial function.

Thus recast variational equation into the form

$$u: \int T(u) v \, d\mathbf{x} = 0 \quad \forall v .$$

- Appeal to Lemma 1.3.37 to conclude $T(u) = 0$, which yields the differential equation.
- Boundary conditions (here = values of u at endpoints) from the definitions of the trial space.

ODE (1.3.38) + boundary conditions (1.2.1) = two-point boundary value problem
(on domain $\Omega = [0, 1]$)

Minimization problem
(1.2.26)

$$\mathbf{u}_* = \operatorname{argmin}_{\mathbf{v} \in V} J(\mathbf{v})$$

①
 \Leftrightarrow

Variational problem
(1.3.12)

$$\mathbf{a}(\mathbf{u}; \mathbf{v}) = f(\mathbf{v}) \quad \forall \mathbf{v}$$

②
 \Leftrightarrow

Two-point BVP

$$F(\mathbf{u}, \mathbf{u}', \mathbf{u}'') = \mathbf{f} , \\ \mathbf{u}(0), \mathbf{u}(1) \text{ fixed .}$$

①: equivalence (“ \Leftrightarrow ”) holds if minimization problem has unique solution

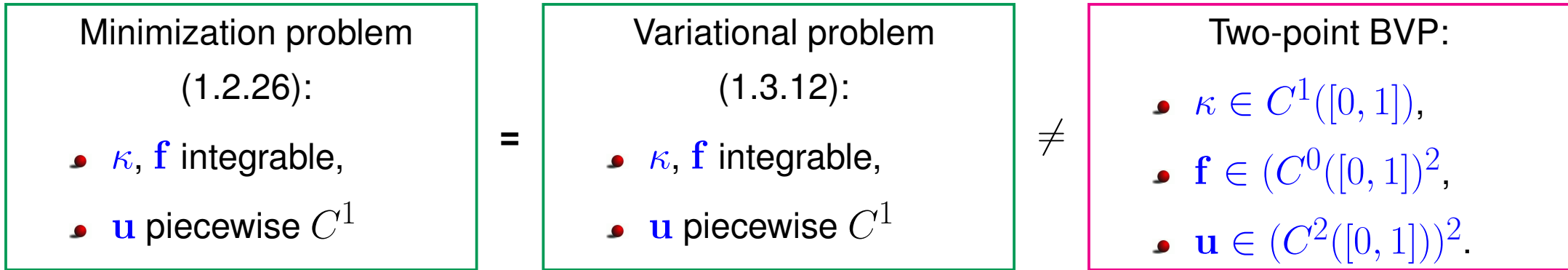
②: meaningful two-point BVP stipulates extra regularity (smoothness) of \mathbf{u} , see Rem. 1.3.39.

Terminology: $\left\{ \begin{array}{l} \text{minimization problem (1.2.26)} \\ \text{variational problem (1.3.12)} \end{array} \right\}$ is called the **weak form** of the string model,

Two-point boundary value problem (1.3.38), (1.2.1) is called the **strong form** of the string model.

A solution \mathbf{u} of (1.3.38), for which all occurring derivatives are continuous is called a **classical solution** of the two-point BVP.

Remark 1.3.39 (Extra regularity requirements). \rightarrow Ex. 1.3.33



\Rightarrow formulation as a classical two-point BVP imposes (unduly) restrictive smoothness on solution and coefficient functions.



Lemma 1.3.40 (Classical solutions are weak solutions).
 For $\kappa \in C^1([0, 1])$, any classical solution of (1.3.38) also solves (1.3.12).

Proof. (“Derivation of (1.3.38) reversed”)

Multiply (1.3.38) with $v \in C_{0,\text{pw}}^1([0, 1])$ and integrate over $[0, 1]$. Then push a derivative onto v by using (1.3.36). \square

1.4 Simplified model

Setting: taut string

$$L \ll \|\mathbf{u}(0) - \mathbf{u}(1)\| . \quad (1.4.1)$$

→ expected: $\|\mathbf{u}'_*(\xi)\| \gg L$ for all $0 \leq \xi \leq 1$ for solution \mathbf{u}_* of (1.2.26)

“Intuitive asymptotics”:

- renormalize stiffness $\kappa \rightarrow \tilde{\kappa} := \frac{\kappa}{L}$, $[\tilde{\kappa}] = \text{Nm}^{-1}$
- suppress equilibrium length: $L = 0$ in (1.2.26).

Simplified equilibrium model:

$$\tilde{\mathbf{u}}_* = \underset{\mathbf{u} \in (C_{\text{pw}}^1([0,1]))^2 \ \& \ (1.2.1)}{\operatorname{argmin}} \underbrace{\int_0^1 \frac{1}{2} \tilde{\kappa}(\xi) \|\mathbf{u}'(\xi)\|^2 - \mathbf{f}(\xi) \cdot \mathbf{u}(\xi) \, d\xi}_{=:\tilde{J}(\mathbf{u})} . \quad (1.4.2)$$

= a **quadratic** minimization problem in a **function space** !

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 1.4.4 (Quadratic minimization problem).

The functional (= mapping from a function space to \mathbb{R}) \tilde{J} from (1.4.2) has the structure

$$\tilde{J}(\mathbf{u}) = \frac{1}{2} \mathbf{a}(\mathbf{u}, \mathbf{u}) - \ell(\mathbf{u}) ,$$

with a symmetric bilinear form $\mathbf{a} : V \times V \mapsto \mathbb{R}$ and a linear form $\ell : V \mapsto \mathbb{R}$.

Minimization problems for functionals of this form have been dubbed **quadratic minimization problems**.



☛ variational problem corresponding to (1.4.2): use

$$\|\mathbf{x} + t\mathbf{h}\|^2 = \|\mathbf{x}\|^2 + 2t\mathbf{x} \cdot \mathbf{h} + t^2 \|\mathbf{h}\|^2 = \|\mathbf{x}\|^2 + 2t\mathbf{x} \cdot \mathbf{h} + O(t^2) .$$

▶
$$\lim_{t \rightarrow 0} \frac{\tilde{J}(\mathbf{u} + t\mathbf{v}) - \tilde{J}(\mathbf{u})}{t} = \int_0^1 \tilde{\kappa}(\xi) \mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = 0, \quad \mathbf{v} \in (C_{\text{pw},0}^1([0, 1]))^2 .$$

Variational equation satisfied by solution $\tilde{\mathbf{u}}_*$ of (1.4.2):

$$\int_0^1 \tilde{\kappa}(\xi) \mathbf{u}'_*(\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = 0 \quad \forall \mathbf{v} \in (C_{\text{pw},0}^1([0, 1]))^2 . \quad (1.4.5)$$

Remark 1.4.6 (Linear variational problems). \rightarrow Rem. 1.3.21

(1.4.5) has the structure (1.3.24)

$$u \in V: \quad \mathbf{a}(u, v) = \ell(v) \quad \forall v \in V_0, \quad (1.4.7)$$

where now

- $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ is a **bilinear form** (\rightarrow Def. 1.3.23), that is, linear in *both* arguments.

In general, *quadratic* minimization problems give rise to *linear* variational problems.

This can be confirmed by an elementary computation:

$$\begin{aligned} J(\mathbf{u}) &= \frac{1}{2}\mathbf{a}(\mathbf{u}, \mathbf{u}) - \ell(\mathbf{u}) \\ \blacktriangleright \lim_{t \rightarrow 0} \frac{J(\mathbf{u} + t\mathbf{v}) - J(\mathbf{u})}{t} &= \lim_{t \rightarrow 0} \frac{t\mathbf{a}(\mathbf{u}, \mathbf{v}) + \frac{1}{2}t^2\mathbf{a}(\mathbf{v}, \mathbf{v}) - t\ell(\mathbf{v})}{t} = \mathbf{a}(\mathbf{u}, \mathbf{v}) - \ell(\mathbf{v}), \end{aligned}$$

where the bilinearity of \mathbf{a} and the linearity of ℓ was crucial, see Rem. 1.4.4.



Corresponding two-point boundary value problem: by integration by parts, see (1.3.36),

$$\int_0^1 \tilde{\kappa}(\xi) \mathbf{u}'_* (\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = \int_0^1 \left\{ -\frac{d}{d\xi} \left(\tilde{\kappa}(\xi) \frac{d}{d\xi} \mathbf{u}(\xi) \right) - \mathbf{f}(\xi) \right\} \cdot \mathbf{v}(\xi) \, d\xi$$

$$\forall \mathbf{v} \in (C_{\text{pw},0}^1([0, 1]))^2 .$$

Then use Lemma 1.3.37.

If $\kappa \in C^1$, $f \in C^0$, then a C^2 -solution of (1.4.5) solves the two-point BVP

$$-\frac{d}{d\xi} \left(\tilde{\kappa}(\xi) \frac{d\mathbf{u}}{d\xi}(\xi) \right) = \mathbf{f}(\xi), \quad 0 \leq \xi \leq 1,$$

$$\mathbf{u}(0) = \begin{pmatrix} a \\ u_a \end{pmatrix}, \quad \mathbf{u}(1) = \begin{pmatrix} b \\ u_b \end{pmatrix} . \tag{1.4.8}$$

Special setting:

“gravitational force” $\mathbf{f}(\xi) = -g(\xi)\mathbf{e}_2$

(1.4.2) decouples into two minimization problems for the components of \mathbf{u} !

$$(1.4.2) \Rightarrow \begin{aligned} \tilde{u}_{1,*} &= \operatorname{argmin}_{u \in C_{\text{pw}}^1([0,1]), u(0)=a, u(1)=b} \frac{1}{2} \int_0^1 \tilde{\kappa}(\xi) (u'(\xi))^2 \, d\xi, \\ \tilde{u}_{2,*} &= \operatorname{argmin}_{u \in C_{\text{pw}}^1([0,1]), u(0)=u_a, u(1)=u_b} \int_0^1 \frac{1}{2} \tilde{\kappa}(\xi) (u'(\xi))^2 + g(\xi) u(\xi) \, d\xi. \end{aligned} \quad (1.4.9)$$

The minimization problem for $\tilde{u}_{1,*}$ has a closed-form solution:

$$\tilde{u}_{1,*}(\xi) = a + \frac{b-a}{\int_0^1 \tilde{\kappa}^{-1}(\tau) \, d\tau} \int_0^\xi \tilde{\kappa}^{-1}(\tau) \, d\tau, \quad 0 \leq \xi \leq 1. \quad (1.4.10)$$

 R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

This solution can easily be found by converting the minimization problem to a 2-point boundary value problem as was done above, *cf.* (1.4.5), (1.4.8).

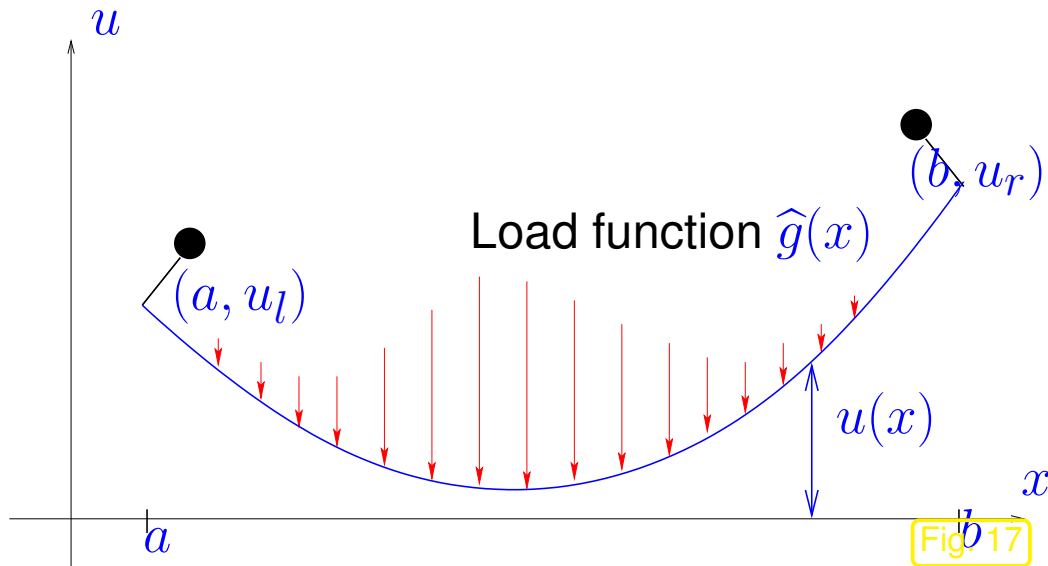
The minimization problem for $\tilde{u}_{2,*}$ leads to the *linear* variational problem, *cf.* (1.4.5)

$$\begin{aligned} \tilde{u}_{2,*} &\in C_{\text{pw}}^1([0,1]) \\ \tilde{u}_{2,*}(0) &= u_a, \quad \tilde{u}_{2,*}(1) = u_b \end{aligned} : \int_0^1 \tilde{\kappa}(\xi) \tilde{u}'_{2,*}(\xi) v'(\xi) \, d\xi = - \int_0^1 g(\xi) v(\xi) \, d\xi \quad \forall v \in C_{0,\text{pw}}^1([0,1]).$$

(1.4.11)

Remark 1.4.12 (Graph description of string shape).

Focus: situation with vertical gravitational force, see (1.4.10), (1.4.11)



Describe shape of string through **graph** of displacement function $\hat{u}_* = \hat{u}_*(x)$, $\hat{u} : [a, b] \mapsto \mathbb{R}$ (physical units $[\hat{u}] = 1\text{m}$).

► **boundary conditions:**

$$u_*(a) = u_a \quad , \quad u_*(b) = u_b \quad . \quad (1.4.13)$$

►
$$\hat{u}(x) = \tilde{u}_{2,*}(\Phi^{-1}(x)) \quad \text{with} \quad \Phi(\xi) := \tilde{u}_{1,*}(\xi) \quad . \quad (1.4.14)$$

Here $\tilde{u}_{1,*}(\xi)$, $\tilde{u}_{2,*}(\xi)$ are the components of the curve description of the equilibrium shape of the string, see Sect. 1.2.1:

$$\mathbf{u}_*(\xi) = \begin{pmatrix} \tilde{u}_{1,*}(\xi) \\ \tilde{u}_{2,*}(\xi) \end{pmatrix} , \quad 0 \leq \xi \leq 1 \quad .$$

Of course, the graph description is possible only for special string shapes. It also hinges on the choice of suitable coordinates.


Note: $\xi \mapsto \Phi(\xi)$ is monotone, $\Phi'(\xi) \neq 0$ for all $0 \leq \xi \leq 1$, $\Phi(0) = a$, $\Phi(1) = b$.

By chain rule [32, Thm. 5.1.3]:

$$v(\xi) = \widehat{v}(\Phi(\xi)) \quad \Rightarrow \quad v'(\xi) = \frac{d\widehat{v}}{dx}(x)\Phi'(\xi), \quad x := \Phi(\xi). \quad (1.4.17)$$

Recall: transformation formula for integrals in one dimension (substitution rule, $x := \Phi(\xi)$, “ $dx = \Phi'(\xi)d\xi$ ”):

$$q \in C_{\text{pw}}^0([0, 1]): \quad \int_0^1 q(\xi) \, d\xi = \int_{a=\Phi(0)}^{b=\Phi(1)} \widehat{q}(x) \left| \frac{1}{\Phi'(\Phi^{-1}(x))} \right| dx, \quad \widehat{q}(x) := q(\Phi^{-1}(x)). \quad (1.4.18)$$


← (1.4.17) & (1.4.18)

Variational problem of taut string problem described as a function of spatial coordinate:

$$\begin{aligned}
 \int_0^1 \tilde{\kappa}(\xi) \tilde{u}'_{2,*}(\xi) v'(\xi) \, d\xi &= \int_a^b \tilde{\kappa}(\Phi^{-1}(x)) \Phi'(\xi) \frac{d\hat{u}}{dx}(x) \Phi'(\xi) \frac{d\hat{v}}{dx}(x) \frac{1}{|\Phi'(\xi)|} \, dx \\
 &= \int_a^b \underbrace{\tilde{\kappa}(\Phi^{-1}(x)) |\Phi'(\Phi^{-1}(x))|}_{=:\hat{\sigma}(x)} \frac{d\hat{u}}{dx}(x) \frac{d\hat{v}}{dx}(x) \, dx, \\
 - \int_0^1 g(\xi) v(\xi) \, d\xi &= - \int_a^b \underbrace{\frac{g(\Phi^{-1}(x))}{|\Phi'(\Phi^{-1}(x))|}}_{=:\hat{g}(x), [\hat{g}] = \text{Nm}^{-1}} \hat{v}(x) \, dx.
 \end{aligned}$$

► *Linear* variational problem in physical space coordinate on spatial domain $\Omega = [a, b]$:

$$\hat{u}_* \in C_{\text{pw}}^1([a, b]), \quad \hat{u}_*(a) = u_a, \quad \hat{u}_*(b) = u_b \quad : \quad \int_a^b \hat{\sigma}(x) \frac{d\hat{u}_*}{dx}(x) \frac{d\hat{v}}{dx}(x) \, dx = - \int_a^b \hat{g}(x) \hat{v}(x) \, dx \quad \forall \hat{v} \in C_{0,\text{pw}}^1([a, b]).$$

(1.4.19)

(assuming $\hat{\sigma} \in C^1([a, b])$) Two-point BVP

$$(1.4.19) \Rightarrow \begin{cases} \frac{d}{dx} \left(\hat{\sigma}(x) \frac{d\hat{u}_*}{dx}(x) \right) = \hat{g}(x), & a \leq x \leq b, \\ \hat{u}_*(a) = u_a, & \hat{u}_*(b) = u_b. \end{cases} \quad (1.4.20)$$



1.5 Discretization

Goal: “computation” of a/the solution $\mathbf{u} : [0, 1] \mapsto \mathbb{R}^2$ of $\begin{cases} \text{minimization problem (1.2.26)} \\ \text{variational problem (1.3.12)} \\ \text{two-point BVP (1.3.38) \& (1.2.1)} \end{cases}$

a function: infinite amount of information, see [21, Rem. 3.1.3].

! Well, just provide a *formula* for u (**analytic solution**):



in general elusive for the above problems

Only option:

Numerical algorithm

Computer →

approximate solution

Finitely many floating point operations

Numerical algorithms can only operate on discrete models

Continuous (PDE) model
("∞-dimensional")

Discretization →

Discrete model
("finitely many unknowns")

as small as possible
(only a few unknowns)

as accurate as possible
(good approximation (*))

as faithful as as possible
(structure preserving)

(*): needs a measure for quality of a solution, usually a **norm** of the **error**, error = difference of exact/analytic and approximate solution.

Remark 1.5.2 (“Physics based” discretization).

Mass-spring model (\rightarrow Sect. 1.2.2) = discretization of the minimization problem (1.2.26) describing the elastic string.

This discretization may be called “physics based”, because it is inspired by the (physical) context of the model.

Note: Other approaches to discretization discussed below will lead to equations resembling the mass-spring model, see 1.5.1.2.



This section will present a few strategies on how to derive discrete models for the problem of computing the shape of an elastic string. The different approaches start from different formulations, some target the minimization problem (1.2.26), or, equivalently, the variational problem (1.3.12), while others tackle the ODE (1.3.38) together with the boundary conditions (1.2.1).

Remark 1.5.4 (Timestepping for ODEs).

For initial value problems for ODEs, whose solutions are functions, too, we also face the problem of discretization: timestepping methods compute a finite number of approximate values of the solutions at discrete instances in time, see [21, Ch. 12].



Remark 1.5.6 (Coefficients/data in procedural form).

For the elastic string model (\rightarrow Sect. 1.2.3) the stiffness $\kappa(\xi)$, and force field \mathbf{f} may not be available in closed form (as formulas).

Instead they are usually given in **procedural form**:

```
function k = kappa(xi);  
function f = force(xi);
```

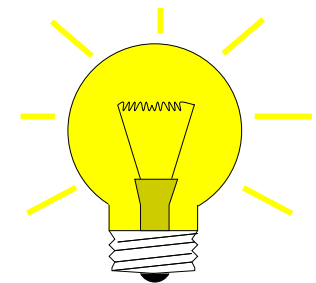
because they may be obtained

- as results of another computation,
- by interpolation from a table.

▶ viable discretizations must be able to deal with data in procedural form!



1.5.1 Ritz-Galerkin discretization



Simple idea of first step of **Ritz-Galerkin discretization**

In $\left\{ \begin{array}{l} \text{minimization problem, e.g., (1.2.26)} \\ \text{variational problem, e.g. (1.3.12)} \end{array} \right.$

↕

replace function space V/V_0 with
finite dimensional subspace $V_N/V_{N,0}$

Note that a subscript tag N distinguishes “discrete functions/quantities”, that is, functions/operators etc. that are associated with a finite dimensional space. In some contexts, N will also be an integer designating the dimension of a finite dimensional space.

Formal presentation: V, V_0 : (affine) function spaces, $\dim V_0 = \infty$,
 $V_N, V_{N,0}$: subspaces $V_N \subset V, V_{N,0} \subset V_0, N := \dim V_{N,0}, \dim V_N < \infty$.

Ritz-Galerkin discretization of minimization problem for functional $J : V \mapsto \mathbb{R}$:

Continuous minimization problem

$$u = \operatorname{argmin}_{v \in V} J(v) . \quad (1.5.7)$$

Galerkin disc. \rightarrow

Discrete minimization problem

$$u_N = \operatorname{argmin}_{v_N \in V_N} J(v_N) . \quad (1.5.8)$$

Ritz-Galerkin discretization of abstract (non-linear) variational problem (1.3.24), see Rem. 1.3.21

Continuous variational problem

$$u \in V: \mathbf{a}(u; v) = \ell(v) \quad \forall v \in V_0. \quad (1.5.9)$$

Galerkin disc. \rightarrow

Discrete variational problem

$$u_N \in V_N: \mathbf{a}(u_N; v_N) = \ell(v_N) \quad \forall v_N \in V_{N,0}. \quad (1.5.10)$$

Terminology: $u_N \in V_N$ satisfying (1.5.8)/(1.5.10) is called a **Galerkin solution** of (9.2.21)/(9.3.62)

V_N is called the **(Galerkin) trial space**, $V_{N,0}$ is the **(Galerkin) test space**.

Remark 1.5.13 (Relationship between discrete minimization problem and discrete variational problem).

In Sect. 1.3.1 we discovered the **equivalence**

Continuous minimization problem
(9.2.21)

\iff

Continuous variational problem
(9.3.62)

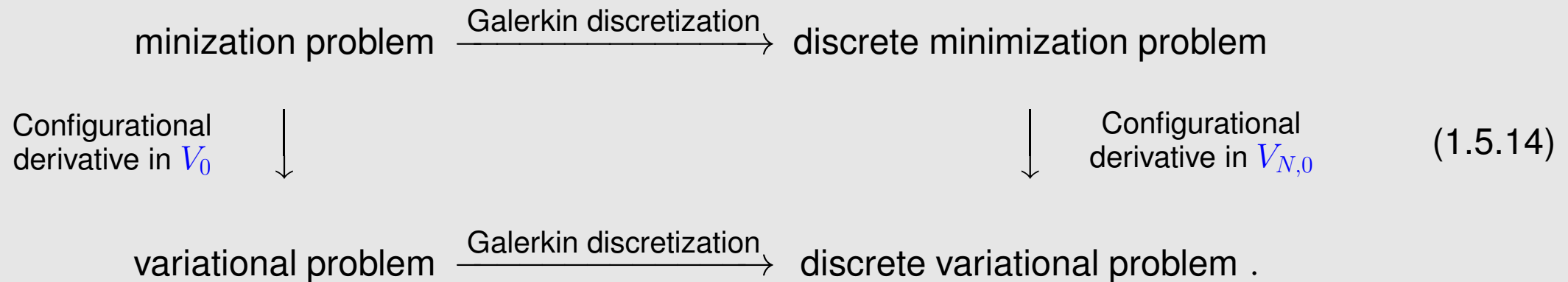
Now it seems that we have *two different* strategies for Galerkin discretization:

1. Ritz-Galerkin discretization via the discrete minimization problem (1.5.8),
2. Ritz-Galerkin discretization based on the discrete variational problem (1.5.10).

However,

the above equivalence extends to the discrete problems!

More precisely, we have the *commuting relationship*:



The commuting diagram means that the same discrete variational problem is obtained no matter whether

1. the minimization problem is first restricted to a finite dimensional subspace and the result is converted into a variational problem according to the recipe of Sect. 1.3.1.

2. or whether the variational problem derived from the minimization problem is restricted to the subspace.

To see this, understand that the manipulations of Sect. 1.3.1 can be carried out for infinite and finite dimensional function spaces alike.



Remark 1.5.15 (Offset functions and Ritz-Galerkin discretization).

Often: $V = u_0 + V_0$, with offset function $u_0 \rightarrow$ Rem. 1.3.32, (1.3.29)

If u_0 is sufficiently simple, we may choose a trial space $V_N = u_0 + V_{N,0}$

➤ Discrete variational problem analogous to (1.3.32)

$$w_N \in V_{N,0}: \quad \mathbf{a}(u_0 + w_N; v_N) = \ell(v_N) \quad \forall v_N \in V_{N,0} \quad \mapsto \quad u_N := w_N + u_0 . \quad (1.5.18)$$

In the case of a linear variational problem (\rightarrow Rem. 1.4.6), that is, a bilinear form \mathbf{a} , we have

$$(1.5.18) \quad \Leftrightarrow \quad \mathbf{a}(w_N, v_N) = \ell(v_N) - \mathbf{a}(u_0, v_N) \quad \forall v_N \in V_{N,0} . \quad (1.5.19)$$

Below we will always make the assumption

$$V = u_0 + V_0.$$



However, a computer is clueless about a concept like “finite dimensional subspace”. What it can process are arrays of floating point numbers.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

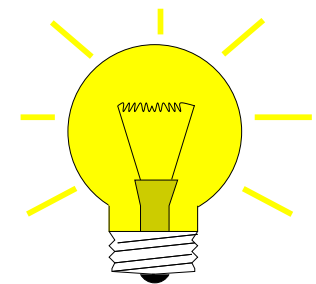
- Idea:
- choose **basis** $\mathfrak{B}_N = \{b_N^1, \dots, b_N^N\}$ of $V_{N,0}$: $V_{N,0} = \text{Span} \{\mathfrak{B}_N\}$
 - insert basis representation into minimization problem (1.5.8)

$$v_N \in V_{N,0} \Rightarrow v_N = u_0 + \nu_1 b_N^1 + \dots + \nu_N b_N^N, \quad \nu_i \in \mathbb{R}, \quad (1.5.20)$$

and variational equation (1.5.10)

$$v_N \in V_{N,0} \Rightarrow v_N = \nu_1 b_N^1 + \dots + \nu_N b_N^N, \quad \nu_i \in \mathbb{R}, \quad (1.5.21)$$

$$u_N \in V_N \Rightarrow u_N = u_0 + \mu_1 b_N^1 + \dots + \mu_N b_N^N, \quad \mu_i \in \mathbb{R}. \quad (1.5.22)$$



Remark 1.5.24 (Ordered basis of test space).

Once we have chosen a basis \mathfrak{B} and **ordered** it, as already indicated in the notation above, the test space $V_{N,0}$ can be identified with \mathbb{R}^N : a coefficient vector $\vec{\mu} = (\mu_1, \dots, \mu_N)^T \in \mathbb{R}^N$ provides a *unique* characterization of a function $u \in V_{N,0}$ (basis property)

$$u = \sum_{j=1}^N \mu_j b_N^j .$$



Discrete minimization problem

$$u_N = \operatorname{argmin}_{v_N \in V} J(v_N) . \quad (1.5.8)$$

Basis
representation \rightarrow

Minimization problem on \mathbb{R}^N

$$\vec{\mu} = \operatorname{argmin}_{\vec{\nu} \in \mathbb{R}^N} F(\vec{\nu}) , \quad (1.5.25)$$

$$F(\vec{\nu}) := J(u_0 + \nu_1 b_N^1 + \dots + \nu_N b_N^N) .$$

amenable to classical optimization
techniques

notation: $\vec{\nu}, \vec{\mu} \hat{=}$ vectors of coefficients $(\nu_i)_{i=1}^N, (\mu_i)_{i=1}^N$, in basis representation of functions $v_N, u_N \in V_N$ according to (1.5.20).

Discrete variational problem

$$u_N \in V_N: \mathbf{a}(u_N; v_N) = \ell(v_N) \\ \forall v_N \in V_{N,0} . \quad (1.5.10)$$

Basis
→
representation

System of equations

$$\mathbf{a}(u_0 + \sum_{j=1}^N \mu_j b_N^j; b_N^k) = \ell(b_N^k) \\ \forall k = 1, \dots, N . \quad (1.5.26)$$

use techniques for linear/non-linear
systems of equations, see [21, Ch. 2],
[21, Ch. 4].

Note that we owe the above equivalence of the discrete variational problem (left) and the system of equations (right) to the fact that $\mathbf{a}(u; v)$ is linear in its second argument, see (3.7.1), because

$$\ell \text{ linear form on } V_{N,0} \Rightarrow \left\{ \ell(b_N^j) = 0 \forall j \Leftrightarrow \ell(v_N) = 0 \forall v_N \in V_{N,0} \right\},$$

and $v \mapsto \mathbf{a}(u; v)$ is a linear form, see Def. 1.3.23.

The choice of the basis \mathfrak{B} has no impact on the (set of) Galerkin solutions of (1.5.10)!

Below, we apply Galerkin approaches to

- (1.4.19) as an example for the treatment of a *linear* variational problem:

$$\begin{aligned} & u \in C_{\text{pw}}^1([a, b]), \\ & u(a) = u_a, \quad u(b) = u_b \end{aligned} : \int_a^b \sigma(x) \frac{du}{dx}(x) \frac{dv}{dx}(x) dx = - \int_a^b g(x) v(x) dx \quad \forall v \in C_{0,\text{pw}}^1([a, b]).$$

(1.4.19)

Here:

spatial domain $\Omega = [a, b]$, linear offset function $u_0(x) = \frac{b-x}{b-a}u_a + \frac{x-a}{b-a}u_b$,

function space $V_0 = C_{0,\text{pw}}^1([a, b])$.

- (1.3.12) to demonstrate its use in the case of a non-linear variational equation:

$$\begin{aligned} & \mathbf{u} \in C_{\text{pw}}^1([0, 1]) \\ & \mathbf{u}(0), \mathbf{u}(1) \text{ from (1.2.1)} \end{aligned} : \int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = 0$$

$$\forall \mathbf{v} \in (C_{0,\text{pw}}^1([0, 1]))^2. \quad (1.3.12)$$

Here: parameter domain $\Omega = [0, 1]$, linear offset function $\mathbf{u}_0(\xi) = \xi \mathbf{u}(0) + (1 - \xi) \mathbf{u}(1)$,
function space $V_0 = (C_{0,\text{pw}}^1([a, b]))^2$.

1.5.1.1 Spectral Galerkin scheme

A simple function space (widely used for interpolation, see [21, Ch. 3], and approximation, see [21, Sec. 9.1]): for interval $\Omega \subset \mathbb{R}$

$$V_{N,0} = \mathcal{P}_p(\mathbb{R}) \cap C_0^0(\Omega)$$

$\hat{=}$ space of univariate **polynomials** of degree $\leq p$ vanishing at endpoints of Ω ,

(1.5.27)

$$N := \dim V_N = p - 1$$

[21, Sect. 3.2] for more information.

Obvious: choice (1.5.27) guarantees $V_N \subset C_{\text{pw},0}^1(\Omega)$ (even $V_{N,0} \subset C^\infty(\Omega)$)

Please note that $V_{N,0}$ is a space of *global* polynomials on Ω .

Example 1.5.28 (Spectral Galerkin discretization of linear variational problem).

Targetted: linear variational problem (1.4.19) with

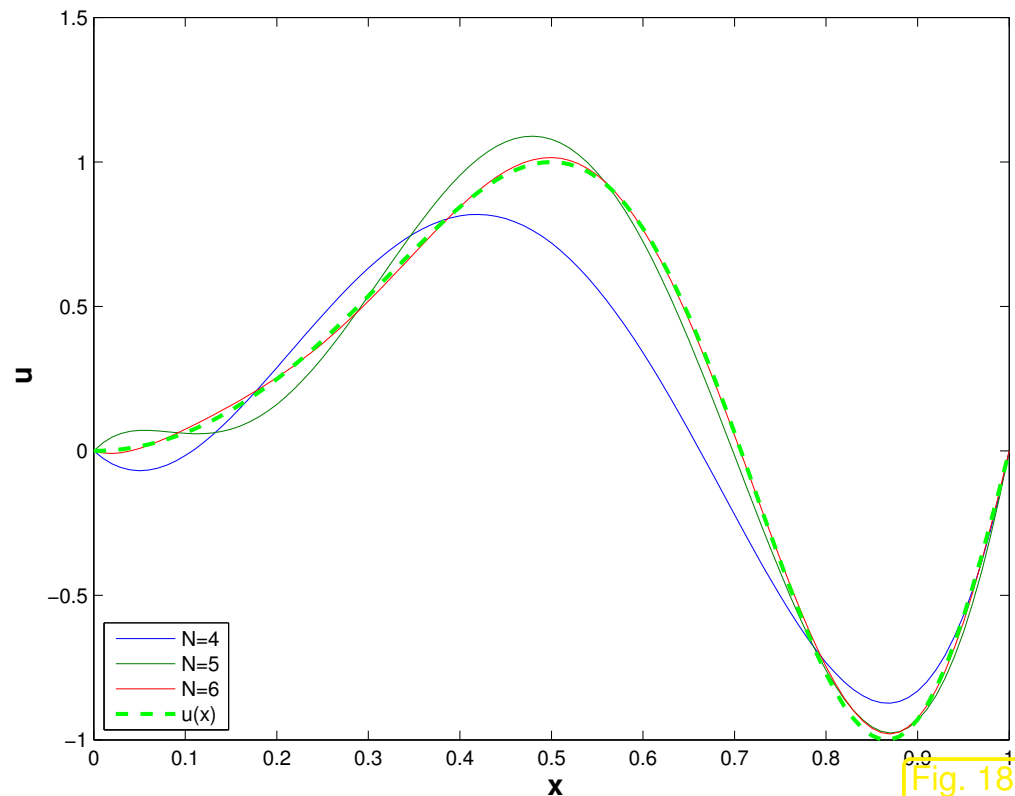
- $a = 0, b = 1 \quad \triangleright \quad \text{domain } \Omega =]0, 1[$,
 - constant coefficient function $\sigma \equiv 1$,
 - load $g(x) = -4\pi(\cos(2\pi x^2) - 4\pi x^2 \sin(2\pi x^2))$,
 - boundary values $u_a = u_b = 0$.
- $\blacktriangleright \quad u(x) = \sin(2\pi x^2), \quad 0 < x < 1.$
- because $\frac{d^2 u}{dx^2}(x) = g(x)$.

\blacktriangleright Concrete variational problem

$$u \in C_{0,\text{pw}}^1([0, 1]): \quad \int_0^1 \frac{du}{dx}(x) \frac{dv}{dx}(x) \, dx = - \int_0^1 g(x)v(x) \, dx \quad \forall v \in C_{0,\text{pw}}^1([0, 1]). \quad (1.5.29)$$

Polynomial spectral Galerkin discretization, degree $p \in \{4, 5, 6\}$.

Plots of approximate/exact solutions

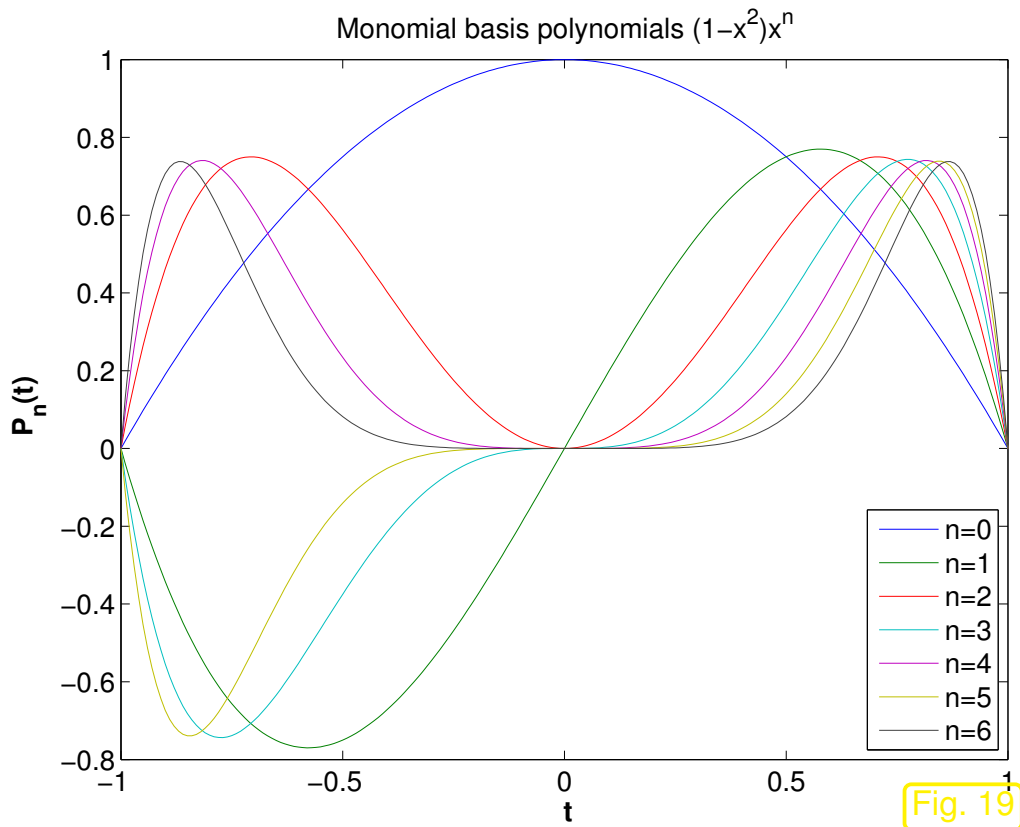


Remark 1.5.30 (Choice of basis for polynomial spectral Galerkin methods).

Sought: (ordered) basis of $V_{N,0} := C_0^1([-1, 1]) \cap \mathcal{P}_p(\mathbb{R})$

❶ “Tempting”: monomial-type basis

$$V_{N,0} = \text{Span} \left\{ 1 - x^2, x(1 - x^2), x^2(1 - x^2), \dots, x^{p-2}(1 - x^2) \right\}. \quad (1.5.31)$$



◁ Monomial basis polynomials

Beware: ill-conditioned !

→ Ex. 1.5.68 below

② “Popular”: **integrated Legendre polynomials**

$$V_{N,0} = \text{Span} \left\{ x \mapsto M_n(x) := \int_{-1}^x P_n(\tau) d\tau, n = 1, \dots, p - 1 \right\}, \quad (1.5.32)$$

where $P_n \hat{=}$ n -th Legendre polynomial.

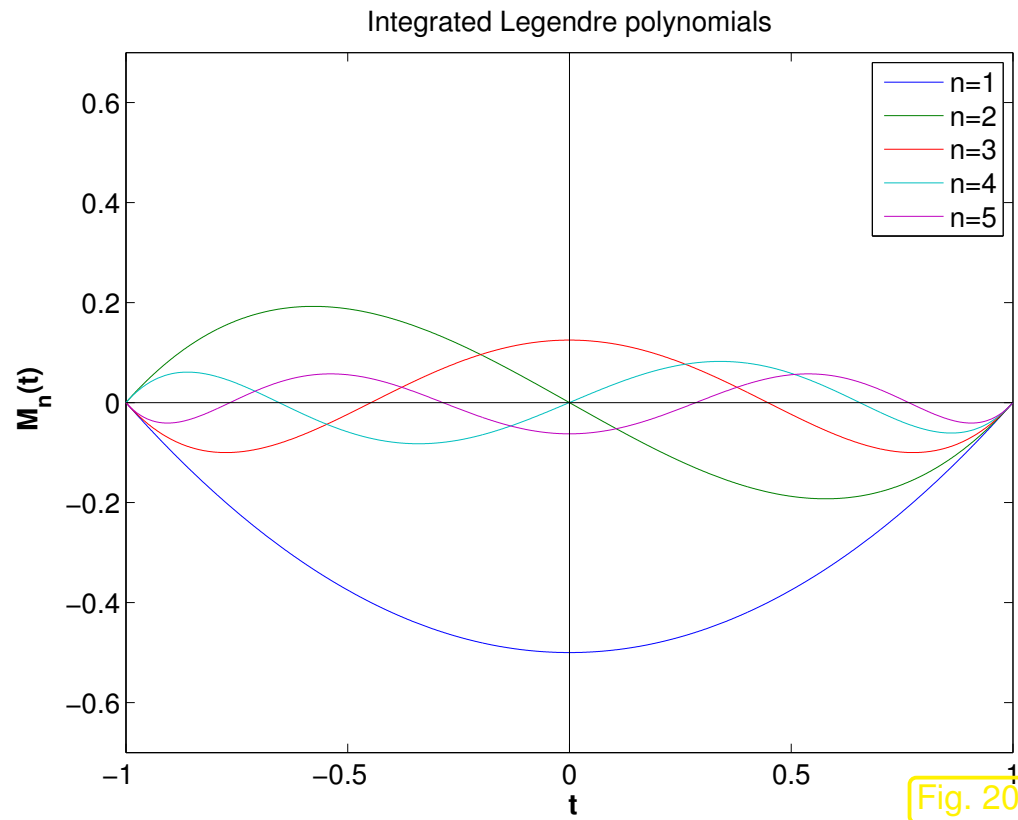


Fig. 20

◁ integrated Legendre polynomials M_1, \dots, M_5

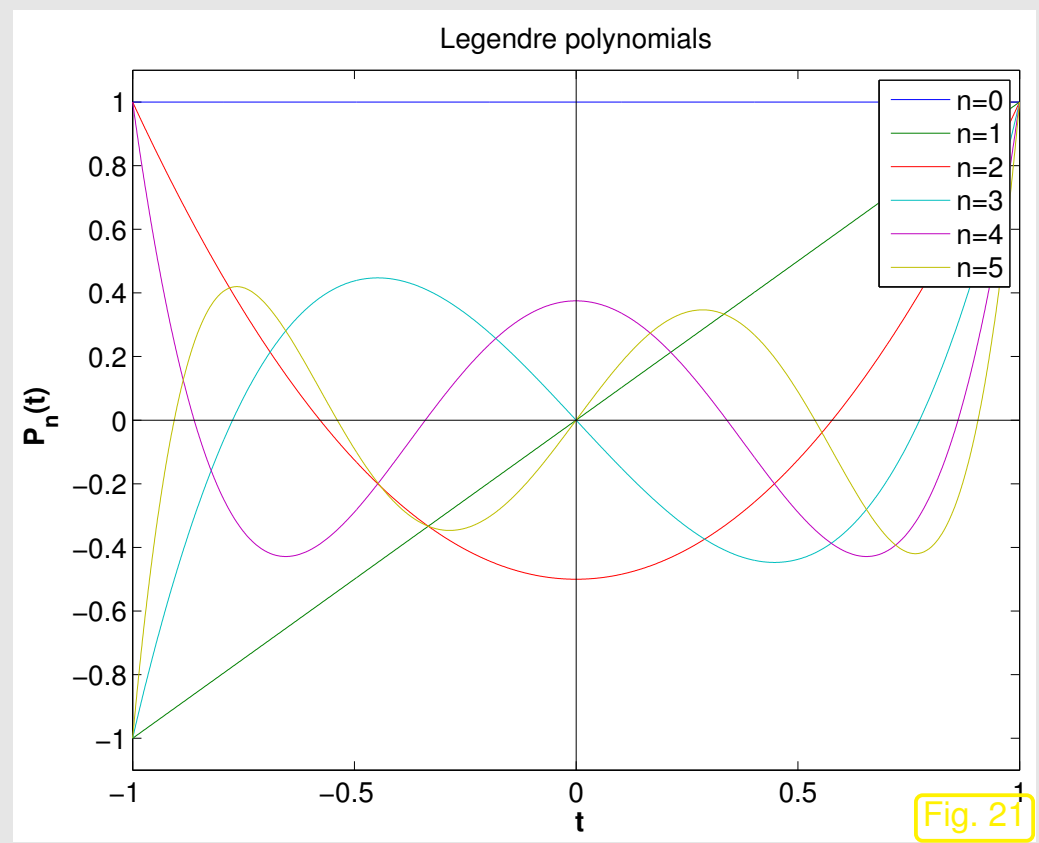
Definition 1.5.40 (Legendre polynomials). \rightarrow [21, Def. 10.4.12]

The n -th Legendre polynomial P_n , $n \in \mathbb{N}_0$, is defined by (Rodriguez formula)

$$P_n(x) := \frac{1}{n!2^n} \frac{d^n}{dx^n} [(x^2 - 1)^n] .$$

Legendre polynomials P_0, \dots, P_5

$$\begin{aligned}
 P_0(x) &= 1, \\
 P_1(x) &= x, \\
 P_2(x) &= \frac{3}{2}x^2 - \frac{1}{2}, \\
 P_3(x) &= \frac{5}{2}x^3 - \frac{3}{2}x, \\
 P_4(x) &= \frac{35}{8}x^4 - \frac{15}{4}x^2 + \frac{3}{8}.
 \end{aligned}$$



Some facts about Legendre polynomials:

- Symmetry:

$$P_n \text{ is } \begin{cases} \text{even} \\ \text{odd} \end{cases} \text{ for } \begin{cases} \text{even } n \\ \text{odd } n \end{cases}, \quad P_n(1) = 1, \quad P_n(-1) = (-1)^n. \tag{1.5.41}$$

- Orthogonality

$$\int_{-1}^1 P_n(x)P_m(x)dx = \begin{cases} \frac{2}{2n+1} & \text{, if } m = n, \\ 0 & \text{else.} \end{cases} \tag{1.5.42}$$

3-term recursion

$$P_{n+1}(x) := \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x) \quad , \quad P_0 := 1 \quad , \quad P_1(x) := x \quad . \quad (1.5.43)$$

This formula paves the way for the efficient evaluation of all Legendre polynomials at many (quadrature) points, see Code 1.5.43.

Code 1.5.44: Computation of Legendre polynomials based on 3-term recursion (1.5.43)

```

1 function V= legendre (n, x)
2 % Computes values of Legendre polynomials up to degree n
3 % in the points xj passed in the row vector x.
4 % Exploits the 3-term recursion (1.5.43) for Legendre polynomials
5 V = ones(size(x)); V = [V; x];
6 for j=1:n-1
7     V = [V; ((2*j+1)/(j+1)) .* x .* V(end, :) - j/(j+1) * V(end-1, :)] ;
8 end

```

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Representation of derivatives and primitives, cf. Code 1.5.46:

$$P_n(x) = \left(\frac{d}{dx} P_{n+1}(x) - \frac{d}{dx} P_{n-1}(x) \right) / (2n+1) \quad , \quad n \in \mathbb{N} \quad , \quad (1.5.45)$$

$$\blacktriangleright \quad M_n(x) = \frac{1}{2n+1} (P_{n+1}(x) - P_{n-1}(x)) \quad \text{and} \quad \frac{dM_n}{dx} = P_n \quad . \quad (1.5.46)$$

Code 1.5.47: Computation of (integrated) Legendre polynomials using (1.5.43) and (1.5.46)

```

1 function [V,M] = intlegpol(n,x)
2 % Computes values of the first n+1 Legendre polynomials P_n (returned in
3 % matrix V) and the first n-1 integrated Legendre polynomials
4 % M_n (returned in matrix M) in the points x_j passed in the
5 % row vector x. Uses the recursion formulas (1.5.43) and
6 % (1.5.46)
7 V = ones(size(x)); V = [V; x];
8 for j=1:n-1, V = [V; ((2*j+1)/(j+1)).*x.*V(end, :) -
   j/(j+1)*V(end-1, :)]; end
9 M = diag(1./(2*(1:n-1)+1)) * (V(3:n+1, :) - V(1:n-1, :));

```

R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

Remark 1.5.48 (Transformation of basis functions).

On a “general domain $\Omega = [a, b]$ ”, we obtain the basis function by a so-called affine transformation of the basis functions on $[-1, 1]$, cf. [21, Rem. 10.1.3], e.g., in the case of integrated Legendre polynomials as basis functions on $\Omega = [a, b]$ we use the basis functions

$$b_N^i(x) = M_i \left(2 \frac{x - a}{b - a} - 1 \right), \quad a \leq x \leq b. \tag{1.5.51}$$

Note the effect of this transformation on the derivative (chain rule!):

$$\frac{db_N^i}{dx}(x) = \frac{dM_i}{dx} \left(2 \frac{x-a}{b-a} - 1 \right) \cdot \frac{2}{b-a} = P_i \left(2 \frac{x-a}{b-a} - 1 \right) \cdot \frac{2}{b-a}. \quad (1.5.52)$$



Remark 1.5.53 (Spectral Galerkin discretization with quadrature).

Consider the linear variational problem, *cf.* (1.4.19),

$$u \in C_{0,\text{pw}}^1([a, b]): \int_a^b \sigma(x) \frac{du}{dx}(x) \frac{dv}{dx}(x) dx = \int_a^b g(x)v(x) dx \quad \forall v \in C_{0,\text{pw}}^1([a, b]). \quad (1.5.54)$$

Assume: σ, g only given in procedural form, see Rem. 1.5.6.

► Analytic evaluation of integrals becomes impossible even if u, v polynomials !

Only remaining option:

Numerical quadrature, see [21, Ch. 10]

► Replace integral with m -point **quadrature formula** on $[a, b]$, $m \in \mathbb{N} \rightarrow$ [21, Sect. 10.1]:

$$\int_a^b f(t) dt \approx Q_n(f) := \sum_{j=1}^m \omega_j^m f(\zeta_j^m). \quad (1.5.58)$$

$$\omega_j^n: \text{ quadrature weights } , \quad \zeta_j^n: \text{ quadrature nodes } \in [a, b]. \quad (1.5.59)$$

(1.5.54) ► discrete variational problem with quadrature:

$$u_N \in V_N: \quad \sum_{j=1}^m \omega_j^m \sigma(\zeta_j^m) \frac{du_N}{dx}(\zeta_j^m) \frac{dv_N}{dx}(\zeta_j^m) = \sum_{j=1}^m \omega_j^m g(\zeta_j^m) v(\zeta_j^m) \quad \forall v \in V_N. \quad (1.5.60)$$

Popular (global) quadrature formulas: **Gauss quadrature** \rightarrow [21, Sect. 10.4]

Important: Accuracy of quadrature formula and computational cost (no. m of quadrature nodes) have to be balanced.

Remark 1.5.61 (Implementation of spectral Galerkin discretization for linear 2nd-order two-point BVP).

Setting:

- linear variational problem (1.5.54) $\triangleright u_0 = 0$,
- coefficients σ, g in procedural form, see Rem. 1.5.6,
- approximation of integrals by p -point Gaussian quadrature formula,
- polynomial spectral Galerkin discretization, degree $\leq p, p \geq 2$,
- basis \mathcal{B} : integrated Legendre polynomials, see (1.5.32):

$$V_{N,0} = \text{Span} \{ M_n, n = 1, \dots, p-1 \}, \quad M_n \hat{=} \text{integrated Legendre polynomials} .$$

Trial expression, *cf.* (1.5.20)

$$u_N = \mu_1 M_1 + \mu_2 M_2 + \dots + \mu_N M_N, \quad \mu_i \in \mathbb{R}, \quad N := p-1 .$$

Note: by definition $\frac{d}{dx}M_n = P_n$.

From (1.5.60) with (1.5.61)

$$\sum_{j=1}^m \omega_j^m \sigma(\zeta_j^m) \sum_{l=1}^N \mu_l P_l(\zeta_j^m) P_k(\zeta_j^m) = \underbrace{\sum_{j=1}^m \omega_j^m g(\zeta_j^m) M_k(\zeta_j^m)}_{=:\varphi_k}, \quad k = 1, \dots, N. \quad (1.5.62)$$

$$\sum_{l=1}^N \left(\sum_{j=1}^m \omega_j^m \sigma(\zeta_j^m) P_l(\zeta_j^m) P_k(\zeta_j^m) \right) \mu_l = \varphi_k, \quad k = 1, \dots, N. \quad (1.5.63)$$

$$\boxed{\mathbf{A} \vec{\mu} = \vec{\varphi}} \quad \text{with} \quad (\mathbf{A})_{kl} := \sum_{j=1}^m \omega_j^m \sigma(\zeta_j^m) P_l(\zeta_j^m) P_k(\zeta_j^m), \quad k, l = 1, \dots, N, \quad (1.5.64)$$

$$\vec{\mu} = (\mu_l)_{l=1}^N \in \mathbb{R}^N, \quad \vec{\varphi} = (\varphi_k)_{k=1}^N \in \mathbb{R}^N.$$

A linear system of equations !

The Galerkin discretization of a *linear* variational problem leads to a *linear* system of equations.

Code 1.5.65: Polynomial spectral Galerkin solution of (1.5.54)

```

1 function u = lin2pbvpspecgalquad(sigma, g, N, x)
2 % Polynomial spectral Galerkin discretization of linear 2nd-order two-point BVP
3 %  $-\frac{d}{dx}(\sigma(x)\frac{du}{dx}) = g(x)$ ,  $u(0) = u(1) = 0$  on  $\Omega = [0,1]$ . Trial space of dimension  $N$ .
4 % Values of approximate solution in points  $x_j$  are returned in the row vector u
5 m = N+1; % Number of quadrature nodes
6 [zeta, w] = gaussquad(m); % Obtain Gauss quadrature nodes w.r.t [-1,1]
7 % Compute values of (integrated) Legendre polynomials at Gauss nodes
8 [V, M] = intlegpol(N+1, zeta');
9 % Note that the 2-point boundary value problem is posed on [0,1], which entails
10 % transforming the quadrature rule to this interval, achieved by the following
11 % transformation, see [21, Rem. 10.1.3] and the related Remark 1.5.48.
12 zeta = (zeta'+1)/2;
13 omega = w' .* sigma(zeta) * 2; % Modified quadrature weights
14 A = V(2:N+1, :) * diag(omega) * V(2:N+1, :)'; % Assemble Galerkin matrix
15 phi = M * (0.5 * w' .* g(zeta)'); % Assemble right hand side vector
16 mu = A \ phi; % Solve linear system
17 % Compute values of integrated Legendre polynomials at output points
18 [V, M] = intlegpol(N+1, 2*x-1); u = mu' * M;

```

Code 1.5.67: MATLAB driver script creating plots of Ex. 1.5.28

```

1 % MATLAB script: Driver routine for polynomial spectral Galerkin
  discretization of linear 2nd-order
2 % two-point boundary value problem
3 clear all;
4 % Coefficient functions (function handles, see MATLAB help)
5 sigma = @(x) ones(size(x));
6 g =      @(x) -4*pi*(cos(2*pi*x.^2) - 4*pi*x.^2.*sin(2*pi*x.^2));
7 x = 0:0.01:1; % Evaluation points
8 % Computation with trial space of dimension 4,5,6
9 N = 4; U = [lin2pbvpspecgalquad(sigma,g,N,x); ...
10            lin2pbvpspecgalquad(sigma,g,N+1,x); ...
11            lin2pbvpspecgalquad(sigma,g,N+2,x)];
12 % Graphical output
13 figure('name','Polynomial spectral Galerkin');
14 plot(x,U); hold on;
15 plot(x,sin(2*pi*x.^2),'g--','linewidth',2);
16 xlabel('{\bf x}','fontsize',14);
17 ylabel('{\bf u}','fontsize',14);
18 legend('N=4','N=5','N=6','u(x)','location','southwest');
19 print -depsc2 '../.../Slides/NPDEpics/specgallinsol.eps';

```

Example 1.5.68 (Conditioning of spectral Galerkin system matrices).

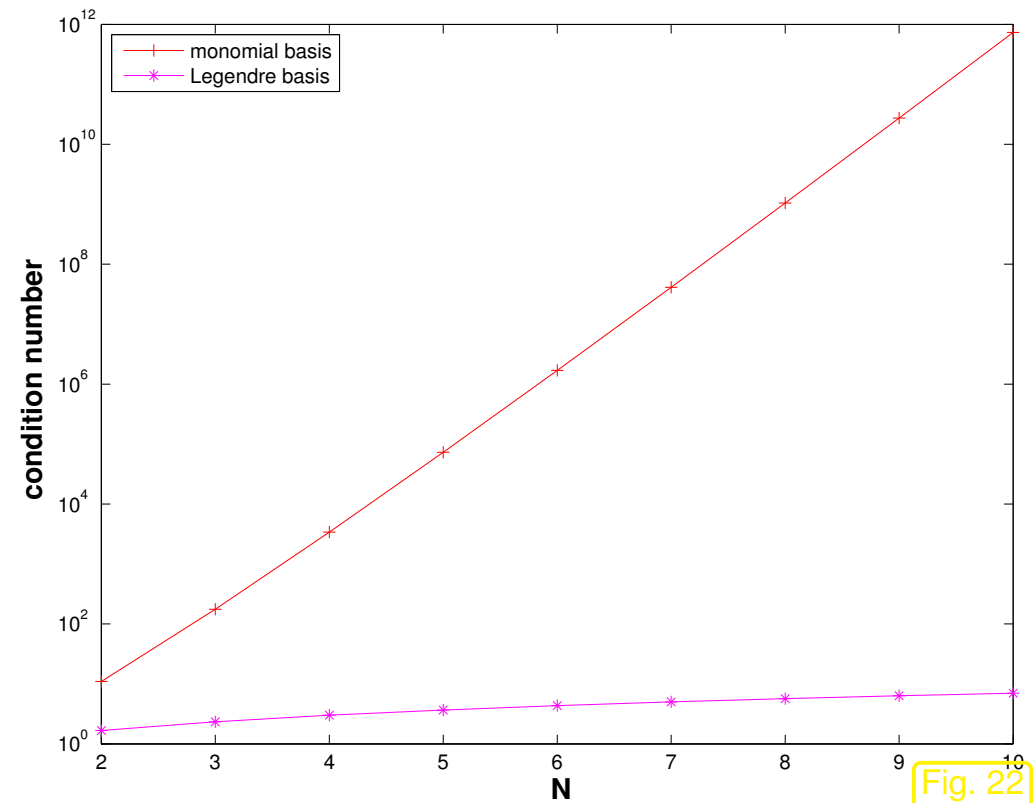
Finally we can provide a rationale for preferring integrated Legendre polynomials to plain monomials for polynomial spectral Galerkin discretization: the argument is based on condition number of the system matrix from (1.5.64).

- Linear variational problem (1.5.29) with bilinear form

$$\mathbf{a}(u, v) = \int_0^1 \frac{du}{dx}(x) \frac{dv}{dx}(x) \, dx, \quad u, v \in C_{0,\text{pw}}^1([0, 1]).$$

- Choice of basis functions for Galerkin trial/test space $V_{N,0} := \mathcal{P}_p(\mathbb{R}) \cap C_0^0([0, 1])$: monomial basis (1.5.31), integrated Legendre polynomials (1.5.32).

Monitored: condition number (w.r.t. Euclidean matrix norm \rightarrow [21, Def. 2.5.26]) of Galerkin matrices



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Recall that a condition number of 10^m involves a loss of m digits w.r.t. the precision guaranteed for the right hand side of the linear system. Thus, using the monomial basis for $N > 10$ may no longer produce reliable results.



Example 1.5.69 (Implementation of spectral Galerkin discretization for elastic string problem).

Targetted: *non-linear* variational equation on domain $\Omega = [0, 1]$

$$\int_0^1 \frac{\kappa(\xi)}{L} \left(1 - \frac{L}{\|\mathbf{u}'(\xi)\|} \right) \mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = 0 \quad \forall \mathbf{v} \in (C_{0,\text{pw}}^1([0, 1]))^2. \quad (1.3.12)$$

- Data κ , \mathbf{f} given in procedural form, see Rem. 1.5.6.
- Spectral Galerkin discretization of “curve space” $(C_{0,\text{pw}}^1([0, 1]))^2$: *component-wise discretization*

Use basis $\mathfrak{B} = \{M_n\}_{n=1}^K$, $K \in \mathbb{N}$, of integrated Legendre polynomials, see (1.5.32) \triangleright basis representation, *cf.* (1.5.61)

$$\mathbf{u}_N(\xi) = \underbrace{\mathbf{u}(0)(1 - \xi) + \mathbf{u}(1)\xi}_{=: \mathbf{u}_0(\xi) \text{ (offset function)}} + \binom{\mu_1}{\mu_{K+1}} M_1(\xi) + \cdots + \binom{\mu_K}{\mu_{2K}} M_K(\xi). \quad (1.5.72)$$

- Approximate evaluation of integrals by m -point Gaussian quadrature on $[0, 1]$, $m := K + 1$ below: nodes ζ_j , weights ω_j , $j = 1, \dots, m$.

In analogy to (1.5.62) we arrive at the *non-linear* system of equations: $(M'_k = P_k!)$

$$\sum_{j=1}^m s_j (b - a + \sum_{l=1}^K \mu_l P_l(\zeta_j)) \cdot P_k(\zeta_j) = \sum_{j=1}^m \omega_j f_1(\zeta_j) \cdot M_k(\zeta_j), \quad k = 1, \dots, K,$$

$$\sum_{j=1}^m s_j (u_b - u_a + \sum_{l=1}^K \mu_{K+l} P_l(\zeta_j)) \cdot P_k(\zeta_j) = \sum_{j=1}^m \omega_j f_2(\zeta_j) \cdot M_k(\zeta_j), \quad k = 1, \dots, K,$$

with $s_j := \omega_j \kappa(\zeta_j) \left(\frac{1}{L} - \frac{1}{\|\mathbf{u}'_N(\zeta_j)\|} \right)$ ($s_j = s_j(\vec{\mu})$!).

$\hat{=} N := 2K$ equations for N unknowns μ_1, \dots, μ_N ; rewrite in compact form:

$$\begin{pmatrix} \mathbf{R}(\vec{\mu}) & 0 \\ 0 & \mathbf{R}(\vec{\mu}) \end{pmatrix} \vec{\mu} = \begin{pmatrix} \vec{\varphi}_1(\vec{\mu}) \\ \vec{\varphi}_2(\vec{\mu}) \end{pmatrix}, \quad (1.5.73)$$

with $\mathbf{R}(\vec{\mu}) \in \mathbb{R}^{K,K}$, $(\mathbf{R}(\vec{\mu}))_{k,l} := \sum_{j=1}^m s_j(\vec{\mu}) P_l(\zeta_j) P_k(\zeta_j)$,

$$(\vec{\varphi}_1(\vec{\mu}))_k = \sum_{j=1}^m \omega_j f_1(\zeta_j) \cdot M_k(\zeta_j) - (b - a) \sum_{j=1}^m s_j(\vec{\mu}) P_k(\zeta_j),$$

$$(\vec{\varphi}_2(\vec{\mu}))_k = \sum_{j=1}^m \omega_j f_2(\zeta_j) \cdot M_k(\zeta_j) - (u_b - u_a) \sum_{j=1}^m s_j(\vec{\mu}) P_k(\zeta_j).$$

Iterative solution of (1.5.73) by **fixed point iteration**:

Initial guess $\vec{\mu}^{(0)} \in \mathbb{R}^N$; $k = 0$;

repeat

$k \leftarrow k + 1$;

Solve the *linear* system of equations

$$\begin{pmatrix} \mathbf{R}(\vec{\mu}^{(k-1)}) & 0 \\ 0 & \mathbf{R}(\vec{\mu}^{(k-1)}) \end{pmatrix} \vec{\mu}^{(k)} = \begin{pmatrix} \vec{\varphi}_1(\vec{\mu}^{(k-1)}) \\ \vec{\varphi}_2(\vec{\mu}^{(k-1)}) \end{pmatrix};$$

until $\|\vec{\mu}^{(k)} - \vec{\mu}^{(k-1)}\| \leq \text{tol} \cdot \|\vec{\mu}^{(k)}\|$

Code 1.5.74: Polynomial spectral Galerkin discretization of elastic string variational problem

```

1 function [vu,figsol] = stringspecgal(kappa,f,L,u0,u1,K,xi,tol)
2 % Solving the non-linear variational problem (1.3.12) for the elastic string by
  % means of polynomial
3 % spectral Galerkin discretization based on K integrated Legendre polynomials.
  % Approximate
4 % evaluation of integrals by means of Gaussian quadrature.
5 % kappa, f are handles of type @(xi) providing the coefficient function
6 % kappa and the force field f. The column vectors u0 and u1 pass the
7 % pinning points. M is the number of mesh cells, tol specifies the tolerance
  % for the
8 % fixed point iteration. return value: 2 x length(xi)-matrix of node
9 % positions for curve parameter values passed in the row vector xi.
10 if (nargin < 8), tol = 1E-2; end

```

```
11 m = K+1; % Number of quadrature nodes
12 [zeta,w] = gaussquad(m); % Obtain Gauss quadrature nodes w.r.t [-1,1]
13 % Compute values of (integrated) Legendre polynomials at Gauss nodes and
    evaluation points
14 [V,M] = intlegpol(K+1,zeta');
15 [Vx,Mx] = intlegpol(K+1,2*xi-1); Mx = [1-xi;Mx;xi]; %
16 % Compute right hand side based on m-point Gaussian quadrature on [0,1].
17 force = f((zeta'+1)/2); phi = M*(0.5*[w';w'].*force)';
18 sv = kappa((zeta'+1)/2); % Values of coefficient function  $\kappa$  at Gauss
    points in [0,1].
19 % mu is an  $2 \times (K+2)$ -matrix, containing the vectorial basis expansion
    coefficients
20 % of  $\mathbf{u}_N$ . The first and last column are contributions of the two functions
21 %  $\xi \mapsto (1-\xi)$  and  $\xi \mapsto \xi$ , which represent the offset function.
22 % Initial guess for fixed point iteration: straight string
23 mu = [u0, zeros(2,K), u1];
24 figsol = figure; hold on;
25 for k=1:100 % loop for fixed point iteration, maximum 100 iterations
26 % Plot shape of string
27 vu = mu*Mx; plot(vu(1,:),vu(2,:), '--g'); drawnow;
28 title(sprintf('K = %d, iteration #%d',K,k));
29 xlabel('\bf x_1'); ylabel('\bf x_2');
30 % Compute values of derivatives of  $\mathbf{u}_N$  and  $\|\mathbf{u}'_N\|$  at Gauss points
31 up = mu(:,2:K)*V(2:K,:) + repmat(u1-u0,1,m);
32 lup = sqrt(up(1,:).^2 + up(2,:).^2);
```

```

33 s = 0.5*(w') .* sv .* (1/L - 1./lup); % Initialization of s_j
34 % Modification of right hand side due to offset function
35 phi1 = phi(:,1) + (2*(u1(1)-u0(1))*V(2:K+1,:) * s');
36 phi2 = phi(:,2) + (2*(u1(2)-u0(2))*V(2:K+1,:) * s');
37 % Assemble K x K-matrix blocks R of linear system
38 R = 4*V(2:K+1,:) * diag(s) * V(2:K+1,:)' ;
39 mu_new = [u0, [(R\phi1)'; (R\phi2)'], u1];
40 % Check simple termination criterion for fixed point iteration.
41 if (norm(mu_new - mu, 'fro') < tol*norm(mu_new, 'fro') / K)
42     vu = mu*Mx; fig = plot(vu(1,:), vu(2,:), 'r--');
43     legend(fig, 'spectral Galerkin
44         solution', 'location', 'southeast'); break; end
45 mu = mu_new;
end

```

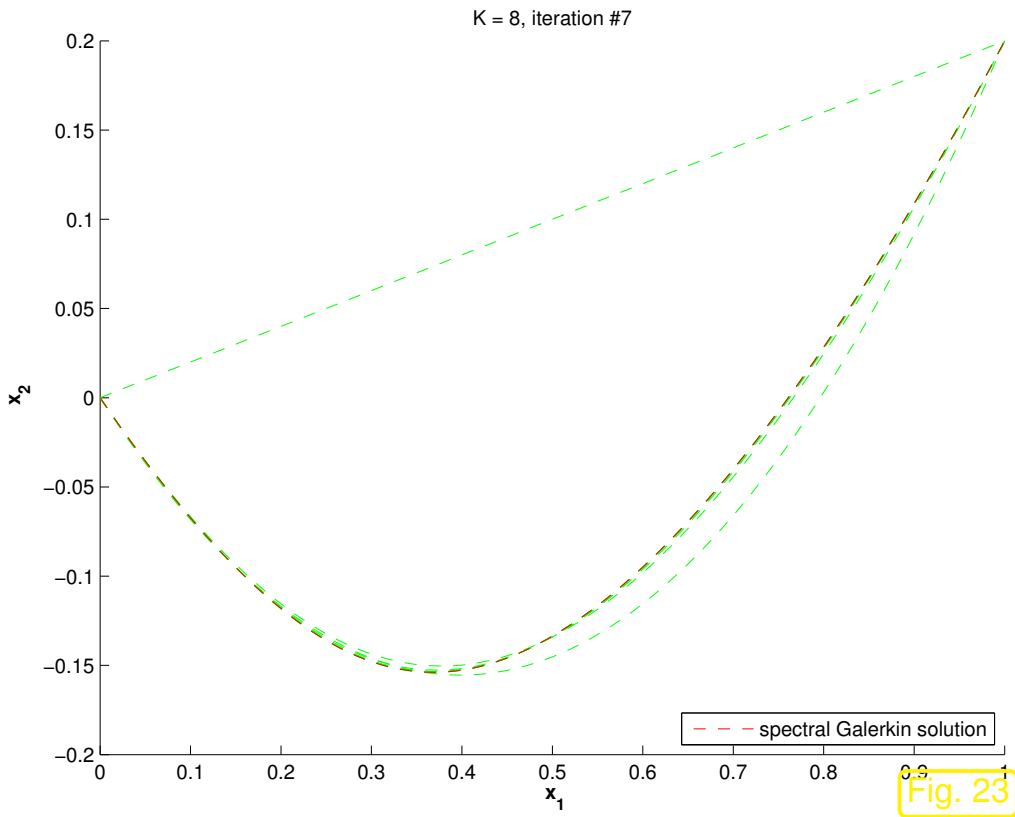


Example 1.5.75 (Spectral Galerkin discretization for elastic string simulation).

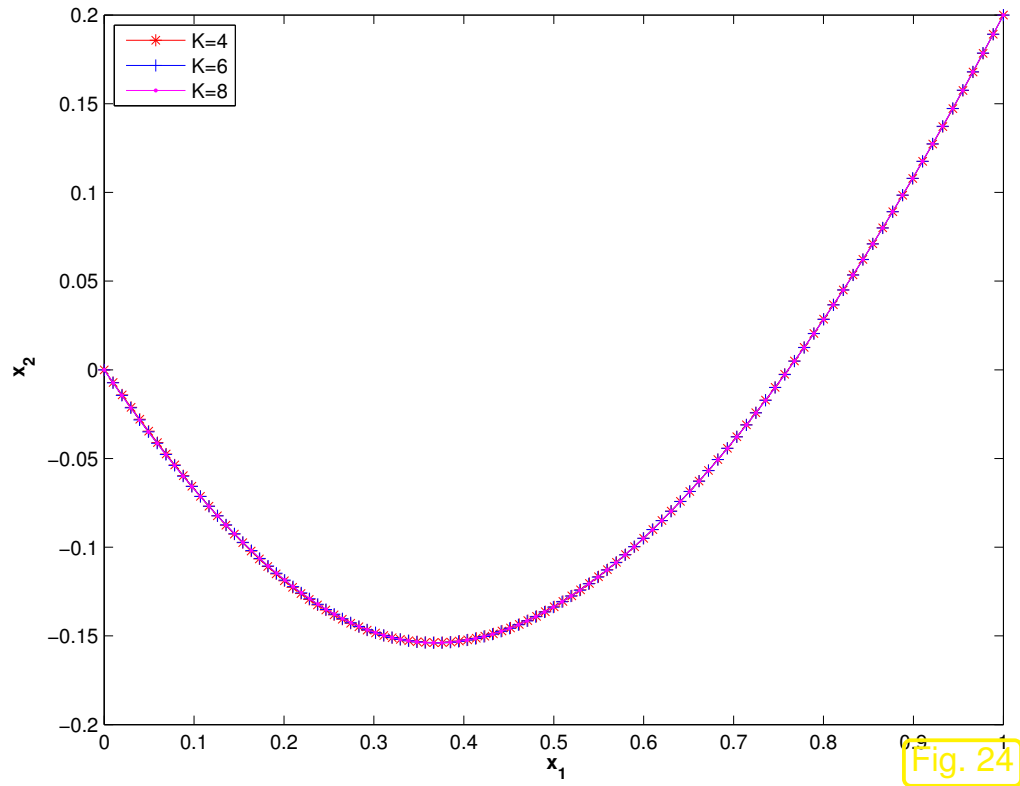
Test of polynomial spectral Galerkin method for elastic string problem, algorithm of Ex. 1.5.69, Code 1.5. with

- pinning positions $\mathbf{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\mathbf{u}(1) = \begin{pmatrix} 1 \\ 0.2 \end{pmatrix}$,

- equilibrium length $L = 0.5$,
- constant coefficient function $\kappa \equiv 1N$,
- gravitational force field $\mathbf{f}(\xi) = -\begin{pmatrix} 0 \\ 2 \end{pmatrix}$.



iteration history



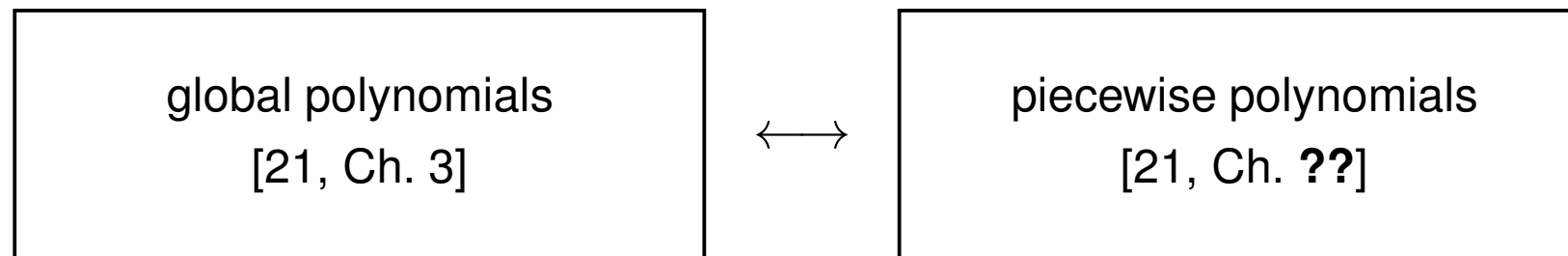
solutions \mathbf{u}_N for different resolutions

Observation: “Visual convergence” as polynomial degree is increased.



1.5.1.2 Linear finite elements

Two ways to approximate functions by polynomials:

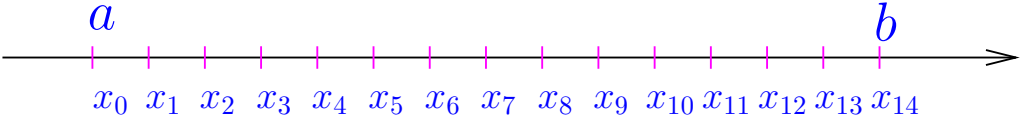


The spectral polynomial Galerkin approach presented in Sect. 1.5.1.1 relies on global polynomials. Now let us examine the use of *piecewise polynomials*.

Preliminaries: piecewise polynomials have to be defined w.r.t. partitioning of the domain $\Omega \subset \mathbb{R}$

➤ $\Omega = [a, b]$ equipped with **nodes** ($M \in \mathbb{N}$)

$\mathcal{X} := \{a = x_0 < x_1 < \dots < x_{M-1} < x_M = b\}$.



➤ **mesh/grid**

$$\mathcal{M} := \{]x_{j-1}, x_j[: 1 \leq j \leq M\} .$$

Special case:

equidistant mesh: $x_j := a + jh$, $h := \frac{b-a}{M}$.

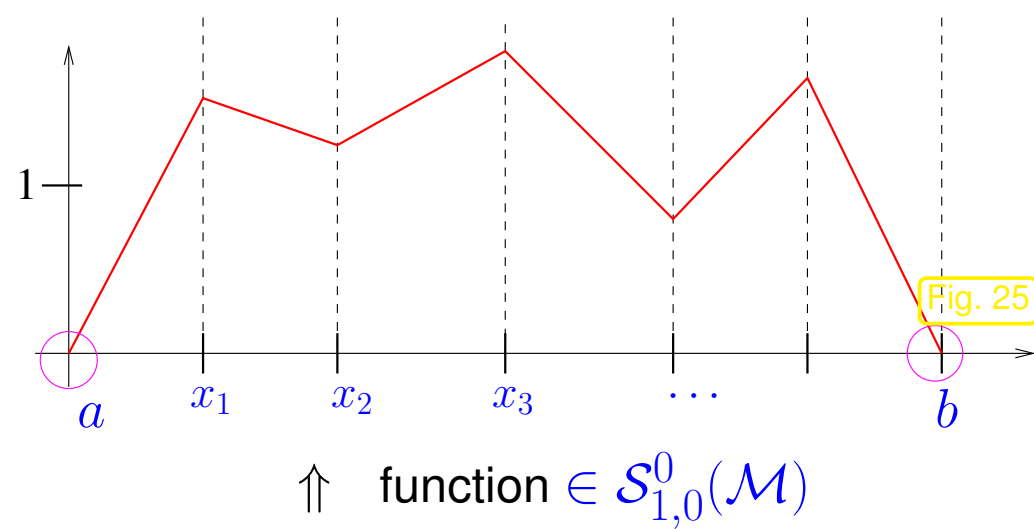
☞ $]x_{j-1}, x_j]$, $j = 1, \dots, M$, $\hat{=}$ **cells** of \mathcal{M} , **cell size** $h_j := |x_j - x_{j-1}|$, $j = 1, \dots, M$
meshwidth $h_{\mathcal{M}} := \max_j |x_j - x_{j-1}|$

Recall from Sect. 1.3.2: merely continuous, piecewise C^1 trial and test functions provide valid trial/test functions!

Simplest choice for test space

$$V_N = \mathcal{S}_{1,0}^0(\mathcal{M}) := \left\{ v \in C^0([0, 1]) : v|_{[x_{i-1}, x_i]} \text{ linear, } \right. \\ \left. i = 1, \dots, M, v(a) = v(b) = 0 \right\}$$

$$N := \dim V_N = M - 1$$

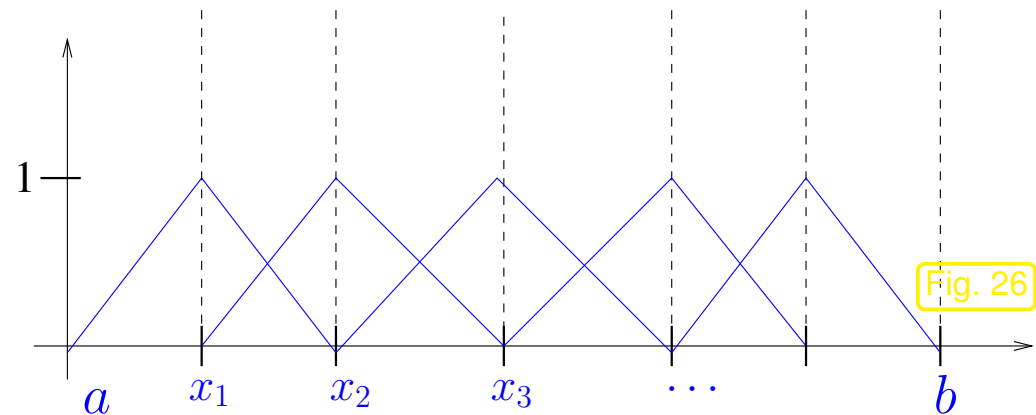


Choice of (ordered) basis \mathfrak{B}_N of V_N ?

1D “tent functions”

$$\mathfrak{B} = \{b_N^1, \dots, b_N^{M-1}\}, \quad (1.5.76)$$

$$b_N^j(x_i) = \delta_{ij} := \begin{cases} 1 & , \text{ if } i = j, \\ 0 & , \text{ if } i \neq j, \end{cases} \quad (1.5.77)$$



$$\blacktriangleright \frac{db_N^j}{dx}(x) = \begin{cases} \frac{1}{h_j} & , \text{ if } x_{j-1} \leq x \leq x_j, \\ -\frac{1}{h_{j+1}} & , \text{ if } x_j < x \leq x_{j+1}, \\ 0 & \text{elsewhere.} \end{cases} \quad (\text{piecewise derivative!}) \quad (1.5.79)$$

Remark 1.5.81 (Benefit of variational formulation of BVPs).

The possibility of using simple piecewise linear trial and test functions is a clear benefit of the variational formulation that can accommodate merely piecewise continuously differentiable functions, see Sect. 1.3.2.

Below, in Sect. 1.5.2 we will learn about a method that targets the strong form of the 2-point BVP and, thus, has to impose more regularity on the trial functions.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs



SAM, ETHZ

❶ simplest case: linear variational problem with constant stiffness coefficient

$$u \in C_{0,\text{pw}}^1([a, b]): \int_a^b \frac{du}{dx}(x) \frac{dv}{dx}(x) dx = \int_a^b g(x)v(x) dx \quad \forall v \in C_{0,\text{pw}}^1([a, b]) .$$

Discrete variational problem with $u_N = \mu_1 b_N^1 + \dots + \mu_N b_N^N$:

$$\int_a^b \sum_{l=1}^N \mu_l \frac{db_N^l}{dx}(x) \frac{db_N^k}{dx}(x) dx = \int_a^b g(x) b_N^k(x) dx \quad k = 1, \dots, N .$$

$$\sum_{l=1}^N \left(\int_a^b \frac{db_N^l}{dx}(x) \frac{db_N^k}{dx}(x) dx \right) \mu_l = \underbrace{\int_a^b g(x) b_N^k(x) dx}_{=:\varphi_k}, k = 1, \dots, N.$$



$\mathbf{A}\vec{\mu} = \vec{\varphi}$ with $(\mathbf{A})_{kl} := \int_a^b \frac{db_N^l}{dx}(x) \frac{db_N^k}{dx}(x) dx, k, l = 1, \dots, N,$
 $\vec{\mu} = (\mu_l)_{l=1}^N \in \mathbb{R}^N, \vec{\varphi} = (\varphi_k)_{k=1}^N \in \mathbb{R}^N.$

A linear system of equations, cf. Rem. 1.5.61!

▷ system matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{M-1, M-1}, a_{ij} := \int_a^b \frac{db_N^i}{dx}(x) \frac{db_N^j}{dx}(x) dx, 1 \leq i, j \leq N$

piecewise derivatives

▷ r.h.s. vector $\vec{\varphi} \in \mathbb{R}^{M-1}, \varphi_k := \int_a^b g(x) b_N^k(x) dx, k = 1, \dots, N.$

The detailed computations start with the evident fact that

$$|i - j| \geq 2 \quad \Rightarrow \quad \frac{b_N^j}{dx}(x) \cdot \frac{b_N^i}{dx}(x) = 0 \quad \forall x \in [a, b],$$

because there is *no overlap* of the *supports* of the two basis functions.

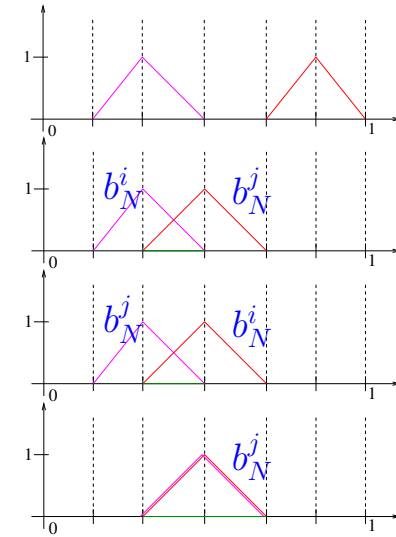
Definition 1.5.83 (Support of a function).

The *support* of a function $f : \Omega \mapsto \mathbb{R}$ is defined as

$$\text{supp}(f) := \overline{\{\mathbf{x} \in \Omega : f(\mathbf{x}) \neq 0\}}.$$

In addition, we use that the gradients of the tent functions are piecewise constant, see (1.5.79).

$$\int_0^1 \frac{db_N^j}{dx}(x) \frac{db_N^i}{dx}(x) dx = \begin{cases} 0 & , \text{if } |i - j| \geq 2 \\ -\frac{1}{h_{i+1}} & , \text{if } j = i + 1 \\ -\frac{1}{h_i} & , \text{if } j = i - 1 \\ \frac{1}{h_i} + \frac{1}{h_{i+1}} & , \text{if } 1 \leq i = j \leq M - 1 \end{cases} \rightarrow$$



A symmetric, positive definite and tridiagonal:

$$\mathbf{A} = \begin{pmatrix} \frac{1}{h_1} + \frac{1}{h_2} & -\frac{1}{h_2} & 0 & & 0 \\ -\frac{1}{h_2} & \frac{1}{h_2} + \frac{1}{h_3} & -\frac{1}{h_3} & & \\ 0 & \dots & \dots & \dots & \\ & & \dots & \dots & 0 \\ & & & \dots & -\frac{1}{h_{M-1}} \\ 0 & & 0 & -\frac{1}{h_{M-1}} & \frac{1}{h_{M-1}} + \frac{1}{h_M} \end{pmatrix} \quad (1.5.84)$$

notation: $h_j := |x_j - x_{j-1}|$ local meshwidth, cell size

Natural choice: piecewise polynomial trial/test spaces \longleftrightarrow composite quadrature rule

e.g, composite trapezoidal rule:
$$\varphi_k = \int_0^1 g(x) b_N^k(x) dx \approx \frac{1}{2}(h_k + h_{k+1})g(x_k), \quad 1 \leq k \leq N .$$
 (1.5.85)

For equidistant mesh with uniform cell size $h > 0$ we arrive at the linear system of equations:

$$\frac{1}{h} \begin{pmatrix} 2 & -1 & 0 & & & & 0 \\ -1 & 2 & -1 & & & & \\ 0 & \cdots & \cdots & \cdots & & & \\ & & & \cdots & \cdots & \cdots & 0 \\ & & & & -1 & 2 & -1 \\ 0 & & & & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_N \end{pmatrix} = h \begin{pmatrix} g(x_1) \\ \vdots \\ g(x_N) \end{pmatrix} .$$
 (1.5.86)

 R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

② case: linear variational problem with variable stiffness, cf. (1.4.19)

$$u \in C_{0,\text{pw}}^1([a, b]): \int_a^b \sigma(x) \frac{du}{dx}(x) \frac{dv}{dx}(x) dx = \int_a^b g(x)v(x) dx \quad \forall v \in C_{0,\text{pw}}^1([a, b]) .$$

Discrete variational problem with $u_N = \mu_1 b_N^1 + \dots + \mu_N b_N^N$:

$$\int_a^b \sigma(x) \sum_{l=1}^N \mu_l \frac{db_N^l}{dx}(x) \frac{db_N^k}{dx}(x) dx = \int_a^b g(x) b_N^k(x) dx \quad k = 1, \dots, N. \quad (1.5.87)$$

Here: numerical quadrature required for both integrals

Choice: • composite midpoint rule for left hand side integral \rightarrow [21, Sect. 10.3]

$$\int_a^b f(x) dx \approx \sum_{j=1}^M h_j f(m_j), \quad m_j := \frac{1}{2}(x_j + x_{j-1}). \quad (1.5.88)$$

• composite trapezoidal rule [21, Eq. 10.3.3] for right hand side integral, see (1.5.85).

Assumption: $\sigma \in C_{\text{pw}}^0([a, b])$ with jumps *only* at grid nodes x_j

(1.5.87)



$$\sum_{l=1}^N \underbrace{\left(\sum_{j=1}^M h_j \sigma(m_j) \frac{db_N^l}{dx}(m_j) \frac{db_N^k}{dx}(m_j) \right)}_{=(\mathbf{A})_{k,l}} \mu_l = \underbrace{\frac{1}{2}(h_{k+1} + h_k)g(x_k)}_{=:\varphi_k}, \quad k = 1, \dots, N,$$

$$\begin{aligned} &\Updownarrow \\ &\mathbf{A}\vec{\mu} = \vec{\varphi}. \end{aligned}$$

Resulting linear system of equations equidistant mesh with uniform cell size $h > 0$

$$\frac{1}{h} \begin{pmatrix} \sigma_1 + \sigma_2 & -\sigma_2 & 0 & & & & 0 \\ -\sigma_2 & \sigma_2 + \sigma_3 & -\sigma_3 & & & & 0 \\ 0 & \dots & \dots & \dots & & & 0 \\ & & & \dots & \dots & \dots & 0 \\ & & & & -\sigma_{M-2} & \sigma_{M-2} + \sigma_{M-1} & -\sigma_{M-1} \\ 0 & & & & 0 & -\sigma_{M-1} & \sigma_{M-1} + \sigma_M \end{pmatrix} \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_N \end{pmatrix} = h \begin{pmatrix} g(x_1) \\ \vdots \\ g(x_N) \end{pmatrix}, \quad (1.5.89)$$

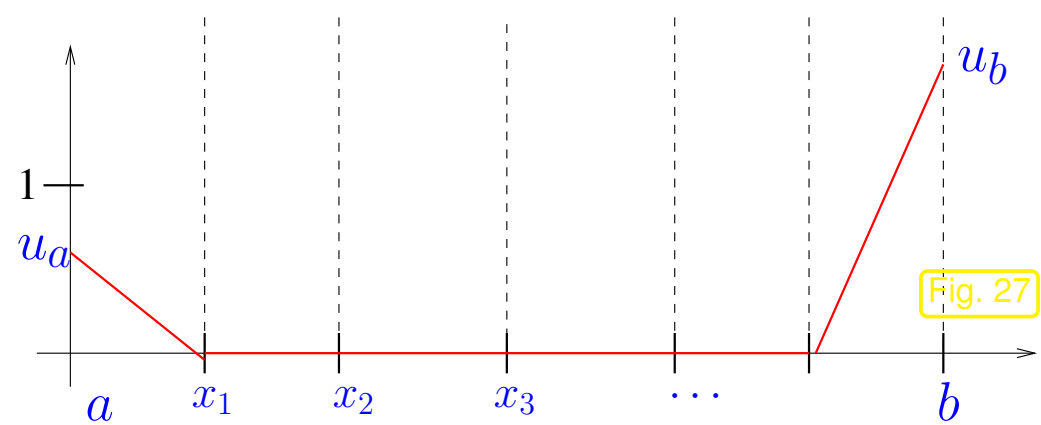
with $\sigma_j = \sigma(m_j)$, $j = 1, \dots, m$.

Remark 1.5.90 (Offset function for finite element Galerkin discretization).

In the case of general boundary conditions

$$u(a) = u_a, \quad u(b) = u_b$$

use *piecewise linear* offset function



$$u_0(x) = \begin{cases} u_a \left(1 - \frac{x-a}{h_1}\right) & , \text{ if } a \leq x \leq x_1, \\ u_b \left(1 - \frac{b-x}{h_M}\right) & , \text{ if } x_{M-1} \leq x \leq b, \\ 0 & \text{elsewhere.} \end{cases} \quad (1.5.91)$$

Benefits of this choice of offset function:

- u_0 is a *simple* function (since p.w. linear),
- u_0 is *locally supported*: contributions from u_0 will alter only first and last component of right hand side vector. To understand why, recall (1.5.19) and verify that $a(u_0, b_N^j) \neq 0$ only for $j = 1, M-1$.

Example 1.5.92 (Linear finite element Galerkin discretization for elastic string model).

Targetted: *non-linear* variational equation on domain $\Omega = [0, 1]$

$$\int_0^1 \frac{\kappa(\xi)}{L} \left(1 - \frac{L}{\|\mathbf{u}'(\xi)\|} \right) \mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = 0 \quad \forall \mathbf{v} \in (C_{0,\text{pw}}^1([0, 1]))^2. \quad (1.3.12)$$

- Data κ, \mathbf{f} given in procedural form, see Rem. 1.5.6.
- trial space $V_{N,0} = (\mathcal{S}_{1,0}^0(\mathcal{M}))^2$ on equidistant mesh \mathcal{M} , meshwidth $h := \frac{1}{M}$.

- Basis: 1D tent functions, lexicographic ordering

$$\mathfrak{B} = \left\{ \begin{pmatrix} b_N^1 \\ 0 \end{pmatrix}, \begin{pmatrix} b_N^2 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} b_N^{M-1} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ b_N^1 \end{pmatrix}, \begin{pmatrix} 0 \\ b_N^2 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ b_N^{M-1} \end{pmatrix} \right\}.$$

- Evaluation of right hand side by composite trapezoidal rule (1.5.85).
- Evaluation left hand side by composite midpoint rule (1.5.88).

Preliminary consideration: the derivative of

$$\mathbf{u}_N := \mathbf{u}_0 + \mu_1 \begin{pmatrix} b_N^1 \\ 0 \end{pmatrix} + \dots + \mu_{M-1} \begin{pmatrix} b_N^{M-1} \\ 0 \end{pmatrix} + \mu_M \begin{pmatrix} 0 \\ b_N^1 \end{pmatrix} + \dots + \mu_{2M-2} \begin{pmatrix} 0 \\ b_N^{M-1} \end{pmatrix} \quad (1.5.101)$$

with offset function according to Rem. 1.5.90 is piecewise constant on \mathcal{M} :

$$\begin{aligned} \text{in }]x_{j-1}, x_j[: \quad s_j(\vec{\mu}) &:= \mathbf{u}'_N(\xi) = \frac{\mathbf{u}(x_j) - \mathbf{u}(x_{j-1})}{h} \\ &= \frac{1}{h} \cdot \begin{cases} \begin{pmatrix} \mu_j - \mu_{j-1} \\ \mu_{j+M-1} - \mu_{j+M-2} \end{pmatrix} & , \text{ if } 2 \leq j \leq M-1, \\ \begin{pmatrix} \mu_1 \\ \mu_M \end{pmatrix} - \mathbf{u}(0) & , \text{ if } j = 1, \\ \mathbf{u}(1) - \begin{pmatrix} \mu_{M-1} \\ \mu_{2M-2} \end{pmatrix} & , \text{ if } j = M. \end{cases} \end{aligned} \quad (1.5.102)$$

$$\text{Set:} \quad r_j = r_j(\vec{\mu}) := h \frac{\kappa(m_j)}{L} \left(1 - \frac{L}{\|s_j(\vec{\mu})\|} \right)$$

Single row non-linear system of equations arising from Galerkin finite element discretization:

$$\text{row } 1: \quad (r_1 + r_2)\mu_1 - r_2\mu_2 = hf_1(h) + r_1a, \quad (1.5.103)$$

$$\text{row } j: \quad -r_j\mu_j + (r_j + r_{j+1})\mu_{j+1} - r_{j+1}\mu_{j+2} = f_1(jh), \quad 2 \leq j < M-1, \quad (1.5.104)$$

$$\text{row } M-1: \quad -r_{M-1}\mu_{M-2} + (r_{M-1} + r_M)\mu_{M-1} = hf_1((M-1)h) + r_Mb, \quad (1.5.105)$$

$$\text{row } M: \quad (r_1 + r_2(\vec{\mu}))\mu_M - r_2\mu_{M+1} = hf_2(h) + r_1u_a, \quad (1.5.106)$$

$$\text{row } j: \quad -r_j\mu_{j+M-1} + (r_j + r_{j+1})\mu_{j+M} - r_{j+1}\mu_{j+M+1} = f_2(jh), \quad 2 \leq j < M-1, \quad (1.5.107)$$

$$\text{row } M-1: \quad -r_{M-1}\mu_{2M-3} + (r_{M-1} + r_M)\mu_{2M-2} = hf_2((M-1)h) + r_Mu_b. \quad (1.5.108)$$

Here the dependence $r_j = r_j(\vec{\mu})$ has been suppressed to simplify the notation.

Please study the derivation of (1.5.89) in order to understand how (1.5.103)-(1.5.108) arise.

These equations can be written in a more compact form:

$$(1.5.103)-(1.5.108) \Leftrightarrow \boxed{\begin{pmatrix} \mathbf{R}(\vec{\mu}) & 0 \\ 0 & \mathbf{R}(\vec{\mu}) \end{pmatrix} \vec{\mu} = \begin{pmatrix} \vec{\varphi}_1(\vec{\mu}) \\ \vec{\varphi}_2(\vec{\mu}) \end{pmatrix}}. \quad (1.5.109)$$

with

$$\mathbf{R}(\vec{\mu}) := \begin{pmatrix} r_1 + r_2 & -r_2 & 0 & & & 0 \\ -r_2 & r_2 + r_3 & -r_3 & & & 0 \\ 0 & \ddots & \ddots & \ddots & & 0 \\ & & \ddots & \ddots & \ddots & 0 \\ & & & -r_{M-2} & r_{M-2} + r_{M-1} & -r_{M-1} \\ 0 & & & 0 & -r_{M-1} & r_{M-1} + r_M \end{pmatrix} \in \mathbb{R}^{M-1, M-1},$$

$$(\vec{\varphi}_1)_j := hf_1(hj) \quad , \quad (\vec{\varphi}_2)_j := hf_2(hj) \quad , \quad j = 1, \dots, M-1.$$

Dependence of the right hand side vector on the solution $\vec{\mu}$ is due to the offset function technique, see Rem. 1.5.90.

Iterative solution of (1.5.109) by **fixed point iteration**, see Ex. 1.5.69

Initial guess $\vec{\mu}^{(0)} \in \mathbb{R}^N$; $k = 0$;

repeat

$k \leftarrow k + 1$;

Solve the *linear* system of equations

$$\begin{pmatrix} \mathbf{R}(\vec{\mu}^{(k-1)}) & 0 \\ 0 & \mathbf{R}(\vec{\mu}^{(k-1)}) \end{pmatrix} \vec{\mu}^{(k)} = \begin{pmatrix} \vec{\varphi}_1(\vec{\mu}^{(k-1)}) \\ \vec{\varphi}_2(\vec{\mu}^{(k-1)}) \end{pmatrix};$$

until $\|\vec{\mu}^{(k)} - \vec{\mu}^{(k-1)}\| \leq \text{tol} \cdot \|\vec{\mu}^{(k)}\|$

Code 1.5.110: Linear finite element discretization of elastic string variational problem

```

1 function [vu, Jrec, figsol, figerg] =
   stringlinfem(kappa, f, L, u0, u1, M, tol)
2 % Solving the non-linear variational problem (1.3.12) for the elastic string by
   means of piecewise
3 % linear finite elements on an equidistant mesh with M-1 interior nodes.
4 % kappa, f are handles of type @(xi) providing the coefficient function
5 % kappa and the force field f. u0 and u1 pass the pinning points.
6 % M is the number of mesh cells, tol specifies the tolerance for the fixed
   point

```

```
7 % iteration. return value:  $2 \times (M+1)$ -matrix of node positions
8 if (nargin < 7), tol = 1E-2; end
9 h = 1/M; % meshwidth
10 phi = h*f(h*(1:M-1)); % Right hand side vector
11
12 % Initial guess: straight string, condition  $L > \|\mathbf{u}(0) - \mathbf{u}(1)\|$ .
13 if (L >= norm(u1-u0)), error('String must be tense'); end
14 vu_new = u0*(1-(0:1/M:1))+u1*(0:1/M:1);
15 % Meaning of components of vu:  $vu(1,2:M) \leftrightarrow \mu_1, \dots, \mu_{M-1}$ ,  $vu(2,2:M) \leftrightarrow$ 
     $\mu_M, \dots, \mu_{2M-2}$ .
16 figsol = figure; Jrec = []; hold on;
17 for k=1:100 % loop for fixed point iteration, maximum 100 iterations
18     vu = vu_new;
19 % Plot shape of string
20     plot(vu(1,:),vu(2:,:), '--g'); drawnow;
21     title(sprintf('M = %d, iteration #%d',M,k));
22     xlabel('{\bf x_1}'); ylabel('{\bf x_2}');
23 %Compute the cell values  $s_j, r_j, j=1, \dots, M$ , see (1.5.102).
24     d = (vu(:,2:end) - vu(:,1:end-1))/h;
25     s = sqrt(d(1,:).^2 + d(2,:).^2);
26     r = kappa(h*((1:M)-0.5)).*(1/L - 1./s)/h;
27 % Compute total potential energy
28     Jel = h/(2*L)*kappa(h*((1:M)-0.5)).*((s-L).^2)';
29     Jf = - (phi(1,:)*vu(1,2:M)'+phi(2,:)*vu(2,2:M)');
30     Jrec = [Jrec; k , Jel, Jf, Jel+Jf];
```

```
31 % Assemble triadiagonal matrix  $\mathbf{R} = \mathbf{R}(\vec{\mu})$ 
32 R = gallery('tridiag', -r(2:M-1), r(1:M-1)+r(2:M), -r(2:M-1));
33 % modify right hand side in order to take into account pinning conditions
34 phi1 = phi(1,:); phi1(1) = phi1(1) + r(1)*u0(1); phi1(M-1) =
    phi1(M-1) + r(M)*u1(1);
35 phi2 = phi(2,:); phi2(1) = phi2(1) + r(1)*u0(2); phi2(M-1) =
    phi2(M-1) + r(M)*u1(2);
36 % Solve linear system and compute new iterate
37 vu_new = [u0, [(R\phi1')'; (R\phi2')'], u1];
38 % Check simple termination criterion for fixed point iteration.
39 if (norm(vu_new - vu, 'fro') < tol*norm(vu_new, 'fro')/M)
40     plot(vu(1,:), vu(2,:), 'r-*'); break; end
41 end
42 % Plot of total potential energy in the course of the iteration
43 figerg = figure('name', 'total potential energy');
44 title(sprintf('elastic string, M = %d', M));
45 plot(Jrec(:,1), Jrec(:,4), 'm-*', ...
46     Jrec(:,1), Jrec(:,2), 'b-+', ...
47     Jrec(:,1), Jrec(:,3), 'g-^');
48 xlabel('\bf no. of iteration step'); ylabel('\bf energy');
49 legend('total potential energy', 'elastic energy', 'energy in force
    field', 'location', 'east');
```

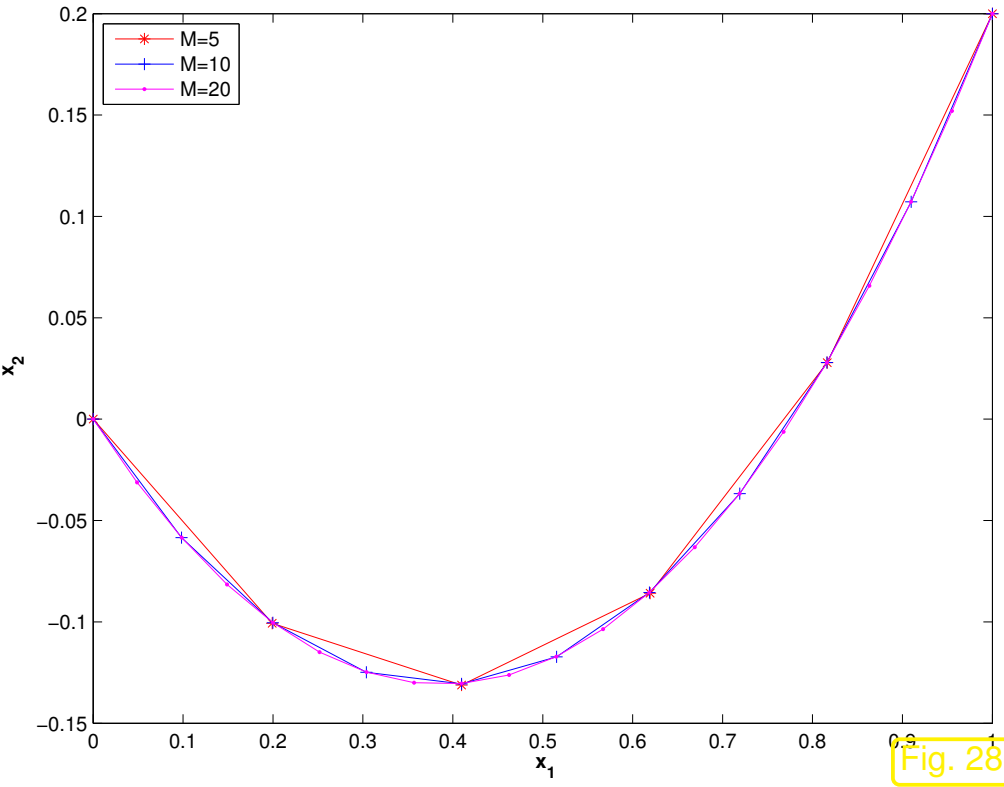


Example 1.5.111 (Elastic string shape by finite element discretization).

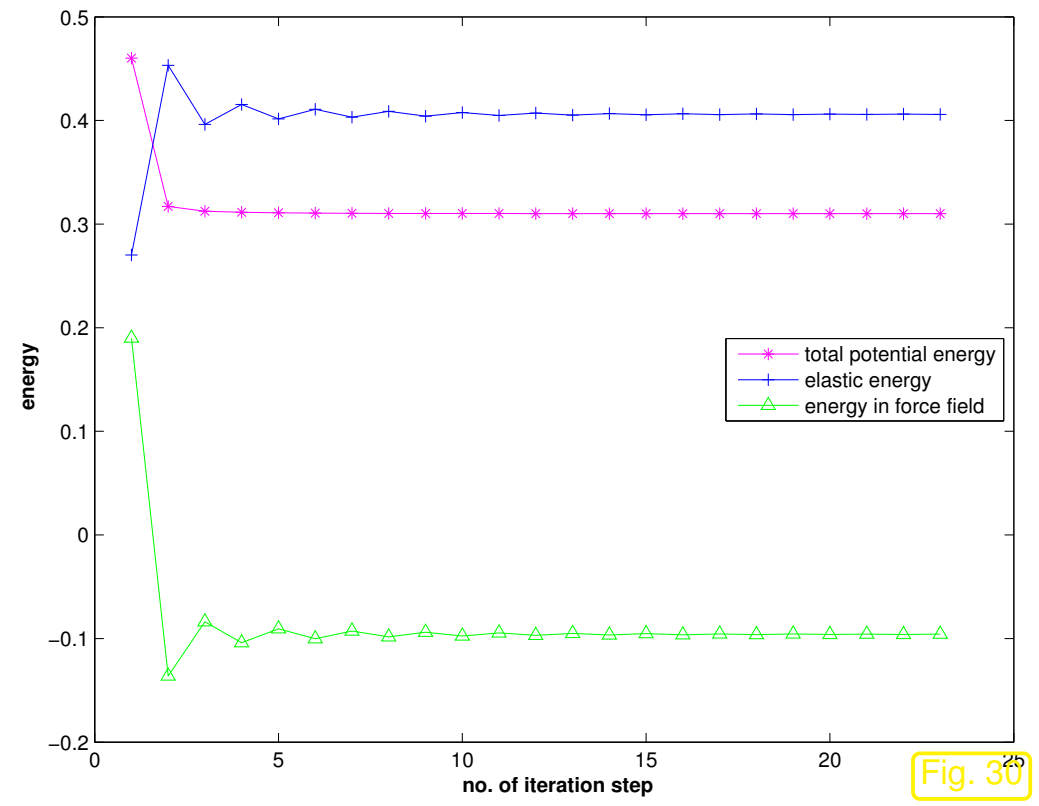
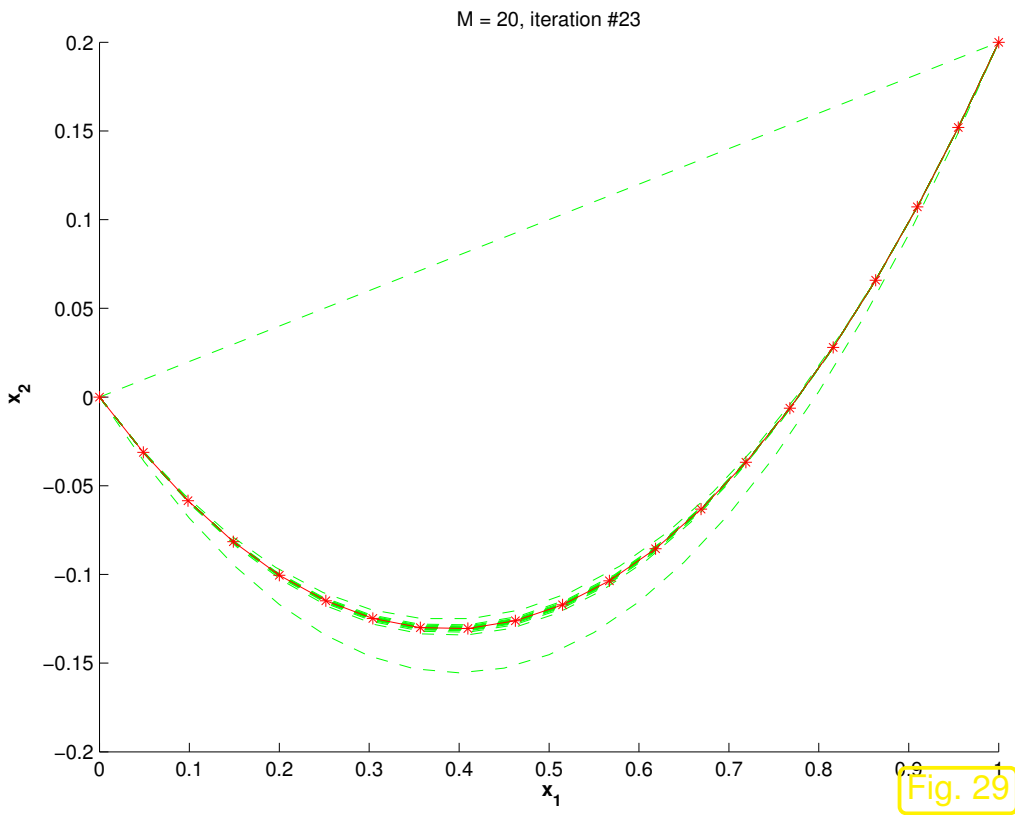
- Linear finite element discretization of (1.3.12), see Ex. 1.5.92, Code 1.5.109.
- $\kappa \equiv 1, L = 0.5, \mathbf{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mathbf{u}(1) = \begin{pmatrix} 1 \\ 0.2 \end{pmatrix}$
- gravitational force field $\mathbf{f}(\xi) = -\begin{pmatrix} 0 \\ 2 \end{pmatrix}$.

Piecewise linear finite element solution of (1.3.12),
equidistant meshes with M cells, $M = 5, 10, 20 \triangleright$

“Visual convergence” of computed polygon approximating the string shape.



Convergence of fixed point iteration ($M = 20$):



1.5.2 Collocation

Targetted:

Two-point BVP = ODE $\mathcal{L}(u) = f$ + boundary conditions

Note: In contrast to the Galerkin approach, collocation techniques do not tackle the weak form of a boundary value problem, but rather the “classical”/strong form.

- Idea: ❶ seek solution in **finite-dimensional** trial space $V_{N,0}$, $N := \dim V_{N,0} < \infty$
 ❷ pick **collocation nodes** $\mathcal{N} := \{x_1, \dots, x_N\} \subset \Omega$ such that \mathbf{x}

“point evaluation”
$$\begin{cases} V_{N,0} \mapsto \mathbb{R}^N \\ v \mapsto (v(x_j))_{j=1}^N \end{cases} \quad (1.5.112)$$

is a *bijjective* linear mapping.

Collocation conditions: $u_N \in V_N: \mathcal{L}(u_N)(x_j) = f(x_j), \quad j = 1, \dots, N.$ (1.5.113)

- ❸ choose ordered **basis** $\mathfrak{B} = \{b_N^1, \dots, b_N^N\}$ of $V_{N,0}$ & plug basis representation

$$u_N = u_0 + \mu_1 b_N^1 + \dots + \mu_N b_N^N \quad (u_0 \hat{=} \text{offset function, cf. Rem. 1.5.15})$$

into collocation conditions (1.5.113)

►
$$\vec{\mu} = (\mu_l)_{l=1}^N: \mathcal{L}(u_0 + \mu_1 b_N^1 + \dots + \mu_N b_N^N)(x_j) = f(x_j), \quad j = 1, \dots, N. \quad (1.5.114)$$

In general: (1.5.114) is a non-linear system of equation (N equations for N unknowns μ_1, \dots, μ_N).

Note: bijectivity of point evaluation (1.5.112) \Rightarrow $\#\{\text{nodes}\} = \dim V_{N,0}$

Below: detailed discussion for *linear* two point boundary value problem

$$\begin{aligned} \mathcal{L}(u) &:= -\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right) = g(x), \quad a \leq x \leq b, \\ u(a) &= u_a, \quad u(b) = u_b, \end{aligned} \tag{1.5.117}$$

on domain $\Omega = [a, b]$, related to variational problem (1.4.19).

Remark 1.5.119 (Smoothness requirements for collocation trial space).

For two-point BVP (1.5.117) consider space $V_{N,0} := \mathcal{S}_{1,0}^0(\mathcal{M})$ of \mathcal{M} -piecewise linear finite element functions. \rightarrow Sect. 1.5.1.2

Note: $v_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$ is *not differentiable* in nodes x_j of the mesh.

► Natural choice collocation points = nodes of the mesh is *not possible!*
(because for $v_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$ the function $\mathcal{L}(v_N)$ is discontinuous in the nodes of the mesh)

► Assuming $\sigma \in C^1([a, b])$ global continuity of $\mathcal{L}(v_N)$ entails $V_{N,0} \subset C^2([a, b])$, *cf.*
Sect. 1.5.2.2. △

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 1.5.121 (Collocation: smoothness requirements for coefficients).

For 2-point BVP (1.5.117): σ must be differentiable in collocation nodes, with known values $\frac{d\sigma}{dx}(x_j)$, $j = 1, \dots, N$, in the sense of Rem. 1.5.6: extra difficulty when σ given in procedural



1.5.2.1 Spectral collocation

Focus: *linear* two point boundary value problem (1.5.117)

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

trial space for polynomial spectral collocation:

$$V_{N,0} = \mathcal{P}_p(\mathbb{R}) \cap C_0^2([a, b]), \quad p \geq 2. \quad (1.5.122)$$

= polynomials of degree $\leq p$, vanishing at endpoints of domain, $N := \dim V_{N,0} = p - 1$.

➤ same trial space as for polynomial spectral Galerkin approach, see Sect. 1.5.1.1.

Discussion: polynomial spectral collocation for two-point BVP (1.5.117)

- offset function $u_0(x) := \frac{b-x}{b-a}u_a + \frac{x-a}{b-a}u_b$.
- Basis $\mathfrak{B} := \{b_N^j := M_j\}$ consisting of integrated Legendre polynomials, see (1.5.32).

Note:

\mathcal{L} from (1.5.117) is a **linear differential operator!**

Terminology: A differential operator is a mapping on a function space involving only values of the function argument and some of its derivatives in the same point.

A differential operator \mathcal{L} is **linear**, if

$$\mathcal{L}(\alpha u + \beta v) = \alpha \mathcal{L}(u) + \beta \mathcal{L}(v) \quad \forall \alpha, \beta \in \mathbb{R}, \quad \forall \text{functions } u, v \quad (1.5.126)$$

$$(1.5.114) \quad \stackrel{(1.5.126)}{\implies} \quad \sum_{l=1}^N \mathcal{L}(b_N^l)(x_k) \mu_l = f(x_k) - \mathcal{L}(u_0)(x_k), \quad k = 1, \dots, N. \quad (1.5.127)$$

$$\mathbf{A} \vec{\mu} = \vec{\varphi}, \quad \begin{aligned} & \updownarrow \\ & (\mathbf{A})_{k,l} := \mathcal{L}(b_N^l)(x_k), \quad k, l \in \{1, \dots, N\}, \\ & \varphi_k := f(x_k) - \mathcal{L}(u_0)(x_k), \quad k \in \{1, \dots, N\}. \end{aligned} \quad (1.5.128)$$

An $N \times N$ linear system of equations

For BVPs featuring *linear* differential operators, collocation invariably leads to a *linear* system of equations for the unknown coefficients of the basis representation of the collocation solution.

Remark 1.5.129 (Bases for polynomial polynomial spectral collocation).

Same choices as for spectral Galerkin methods, see Rem. 1.5.30.



Remark 1.5.130 (Collocation points for polynomial spectral collocation).

Rule of thumb (without further explanation, see [19]):

choose collocation points x_j , $j = 1, \dots, N$ such that the induced Lagrangian interpolation operator (\rightarrow [21, Thm. 3.3.7]) has a small ∞ -norm, see [21, Lemma 3.5.5].

Popular choice (due to [21, Eq. 9.2.15]): **Chebyshev nodes**

$$x_k := a + \frac{1}{2}(b - a) \left(\cos\left(\frac{2k - 1}{2N} \pi\right) + 1 \right), \quad k = 1, \dots, N. \quad (1.5.131)$$



Code 1.5.132: Computation of derivatives of Legendre polynomials using (1.5.45)

```

1 function [V,M,D] = dilegpol(n,x)
2 % Computes values of the first n+1 Legendre polynomials (returned in matrix V)
3 % the first n-1 integrated Legendre polynomials (returned in matrix M), and
  the
4 % first n+1 first derivatives of Legendre polynomials in the points x_j passed
5 % in the row vector x.
6 % Uses the recursion formulas (1.5.43) and (1.5.32)
7 V = ones(size(x)); V = [V; x];
8 % recursion (1.5.43) for Legendre polynomials
9 for j=1:n-1, V = [V; ((2*j+1)/(j+1)).*x.*V(end, :) -
   j/(j+1)*V(end-1, :)]; end
10 % Formula (1.5.32) for integrated Legendre polynomials
11 M = diag(1./(2*(1:n-1)+1)).*(V(3:n+1, :) - V(1:n-1, :));
12 % Recursion formula (1.5.45) for derivatives of Legendre polynomials

```

```

13 if (nargout < 3)
14     D = [zeros(size(x)); ones(size(x))];
15     for j=1:n-1, D = [D; (2*j+1)*V(j+1,:) + D(j,:)]; end
16 end

```

Code 1.5.133: Spectral collocation for linear 2nd-order two-point BVP

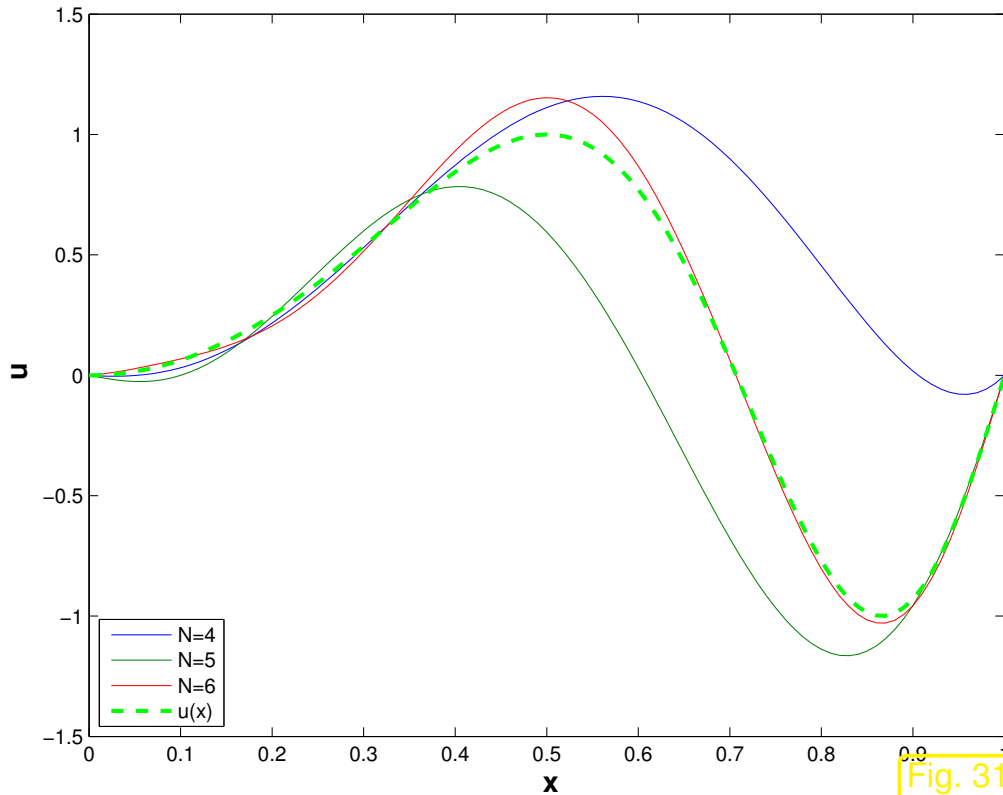
```

1 function u = linspeccol(g,N,x)
2 % Polynomial spectral collocation discretization of linear 2nd-order two-point
   BVP
3 %  $-\frac{d^2u}{dx^2} = g(x)$ ,  $u(0) = u(1) = 0$ 
4 % on  $\Omega = [0, 1]$ . Trial space of dimension  $N$ , collocation in Chebychev nodes.
5 % Values of approximate solution in points  $x_j$  are returned in the row vector  $u$ 
6 cn = cos((2*(1:N)-1)*pi/(2*N)); % Chebychev nodes, see (1.5.131)
7 [V,M,D] = dilegpol(N+1,cn); % Obtain values of (2nd
   derivatives) of  $M_m$ 
8 mu = (-4*D(2:N+1,:))' \ (g(0.5*(cn+1)))'; % Solve collocation system
9 % Compute values of integrated Legendre polynomials at output points
10 [V,M] = dilegpol(N+1,2*x-1); u = mu'*M;

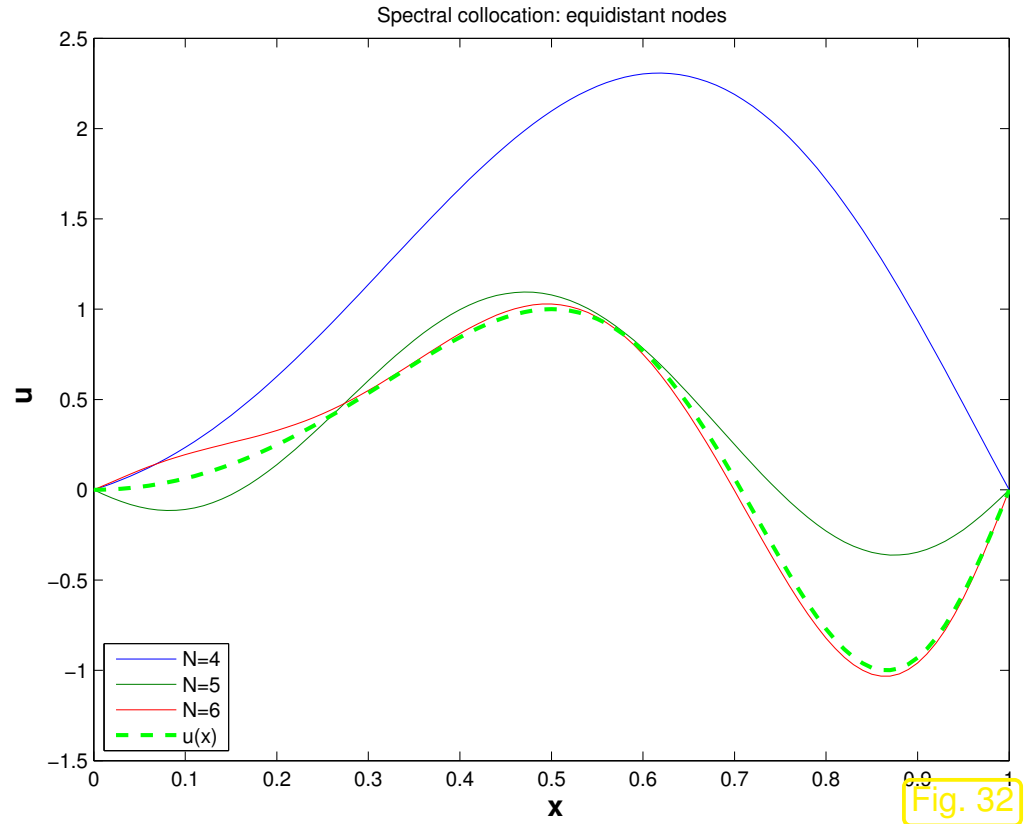
```

Example 1.5.134 (Polynomial spectral collocation for 2-point BVP).

Setting of Ex. 1.5.28, spectral polynomial collocation, on , $N = 5, 7, 10$, basis from integrated Legendre polynomials, plot of solution u_N .



Collocation in Chebychev nodes



Collocation in equidistant nodes

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



1.5.2.2 Spline collocation

Analogous to Sect. 1.5.1.2: now collocation based on *piecewise polynomials*

Rem. 1.5.119 \triangleright for BVP (1.5.117) smoothness $V_{N,0} \subset C^2([a, b])$ is required.

Which piecewise polynomial spaces offer this kind of smoothness ?

Recall [21, Def. 3.8.1], *cf.* [21, Sect. 3.8.1]:

Definition 1.5.135 (Cubic spline).

$s :]a, b[\mapsto \mathbb{R}$ is a **cubic spline** function w.r.t. the **node set** $\mathcal{T} := \{a = x_0 < x_1 < x_2 < \dots < x_{M-1} < x_M = b\}$, if

- (i) $s \in C^2([a, b])$ (twice continuously differentiable),
- (ii) $s|_{]x_{j-1}, x_j[} \in \mathcal{P}_3(\mathbb{R})$ (**piecewise cubic polynomial**)

 notation: $\mathcal{S}_{3, \mathcal{T}} \hat{=}$ vector space of cubic splines on node set \mathcal{X}

Known:

$$\dim \mathcal{S}_{3, \mathcal{T}} = \#\mathcal{T} + 2 = M + 3$$

► Trial space for collocation for 2-point BVP (1.5.117)

natural cubic splines: $V_{N,0} := \left\{ s \in \mathcal{S}_{3, \mathcal{T}} : \begin{array}{l} s''(a) = s''(b) = 0, \\ s(a) = s(b) = 0 \end{array} \right\} \Rightarrow \dim N := V_N = M - 1$,

Choice of collocation nodes:

collocation nodes for cubic spline collocation = spline nodes x_j : $\mathcal{N} = \mathcal{T}$

Example 1.5.136 (Cubic spline collocation discretization of 2-point BVP).

Setting of Ex. 1.5.28

Cubic spline collocation with equidistant nodes, $M = 5, 7, 12$

Solution u_N \triangleright

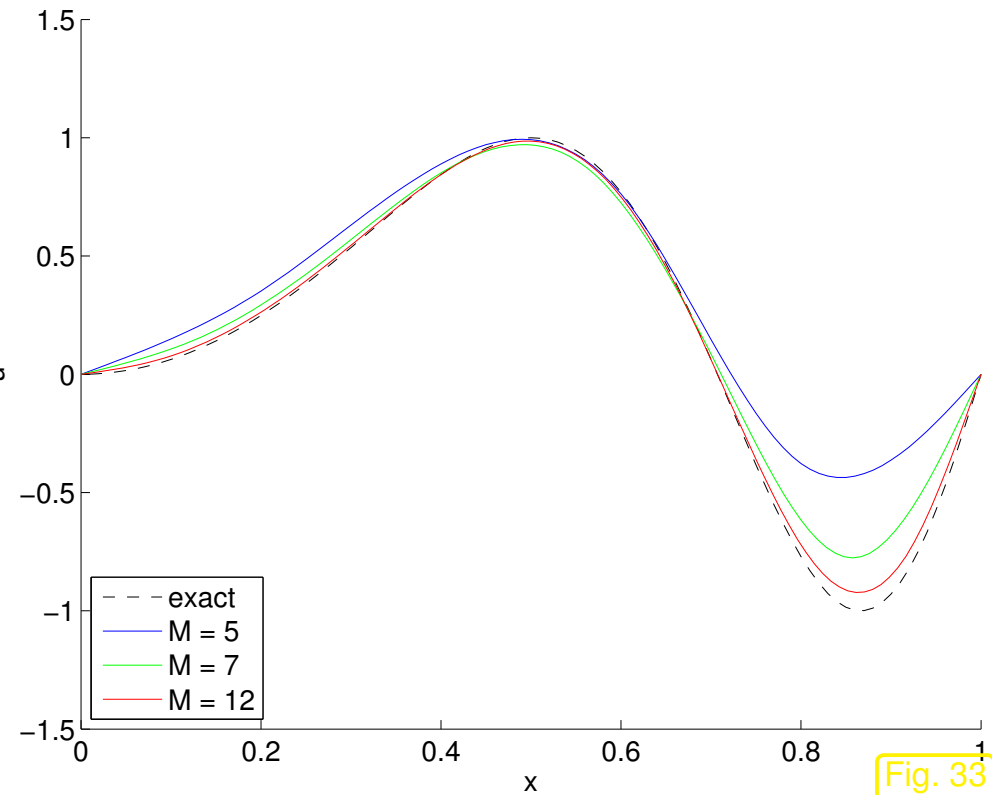


Fig. 33



1.5.3 Finite differences

Focus: 2nd-order linear two-point BVP

$$\mathcal{L}(u) := -\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right) = g(x), \quad a \leq x \leq b, \quad (1.5.117)$$

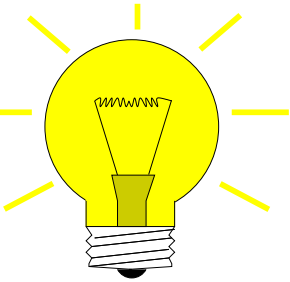
$$u(a) = u_a \quad , \quad u(b) = u_b \quad ,$$

Idea:

Replace derivatives \longrightarrow difference quotients

(in finitely many special points = nodes of a mesh)

E.g.
$$\frac{d^2u}{dx^2}(x) \approx \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} \quad , \quad h > 0 \text{ "small"} \quad . \quad (1.5.137)$$



Setting as in Sect. 1.5.1.2:

➤ $\Omega = [a, b]$ equipped with **nodes** ($M \in \mathbb{N}$)

$\mathcal{X} := \{a = x_0 < x_1 < \dots < x_{M-1} < x_M = b\}$.

➤ **mesh/grid**

$$\mathcal{M} := \{]x_{j-1}, x_j[: 1 \leq j \leq M\} .$$

Special case:

equidistant mesh: $x_j := a + jh$, $h := \frac{b-a}{M}$.

☞ $]x_{j-1}, x_j]$, $j = 1, \dots, M$, $\hat{=}$ **cells** of \mathcal{M} , **cell size** $h_j := |x_j - x_{j-1}|$, $j = 1, \dots, M$
meshwidth $h_{\mathcal{M}} := \max_j |x_j - x_{j-1}|$

① replacement of outer derivative ($x_{j-1/2} = \frac{1}{2}(x_j + x_{j-1})$):

$$\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right) \Big|_{x=x_j} \approx \frac{2}{h_{j-1} + h_j} \left(\sigma(x_{j+1/2}) \frac{du}{dx}(x_{j+1/2}) - \sigma(x_{j-1/2}) \frac{du}{dx}(x_{j-1/2}) \right) .$$

② replacement of inner derivative, e.g.,

$$\frac{du}{dx}(x_{j+1/2}) \approx \frac{u(x_{j+1}) - u(x_j)}{h_j} .$$

$$-\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right) \Big|_{x=x_j} = \frac{\sigma(x_{j-1/2}) \frac{u(x_j) - u(x_{j-1})}{h_{j-1}} - \sigma(x_{j+1/2}) \frac{u(x_{j+1}) - u(x_j)}{h_j}}{\frac{1}{2}(h_{j-1} + h_j)} .$$

(1.5.138)

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

On equidistant mesh, uniform meshwidth $h_j = h > 0$, $j = 1, \dots, M$:

$$\begin{aligned} & -\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right) \Big|_{x=x_j} \\ &= \frac{1}{h^2} \left(-\sigma(x_{j+1/2})u(x_{j+1}) + (\sigma(x_{j+1/2}) + \sigma(x_{j-1/2}))u(x_j) - \sigma(x_{j-1/2})u(x_{j-1}) \right) . \end{aligned} \quad (1.5.139)$$

Unknowns in finite difference method:

$$\mu_l = u(x_l), \quad l = 1, \dots, M - 1$$

$$-\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right) = g(x), a \leq x \leq b.$$

← restriction to \mathcal{X} , use (1.5.139)

$$\frac{-\sigma(x_{j+1/2})\mu_{j+1} + (\sigma(x_{j+1/2}) + \sigma(x_{j-1/2}))\mu_j - \sigma(x_{j-1/2})\mu_{j-1}}{h^2} = g(x_j), \quad j = 1, \dots, M-1. \quad (1.5.140)$$

↕

$$(\mathbf{A})_{jl} = h^{-2} \cdot \begin{cases} 0 & , \text{ if } |j - l| > 1, \\ -\sigma(x_{j+1/2}) & , \text{ if } j = l - 1, \\ \sigma(x_{j-1/2}) + \sigma(x_{j+1/2}) & , \text{ if } j = l, \\ -\sigma(x_{l+1/2}) & , \text{ if } l = j - 1. \end{cases} \quad (1.5.141)$$

$$\mathbf{A}\vec{\mu} = \vec{\varphi}, \quad \text{with}$$

$$\varphi_j = \begin{cases} g(x_1) + \sigma(x_{1/2})u_a & , \text{ if } j = 1, \\ g(x_j) & , \text{ if } 1 < j < M-1, \\ g(x_{M-1}) + \sigma(x_{M-1/2})u_b & , \text{ if } j = M-1. \end{cases}$$

An $(M-1) \times (M-1)$ linear system of equations

(Up to scaling with h) the finite difference approach and the linear finite element Galerkin scheme (\rightarrow Sect. 1.5.1.2) yield the *same* system matrix for the BVP (1.5.117) and its associated variational problem (1.4.19), *cf.* (1.5.141) and (1.5.89).

1.6 Convergence

For elastic string model (1.2.26)/(1.3.12), taut string model in graph description (1.4.19) with exact solution $\mathbf{u} : [0, 1] \mapsto \mathbb{R}^2$ or $u : [a, b] \mapsto \mathbb{R}$, respectively:

Discretization schemes (Galerkin approach, Sect. 1.5.1 collocation methods, Sect. 1.5.2)	\longrightarrow	Approximate solution $\mathbf{u}_N : [0, 1] \mapsto \mathbb{R}^2 / u_N : [a, b] \mapsto \mathbb{R}$ (functions $\in V_N$)
--	-------------------	--

Desirable: approximation u_N “close to” exact solution u : *rigorous meaning ?*



How to measure **discretization error** $u - u_N$?

Remark 1.6.2 (Grid functions).

Note: for finite differences (\rightarrow Sect. 1.5.3) we get no solution function, only **grid function** $\mathcal{X} \mapsto \mathbb{R}$ (“point values”)

☞ reconstruction of a function through **postprocessing**, e.g., linear interpolation



Remark 1.6.4.

We encountered the issues of *convergence of approximate solutions* before:

- Numerical quadrature [21, Ch. 10]: study of **asymptotic** behavior of quadrature error
- Numerical integration [21, Ch. 12]: discretization error of single step methods



1.6.1 Norms on function spaces

Tools for measuring discretization errors: **norms** on function spaces/grid function spaces

Reminder → [21, Sect. 2.5.1]

Definition 1.6.6 (Norm).

A *norm* $\|\cdot\|_V$ on an \mathbb{R} -vector space V is a mapping $\|\cdot\|_V : V \mapsto \mathbb{R}_0^+$, such that

$$\text{(definiteness)} \quad \|v\|_V = 0 \iff v = 0 \quad \forall v \in V \quad \text{(N1)}$$

$$\text{(homogeneity)} \quad \|\lambda v\|_V = |\lambda| \|v\|_V \quad \forall \lambda \in \mathbb{R}, \forall v \in V, \quad \text{(N2)}$$

$$\text{(triangle inequality)} \quad \|w + v\|_V \leq \|w\|_V + \|v\|_V \quad \forall w, v \in V. \quad \text{(N3)}$$

Next: important norms on function spaces, cf. [21, Eq. 3.5.2], [21, Eq. 3.5.3], [21, Eq. 3.5.4]:

Definition 1.6.7 (Supremum norm).

The *supremum norm* of an (essentially) bounded function $\mathbf{u} : \Omega \mapsto \mathbb{R}^n$ is defined as

$$\|\mathbf{u}\|_\infty \quad (= \|\mathbf{u}\|_{L^\infty(\Omega)}) := \sup_{x \in \Omega} \|\mathbf{u}(x)\| \quad , \quad \mathbf{u} \in (L^\infty(\Omega))^n . \quad (1.6.8)$$

- $L^\infty(\Omega)$ denotes the vector space of essentially bounded functions. It is the instance for $p = \infty$ of an L^p -space.
- The notation $\|\cdot\|_\infty$ hints at the relationship between the supremum norm of functions and the maximum norm for vectors in \mathbb{R}^n .
- For $n = 1$ the Euclidean vector norm in the definition reduces to the modulus $|u(x)|$.
- The norm $\|\mathbf{u} - \mathbf{u}_N\|_{L^\infty(\Omega)}$ measures the maximum distance of the function values of \mathbf{u} and \mathbf{u}_N .

Definition 1.6.9 (Mean square norm/ L^2 -norm). \rightarrow Def. 2.2.8

For a function $\mathbf{u} \in (C_{\text{pw}}^0(\Omega))^n$ the **mean square norm/ L^2 -norm** is given by

$$\|\mathbf{u}\|_0 \left(\|\mathbf{u}\|_{L^2(\Omega)} \right) := \left(\int_{\Omega} \|\mathbf{u}(x)\|^2 \, dx \right)^{1/2}, \quad \mathbf{u} \in (L^2(\Omega))^n.$$

- $L^2(\Omega)$ designates the vector space of square integrable functions, another L^p -space (for $p = 2$) and a **Hilbert space**.
- The “0” in the notation $\|\cdot\|_0$ refers to the absence of derivatives in the definition of the norm.
- Obviously, the L^2 -norm is **weaker** than the supremum norm:

$$\|v\|_{L^2([a,b])} \leq \sqrt{|b-a|} \|v\|_{L^\infty([a,b])} \quad \forall v \in C_{\text{pw}}^0([a,b]).$$

In particular, the L^2 -norm of the discretization error may be small despite large deviations of u_N from u , provided that these deviations are very much *localized*.

Relevant error norms suggested by application context/physics!

Remark 1.6.10 (Energy norm).

We consider the model for a homogeneous taut string in physical space, see (1.4.19), with associated total potential energy functional

$$J(u) := \int_a^b \frac{1}{2} \left| \frac{du}{dx}(x) \right|^2 + \widehat{g}(x)u(x) \, dx, \quad u \in C_{0,\text{pw}}^1([a, b]), \quad (1.6.15)$$

where, for the sake of simplicity, we assume $u_a = u_b = 0$.

A manifestly relevant error quantity of interest is the **deviation of energies**

$$E_J := |J(u) - J(u_N)|.$$

We adopt the concise notations introduced for abstract (linear) variational problems in Rems. 1.3.21, 1.4.6:

$$J(u) = \frac{1}{2} \mathbf{a}(u, u) - \ell(u), \quad \mathbf{a}(u, v) := \int_a^b \frac{du}{dx}(x) \frac{dv}{dx}(x) \, dx, \quad \ell(v) := - \int_a^b \widehat{g}(x)v(x) \, dx,$$

where \mathbf{a} is a *symmetric* bilinear form, see Def. 1.3.23.

Assumption: $u_N \in V_{N,0} \hat{=}$ **Galerkin solution** based on discrete trial space $V_{N,0} \subset V_0$.

$$\blacktriangleright \begin{aligned} \mathbf{a}(u, v) &= \ell(v) \quad \forall v \in V_0 := C_{0,\text{pw}}^1([a, b]) , \\ \mathbf{a}(u_N, v_N) &= \ell(v_N) \quad \forall v_N \in V_{N,0} \subset V_0 . \end{aligned} \tag{1.6.16}$$

We can use the defining variational equations for u and u_N to express

$$J(u) - J(u_N) = -\frac{1}{2}(\mathbf{a}(u, u) - \mathbf{a}(u_N, u_N)) \stackrel{(*)}{=} -\frac{1}{2}\mathbf{a}(u + u_N, u - u_N) . \tag{1.6.17}$$

(*) : a straightforward consequence of the bilinearity of \mathbf{a} , see Def. 1.3.23, *c.f.* $a^2 - b^2 = (a+b)(a-b)$ for $a, b \in \mathbb{R}$.

Concretely,

$$\begin{aligned} |J(u) - J(u_N)| &= \frac{1}{2} \left| \int_a^b \frac{d}{dx}(u + u_N) \cdot \frac{d}{dx}(u - u_N) dx \right| \\ &\stackrel{(*)}{\leq} \frac{1}{2} \left(\int_a^b \left| \frac{d}{dx}(u + u_N) \right|^2 dx \right)^{1/2} \left(\int_a^b \left| \frac{d}{dx}(u - u_N) \right|^2 dx \right)^{1/2} . \end{aligned} \tag{1.6.18}$$

(*) : due to Cauchy-Schwarz inequality (2.2.24)

Definition 1.6.19 (H^1 -seminorm). \rightarrow *Def. 2.2.18*

For a function $u \in C_{\text{pw}}^1([a, b])$ the H^1 -seminorm reads

$$|u|_{H^1([a,b])}^2 := \int_a^b \left| \frac{du}{dx}(x) \right|^2 dx . \quad (1.6.20)$$

- $|\cdot|_{H^1([a,b])}$ is merely a **semi-norm**, because it only satisfies norm axioms (N2) and (N3), but fails to be definite: $|\cdot|_{H^1([a,b])} = 0$ for constant functions.

- In the setting of the homogeneous taut string model, we have

$$|u|_{H^1([a,b])}^2 = a(u, u) \quad \blacktriangleright \quad |\cdot|_{H^1([a,b])} \text{ is called the } \mathbf{energy\ norm} \text{ for the model.}$$

More explanations in Sect. 2.1.3.

- On $C_{0,\text{pw}}^1([a, b])$ the semi-norm $|\cdot|_{H^1([a,b])}$ is a genuine norm \rightarrow Def. 1.6.6. See proof of Thm. 2.2.25.

From (1.6.18)

$$\|u - u_N\|_{H^1(\Omega)} \leq \epsilon \quad \blacktriangleright \quad |J(u) - J(u_N)| \leq |u + u_N|_{H^1(\Omega)} |u - u_N|_{H^1(\Omega)} \quad (1.6.22)$$

$$\stackrel{\text{(N3)}}{\leq} (2|u|_{H^1(\Omega)} + \epsilon) \epsilon .$$

☛ estimate of the energy norm of the discretization error paves the way for bounding the energy deviation.

△
R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 1.6.26 (Norms on grid function spaces).

To measure the discretization error for finite difference schemes (\rightarrow Sect. 1.5.3) one may resort to **mesh dependent norms**

$$\text{(discrete) } l^2\text{-norm} \quad : \quad \|\vec{\mu}\|_{l^2(\mathcal{X})}^2 := \sum_{j=0}^M \frac{1}{2} (h_j + h_{j+1}) |\mu_j|^2, \quad (1.6.27)$$

(under convention $h_0 := 0, h_{M+1} := 0$),

(discrete) maximum norm :
$$\|\vec{\mu}\|_{l^\infty(\mathcal{X})} := \max_{j=0,\dots,M} |\mu_j|. \quad (1.6.28)$$

Remark 1.6.30 (Approximate computation of norms).

Standard *testing* of implementations of numerical methods for 2-point BVP: Examine norm of discretization error for test cases with (analytically) known exact solution u .

Even for numerical methods computing $u_N \in V_N \subset V$ (Galerkin methods \rightarrow Sect. 1.5.1, collocation methods \rightarrow Sect. 1.5.2):

usually exact computation of $\|u - u_N\|$ is impossible/very difficult.

Option: approximate evaluation of norm $\|u - u_N\|$

- supremum norm $\|\cdot\|_\infty$: approximation by sampling on discrete point set.
- L^2 -norm, energy norm: numerical quadrature
(Gauss quadrature for spectral schemes, composite quadrature for mesh based schemes)

! Error introduced by approximation of norm must be smaller than discretization error
(➤ use “overkill” quadrature/sampling, cost does not matter much in testing).



1.6.2 Algebraic and exponential convergence

Crucial: convergence is an *asymptotic notion* !

sequence of discrete models \Rightarrow sequence of approximate solutions $(u_N^{(i)})_{i \in \mathbb{N}}$
 \Rightarrow study sequence $(\|u_N^{(i)} - u\|)_{i \in \mathbb{N}}$

created by *variation* of a **discretization parameter**:

Discretization parameters:

- *meshwidth* $h > 0$ for finite differences (\rightarrow Sect. 1.5.3), p.w. linear finite elements
(\rightarrow Sect. 1.5.1.2), spline collocation (\rightarrow Sect. 1.5.2.2)
- *polynomial degree* for spectral collocation (\rightarrow Sect. 1.5.2.1),
spectral Galerkin discretization (\rightarrow Sect. 1.5.1.1)

Example 1.6.31 (Numerical studies of convergence).

Focus: Linear 2-point boundary value problem $-\frac{d^2u}{dx^2} = g(x)$, $u(0) = u(1) = 0$ on $\Omega =]0, 1[$,
variational form (1.5.29),

exact solution $u(x) = \sin(2\pi x^2)$ (\rightarrow setting of Ex. 1.5.28)

- ① finite difference discretization on equidistant mesh, meshwidth $h > 0$ (\rightarrow Sect. 1.5.3)

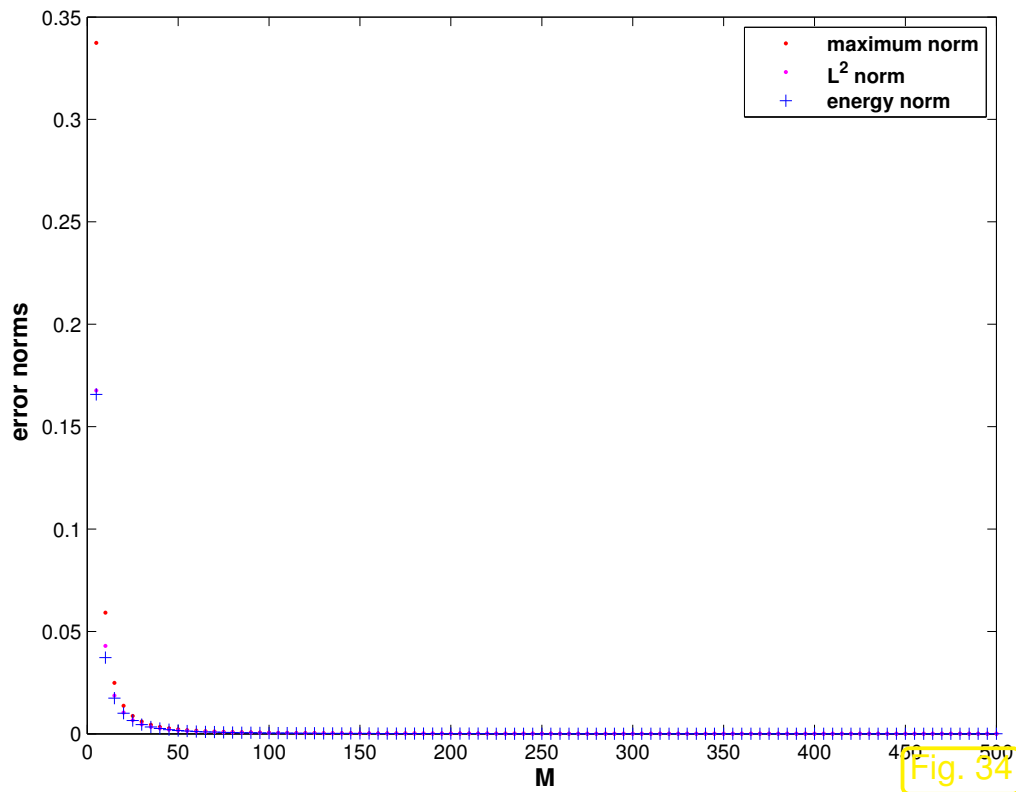


Fig. 34

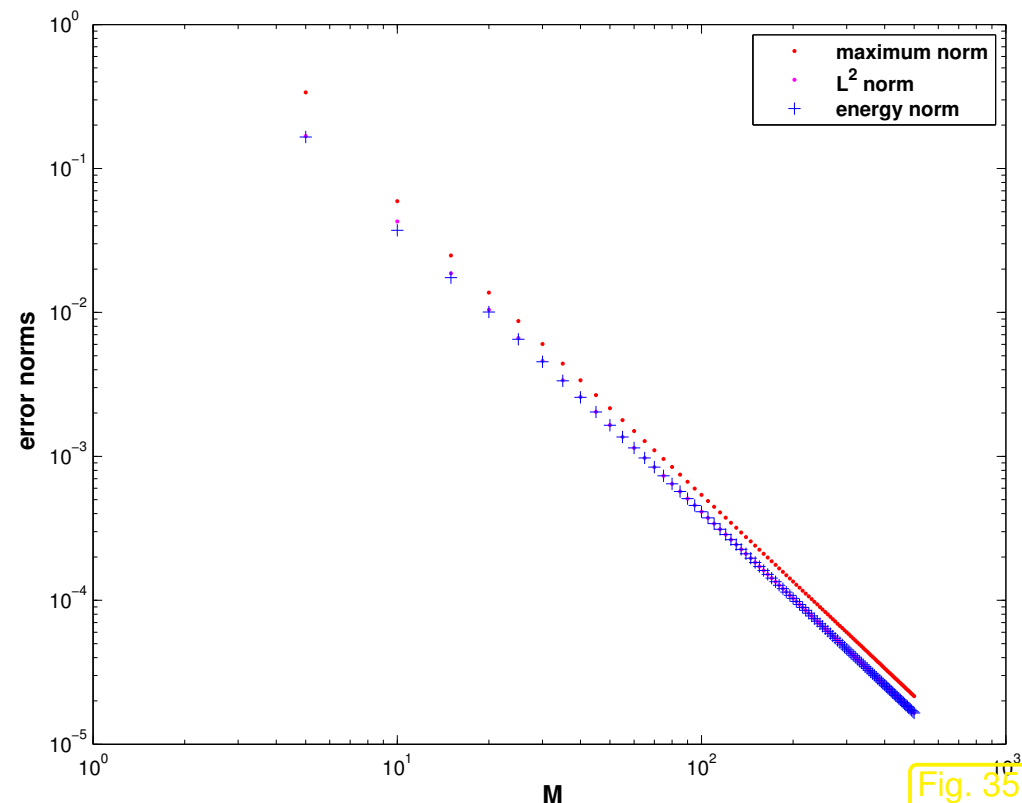


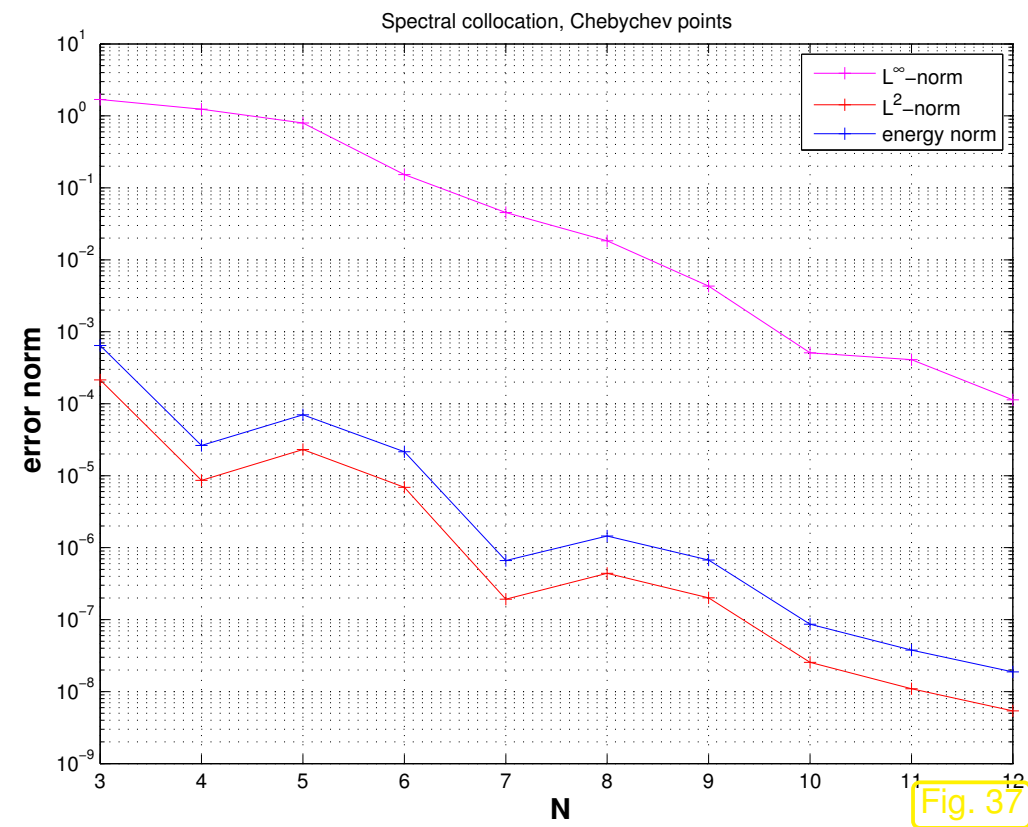
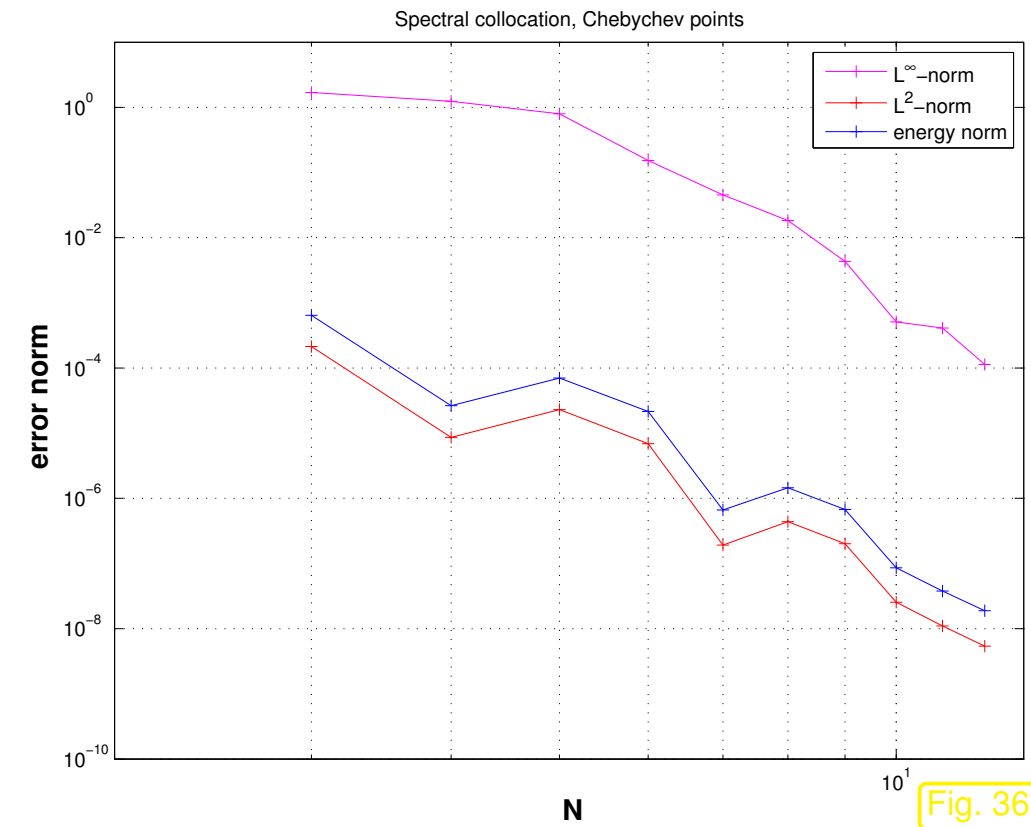
Fig. 35

What is plotted are the discrete versions of the L^2 -norm and supremum norm, see Rem. 1.6.26.

The energy norm of the error was computed according to the formula

$$\text{energy norm}(\text{error})^2 := \sum_{j=1}^M h_j \left| \frac{\mu_j - \mu_{j-1}}{h_j} - \frac{du}{dx}(x_{j-1/2}) \right|^2.$$

Monitored: supremum norm (1.6.8), L^2 -norm (1.6.9) of discretization error $u - u_N$ (approximated by “overkill” Gaussian quadrature with 10^4 nodes)



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

③ Spline collocation on equidistant mesh, meshwidth $h > 0$ (\rightarrow Sect. 1.5.2.2)

Monitored: supremum norm (1.6.8), L^2 -norm (1.6.9) of $u - u_N$ (approximated by sampling on fine grid with 10^4 points)

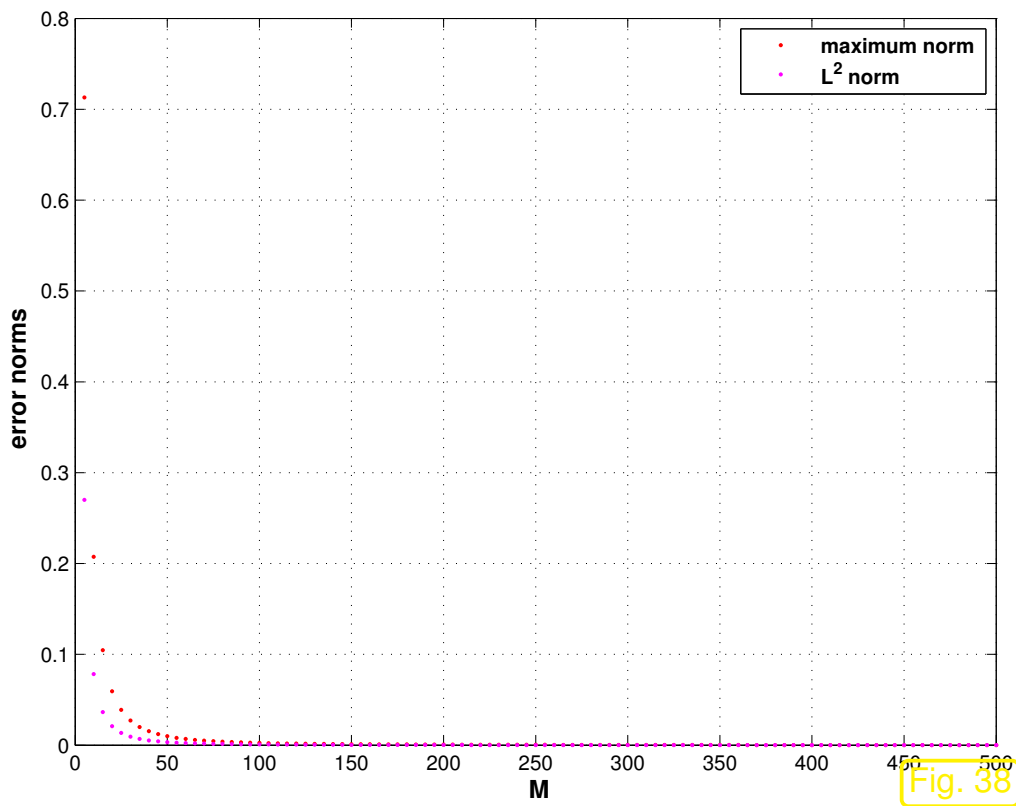


Fig. 38

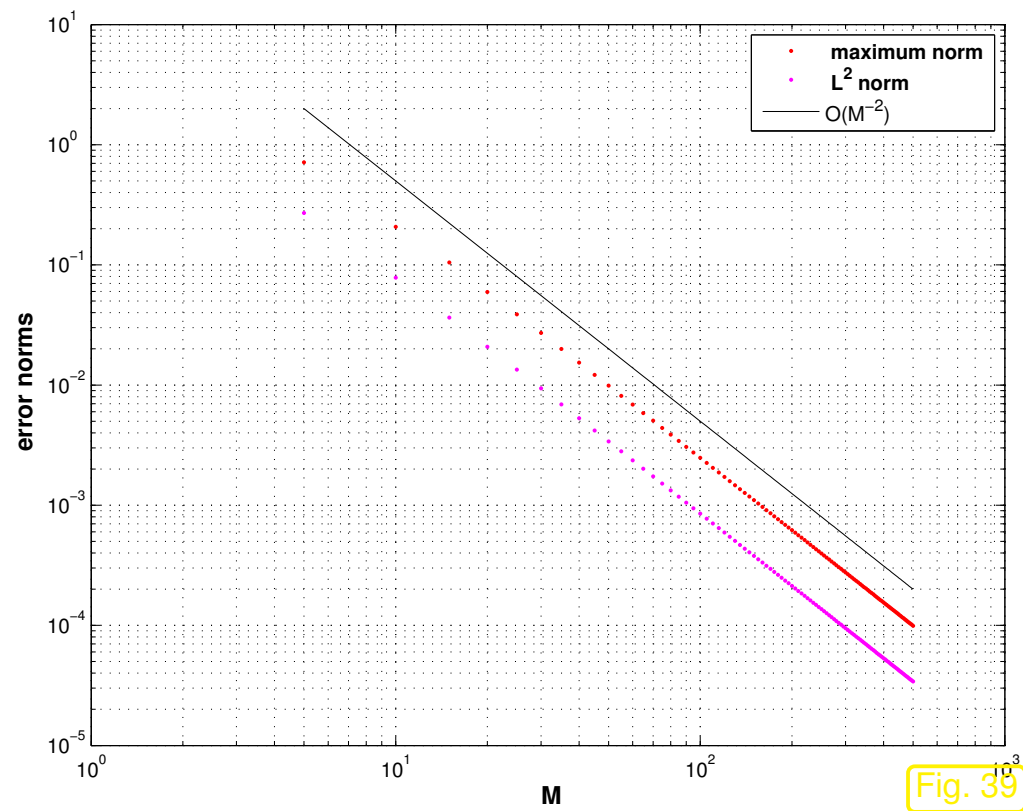


Fig. 39

④ Spectral Galerkin based on degree $p \in \mathbb{N}$ polynomials \rightarrow Sect. 1.5.1.1

Monitored: supremum norm (1.6.8), L^2 -norm (1.6.9) of discretization error $u - u_N$ (approximated by trapezoidal rule on fine grid with 10^4 points)

Spectral Galerkin error convergence

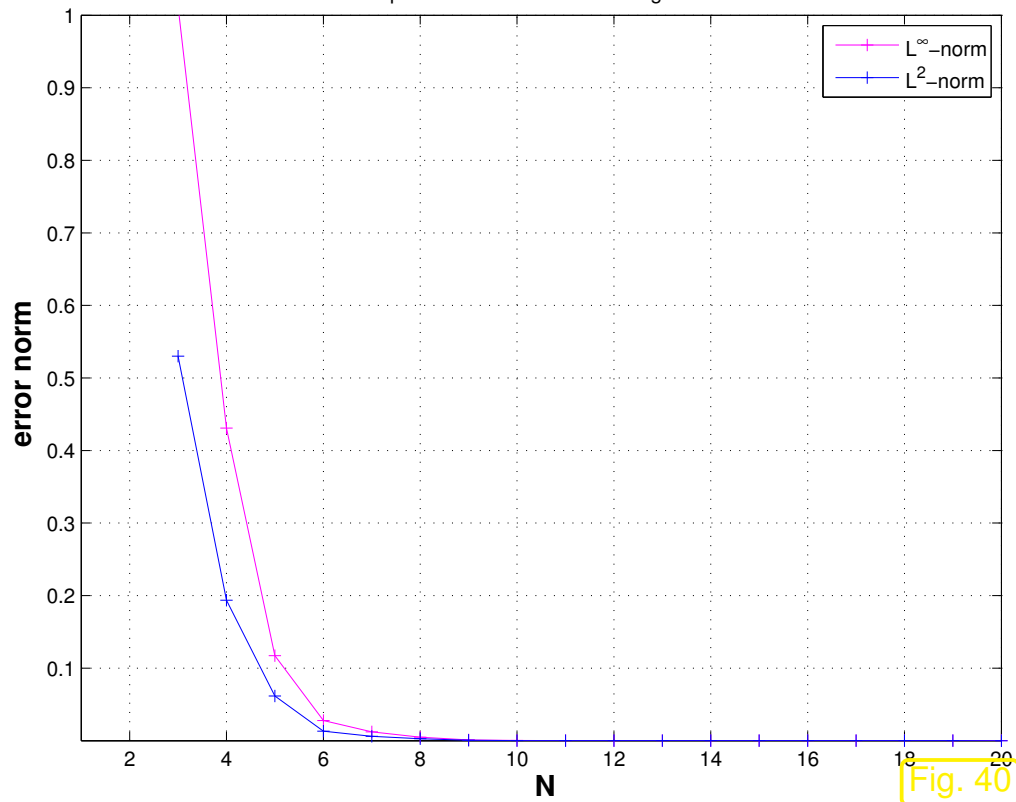


Fig. 40

Spectral Galerkin error convergence

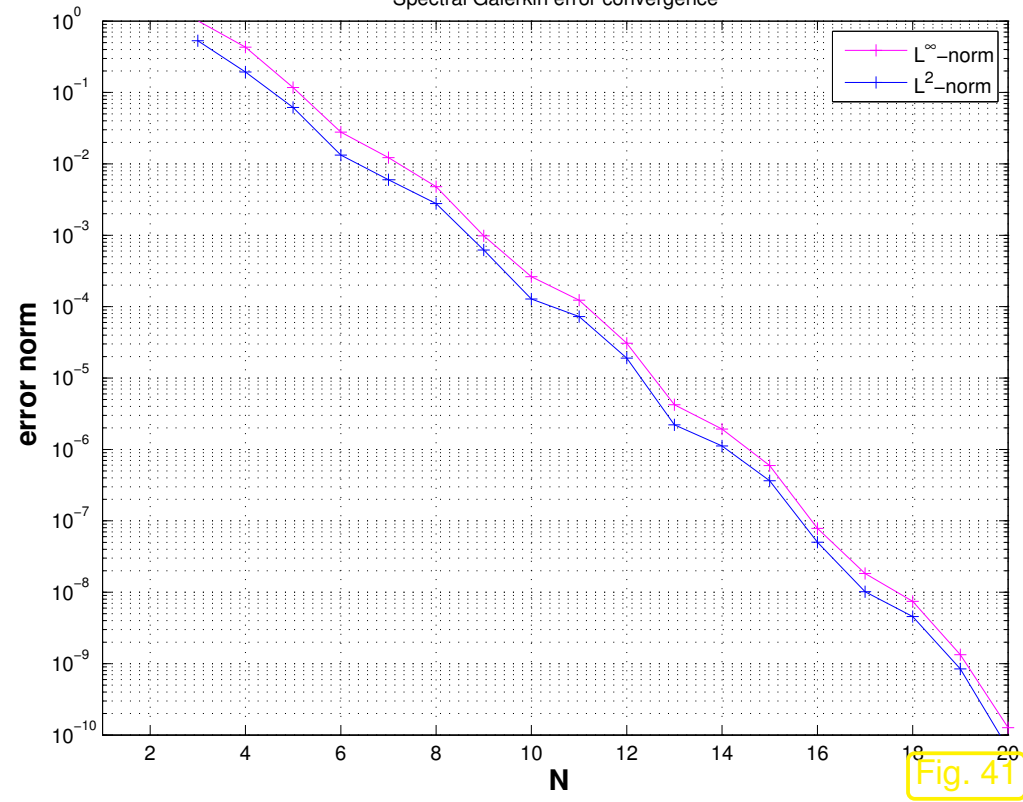


Fig. 41

Observation: \triangleright ‘Empiric convergence” in all cases
 \triangleright different qualitative behavior (of norm of discretization error)



How to compare different discretizations ?

Unified view:

Study $\|u - u_N\|$ as function of number N of unknowns (degrees of freedom)

measure for costs incurred by method

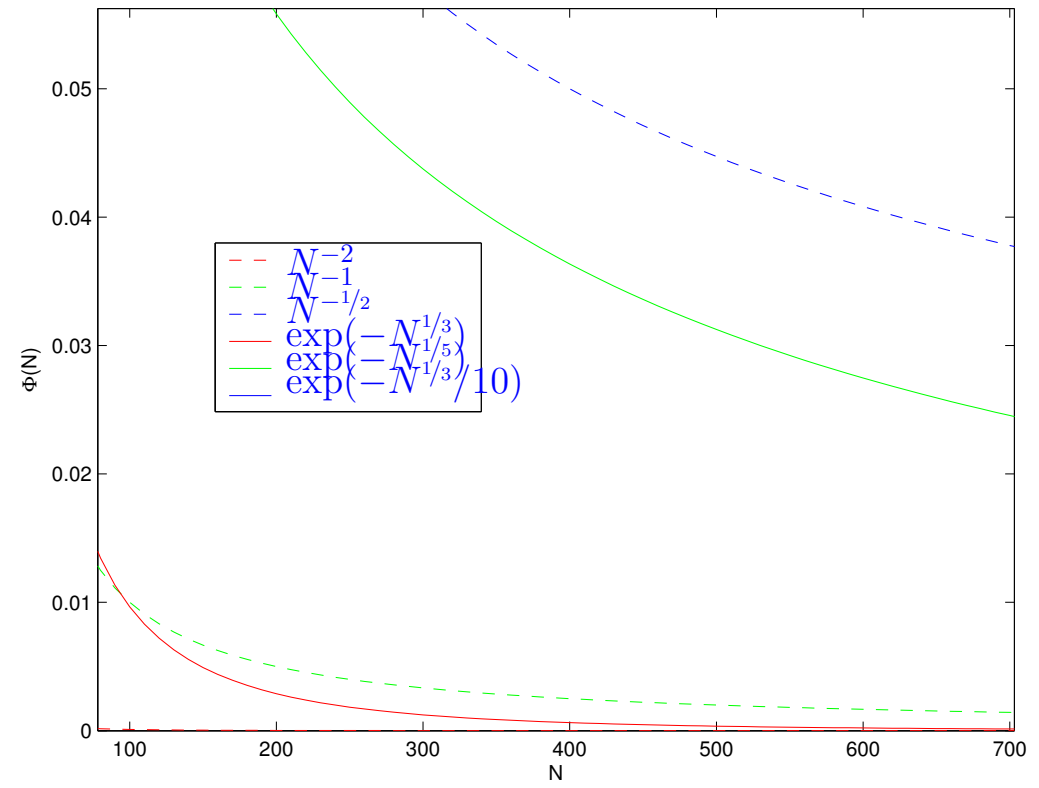
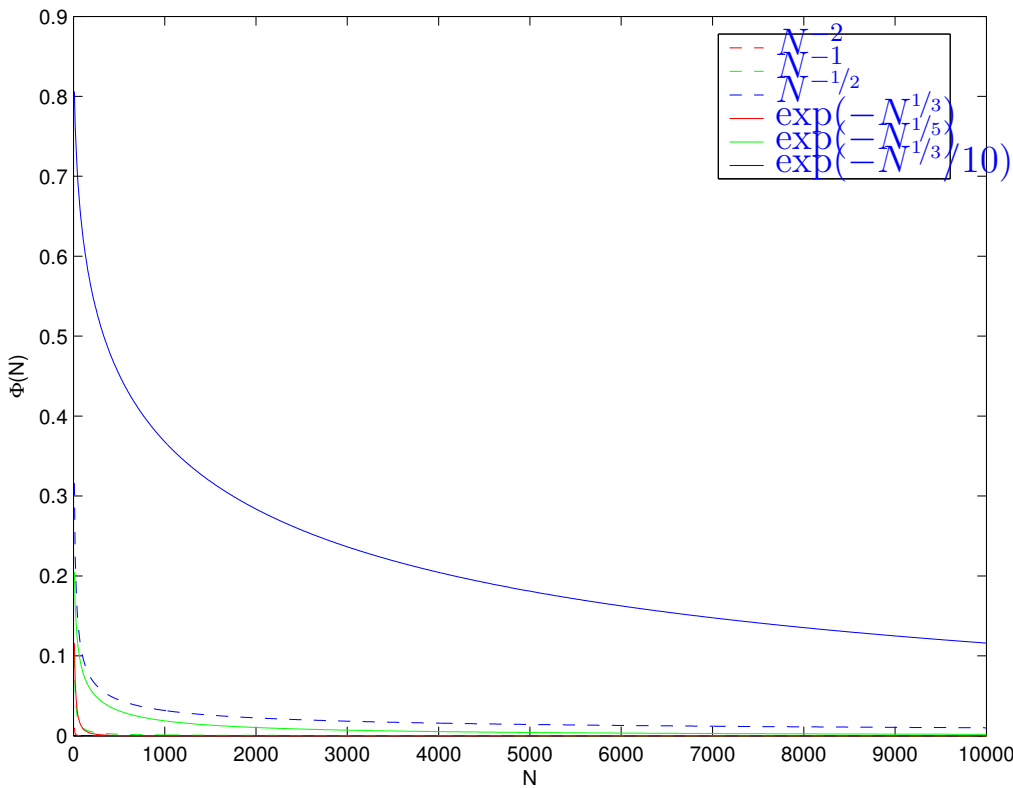
Definition 1.6.32 (Convergence rate). \rightarrow [21, Sect. 9.1], [21, Eq. 9.1.3]

$$\|u - u_N\| = O(N^{-\alpha}), \alpha > 0 \quad :\Leftrightarrow \quad \text{algebraic convergence with rate } \alpha$$

$$\|u - u_N\| = O(\exp(-\gamma N^\delta)), \text{ with } \gamma, \delta > 0 \quad :\Leftrightarrow \quad \text{exponential convergence}$$

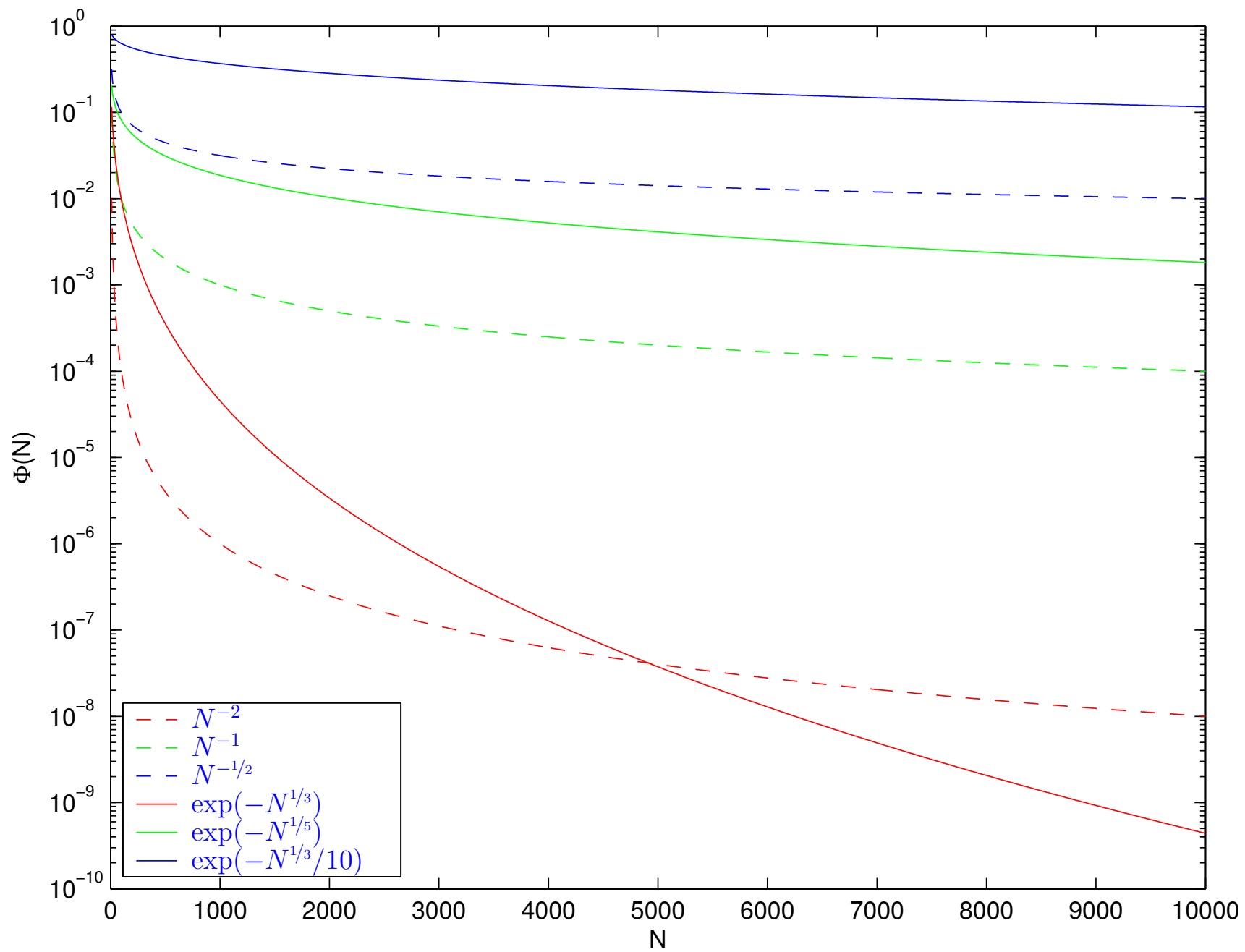
 recall notation (Landau- O):

$$f(N) = O(g(N)) \quad :\Leftrightarrow \quad \begin{array}{l} \exists N_0 > 0, \exists C > 0 \text{ independent of } N \\ \text{such that } |f(N)| \leq Cg(N) \text{ for } N > N_0. \end{array} \quad (1.6.33)$$

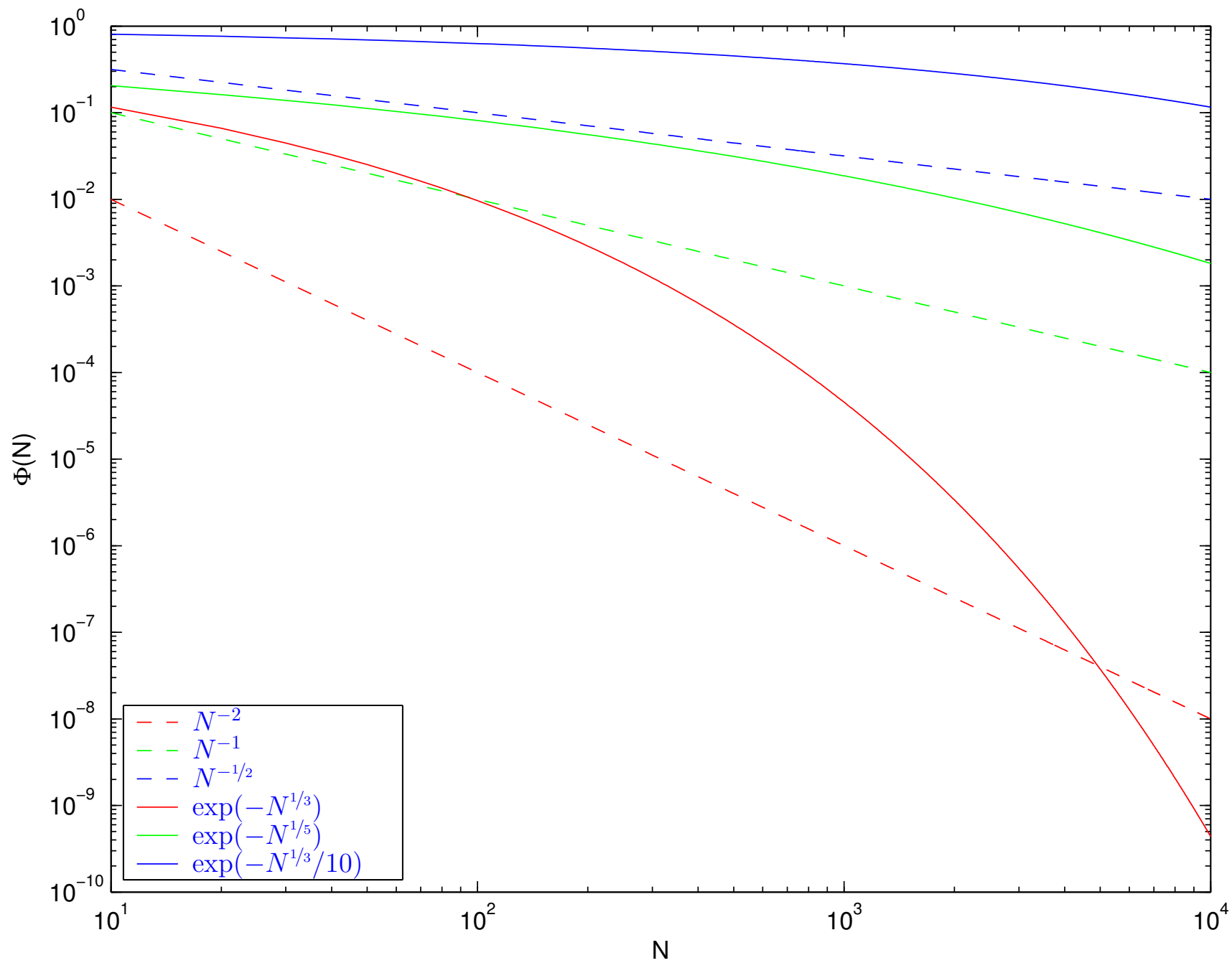


Linear plot of qualitative convergence behavior: algebraic/exponential convergence rates

Exponential convergence will always win (asymptotically)



Log-linear plot of decrease of discretization error for algebraic/exponential convergence rates



Log-log plot of decrease of discretization error for algebraic/exponential convergence rates

Remark 1.6.35 (Exploring convergence experimentally). \rightarrow [21, Rem. 9.1.4]

How to determine qualitative asymptotic convergence from raw norms of discretization error?

Given: data tuples (N_i, ϵ_i) , $i = 1, 2, 3, \dots$, $N_i \hat{=}$ problem sizes, $\epsilon_i \hat{=}$ error norms

1. Conjecture: algebraic convergence: $\epsilon_i \approx C N_i^{-\alpha}$

$$\log(\epsilon_i) \approx \log(C) - \alpha \log N_i \quad (\text{affine linear in log-log scale}).$$

➤ linear regression on data $(\log N_i, \log \epsilon_i)$, $i = 1, 2, 3, \dots$ to determine rate α .

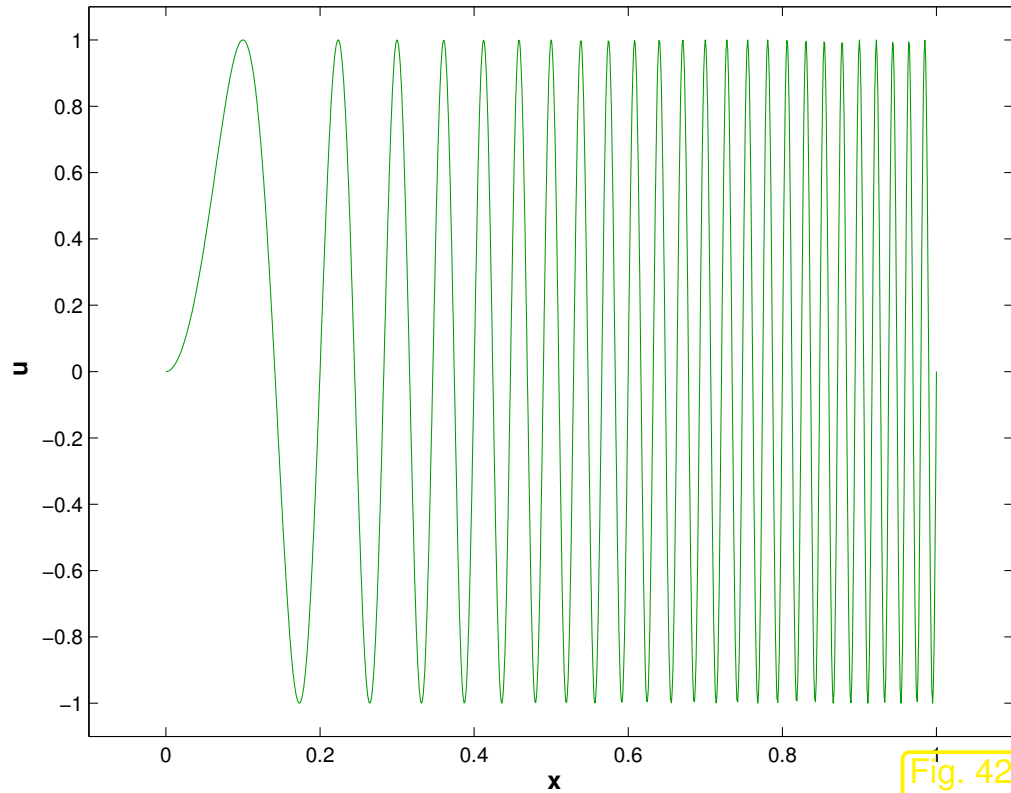
2. Conjecture: exponential convergence: $\epsilon_i \approx C \exp(-\gamma N_i^\delta)$

$$\log \epsilon_i \approx \log(C) - \gamma N_i^\delta.$$

➤ non-linear least squares fit (\rightarrow [21, Sect. 7.5]) to determine δ :

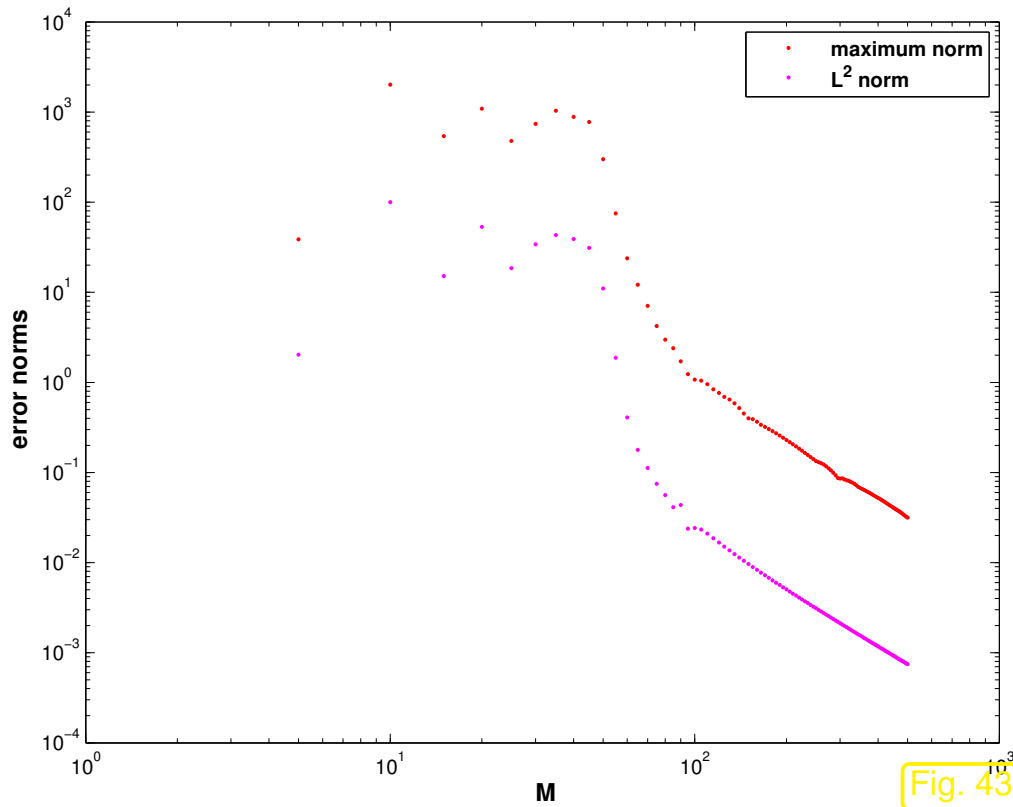
$$(c, \gamma, \delta) = \operatorname{argmin} \left\{ \sum_i |\log \epsilon_i - c + \gamma N_i^\delta|^2 \right\},$$

residual \leftrightarrow validity of conjecture. This can be done by a short MATLAB code (\rightarrow exercise)

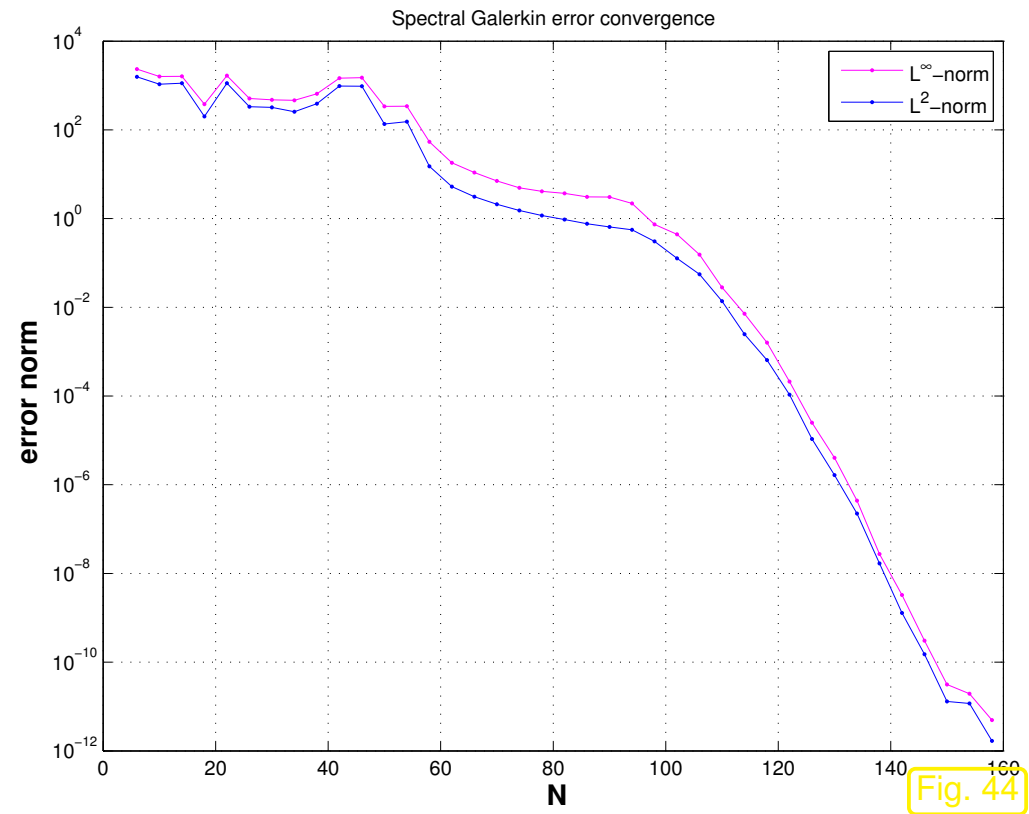
Example 1.6.36 (Asymptotic nature of convergence).

- 2-point BVP $-\frac{d^2u}{dx^2} = g(x)$, $u(0) = u(1) = 0$,
 $\Omega =]0, 1[$,
 $\triangleleft u(x) = \sin(50\pi x^2)$
- ❶ finite difference discretization on equidistant
 mesh, meshwidth $h > 0$ (\rightarrow Sect. 1.5.3)
- ❷ Spectral Galerkin based on degree $p \in \mathbb{N}$
 polynomials \rightarrow Sect. 1.5.1.1

Evaluations as in Ex. 1.6.31



① Finite Difference Method



② Spectral Galerkin Method

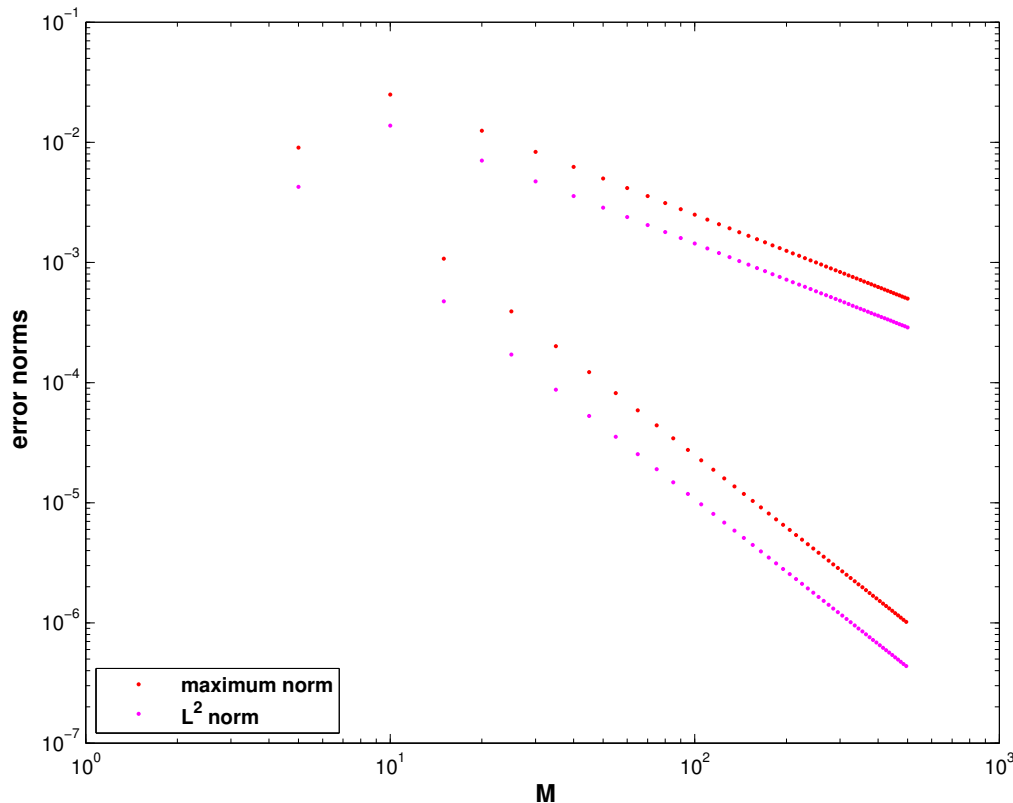
Observation: for $h \rightarrow 0$, $p \rightarrow \infty$, algebraic convergence of the finite difference solution, and exponential convergence of the spectral Galerkin solution emerge. This is the “typical” asymptotic behavior of the discretization error norms for these discretization methods.

However, the onset of asymptotic convergence occurs only for rather small meshwidth or large p , respectively, beyond thresholds that may never be reached in a computation. During a long pre-asymptotic phase the error is hardly reduced when increasing the resolution of the discretization.

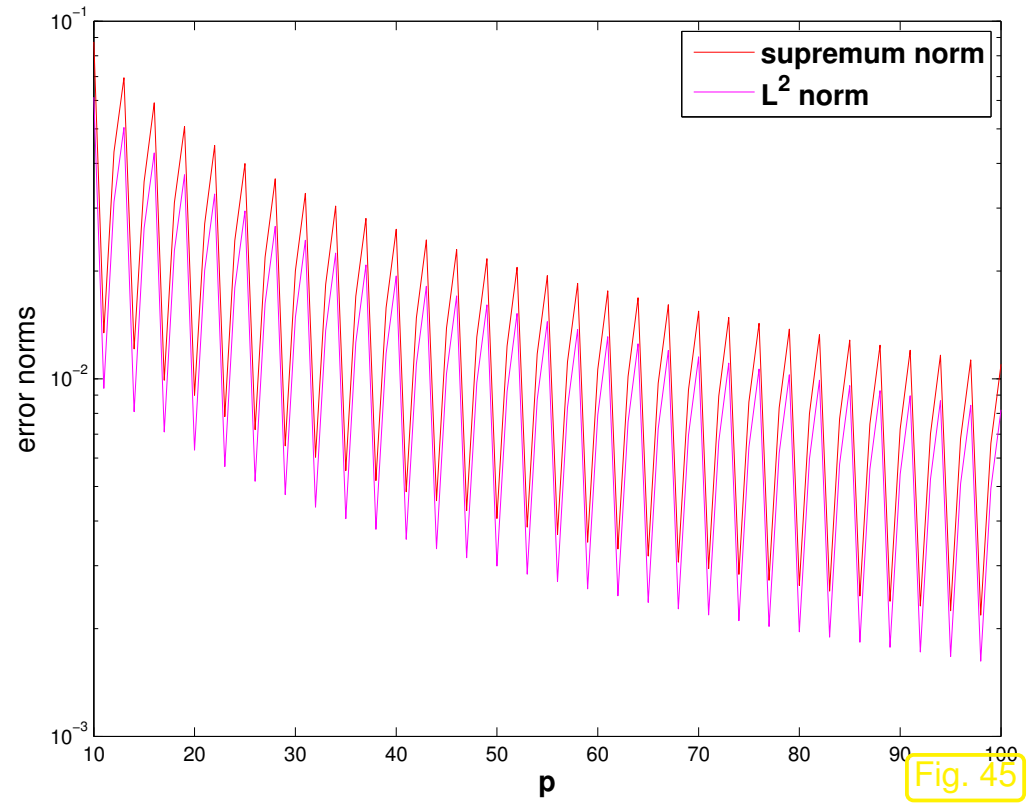
Example 1.6.37 (Convergence and smoothness of solution).

- $\Omega =]0, 1[$ (for finite differences), $\Omega =]-1, 1[$ (for spectral Galerkin), exact solution of 2-point BVP for ODE $-\frac{d^2u}{dx^2} = g(x)$,

$$u(x) = \begin{cases} \frac{3}{4} - x^2 & , \text{if } |x| < \frac{1}{2} , \\ 1 - |x| & , \text{if } |x| \geq \frac{1}{2} . \end{cases} \quad \Leftrightarrow \quad g(x) = \begin{cases} 2 & , \text{if } |x| < \frac{1}{2} , \\ 0 & \text{elsewhere} . \end{cases}$$



① Finite Difference Method



② Spectral Galerkin Method

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

- Observations:
- no more exponential convergence of spectral Galerkin
 - FD: different rate of algebraic convergence for even/odd M !



Qualitative asymptotic convergence also depends on data !

Learning Outcomes

You should be able to ...

- formulate simple mechanical problems as energy minimization problems

- derive **variational formulations** using the calculus of variations
- know that quadratic minimization problems lead to linear variational equations
- derive a **two-point boundary value problem** from a variational equation
- understand different smoothness requirements on the solutions for different problem formulations
- understand the principle of **Ritz-Galerkin discretization** and appreciate the impact of choice of basis.
- apply Ritz-Galerkin approach based on both piecewise polynomials and global polynomials to discretize variational problems in one dimension.
- obtain the linear system of equation that arises from a prescribed Ritz-Galerkin discretization of a linear variational problem.
- understand the physical relevance of the H^1 seminorm
- detect algebraic and exponential **convergence** in numerical experiments.

2

Second-order Scalar Elliptic Boundary Value Problems

Preface

The previous chapter discussed the transformation of a minimization problem on a function space via a variational problem to a differential equation. To begin with, in Sect. 2.1–Sect. 2.4, this chapter revisits this theme for models that naturally rely on function spaces over domains in two and three spatial dimensions. Thus the transformation leads to genuine partial differential equations.

Sect. 2.2 ventures into the realm of Sobolev spaces, which provide the framework for rigorous mathematical investigation of variational equations. However, we will approach Sobolev spaces as “spaces

of physically meaningful solutions” or “spaces of solutions with finite energy”. From this perspective dealing with Sobolev spaces will be reduced to dealing with their norms.

In Sect. 2.5, we change tack and consider a physical phenomenon (heat conduction) where modelling naturally leads to partial differential equations. On this occasion, we embark on a general discussion of boundary conditions in Sect. 2.6.

Then the fundamental class of second-order elliptic boundary value problems is introduced. Appealing to “intuitive knowledge” about the physical systems underlying the models, key properties of their solutions are presented in Sect. 2.7.

In 2.6 Sect. in the context of stationary heat conduction we introduce the whole range of standard boundary conditions for 2nd-order elliptic boundary value problems. Their discussion in variational context will be resumed in Sect. 2.9.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Supplementary and further reading:

An excellent *mathematical* introduction to partial differential equations is Evans' book [15]. Chapter 2 gives a very good idea about fundamental properties of various simple PDEs. Chapters 6 and 7 fit the scope of this course chapter, but go way beyond it in terms of mathematical depth.

Remark 2.0.2 (Boundary value problems (BVPs)).

The traditional concept of a boundary value problem for a partial differential equation:

Boundary value problem (BVP)

Given a partial differential operator \mathcal{L} , a **domain** $\Omega \subset \mathbb{R}^d$, a boundary differential operator \mathcal{B} , **boundary data** g , and a **source term** f , seek a function $u : \Omega \mapsto \mathbb{R}^n$ such that

$$\begin{aligned}\mathcal{L}(u) &= f \quad \text{in } \Omega, \\ \mathcal{B}(u) &= g \quad \text{on part of (or all) boundary } \partial\Omega.\end{aligned}$$

Terminology: boundary value problem is **scalar** $:\Leftrightarrow n = 1$
(in this case the unknown is a real valued function)

What does **elliptic** mean ?

Mathematical theory of PDEs distinguishes three main classes of boundary value problems (**BVPs**) for partial differential equations (**PDE**):

- **Elliptic BVPs** (\triangleright “equilibrium problems”, as discussed in Sects. 1.2.3, 2.1.1, 2.1.2)
- **Parabolic initial boundary value problems** (IBVPs) (\triangleright evolution towards equilibrium)
- **Hyperbolic IBVPs**, among them wave propagation problems and conservation laws (\triangleright transport/propagation)

The rigorous mathematical definition is complicated and often fails to reveal fundamental properties of, e.g., solutions that are intuitively clear against the backdrop of the physics modelled by a certain PDE. Further discussion of classification in [4, § 1] and [18, Ch. 1].

\triangleright In the spirit of Sect. 1.1

Structural properties of a BPV inherited from the modelled system are more important than formal mathematical classification.

2.1 Equilibrium models

We only consider stationary systems. Then, frequently, see Sect. 1.2.2

equilibrium = minimal energy configuration of a system

Example: elastic string model of Sect. 1.2 (minimization of energy functional $J(\mathbf{u})$, see (1.2.26))

Now we study minimization problems for energy functional on spaces of functions $\Omega \mapsto \mathbb{R}$, where $\Omega \subset \mathbb{R}^d$ is a bounded (spatial) domain and $d = 2, 3$.

2.1.1 Taut membrane

Recall: energy functional for pinned *taut* string under gravitational load \hat{g} , see (1.4.9), in terms of displacement, see Fig. 17:

$$J(u) := \frac{1}{2} \int_a^b \hat{\sigma}(x) \left| \frac{du}{dx}(x) \right|^2 - \hat{g}(x)u(x) \, dx, \quad \begin{aligned} u &\in C_{\text{pw}}^1([a, b]), \\ u(a) &= u_a, \quad u(b) = u_b. \end{aligned}$$

“2D generalization” of an elastic string \triangleright elastic membrane.

Taut drum membranes



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Shape of membrane



Graph of $u : \Omega \mapsto \mathbb{R}$

“membrane” on spatial domain $\Omega =]0, 1[^2$

(--- $\hat{=}$ boundary data)

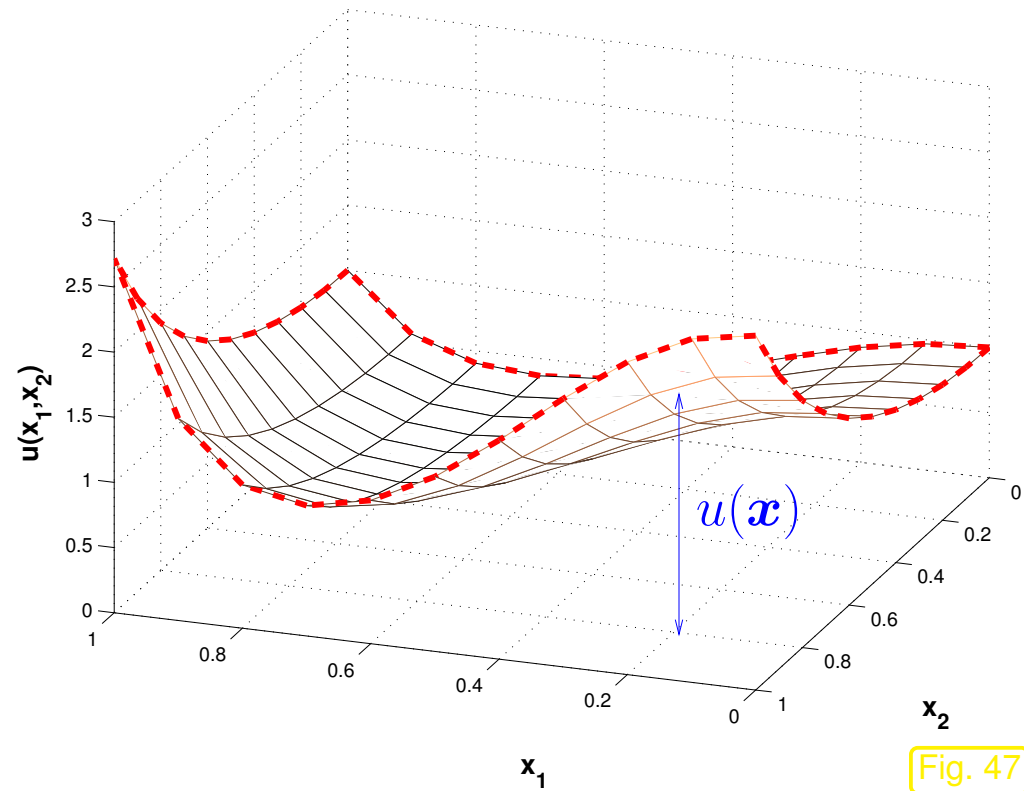



Fig. 47

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 2.1.1 (Spatial domains).

General assumptions on spatial domains $\Omega \subset \mathbb{R}^d$:

 $d = 1, 2, 3 \hat{=}$ “dimension” of domain

- Ω is bounded

$$\text{diam}(\Omega) := \sup\{\|\mathbf{x} - \mathbf{y}\| : \mathbf{x}, \mathbf{y} \in \Omega\} < \infty ,$$

- Ω has piecewise smooth boundary $\partial\Omega$

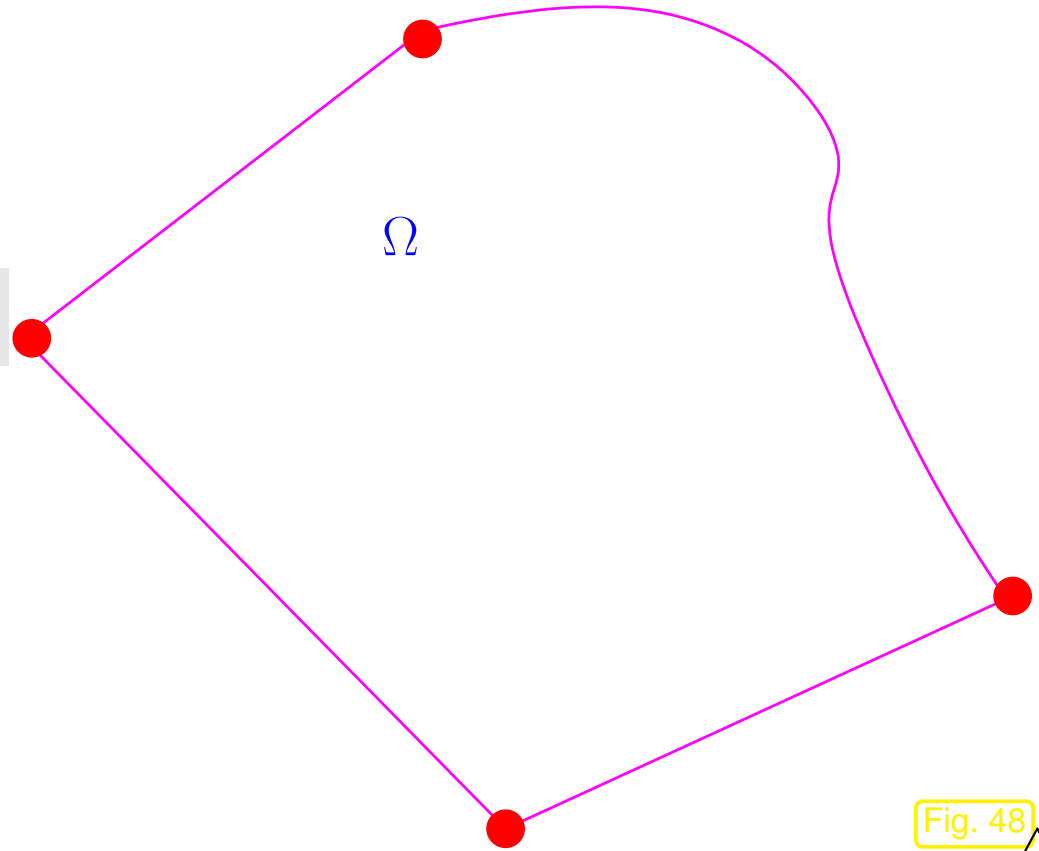



Fig. 48 

Pinning conditions (**boundary conditions**), cf. (1.2.1), (1.4.13):

$$\begin{array}{l} \text{fix} \quad u(\mathbf{x}) = g(\mathbf{x}) \quad \mathbf{x} \in \partial\Omega \\ \quad \quad \quad \updownarrow \\ \quad \quad \quad u|_{\partial\Omega} = g \quad \text{on } \partial\Omega . \end{array} \quad \text{for some } g \in C^0(\partial\Omega) . \quad (2.1.2)$$

 notation: $\partial\Omega \hat{=}$ boundary of Ω

(2.1.2) means that the displacement of the membrane over $\partial\Omega$ is provided by a prescribed *continuous* function $g : \partial\Omega \mapsto \mathbb{R}$: the membrane is clamped into a rigid frame.

Intuition: g has to be continuous, unless the membrane is to be torn!
(Further discussion in Rem. 2.9.7)

configuration space $V = \left\{ \begin{array}{l} \text{continuous functions } u \in C^0(\Omega), \\ \text{with } u|_{\partial\Omega} = g. \end{array} \right\}$

Think of the membrane as a grid of taut strings. Together with Rem. 1.4.12 this justifies the following expression for its total potential energy.

Potential energy of a taut membrane (described by $u \in C^0(\Omega)$) under vertical loading:

$$J_M(u) := \int_{\Omega} \frac{1}{2} \sigma(\mathbf{x}) \|\mathbf{grad} u\|^2 - f(\mathbf{x})u(\mathbf{x}) \, d\mathbf{x}, \tag{2.1.3}$$

elastic energy

potential energy in force field

Recall the definition of the **gradient** of a function $F : \Omega \subset \mathbb{R}^d \mapsto \mathbb{R}, F(\mathbf{x}) = F(x_1, \dots, x_d)$, see [32, Kap. 7], [21, Eq. 5.1.8]:

$$\mathbf{grad} F(\mathbf{x}) := \begin{pmatrix} \frac{\partial F}{\partial x_1} \\ \vdots \\ \frac{\partial F}{\partial x_d} \end{pmatrix} .$$

Note: the gradient at \mathbf{x} is a column vector of first *partial derivatives*,
read $\mathbf{grad} F(\mathbf{x})$ as $(\mathbf{grad} F)(\mathbf{x})$; $\mathbf{grad} F$ is a vector valued function $\Omega \mapsto \mathbb{R}^d$.

Also in use (but not in this course) is the “ ∇ -notation”: $\nabla F(\mathbf{x}) := \mathbf{grad} F(\mathbf{x})$.

Note that

$$\sigma(\mathbf{x}) \|\mathbf{grad} u\|^2 = \sigma(x_1, x_2) \left| \frac{\partial u}{\partial x_1}(x_1, x_2) \right|^2 + \sigma(x_1, x_2) \left| \frac{\partial u}{\partial x_2}(x_1, x_2) \right|^2 ,$$

which justifies calling the taut membrane a “two-dimensional string under tension”.

- with
- $u : \Omega \mapsto \mathbb{R} \hat{=} \text{displacement function, see Fig. 47, } [u] = \text{m,}$
 - $f : \Omega \mapsto \mathbb{R} \hat{=} \text{force density (pressure), } [f] = \text{N m}^{-2},$
 - $\sigma : \Omega \mapsto \mathbb{R}^+ \hat{=} \text{stiffness, } [\sigma] = \text{J.}$

Displacement of taut membrane in **equilibrium** achieves minimal potential energy, *cf.* (1.2.26)

$$u_* = \underset{u \in V}{\operatorname{argmin}} J_M(u). \quad (2.1.4)$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 2.1.6 (Minimal regularity of membrane displacement).

Smoothness required for u, f to render $J_M(u)$ from (2.1.3) meaningful, *cf.* Sect. 1.3.2:

- $u \in C_{\text{pw}}^1(\Omega)$ is sufficient for displacement u ,
- $\sigma, f \in C_{\text{pw}}^0(\Omega)$ already allows integration.

2.1.2 Electrostatic fields

- metal body in metal box
- prescribed voltage drop body—box

Sought: electric field $\mathbf{E} : \Omega \mapsto \mathbb{R}^3$ in $\Omega \subset \mathbb{R}^3$
($\Omega \hat{=}$ blue region \triangleright)

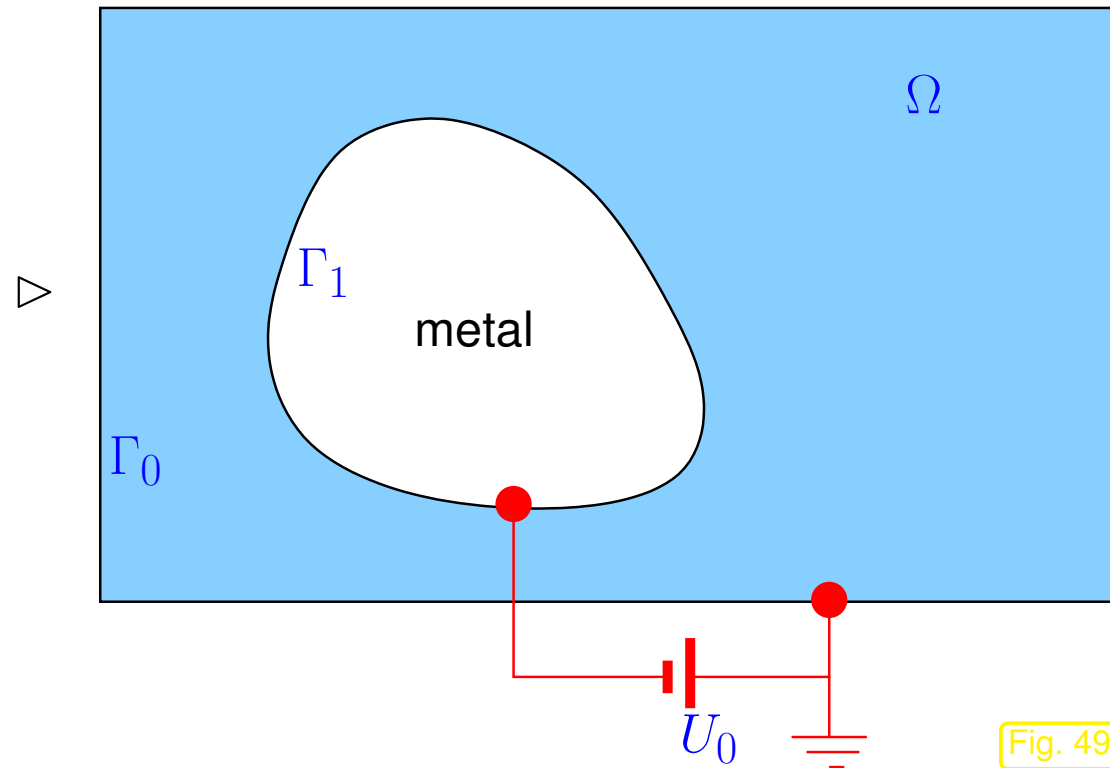
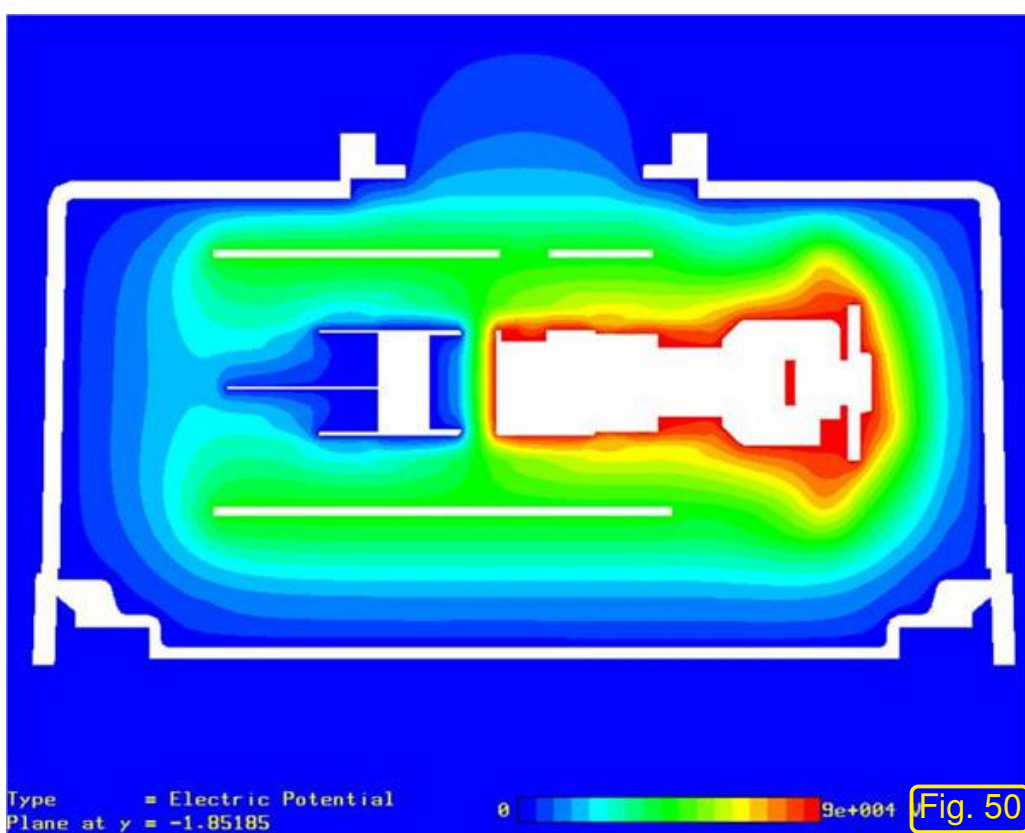


Fig. 49



From Maxwell's equations, static case:

$$\mathbf{E} = -\operatorname{grad} u, \quad (2.1.7)$$

where $u : \Omega \mapsto \mathbb{R} \hat{=} \text{electric (scalar) potential}$,
 $[u] = 1\text{V}$

◁ Electric potential in technical device

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Electromagnetic field energy: (electrostatic setting)

$$J_E(\mathbf{E}) = \frac{1}{2} \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{E}(\mathbf{x})) \cdot \mathbf{E}(\mathbf{x}) \, d\mathbf{x} = \frac{1}{2} \int_{\Omega} (\epsilon(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} u(\mathbf{x}) \, d\mathbf{x}, \quad (2.1.8)$$

where $\epsilon : \Omega \mapsto \mathbb{R}^{3,3} \hat{=} \text{dielectric tensor}$, $\epsilon(\mathbf{x})$ symmetric, $[\epsilon] = \frac{\text{As}}{\text{Vm}}$.

- Symmetry of the dielectric tensor can always be assumed: if $\epsilon(\mathbf{x})$ was not symmetric, then replacing it with $\frac{1}{2}(\epsilon(\mathbf{x})^T + \epsilon(\mathbf{x}))$ will yield exactly the same field energy.
- In terms of partial derivatives and tensor components $\epsilon(\mathbf{x}) = (\epsilon_{ij})_{i,j=1}^3$ we have

$$(\epsilon(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} u(\mathbf{x}) = \sum_{i=1}^3 \sum_{j=1}^3 \epsilon_{ij}(\mathbf{x}) \frac{\partial u}{\partial x_i}(\mathbf{x}) \frac{\partial u}{\partial x_j}(\mathbf{x}) .$$

Fundamental property of dielectric tensor (for “normal” materials):

$$\exists 0 < \epsilon^- \leq \epsilon^+ < \infty: \quad \epsilon^- \|\mathbf{z}\|^2 \leq (\epsilon(\mathbf{x})\mathbf{z}) \cdot \mathbf{z} \leq \epsilon^+ \|\mathbf{z}\|^2 \quad \forall \mathbf{z} \in \mathbb{R}^3, \forall \mathbf{x} \in \Omega . \quad (2.1.9)$$

Terminology: (2.1.9) \Leftrightarrow ϵ is bounded and **uniformly positive definite**

Definition 2.1.12 (Uniformly positive (definite) tensor field).

An matrix-valued function $\mathbf{A} : \Omega \mapsto \mathbb{R}^{n,n}$, $n \in \mathbb{N}$, is called *uniformly positive definite*, if

$$\exists \alpha^- > 0: \quad (\mathbf{A}(\mathbf{x})\mathbf{z}) \cdot \mathbf{z} \geq \alpha^- \|\mathbf{z}\|^2 \quad \forall \mathbf{z} \in \mathbb{R}^n \quad (2.1.13)$$

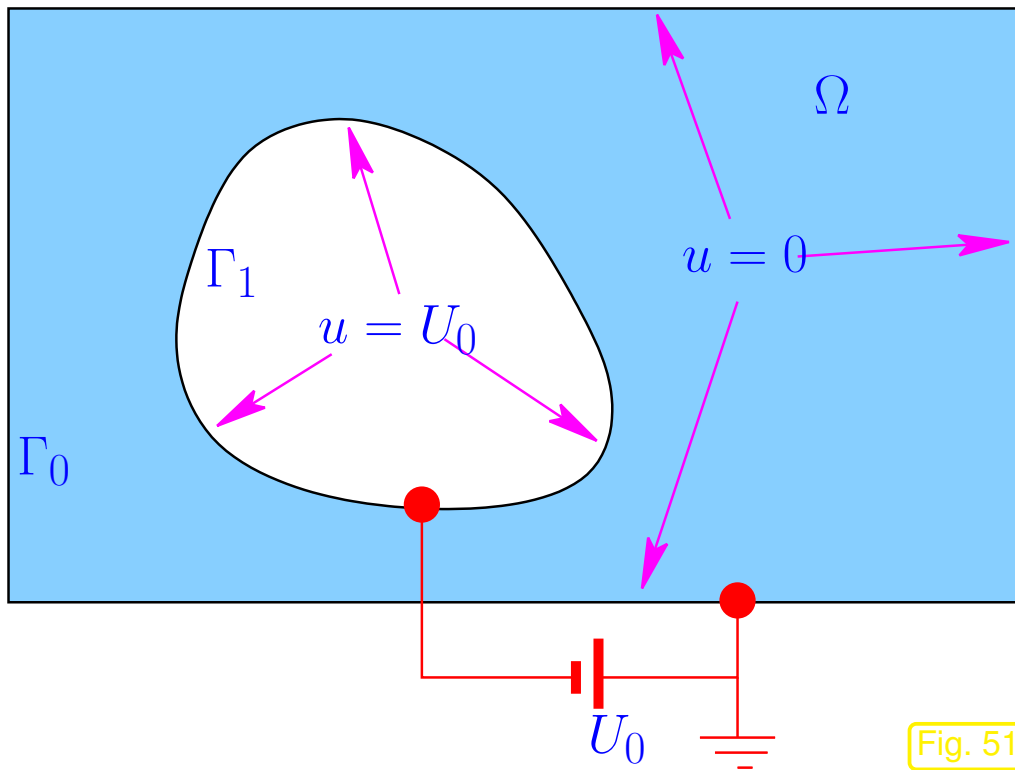
for almost all $\mathbf{x} \in \Omega$, that is, only with the exception of a set of volume zero.

If $\mathbf{A}(\mathbf{x})$ is symmetric, then we have the equivalence, cf. [21, Rem. 5.1.19],

$$(2.1.13) \Leftrightarrow \mathbf{A}(\mathbf{x}) \text{ s.p.d. } (\rightarrow [21, \text{Def. 2.7.9}]) \quad \text{and} \quad \lambda_{\min}(\mathbf{A}(\mathbf{x})) \geq \alpha^- .$$

What is the set/space V of admissible electric scalar potentials ?

Recall: in electrostatics surfaces of conducting bodies are *equipotential surfaces*



In the situation of Fig. 49:

Boundary conditions

$$\begin{aligned} u &= 0 \quad \text{on } \Gamma_0, \\ u &= U_0 \quad \text{on } \Gamma_1. \end{aligned} \tag{2.1.14}$$

$$V = \left\{ u \in C_{pw}^1(\Omega), u \text{ satisfies (2.1.14)} \right\} .$$

to render $J_E(u)$ well defined, cf. Sect. 1.3.2.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Below, the notation $u = U$ will designate the boundary conditions (2.1.14).

Equilibrium condition in electrostatic setting: minimal electromagnetic field energy

$$u_* = \operatorname{argmin}_{u \in V} J_E(u) . \tag{2.1.15}$$

2.1.3 Quadratic minimization problems

Structure of minimization problems (equilibrium problems) encountered above:

$$\text{Sect. 2.1.1} \quad \triangleright \quad u_* = \operatorname{argmin}_{\substack{u \in C_{\text{pw}}^1(\Omega) \\ u=g \text{ on } \partial\Omega}} \underbrace{\frac{1}{2} \int_{\Omega} \sigma(\mathbf{x}) \|\mathbf{grad} u(\mathbf{x})\|^2 - f(\mathbf{x})u(\mathbf{x}) \, d\mathbf{x}}_{=: J_M(u), \text{ see (2.1.3)}}, \quad (2.1.16)$$

$$\text{Sect. 2.1.2} \quad \triangleright \quad u_* = \operatorname{argmin}_{\substack{u \in C_{\text{pw}}^1(\Omega) \\ u=U \text{ on } \partial\Omega}} \underbrace{\frac{1}{2} \int_{\Omega} (\boldsymbol{\epsilon}(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{grad} u(\mathbf{x}) \, d\mathbf{x}}_{=: J_E(u), \text{ see (2.1.8)}}. \quad (2.1.17)$$

Evidently, (2.1.16) and (2.1.17) share a common structure. It is the *same* structure we have already come across in the minimization problem (1.4.2) for the taut string model in Sect. 1.4.

Definition 2.1.18 (Quadratic functional).

A *quadratic functional* on a real vector space V_0 is a mapping $J : V_0 \mapsto \mathbb{R}$ of the form

$$J(u) := \frac{1}{2}\mathbf{a}(u, u) - \ell(u) + c, \quad u \in V_0, \quad (2.1.19)$$

where $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ is a *symmetric bilinear form* (\rightarrow Def. 1.3.23), $\ell : V_0 \mapsto \mathbb{R}$ a *linear form*, and $c \in \mathbb{R}$.

Recall: A bilinear form $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ is *symmetric*, if

$$\mathbf{a}(u, v) = \mathbf{a}(v, u) \quad \forall u, v \in V_0. \quad (2.1.21)$$

If $V_0 = \mathbb{R}^N$ (finite-dimensional case), then a quadratic functional has the general representation

$$J(\mathbf{u}) = \mathbf{u}^T \mathbf{A} \mathbf{u} - \mathbf{b}^T \mathbf{u} + c, \quad \mathbf{A} = \mathbf{A}^T \in \mathbb{R}^{N,N}, \quad \mathbf{b} \in \mathbb{R}^N, \quad c \in \mathbb{R}. \quad (2.1.23)$$

Reminder: quadratic functionals of this forms occur in derivation of steepest descent and conjugate gradient methods for linear systems of equations, see [21, Sect. 5.1.1].

Definition 2.1.26 (Quadratic minimization problem).

A minimization problem

$$w_* = \operatorname{argmin}_{w \in V_0} J(w)$$

is called a **quadratic minimization problem**, if J is a quadratic functional on a real vector space V_0 .

Hey, both (2.1.16) and (2.1.17) are no genuine quadratic minimization problems, because they are posed over affine spaces (= “vector space + offset function”, cf. (1.3.24))!

“Offset function trick”, cf. (1.3.32), resolves the mismatch: for quadratic form J from (2.1.19)

$$\begin{aligned} J(u + u_0) &= \frac{1}{2} \mathbf{a}(u + u_0, u + u_0) - \ell(u + u_0) + c \\ &= \frac{1}{2} \mathbf{a}(u, u) + \underbrace{\mathbf{a}(u, u_0) - \ell(u)}_{=:\tilde{\ell}(u)} + \underbrace{+\frac{1}{2} \mathbf{a}(u_0, u_0) - \ell(u_0) + c}_{=:\tilde{c}} =: \tilde{J}(u), \end{aligned}$$

due to the bilinearity of \mathbf{a} and the linearity of ℓ .

$$\blacktriangleright \operatorname{argmin}_{u \in u_0 + V_0} J(u) = u_0 + \operatorname{argmin}_{w \in V_0} J(w + u_0) = u_0 + \operatorname{argmin}_{w \in V_0} \tilde{J}(w). \quad (2.1.27)$$

For a discussion of quadratic functionals on $\mathbb{R}^n \rightarrow [21, \text{Sect. 5.1.1}]$

Both (2.1.16) and (2.1.17) involve quadratic functionals. To see this apply the “offset function trick” from (2.1.27) in this concrete case: write $u = u_0 + w$ with an offset function u_0 that satisfies the boundary conditions and $w \in C_{0,\text{pw}}^1(\Omega)$, cf. (1.3.32).

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

(2.1.16) \Rightarrow **quadratic minimization problem** (\rightarrow Def. 2.1.26) with, cf. (2.1.19),

$$\mathbf{a}(w, v) = \int_{\Omega} \sigma(\mathbf{x}) \mathbf{grad} w(\mathbf{x}) \cdot \mathbf{grad} v(\mathbf{x}) d\mathbf{x}, \quad \ell(v) := \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) d\mathbf{x} - \mathbf{a}(u_0, v). \quad (2.1.30)$$

(2.1.17) \Rightarrow **quadratic minimization problem** (\rightarrow Def. 2.1.26) with, cf. (2.1.19),

$$\mathbf{a}(w, v) = \int_{\Omega} \mathbf{grad} w(\mathbf{x})^T \boldsymbol{\epsilon}(\mathbf{x}) \mathbf{grad} v(\mathbf{x}) d\mathbf{x}, \quad \ell(v) := \mathbf{a}(u_0, v). \quad (2.1.31)$$

In both cases: $V_0 = C_{0,\text{pw}}^1(\Omega)$

Can we conclude existence and uniqueness of solutions of the minimization problems (2.1.16) and (2.1.17) ?

Let us first tackle the issue of **uniqueness**:

Definition 2.1.32 (Positive definite bilinear form).

A (symmetric) bilinear form $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ on a real vector space V_0 is *positive definite*, if

$$u \in V_0 \setminus \{0\} \iff \mathbf{a}(u, u) > 0 .$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

For the special case $V_0 = \mathbb{R}^n$ any matrix $\mathbf{A} \in \mathbb{R}^{n,n}$ induces a bilinear form via

$$\mathbf{a}(\mathbf{u}, \mathbf{v}) := \mathbf{u}^T \mathbf{A} \mathbf{v} = (\mathbf{A} \mathbf{v}) \cdot \mathbf{u} , \quad \mathbf{u}, \mathbf{v} \in \mathbb{R}^n . \quad (2.1.34)$$

This connects the concept of a symmetric positive definite bilinear form to the more familiar concept of s.p.d. matrices (\rightarrow [21, Def. 2.7.9])

\mathbf{A} s.p.d. $\iff \mathbf{a}$ from (2.1.34) is symmetric, positive definite.

Definition 2.1.35 (Energy norm). *cf. [21, Def. 5.1.1]*

A symmetric positive definite bilinear form $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ (\rightarrow Def. 2.1.32) induces the **energy norm**

$$\|u\|_{\mathbf{a}} := (\mathbf{a}(u, u))^{1/2} .$$

Origin of the term “energy norm” is clear from the connection with potential energy (e.g., in membrane model and in the case of electrostatic fields, see (2.1.30), (2.1.31)), see above.

Next, we have to verify the norm axioms (N1), (N2), and (N3) from Def. 1.6.6:

- (N1) is immediate from Def. 2.1.32,
- (N2) follows from bilinearity of \mathbf{a} ,
- (N3) is a consequence of the **Cauchy-Schwarz** inequality: for any symmetric positive definite bilinear form

$$|\mathbf{a}(u, v)| \leq (\mathbf{a}(u, u))^{1/2} (\mathbf{a}(v, v))^{1/2} . \quad (2.1.37)$$

Example 2.1.38 (Quadratic functionals with positive definite bilinear form in 2D).

Analogy between quadratic functionals with positive definite bilinear form and parabolas:

$$\begin{array}{rcc}
 J(v) = \frac{1}{2}a(v, v) - \ell(v) & & \\
 \updownarrow & & \updownarrow \\
 f(x) = \frac{1}{2}ax^2 - bx & &
 \end{array}$$

with $a > 0!$

graph of quadratic functional $\mathbb{R}^2 \mapsto \mathbb{R}$

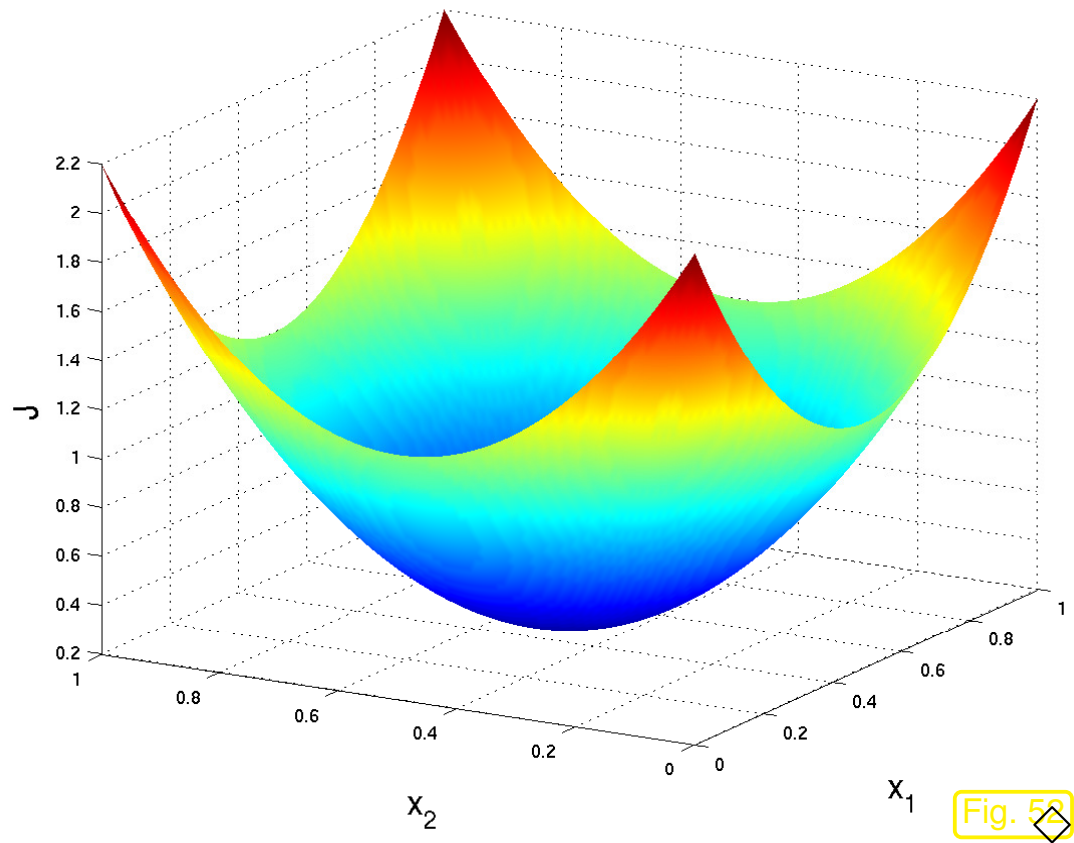


Fig. 52

Theorem 2.1.39 (Uniqueness of solutions of quadratic minimization problems).

If the bilinear form $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ is *positive definite* (\rightarrow Def. 2.1.32), then any solution of

$$u_* = \operatorname{argmin}_{u \in V_0} J(u) \quad , \quad J(u) = \frac{1}{2} \mathbf{a}(u, u) - \ell(u) + c \quad ,$$

is unique for any linear form $\ell : V_0 \mapsto \mathbb{R}$.

Proof. (indirect)

Assume that both $u_* \in V_0$ and $w_* \in V_0$, $u_* \neq w_*$ are *global minimizers* of J on V_0 .

❶ $\varphi(t) := J(tu_* + (1-t)v_*)$ has *two distinct* global minima in $t = 0, 1$.

❷ However $\varphi(t) = t^2 \underbrace{\mathbf{a}(u_* - v_*, u_* - v_*)}_{>0} + t \dots$ is a non-degenerate parabola opening towards $+\infty$, which clearly has a unique global minimum at its apex.

Contradiction between ❶ and ❷ \Rightarrow assumption wrong

□

Under the assumptions of the theorem, the quadratic functional J is **convex**, which is easily seen by considering the second derivative of the function

$$\varphi(t) := J(u + tv) \Rightarrow \ddot{\varphi}(t) = \mathbf{a}(v, v) > 0 \quad , \text{ if } v \neq 0 .$$

? Is $\mathbf{a}(u, v) := \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x}$ positive definite on $V_0 := C_{0,pw}^1(\Omega)$?

1: Since ϵ bounded and uniformly positive definite (\rightarrow Def. 2.1.12, (2.1.9))

$$\epsilon^- \int_{\Omega} \|\mathbf{grad} u(\mathbf{x})\|^2 \, d\mathbf{x} \leq \mathbf{a}(u, u) \leq \epsilon^+ \int_{\Omega} \|\mathbf{grad} u(\mathbf{x})\|^2 \, d\mathbf{x} \quad \forall u . \quad (2.1.42)$$

Hence, it is sufficient to examine the simpler bilinear form

$$d(u, v) := \int_{\Omega} \mathbf{grad} u(\mathbf{x}) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} \quad , \quad u, v \in C_{0,pw}^1(\Omega) . \quad (2.1.43)$$

2: Obviously $d(u, u) = 0 \Rightarrow \mathbf{grad} u = 0 \Rightarrow u \equiv \text{const in } \Omega$

Observe: $u = 0$ on $\partial\Omega \Rightarrow u = 0$

Zero boundary conditions are essential; otherwise one could add constants to the arguments of a without changing its value.

In a finite dimensional setting this is not a moot point, see Fig. 52 for a “visual proof”.

However, infinite dimensional spaces hold a lot of surprises and existence of solutions of quadratic minimization problems becomes a subtle issue, even if the bilinear form is positive definite.

Example 2.1.44 (Non-existence of solutions of positive definite quadratic minimization problem).

We consider the quadratic functional

$$J(u) := \int_0^1 \frac{1}{2}u^2(x) - u(x) \, dx = \frac{1}{2} \int_0^1 (u(\xi) - 1)^2 - 1 \, dx ,$$

on the space

$$V_0 := C_{0,\text{pw}}^0([0, 1])$$

It fits the abstract form from Def. 2.1.18 with

$$\mathbf{a}(u, v) = \int_0^1 u(x)v(x) \, dx \quad , \quad \ell(v) = \int_0^1 v(x) \, dx \, .$$

The function $\varphi(\xi) = \frac{1}{2}\xi^2 - \xi = \frac{1}{2}\xi(1 - 2\xi) = \frac{1}{2}(\xi - 1)^2 - \frac{1}{2}$ has a global minimum at $\xi = 1$ and $\varphi(\xi) - \varphi(1) = \frac{1}{2}(\xi - 1)^2$.

$$\blacktriangleright \quad |\eta - 1| > |\xi - 1| \quad \Rightarrow \quad \varphi(\eta) > \varphi(\xi) \, .$$

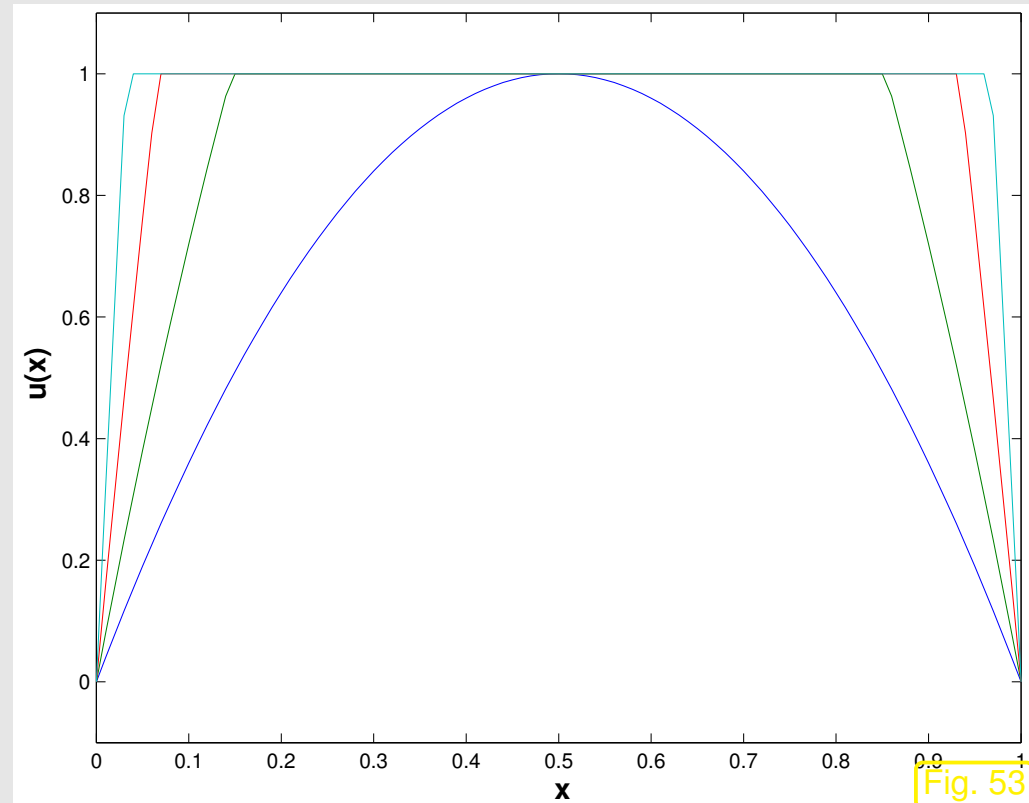
Assume that $u \in V_0$ is a global minimizer of J .

Then

$$w(x) := \min\{1, 2 \max\{u(x), 0\}\} \, , \\ 0 \leq x \leq 1 \, ,$$

is another function $\in C_{0,\text{pw}}^0([0, 1])$, which satisfies

$$u(x) \neq 1 \quad \Rightarrow \quad |w(x) - 1| < |u(x) - 1| \\ \Rightarrow \quad J(w) < J(u) \quad !$$



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Hence, whenever we think we have found a minimizer $\in C_{0,pw}^0([0, 1])$, the formula provides another eligible function for which the value of the functional is even smaller!

The problem in this example seems to be that we have chosen “too small” a function space, *c.f.* Sect. 2.2 below.



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

2.2 Sobolev spaces

Mathematical theory is much concerned about proving existence of suitably defined solutions for minimization problems. As demonstrated in Ex. 2.1.44 this can encounter profound problems.

In this section we will learn about a class of *abstract function spaces* that has been devised to deal with the question of existence of solutions of quadratic minimization problems like (2.1.16) and (2.1.17). We can only catch of glimpse of the considerations; thorough investigation is done in the mathematical field of *functional analysis*.

Consider a quadratic minimization problem (\rightarrow Def. 2.1.26) for a quadratic functional (\rightarrow Def. 2.1.18)

$$J : V_0 \mapsto \mathbb{R} \quad , \quad J(u) = \frac{1}{2}\mathbf{a}(u, u) - \ell(u) \quad ,$$

based on a symmetric positive definite (s.p.d.) bilinear form \mathbf{a} \rightarrow Def. 2.1.32.

It is clear that $J(V_0)$ is bounded from below, if

$$\exists C > 0: \quad |\ell(u)| \leq C \|u\|_{\mathbf{a}} \quad \forall u \in V_0 \quad , \quad (2.2.3)$$

where $\|\cdot\|_{\mathbf{a}}$ is the energy norm induced by \mathbf{a} , see Def. 2.1.35:

$$J(u) = \frac{1}{2}\mathbf{a}(u, u) - \ell(u) \geq \frac{1}{2} \|u\|_{\mathbf{a}}^2 - C \|u\|_{\mathbf{a}} \geq -\frac{1}{2}C^2 \quad .$$

Remark: In mathematical terms (2.2.3) means that ℓ is **continuous** w.r.t. $\|\cdot\|_a$

Under these conditions, the quadratic minimization problem for J should have a (unique, due to Thm. 2.1.39) solution, if it is considered on a space that is “large enough”.

► Note that, due to the variational formulation we have

$$|\ell(v)| = |\mathbf{a}(u_*, v)| \leq \|u_*\|_a \|v\|_a = C \|v\|_a \quad \forall v \in V_0, \quad (2.2.4)$$

where u_* is solution of the minimization problem and $C := \|u_*\|_a$.

Whenever a finite energy solution u_* to a quadratic optimization problem exists, then ℓ must be continuous with $C = \|u_*\|_a < \infty!$

Idea: for a quadratic minimization problem (\rightarrow Def. 2.1.26) with

- symmetric positive definite (s.p.d.) bilinear form \mathbf{a} ,
 - a linear form ℓ that is continuous w.r.t. $\|\cdot\|_{\mathbf{a}}$, see (2.2.3),
- posed over a function space follow the advice:

consider it on the largest space of functions for which \mathbf{a} still makes sense !

(and which complies with boundary conditions)

Choose “ $V_0 := \{\text{functions } v \text{ on } \Omega: \mathbf{a}(v, v) < \infty\}$ ”

Example 2.2.5 (Space of square integrable functions). \rightarrow Ex. 2.1.44

Quadratic functional (related to J from Ex. 2.1.44):

$$J(u) := \int_{\Omega} \frac{1}{2} |u(\mathbf{x})|^2 - u(\mathbf{x}) \, d\mathbf{x} . \quad \left(u \in C_{\text{pw}}^0(\Omega) ? \right) \quad (2.2.6)$$

We follow the above recipe, which suggests to choose

$$\blacktriangleright \quad V_0 := \{v : \Omega \mapsto \mathbb{R} \text{ integrable: } \int_{\Omega} |v(\mathbf{x})|^2 d\mathbf{x} < \infty\} \quad (2.2.7)$$

Definition 2.2.8 (Space $L^2(\Omega)$). \rightarrow Def. 1.6.9

*The function space defined in (2.2.7) is the **space of square-integrable functions** on Ω and denoted by $L^2(\Omega)$.*

It is a normed space with norm $(\|v\|_0 :=) \quad \|v\|_{L^2(\Omega)} := \left(\int_{\Omega} |v(\mathbf{x})|^2 d\mathbf{x} \right)^{1/2}.$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Notation: $L^2(\Omega)$ \leftarrow superscript “2”, because square in the definition of norm $\|\cdot\|_0$

Note: obviously $C_{\text{pw}}^0(\Omega) \subset L^2(\Omega)$.

Remark 2.2.10 (Boundary conditions and $L^2(\Omega)$).

Ex. 2.1.44 vs. Ex. 2.2.5: Crying foul! (boundary conditions $u(0) = u(1) = 0$ in Ex. 2.1.44, but none in Ex. 2.2.5!)

Consider $u \in C^0([0, 1])$ and try to impose boundary values $u_0, u_1 \in \mathbb{R}$ by “altering” u :

$$\tilde{u}(x) = \begin{cases} u(x) + (1 - nx)(u_0 - u(0)) & , \text{ for } 0 \leq x \leq \frac{1}{n} , \\ u(x) & , \text{ for } \frac{1}{n} < x < 1 - \frac{1}{n} , \\ u(x) - n(1 - \frac{1}{n} - x)(u_1 - u(1)) & , \text{ for } 1 - \frac{1}{n} < x \leq 1 . \end{cases}$$

$$\tilde{u}(0) = u_0 , \quad \tilde{u}(1) = u_1 \quad , \quad \|\tilde{u} - u\|_{L^2(]0,1[)}^2 = \frac{1}{3n}(u_0 + u_1 - u(0) - u(1)) \rightarrow 0 \quad \text{for } n \rightarrow \infty$$

Tiny perturbations of a function $u \in L^2(]0, 1[)$ (in terms of changing its L^2 -norm) can make it attain any value at $x = 0$ and $x = 1$.

Mathematically this means that the space $V = \{u \in L^2(]0, 1[) : u(0) = u(1) = 0\}$ is *not* closed in the energy space $L^2(]0, 1[)$, meaning that there exist functions which are not in V but which can be arbitrarily well approximated by elements of V . Ex. 2.1.44 makes this concrete: the solution is approximated better and better but it is never reached because the trial space is too small.

Boundary conditions cannot be imposed in $L^2(\Omega)$!



Remark 2.2.12 (Quadratic minimization problems on Hilbert spaces).

On the function space $V_0 = L^2(\Omega)$ the quadratic minimization problem for the quadratic functional from (2.2.6) can be shown to possess a solution. Instrumental in the proof is the fact that $L^2(\Omega)$ is a **Hilbert space**, that is, a *complete* normed space.

This theory is beyond the scope of this course. For more explanations see [15, Ch. 5 and Sect. 6.2].



Now consider a quadratic minimization problem for the functional, *c.f.* (2.1.16),

$$J(u) := \int_{\Omega} \frac{1}{2} \|\mathbf{grad} u\|^2 - f(\mathbf{x})u(\mathbf{x}) \, d\mathbf{x} \quad \left(u \in C_{0,pw}^1(\Omega) ? \right) \quad (2.2.13)$$

What is the natural function space for this minimization problem? Again, we follow the above recipe, which suggests that we choose

$$\blacktriangleright \quad V_0 := \{v : \Omega \mapsto \mathbb{R} \text{ integrable: } v = 0 \text{ on } \partial\Omega, \int_{\Omega} |\mathbf{grad} v(\mathbf{x})|^2 d\mathbf{x} < \infty\} \quad (2.2.14)$$

Definition 2.2.15 (Sobolev space $H_0^1(\Omega)$).

The space defined in (2.2.14) is the **Sobolev space** $H_0^1(\Omega)$ with norm

$$|v|_{H^1(\Omega)} := \left(\int_{\Omega} \|\mathbf{grad} v\|^2 d\mathbf{x} \right)^{1/2}.$$

Notation: $H_0^1(\Omega)$ ← superscript “1”, because first derivatives occur in norm
 ← subscript “0”, because zero on $\partial\Omega$

Note: $|\cdot|_{H^1(\Omega)}$ is the **energy norm** (\rightarrow Def. 2.1.35) associated with the bilinear form in the quadratic functional J from (2.2.13), *cf.* (2.1.19).

☛ See Rem. 1.6.10 for a discussion of the relevance of the energy norm.

Remark 2.2.17 (Boundary conditions in $H_0^1(\Omega)$).

Rem. 2.2.10 explained why imposing boundary conditions on functions in $L^2(\Omega)$ does not make sense.

Yet, in (2.2.14) zero boundary conditions are required for v !

Discussion parallel to Rem. 2.2.10, but now with the norm $|\cdot|_{H^1(\Omega)}$ in mind: Consider $u \in C^1([0, 1])$ and try to impose boundary values $u_0, u_1 \in \mathbb{R}$ by “altering” u :

$$\tilde{u}(x) = \begin{cases} u(x) + (1 - nx)(u_0 - u(0)) & , \text{ for } 0 \leq x \leq \frac{1}{n} , \\ u(x) & , \text{ for } \frac{1}{n} < x < 1 - \frac{1}{n} , \\ u(x) - n(1 - \frac{1}{n} - x)(u_1 - u(1)) & , \text{ for } 1 - \frac{1}{n} < x \leq 1 . \end{cases}$$

▶ $\tilde{u}(0) = u_0$, $\tilde{u}(1) = u_1$, **BUT** $|\tilde{u} - u|_{H^1([0,1])}^2 = n(u_0 + u_1 - u(0) - u(1)) \rightarrow \infty$ for $n \rightarrow \infty$

▶ Enforcing boundary values at $x = 0$ and $x = 1$ cannot be done without significantly changing the “energy” of the function.

However, the solutions of the quadratic minimization problems (2.1.16), (2.1.17) are to satisfy non-zero boundary conditions. They are sought in a larger Sobolev space, which arises from $H_0^1(\Omega)$ by dispensing with the requirement “ $v = 0$ on $\partial\Omega$ ”.

Definition 2.2.18 (Sobolev space $H^1(\Omega)$).

The Sobolev space

$$H^1(\Omega) := \{v : \Omega \mapsto \mathbb{R} \text{ integrable: } \int_{\Omega} |\mathbf{grad} v(\mathbf{x})|^2 d\mathbf{x} < \infty\}$$

is a normed function space with norm

$$\|v\|_{H^1(\Omega)}^2 := \|v\|_0^2 + |v|_{H^1(\Omega)}^2.$$

► $H^1(\Omega)$ is the “maximal function space” on which both J_M and J_E from (2.1.16), (2.1.17) are defined.

Remark 2.2.20 ($|\cdot|_{H^1(\Omega)}$ -seminorm).

Note that $|\cdot|_{H^1(\Omega)}$ alone is no longer a norm on $H^1(\Omega)$, because for $v \equiv \text{const}$ obviously $|v|_{H^1(\Omega)} = 0$, which violates (N1), cf. the discussion after Def. 1.6.19.



In the introduction to this section we saw that a quadratic functional with s.p.d. bilinear form \mathbf{a} is bounded from below, if its linear form ℓ satisfies the continuity (2.2.3). Now, we discuss this for the quadratic functional J from (2.2.13) in lieu of J_M and J_E .

The quadratic functional J from (2.2.13) involves the linear form

$$\ell(u) := \int_{\Omega} f(\mathbf{x})u(\mathbf{x}) \, d\mathbf{x} . \quad (2.2.23)$$

$f \hat{=}$ load function $\triangleright f \in C_{\text{pw}}^0(\Omega)$ should be admitted.

Crucial question:

Is ℓ from (2.2.23) continuous on $H_0^1(\Omega)$?

\Updownarrow (c.f. (2.2.3))

$\exists C > 0: |\ell(u)| \leq C|u|_{H^1(\Omega)} \quad \forall u \in H_0^1(\Omega) ?$

To begin with, we use the Cauchy-Schwarz inequality (2.1.37) for integrals, which implies

$$|\ell(u)| = \left| \int_{\Omega} f(\mathbf{x})u(\mathbf{x})d\mathbf{x} \right| \leq \left(\int_{\Omega} |f(\mathbf{x})|^2d\mathbf{x} \right)^{1/2} \left(\int_{\Omega} |u(\mathbf{x})|^2d\mathbf{x} \right)^{1/2} = \underbrace{\|f\|_0}_{< \infty} \|u\|_0 . \quad (2.2.24)$$

This reduces the problem to bounding $\|u\|_0$ in terms of $|u|_{H^1(\Omega)}$.

Theorem 2.2.25 (First Poincaré-Friedrichs inequality).

If $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, is bounded, then

$$\|u\|_0 \leq \text{diam}(\Omega) \|\mathbf{grad} u\|_0 \quad \forall u \in H_0^1(\Omega) .$$

Proof. The proof employs a powerful technique in the theoretical treatment of function spaces: exploit **density** of smooth functions (which, by itself, is a deep result).

It boils down to the insight:

In order to establish inequalities between continuous functionals on Sobolev spaces of functions on Ω it often suffices to show the target inequality for smooth functions in $C_0^\infty(\Omega)$ or $C^\infty(\Omega)$, respectively.

 notation: $C_0^\infty(\Omega) \hat{=}$ smooth functions with (compact) support (\rightarrow Def. 1.5.83) *inside* Ω

In the concrete case (note the zero boundary values inherent in the definition of $H_0^1(\Omega)$) we have to establish the first Poincaré-Friedrichs inequality for functions $u \in C_0^\infty(\Omega)$ only.

For the sake of simplicity the proof is elaborated for $d = 1$, $\Omega = [0, 1]$. It merely employs elementary results from calculus throughout, namely the Cauchy-Schwarz inequality (2.2.24) and the fundamental theorem of calculus [32, Satz 6.3.4], see (2.4.3):

$$\forall u \in C_0^\infty([0, 1]): \quad u(x) = \underbrace{u(0)}_{=0} + \int_0^x \frac{du}{dx}(\tau) \, d\tau, \quad 0 \leq x \leq 1.$$

$$\blacktriangleright \quad \|u\|_0^2 = \int_0^1 \left| \int_0^x \frac{du}{dx}(\tau) d\tau \right|^2 dx \stackrel{(2.2.24)}{\leq} \int_0^1 \left(\int_0^x 1 d\tau \cdot \int_0^x \left| \frac{du}{dx}(\tau) \right|^2 d\tau \right) dx \leq \left\| \frac{du}{dx} \right\|_0^2 .$$

Taking the square root finished the proof in 1D. □

The elementary proof in higher dimensions can be found in [18, Sect. 6.2.2] and in even greater generality in [15, Sect. 5.6.1].

\blacktriangleright If $f \in L^2(\Omega)$, then $\ell(u) = \int_{\Omega} f u d\mathbf{x}$ is a continuous linear functional on $H_0^1(\Omega)$.

Here “continuity” has to be read as

$$\boxed{\exists C > 0: |\ell(u)| \leq C|u|_{H^1(\Omega)} \quad \forall u \in H_0^1(\Omega)} \quad (2.2.3)$$

Most concrete results about Sobolev spaces boil down to relationships between their norms. The spaces themselves remain intangible, but the norms are very concrete and can be computed and manipulated as demonstrated above.

Do not be afraid of Sobolev spaces!

It is only the norms that matter for us, the ‘spaces’ are irrelevant!

Sobolev spaces = “concept of convenience”: the minimization problem seeks its own function space.

Minimization problem

$$u = \operatorname{argmin}_{v: \Omega \rightarrow \mathbb{R}} J(v)$$



“Maximal” function space
on which J is defined
(Sobolev space)

Then, why do you bother me with these uncanny “Sobolev spaces” after all ?

- Anyone involved in CSE must be able to understand mathematical publications on numerical methods for PDEs, Those regularly resort to the concept of Sobolev spaces to express their findings.
- The statement that a function belongs to a certain Sobolev space can be regarded as a concise way of describing quite a few of its essential properties.

Let us elucidate the second point:

Theorem 2.2.26. Compatibility conditions for piecewise smooth functions in $H^1(\Omega)$

Let Ω be partitioned into sub-domains Ω_1 and Ω_2 . A function that is continuously differentiable in both sub-domains and continuous up to their boundary, belongs to $H^1(\Omega)$, if and only if u is **continuous** on Ω .

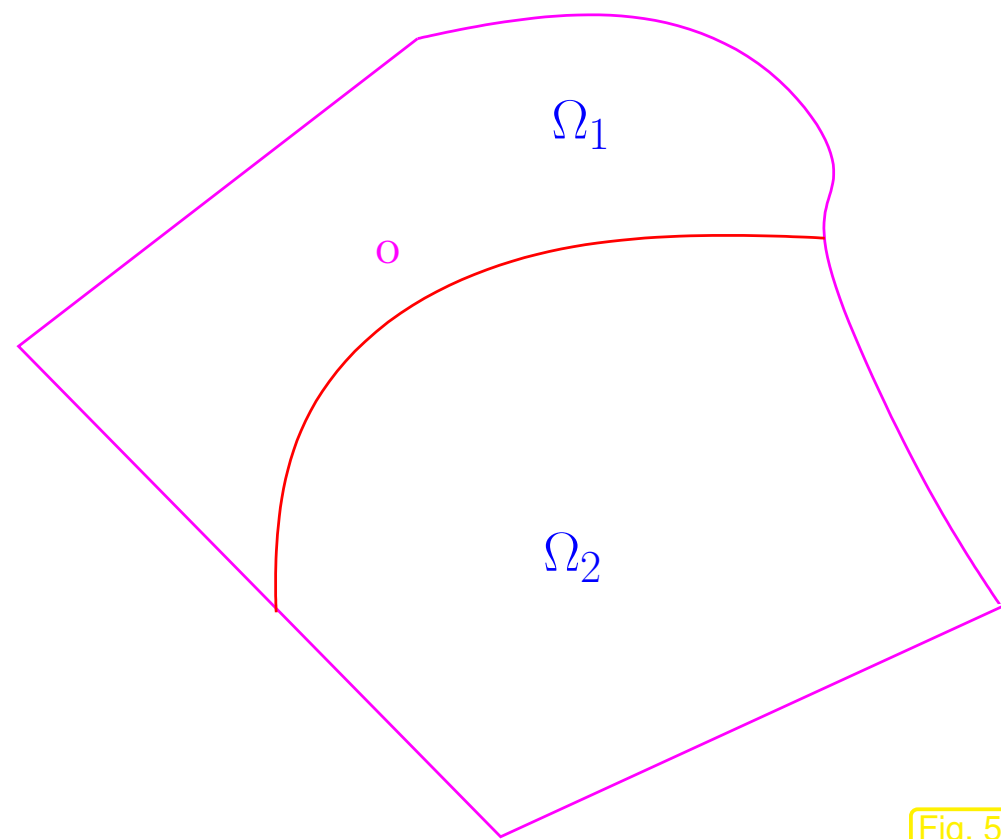
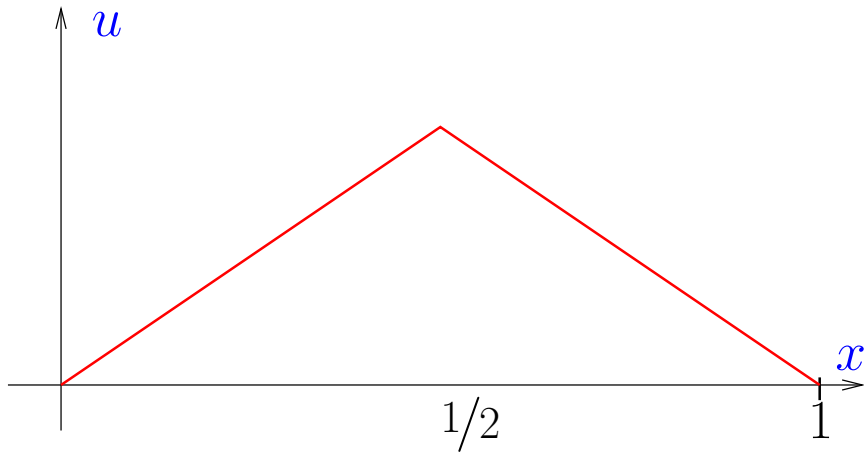


Fig. 54

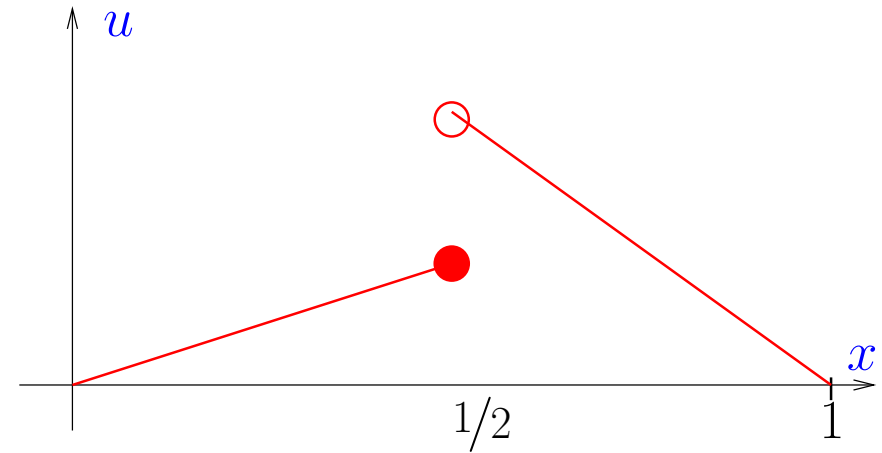
The proof of this theorem requires the notion of *weak derivatives* that will not be introduced in this course.

Example 2.2.27 (Piecewise linear functions (not) in $H_0^1(]0, 1[)$).

We conclude from Thm. 2.2.26:



$$u \in H_0^1(]0, 1[)$$



$$u \notin H_0^1(]0, 1[)$$



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

From Thm. 2.2.26 we conclude

$$C_{\text{pw}}^1([a, b]) \subset H^1(]a, b[) \text{ and } C_{0,\text{pw}}^1([a, b]) \subset H_0^1(]a, b[)$$

Thm. 2.2.26 provides a simple recipe for computing the norm $|u|_{H^1(\Omega)}$ of a piecewise C^1 -function that is continuous in all of Ω .

Corollary 2.2.28 (H^1 -norm of piecewise smooth functions).

Under the assumptions of Thm. 2.2.26 we have for a continuous, piecewise smooth function $u \in C^0(\Omega)$

$$|u|_{H^1(\Omega)}^2 = |u|_{H^1(\Omega_1)}^2 + |u|_{H^1(\Omega_2)}^2 = \int_{\Omega_1} |\mathbf{grad} u(\mathbf{x})|^2 d\mathbf{x} + \int_{\Omega_2} |\mathbf{grad} u(\mathbf{x})|^2 d\mathbf{x} .$$

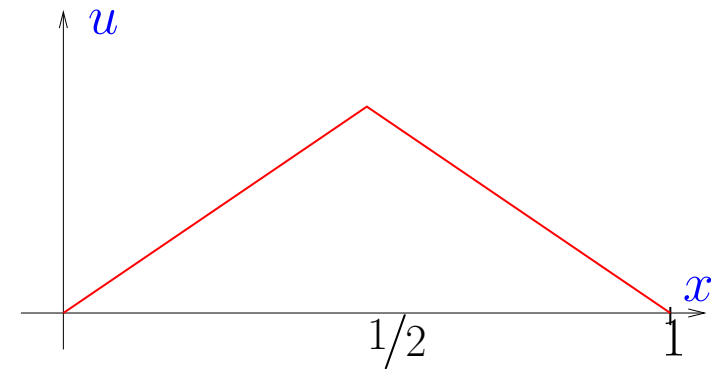
Actually, this is not new, see Sect. 1.3.2: earlier we already evaluated the elastic energy functionals (1.2.25), (1.4.2) for functions in $C_{pw}^1([0, 1])$ by “piecewise differentiation” followed by integration of the resulting discontinuous function.

Example 2.2.29 (Non-differentiable function in $H_0^1(]0, 1[)$).

$d = 1, \Omega =]0, 1[$:

“Tent function”

$$u(x) = \begin{cases} 2x & \text{for } 0 < x < 1/2, \\ 2(1-x) & \text{for } 1/2 < x < 1. \end{cases}$$



Compute

$$|u|_{H^1(\Omega)}^2 = \int_0^1 |u'(x)|^2 dx = 4 < \infty .$$

► Example for a $u \in H_0^1(]0, 1[)$, which is not globally differentiable.

Recall: we cheerfully computed the derivative of a piecewise smooth function already in Sect. 1.5.1.2 when differentiating the basis functions, *cf.* (1.5.79). Now this “reckless” computations have found their rigorous justification.

If you are still feeling uneasy when dealing with Sobolev spaces, do not hesitate to think of the following replacements

$$L^2(\Omega) \rightarrow C_{\text{pw}}^0(\Omega) \quad , \quad H_0^1(\Omega) \rightarrow C_{0,\text{pw}}^1(\Omega) \quad .$$

2.3 Variational formulations

2.3.1 Linear variational problems

Recall: derivation of variational formulation (1.4.5) from taut string minimization problem (1.4.2) in Sect. 1.4.

No surprise: (2.1.16) & (2.1.17) are amenable to the same approach:

Calculus of variations \rightarrow Sect. 1.3.1: “Directional derivative” of J_E :

$$\begin{aligned}
 J_E(u + tv) - J_E(u) &= \frac{1}{2} \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad}(u + tv)) \cdot \mathbf{grad}(u + tv) \, d\mathbf{x} \\
 &\quad - \frac{1}{2} \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} u \, d\mathbf{x} \\
 &\stackrel{(*)}{=} \frac{1}{2} \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} u + 2t(\epsilon(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} v + \\
 &\quad t^2(\epsilon(\mathbf{x}) \mathbf{grad} v) \cdot \mathbf{grad} v - (\epsilon(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} u \, d\mathbf{x} \\
 &= t \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} v \, d\mathbf{x} + O(t^2) \quad \text{for } t \rightarrow 0.
 \end{aligned}$$

(*) : due to the symmetry of $\epsilon(\mathbf{x})$: $(\epsilon \mathbf{grad} u) \cdot \mathbf{grad} v = (\epsilon \mathbf{grad} v) \cdot \mathbf{grad} u$!

$$\blacktriangleright \lim_{t \rightarrow 0} \frac{J_E(u + tv) - J_E(u)}{t} = \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x},$$

for perturbation functions

$$v \in H_0^1(\Omega), \quad \text{see Def. 2.2.15}$$

The requirement $v = 0$ on $\partial\Omega$ reflects the fact that we may not perturb u on the boundary, lest the prescribed boundary values be violated.

As explained in Sect. 1.3.1 (“idea of calculus of variations”), this leads to the following variational problem equivalent to (2.1.17)

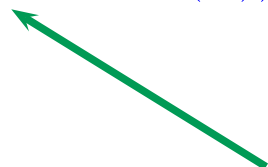
$$\begin{aligned} u \in H^1(\Omega) , \\ u = U \text{ on } \partial\Omega \end{aligned} : \int_{\Omega} (\boldsymbol{\epsilon}(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} = 0 \quad \forall v \in H_0^1(\Omega) . \quad (2.3.3)$$

For the membrane problem (2.1.16) we arrive at

$$\begin{aligned} u \in H^1(\Omega) , \\ u = g \text{ on } \partial\Omega \end{aligned} : \int_{\Omega} \sigma(\mathbf{x}) \mathbf{grad} u(\mathbf{x}) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) . \quad (2.3.4)$$

Both, (2.3.3) and (2.3.4) have a common structure, expressed in the following variational problem:

Variational formulation of 2nd-order elliptic (Dirichlet) minimization problems:

$$\begin{aligned} & u \in H^1(\Omega), \\ & u = g \text{ on } \partial\Omega : \int_{\Omega} (\alpha(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega). \end{aligned} \quad (2.3.5)$$


Symmetric uniformly positive definite **material tensor** $\alpha : \Omega \mapsto \mathbb{R}^{d,d}$

The attribute “Dirichlet” refers to a setting, in which the function u is prescribed on the entire boundary.

Some more explanations and terminology:

- $\Omega \subset \mathbb{R}^d$, $d = 2, 3 \hat{=}$ (spatial) domain, bounded, piecewise smooth boundary
- $g \in C^0(\partial\Omega) \hat{=}$ boundary values (**Dirichlet data**)
- $f \in C_{pw}^0(\Omega) \hat{=}$ loading function, source function
- $\alpha : \Omega \mapsto \mathbb{R}^{d,d} \hat{=}$ material tensor, stiffness function, diffusion coefficient
(uniformly positive definite, bounded \rightarrow Def. 2.1.12)

$$\exists 0 < \alpha^- \leq \alpha^+ : \alpha^- \|z\|^2 \leq (\alpha(x)z) \cdot z \leq \alpha^+ \|z\|^2 \quad \forall z \in \mathbb{R}^d, \quad (2.3.6)$$

for almost all $x \in \Omega$.

Rewriting (2.3.5), using **offset function** u_0 with $u_0 = g$ on $\partial\Omega$, cf. (2.1.27),

$$\begin{aligned} w \in H_0^1(\Omega) : \int_{\Omega} (\alpha(x) \operatorname{grad} w(x)) \cdot \operatorname{grad} v(x) \, dx \\ = \int_{\Omega} f(x)v(x) - (\alpha(x) \operatorname{grad} u_0(x)) \cdot \operatorname{grad} v(x) \, dx \quad \forall v \in H_0^1(\Omega). \end{aligned} \quad (2.3.7)$$



(2.3.7) is a **linear variational problem**, see Rem. 1.4.6

We can lift the above discussion to an abstract level, cf. discussion after (1.4.7):

Variational formulation of a quadratic minimization problem (\rightarrow Def. 2.1.26)

$$J(u) := \frac{1}{2} \mathbf{a}(u, u) + \ell(u) + c \quad \Rightarrow \quad J(u + tv) = J(u) + t(\mathbf{a}(u, v) + \ell(v)) + \frac{1}{2} t^2 \mathbf{a}(v, v),$$

for all $u, v \in V_0$.

► For a quadratic functional (\rightarrow Def. 2.1.26) on real vector space V_0

$$\lim_{t \rightarrow 0} \frac{J(u + tv) - J(u)}{t} = \mathbf{a}(u, v) + \ell(v) . \quad (2.3.12)$$

► **Linear variational problem** (\rightarrow Rem. 1.4.6) arising from quadratic minimization problem for functional $J(u) := \frac{1}{2}\mathbf{a}(u, u) + \ell(u) + c$:

$$w \in V_0: \quad \mathbf{a}(w, v) + \ell(v) = 0 \quad \forall v \in V_0 . \quad (2.3.13)$$

Concretely, for (2.3.7): $V_0 = H_0^1(\Omega)$ and

$$\mathbf{a}(u, v) = \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \mathbf{grad} w(\mathbf{x})) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} , \quad (2.3.14)$$

$$\ell(v) = - \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) + (\boldsymbol{\alpha}(\mathbf{x}) \mathbf{grad} u_0(\mathbf{x})) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} . \quad (2.3.15)$$

Notion of **stability** for a (linear) variational problem (2.3.13):

Lipschitz continuity of (linear) mapping data $\ell \mapsto$ solution w

\iff Is there/what is a constant $C_{\text{stab}} > 0$ such that

$$\|w\|_X \leq C_{\text{stab}} \|\ell\|_Y, \text{ where } w \text{ solves (2.3.13),} \quad (2.3.17)$$

with **suitable/relevant** norms $\|\cdot\|_X, \|\cdot\|_Y$? These norms will be suggested by the modelling background. Their choice will determine existence and value of C_{stab} .

Remark 2.3.18 (Sensitivity of linear variational problems).

Recall a notion introduced in [21, Sect. 2.5.5]:

Sensitivity of a problem (for given data) gauges
impact of small perturbations of the data on the result.

Remember: “Problem” = mapping from data space to solution space, see [21, Sect. 2.5.2].

Here, we define the “problem” as the mapping

$$\begin{cases} \{\text{linear forms on } V_0\} & \mapsto V_0 \\ \ell & \mapsto w \in V_0: \mathbf{a}(w, v) = -\ell(v) \quad \forall v \in V_0. \end{cases} \quad (2.3.20)$$

Undesirable: “sensitive dependence of solution on data”, that is small (in the norm of the data space) perturbations of ℓ translate into huge (in the norm of the solution space) or even “infinite” perturbations of the solution. In this case of an “**ill-posed problem**” inevitable data errors (e.g., due to non-exact measurements) will thwart any attempt to compute an “accurate” (in the norm of the solution space) solution.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Desirable: Lipschitz continuity of problem map with small Lipschitz constant (**well-posed problem**).

Note: the problem map (2.3.20) is **linear** and its Lipschitz constant is given by the smallest value for C_{stab} in (2.3.17).



Consider the particular choice (2.3.14).

How to choose the norms $\|\cdot\|_X$ (on solution space) and $\|\cdot\|_Y$ (on data space) ?

Norm on solution space: **energy norm**: $\|\cdot\|_a$

Norm on r.h.s: Mean square norm (L^2 -norm, \rightarrow Def. 2.2.8) for f ,
 H^1 -semi-norm (\rightarrow Def. 2.2.18) for u_0

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

What will be the impact of a perturbation of ℓ , if we use these norms?

First use the Cauchy-Schwarz inequality (2.2.24) and the uniform positivity (\rightarrow Def. 2.1.12) of α , see (2.3.6):

$$\begin{aligned} |\ell(v)| &\leq \|f\|_0 \|v\|_0 + \alpha^+ \|\mathbf{grad} u_0\|_0 \|\mathbf{grad} v\|_0 \\ &\leq \left(\|f\|_0^2 + (\alpha^+)^2 \|\mathbf{grad} u_0\|_0^2 \right)^{1/2} \|v\|_{H^1(\Omega)} \quad \forall v \in H^1(\Omega). \quad (2.3.24) \end{aligned}$$

Next, we appeal to the lower estimate in (2.3.6) and the first Poincaré-Friedrichs inequality of Thm. 2.2.25:

$$\|v\|_{H^1(\Omega)} \leq \sqrt{1 + \text{diam}^2(\Omega)} |v|_{H^1(\Omega)} \leq \sqrt{\frac{1 + \text{diam}^2(\Omega)}{\alpha^-}} \|v\|_{\mathbf{a}} . \quad (2.3.25)$$

Combine (2.3.24) and (2.3.25),

$$|\ell(v)| \leq \underbrace{\left(\|f\|_0^2 + (\alpha^+)^2 |u_0|_{H^1(\Omega)}^2 \right)^{1/2}}_{=:K(f,u_0)} \sqrt{\frac{1 + \text{diam}^2(\Omega)}{\alpha^-}} \|v\|_{\mathbf{a}} .$$

This enters the estimate for the perturbation of the solution:

$$\begin{aligned} \mathbf{a}(w, v) &= -\ell(v) & \forall v \in V_0 , \\ \mathbf{a}(w + \delta w, v) &= -(\ell + \delta\ell)(v) & \forall v \in V_0 . \end{aligned}$$

\mathbf{a} bilinear
 \implies

(2.3.17)
 \implies

\implies

$$\begin{aligned} \mathbf{a}(\delta w, v) &= -\delta\ell(v) \quad \forall v \in V_0 , \\ \|\delta w\|_{\mathbf{a}} &= \sqrt{\mathbf{a}(\delta w, \delta w)} = \sqrt{|\delta\ell(\delta w)|} \leq (K(\delta f, \delta u_0) \|\delta w\|_{\mathbf{a}})^{1/2} , \\ \|\delta w\|_{\mathbf{a}} &\leq K(\delta f, \delta u_0) . \end{aligned}$$

As in Rem. 1.6.10 for associated quadratic energy functional J :

$$|J(w + \delta w) - J(w)| = \frac{1}{2} |\mathbf{a}(2w + \delta w, \delta w)| \leq \frac{1}{2} \|2w + \delta w\|_{\mathbf{a}} \|\delta w\|_{\mathbf{a}} . \quad (2.3.26)$$

Perturbation estimates in energy norm directly translate into perturbation estimates for the equilibrium energy!

Remark 2.3.27 (Needle loading).

Now we inspect a striking manifestation of instability for a 2nd-order elliptic variational problem caused by a right hand side functional that fails to satisfy (2.2.3).

Consider the taut membrane model, see Sect. 2.1.1 for details, (2.1.16) for the related minimization problem, and (2.3.4) for the associated variational equation.

Let us assume that a needle is poked at the membrane: loading by a force f “concentrated in a point \mathbf{y} ”, often denoted by $f = \delta_{\mathbf{y}}$, $\mathbf{y} \in \Omega$, where δ is the so-called **Dirac delta function** (delta distribution).

In the variational formulation this can be taken into account as follows ($u|_{\partial\Omega} = 0$, $\sigma \equiv 1$ is assumed):

$$u \in H_0^1(\Omega): \underbrace{\int_{\Omega} \mathbf{grad} u(\mathbf{x}) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x}}_{=:a(u,v)} = \underbrace{v(\mathbf{y})}_{=:l(v)} \quad \forall v \in H_0^1(\Omega). \quad (2.3.29)$$

Recall the discussion of Sect. 2.2: is the linear functional ℓ on the right hand side continuous w.r.t. the $H_0^1(\Omega)$ -norm (= energy norm, see Def. 2.1.35) in the sense of (2.2.3)?

Consider the function $v(\mathbf{x}) = \log |\log \|\mathbf{x}\||$, $\mathbf{x} \neq 0$, on $\Omega = \{\mathbf{x} \in \mathbb{R}^2: \|\mathbf{x}\| < \frac{1}{2}\}$.

First, we express this function in **polar coordinates**

(r, φ)

$$x_1 = r \cos \varphi \quad , \quad x_2 = r \sin \varphi \quad \blacktriangleright \quad v(r, \varphi) = \log |\log r| . \quad (2.3.32)$$

Then we recall the expression for the gradient in polar coordinates

$$\mathbf{grad} v(r, \varphi) = \frac{\partial v}{\partial r}(r, \varphi) \mathbf{e}_r + \frac{1}{r} \frac{\partial v}{\partial \varphi}(r, \varphi) \mathbf{e}_\varphi , \quad (2.3.33)$$

where \mathbf{e}_r and \mathbf{e}_φ are orthogonal unit vectors in the polar coordinate directions.

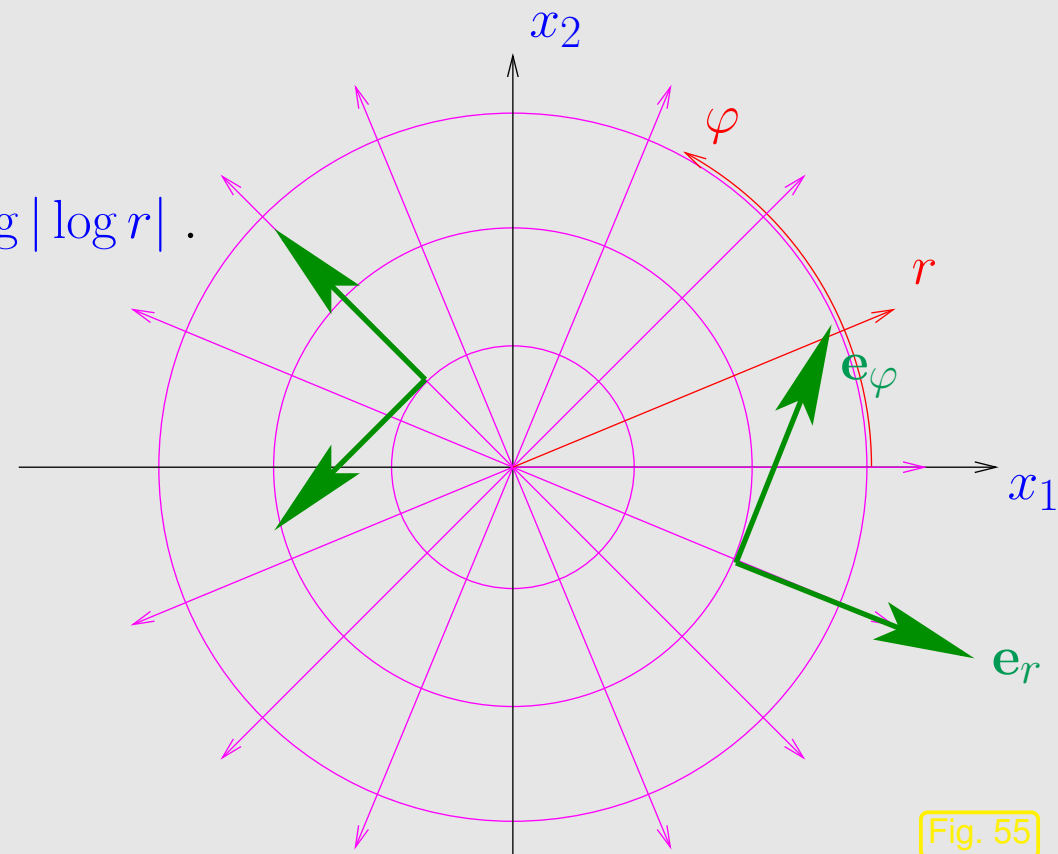


Fig. 55

Also recall integration in polar coordinates, see [32, Bsp. 8.5.3]:

$$\int_{\Omega} v(\mathbf{x}) \, d\mathbf{x} = \int_0^{1/2} \int_0^{2\pi} v(r, \varphi) r \, d\varphi dr . \quad (2.3.36)$$

Using these formulas we try to compute $|v|_{H^1(\Omega)}$,

$$\begin{aligned} \int_{\Omega} \|\mathbf{grad} v(\mathbf{x})\|^2 \, d\mathbf{x} &= \int_0^{1/2} \int_0^{2\pi} \left\| -\frac{1}{\log r r} \mathbf{e}_r \right\|^2 r \, d\varphi dr = 2\pi \int_0^{1/2} \frac{1}{\log^2 r} \cdot \frac{1}{r} \, dr \\ &= [-1/\log r]_0^{1/[2]} = \frac{1}{\log 2} < \infty , \end{aligned}$$

because the improper integral exists. This means that v has “finite elastic energy”, that is $v \in H^1(\Omega)$, see Def. 2.2.18.

On the other hand, $v(0) = \infty$!

$H^1(\Omega)$ contains unbounded functions !

Corollary 2.3.37 (Point evaluation on $H^1(\Omega)$).

The point evaluation $v \mapsto v(\mathbf{y})$, $\mathbf{y} \in \Omega$ is not a continuous linear form on $H^1(\Omega)$.

In view of Eq. 2.2.4, this means that no solution of 2.3.29 with finite energy can exist. The energy must blow up which results in a bursting of the membrane.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

This is the mathematics behind the observation that a needle can easily prick a taut membrane: a point load leads to configurations with “infinite elastic energy”.



Another implication of Cor. 2.3.37:

The quadratic functional $J(u) := \int_{\Omega} \|\mathbf{grad} u\|^2 \, d\mathbf{x} - u(\mathbf{y})$, $\mathbf{y} \in \Omega$
is *not* bounded from below on $H_0^1(\Omega)$!

Thus, it is clear that the attempt to minimize J will run into difficulties. Yet, this is the quadratic functional underlying the variational problem (2.3.29).



2.4 Equilibrium models: Boundary value problems

Recall the derivation of an ODE from a variational problem on a 1D domain (interval) in Sect. 1.3.3:

Tool: **Integration by parts** (1.3.36)

This section elucidates how to extend this approach to domains $\Omega \subset \mathbb{R}^d$, $d \geq 1$ (usually $d = 2, 3$).

Crucial issue: Integration by parts in higher dimensions ?

Remember the origin of integration by parts: fundamental theorem of calculus [32, Satz 6.3.4]: for $F \in C_{\text{pw}}^1([a, b])$, $a, b \in \mathbb{R}$,

$$\int_a^b F'(x) \, dx = F(b) - F(a) , \quad (2.4.3)$$

where $'$ stands for differentiation w.r.t x . This formula is combined with the product rule [32, Satz 5.2.1 (ii)]

$$F(x) = f(x) \cdot g(x) \quad \Rightarrow \quad F'(x) = f'(x)g(x) + f(x)g'(x) . \quad (2.4.4)$$

$$\blacktriangleright \int_a^b f'(x)g(x) + f(x)g'(x) \, dx = f(b)g(b) - f(a)g(a) ,$$

which amounts to (1.3.36).

There is a **product rule** in higher dimensions, see [32, Sect. 7.2]

Lemma 2.4.7 (General product rule).

For all $\mathbf{j} \in (C^1(\Omega))^d$, $v \in C^1(\Omega)$ holds

$$\operatorname{div}(\mathbf{j}v) = v \operatorname{div} \mathbf{j} + \mathbf{j} \cdot \operatorname{grad} v . \quad (2.4.8)$$

An important *differential operator*, see [32, Def. 8.8.1]:

divergence of a C^1 -**vector field** $\mathbf{j} = (f_1, \dots, f_d)^T : \Omega \mapsto \mathbb{R}^d$

$$\operatorname{div} \mathbf{j}(\mathbf{x}) := \frac{\partial f_1}{\partial x_1}(\mathbf{x}) + \dots + \frac{\partial f_d}{\partial x_d}(\mathbf{x}) , \quad \mathbf{x} \in \Omega .$$

A truly fundamental result from differential geometry provides a multidimensional analogue of the fundamental theorem of calculus:

Theorem 2.4.9 (Gauss' theorem). \rightarrow [32, Sect. 8.8]

With $\mathbf{n} : \partial\Omega \mapsto \mathbb{R}^d$ denoting the *exterior unit normal vectorfield* on $\partial\Omega$ and dS indicating integration over a surface, we have

$$\int_{\Omega} \operatorname{div} \mathbf{j}(\mathbf{x}) \, d\mathbf{x} = \int_{\partial\Omega} \mathbf{j}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \, dS(\mathbf{x}) \quad \forall \mathbf{j} \in (C_{\text{pw}}^1(\Omega))^d. \quad (2.4.10)$$

Note: In (2.4.10) integration again allows to relax smoothness requirements, cf. Sect. 1.3.2.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Theorem 2.4.11 (Green's first formula).

For all vector fields $\mathbf{j} \in (C_{\text{pw}}^1(\Omega))^d$ and functions $v \in C_{\text{pw}}^1(\Omega)$ holds

$$\int_{\Omega} \mathbf{j} \cdot \operatorname{grad} v \, d\mathbf{x} = - \int_{\Omega} \operatorname{div} \mathbf{j} v \, d\mathbf{x} + \int_{\partial\Omega} \mathbf{j} \cdot \mathbf{n} v \, dS. \quad (2.4.12)$$

Note that the dependence on the integration variable \boldsymbol{x} is suppressed in the formula (2.4.12) to achieve a more compact notation. The first Green formula could also have been written as

$$\int_{\Omega} \mathbf{j}(\boldsymbol{x}) \cdot (\mathbf{grad} v)(\boldsymbol{x}) \, d\boldsymbol{x} = - \int_{\Omega} (\operatorname{div} \mathbf{j})(\boldsymbol{x}) v(\boldsymbol{x}) \, d\boldsymbol{x} + \int_{\partial\Omega} \mathbf{j}(\boldsymbol{x}) \cdot \mathbf{n}(\boldsymbol{x}) v(\boldsymbol{x}) \, dS(\boldsymbol{x}) . \quad (2.4.12)$$

Proof. (of Thm. 2.4.11) Straightforward from Lemma 2.4.7 and Thm. 2.4.9. □

Now we apply Green's first formula to the variational problem (2.3.5), which covers the membrane model and electrostatics:

The role of \mathbf{j} in (2.4.12) is played by the *vector field* $\alpha \mathbf{grad} u : \Omega \mapsto \mathbb{R}^d$.

$$\int_{\Omega} \underbrace{\alpha(\boldsymbol{x}) \mathbf{grad} u(\boldsymbol{x})}_{=:\mathbf{j}(\boldsymbol{x})} \cdot \mathbf{grad} v(\boldsymbol{x}) \, d\boldsymbol{x}$$

$$= - \int_{\Omega} \operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) v(\mathbf{x}) \, d\mathbf{x} + \int_{\partial\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) v(\mathbf{x}) \, dS(\mathbf{x}) .$$

(2.3.5) \blacktriangleright

$$\begin{aligned}
 & - \int_{\Omega} \operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) v(\mathbf{x}) \, d\mathbf{x} \\
 & + \underbrace{\int_{\partial\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) v(\mathbf{x}) \, dS(\mathbf{x})}_{=0, \text{ since } v|_{\partial\Omega}=0} \\
 & = \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in C_{0,\text{pw}}^1(\Omega) ,
 \end{aligned}$$

(2.4.13)

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

where we have to *assume* that

$u, \boldsymbol{\alpha}$ are sufficiently smooth:

$$\boldsymbol{\alpha} \operatorname{grad} u \in C_{\text{pw}}^1(\Omega)$$

$$\int_{\Omega} (\operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) + f(\mathbf{x})) v(\mathbf{x}) \, d\mathbf{x} = 0 \quad \forall v \in C_{0,\text{pw}}^1(\Omega) .$$

Now we can invoke the multidimensional analogue of the fundamental lemma of the calculus of variations, see Lemm 1.3.37

Lemma 2.4.15 (Fundamental lemma of calculus of variations in higher dimensions).

If $f \in L^2(\Omega)$ satisfies

$$\int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} = 0 \quad \forall v \in C_0^\infty(\Omega) ,$$

then $f \equiv 0$ can be concluded.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

$$(2.3.5) \quad \alpha \operatorname{grad} u \in C_{\text{pw}}^1(\Omega) \quad \blacktriangleright$$

Partial differential equations (PDE)

$$-\operatorname{div}(\alpha(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega .$$

(2.4.16)

Again, for the sake of brevity, dependence $\mathbf{grad} u = \mathbf{grad} u(\mathbf{x})$, $f = f(\mathbf{x})$ is not made explicit in the PDE in (2.4.16).

Remark 2.4.19 (Laplace operator).

If α agrees with a positive *constant*, by rescaling of (2.5.6) we can achieve

$$-\Delta u = f \quad \text{in } \Omega . \quad (2.4.20)$$

$$\Delta = \mathbf{div} \circ \mathbf{grad} = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2} = \text{Laplace operator}$$

► (2.4.20) is called **Poisson equation**, $\Delta u = 0$ in Ω is called **Laplace equation**



Finally:

$$\begin{array}{c}
 \text{PDE (2.4.16)} \\
 \downarrow \\
 -\operatorname{div}(\boldsymbol{\alpha}(\boldsymbol{x}) \operatorname{grad} u) = f \quad \text{in } \Omega
 \end{array}
 \quad + \quad
 \begin{array}{c}
 \text{boundary conditions} \\
 \downarrow \\
 u = g \quad \text{on } \partial\Omega .
 \end{array}
 \quad (2.4.21)$$

(2.4.21) = **second-order elliptic BVP** with **Dirichlet boundary conditions**

Short name for BVPs of the type (2.4.21): **“Dirichlet problem”**

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 2.4.23 (Extra smoothness requirement for PDE formulation).

Same situation as in Sect. 1.3.3, *cf.* Assumption (1.3.34):

Transition from variational equation to PDE requires
extra assumptions on smoothness of solution and coefficients.



Remark 2.4.24 (Membrane with free boundary values).

(Graph description of membrane shape by $u : \Omega \mapsto \mathbb{R}$, see Sect. 2.1.1)

Now: membrane clamped only on a part $\Gamma_0 \subset \partial\Omega$ of its edge.

--- : prescribed boundary values here (Γ_0)

— : “free boundary”

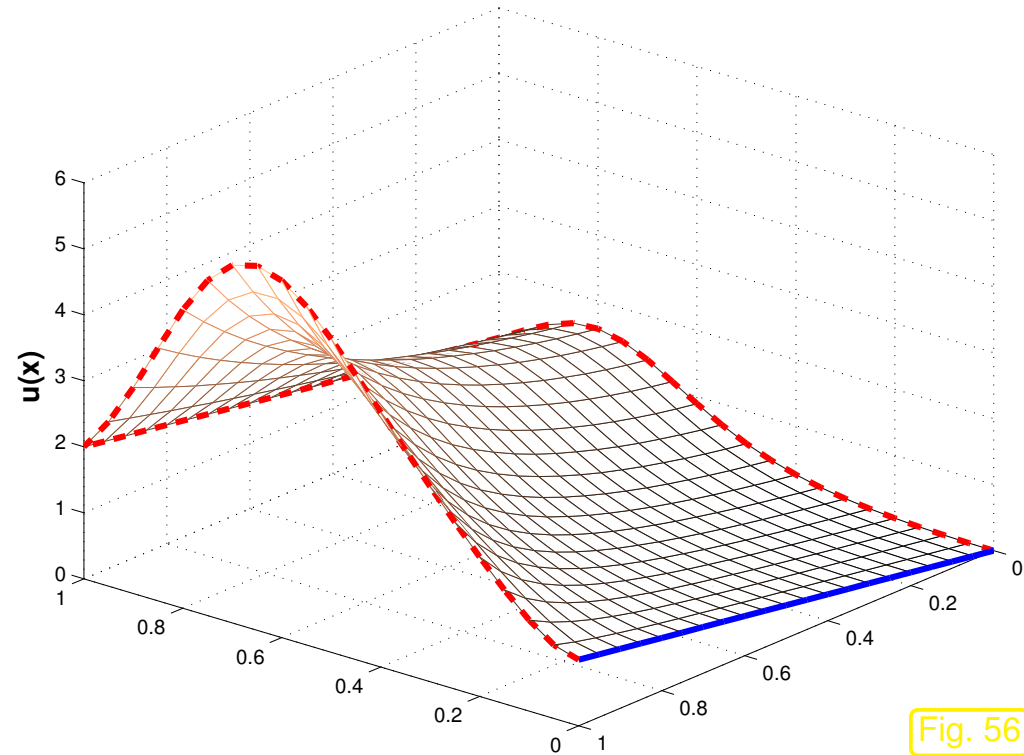


Fig. 56

► Configuration space $V := \{u \in H^1(\Omega) : u|_{\Gamma_0} = g\} \rightarrow$ Def. 2.2.18

Total potential energy as in (2.1.3):

$$J_M(u) := \int_{\Omega} \frac{1}{2} \sigma(\mathbf{x}) \|\mathbf{grad} u\|^2 - f(\mathbf{x})u(\mathbf{x}) \, d\mathbf{x} \quad (2.1.3)$$

► test space in variational formulation

$$V_0 := \{u \in H^1(\Omega) : u|_{\Gamma_0} = 0\}$$

(Remember: test space comprises “admissible perturbations” of configurations, *cf.* Sect. 1.3.1.)

Variational formulation, *c.f.* (2.3.4)

$$\begin{aligned} u \in H^1(\Omega), \\ u = g \text{ on } \Gamma_0 \end{aligned} : \int_{\Omega} \sigma(\mathbf{x}) \mathbf{grad} u(\mathbf{x}) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in V_0. \quad (2.4.29)$$

► Application of Green’s first formula (2.4.12) to (2.4.29) leads to

$$\begin{aligned} - \int_{\Omega} (\operatorname{div}(\sigma(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) + f(\mathbf{x})) v(\mathbf{x}) \, d\mathbf{x} \\ + \int_{\partial\Omega \setminus \Gamma_0} ((\sigma(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})) v(\mathbf{x}) \, dS(\mathbf{x}) \quad \forall v \in V_0. \end{aligned} \quad (2.4.30)$$

Note that, unlike above, the boundary integral term cannot be dropped entirely, because $v \neq 0$ on $\partial\Omega \setminus \Gamma_0$.

Assumption (\rightarrow Rem. 2.4.23): extra smoothness $u \in C_{\text{pw}}^2(\Omega)$, $\sigma \in C_{\text{pw}}^1(\Omega)$

How to deal with the boundary term ?

Idea: **1** First restrict test function v to $C_0^\infty(\Omega)$



Boundary term vanishes !

Then, apply Lemma 2.4.15

$$\blacktriangleright \operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) + f(\mathbf{x}) = 0 \quad \text{in } \Omega . \quad (2.4.31)$$

2 Then test with generic $v \in V_0$, while *making use of* (2.4.31):

$$\blacktriangleright \int_{\partial\Omega \setminus \Gamma_0} ((\sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})) v(\mathbf{x}) \, dS(\mathbf{x}) = 0 \quad \forall v \in V_0 .$$

Lemma 2.4.15 on $\partial\Omega \setminus \Gamma_0$
 \implies

$$(\sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad \text{on } \partial\Omega \setminus \Gamma_0 . \quad (2.4.32)$$

When removing pinning conditions on $\partial\Omega \setminus \Gamma_0$ the equilibrium conditions imply the (homogeneous) **Neumann boundary conditions** $(\sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) = 0$ on $\partial\Omega \setminus \Gamma_0$.



Boundary value problem for membrane clamped at $\Gamma_0 \subset \partial\Omega$

$$\begin{aligned}
 -\operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u) &= f && \text{in } \Omega, && u = g && \text{on } \Gamma_0, \\
 (\sigma(\mathbf{x}) \operatorname{grad} u) \cdot \mathbf{n} &= 0 && \text{on } \partial\Omega \setminus \Gamma_0. && &&
 \end{aligned}
 \tag{2.4.33}$$

(2.4.33) = **Second-order elliptic BVP with Neumann boundary conditions** on $\partial\Omega \setminus \Gamma_0$

Short name for BVPs of the type (2.4.33): **“Mixed Neumann–Dirichlet problem”**



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

2.5 Diffusion models (Stationary heat conduction)

Now we look at a class of physical phenomena, for which models are based on two building blocks

1. a **conservation principle** (of mass, energy, etc.),


2. a **a potential driven flux** of the conserved quantity.

Mathematical modelling for these phenomena naturally involves partial differential equations in the first steps, which are supplemented with boundary conditions. Hence, second-order elliptic boundary value problems arise first, while variational formulations are deduced from them, thus reversing the order of steps followed for equilibrium models in Sects. 2.1–2.4.

In order to keep the presentation concrete, the discussion will target **heat conduction**, about which everybody should have a sound “intuitive grasp”.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

 notation: $\Omega \subset \mathbb{R}^3$: bounded open region occupied by solid object
($\hat{=}$ $\Omega \rightarrow$ **computational domain**)

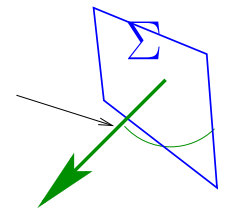
Fundamental concept:

heat flux, modelled by **vector field** $\mathbf{j} : \Omega \mapsto \mathbb{R}^3$

Heat flux = power flux: $[\mathbf{j}] = \frac{\text{W}}{\text{m}^2}$

Vector field $\mathbf{j} : \Omega :=]0, 1[^2 \mapsto \mathbb{R}^3$

normal vector \mathbf{n}



Total heat flux through oriented surface $\Sigma \subset \mathbb{R}^3$

Power $P_\Sigma = \int_\Sigma \mathbf{j} \cdot \mathbf{n} \, dS . \quad (2.5.1)$

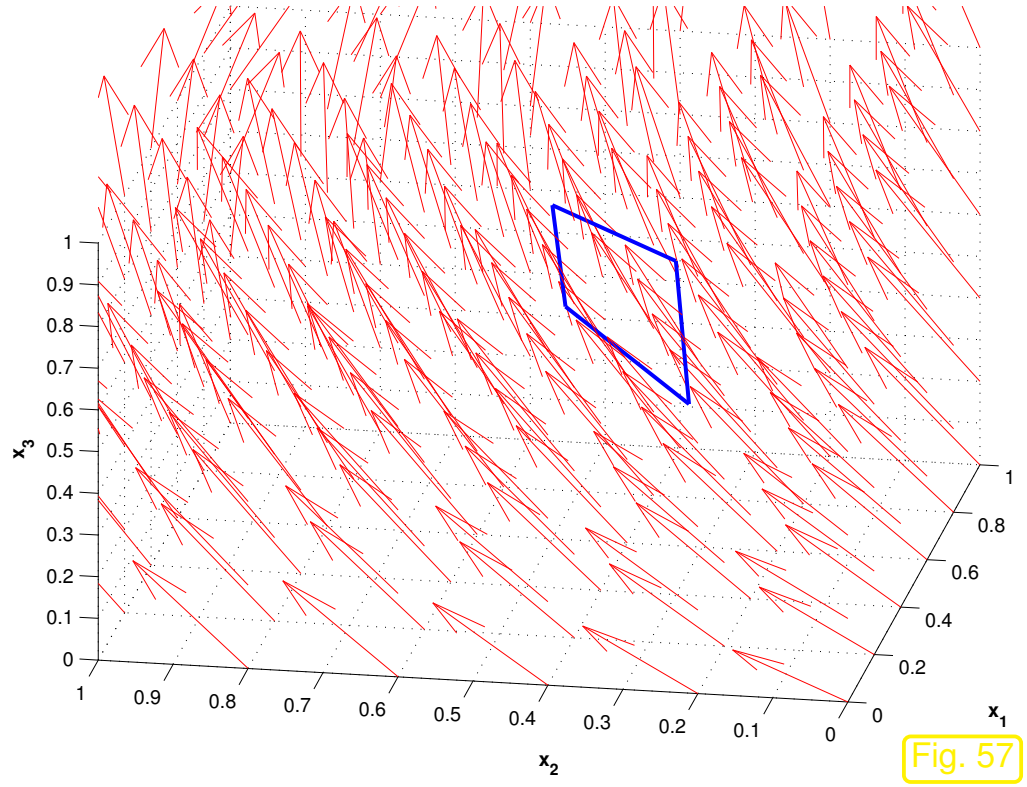


Fig. 57

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs
SAM, ETHZ

P_Σ ($[P_\Sigma] = 1\text{W}$): directed total power flowing through the oriented surface Σ per unit time. Note that the sign of P_Σ will change when flipping the normal of Σ !

Conservation of energy

$$\int_{\partial V} \mathbf{j} \cdot \mathbf{n} \, dS = \int_V f \, d\mathbf{x} \quad \text{for all "control volumes" } V. \quad (2.5.2)$$

power flux through surface of V

heat production inside V

$f = \text{heat source/sink}$ ($[f] = \frac{\text{W}}{\text{m}^3}$), $f = f(\mathbf{x})$ and f can be discontinuous ($f \in C_{\text{pw}}^0(\Omega)$)

- Intuition:
- heat flows from hot zones to cold zones
 - the larger the temperature difference, the stronger the heat flow

Experimental evidence supports this intuition and, for many materials, yields the following quantitative relationship:

Fourier's law

$$\mathbf{j}(\mathbf{x}) = -\kappa(\mathbf{x}) \mathbf{grad} u(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (2.5.3)$$

Meaning of quantities:

$$\begin{aligned} \mathbf{j} &= \text{heat flux} & ([\mathbf{j}] &= 1 \frac{\text{W}}{\text{m}^2}) \\ u &= \text{temperature} & ([u] &= 1 \text{K}) \\ \kappa &= \text{heat conductivity} & ([\kappa] &= 1 \frac{\text{W}}{\text{Km}}) \end{aligned}$$



(2.5.3) \Rightarrow Heat flow from hot to cold regions **linearly proportional** to gradient of temperature

Some facts about the heat conductivity:

- κ :
- $\kappa = \kappa(\mathbf{x})$ for **non-homogeneous** materials (spatially varying heat conductivity)
 - κ can even be discontinuous for composite materials
 - κ may be $\mathbb{R}^{3,3}$ -valued (heat conductivity tensor)

The most general form of the heat conductivity (tensor) enjoys the very same properties as the dielectric tensor introduced in Sect. 2.1.2:

From thermodynamic principles, *cf.* (2.1.9):

$$\exists \kappa^-, \kappa^+ > 0: \quad 0 < \kappa^- \leq \kappa(\mathbf{x}) \leq \kappa^+ < \infty \quad \text{for almost all } \mathbf{x} \in \Omega . \quad (2.5.4)$$

Terminology: (2.5.4) \Leftrightarrow κ is bounded and **uniformly positive**, see Def. 2.1.12.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

From 2.5.2 by Gauss' theorem Thm. 2.4.9

$$\int_V \operatorname{div} \mathbf{j}(\mathbf{x}) \, d\mathbf{x} = \int_V f(\mathbf{x}) \, d\mathbf{x} \quad \text{for all "control volumes" } V \subset \Omega .$$

SAM, ETHZ

Now appeal to another version of the fundamental lemma of the calculus of variations, see Lemma 2.4.15, this time sporting piecewise constant test functions.

► **local form of energy conservation:**

$$\operatorname{div} \mathbf{j} = f \quad \text{in } \Omega . \quad (2.5.5)$$

Combine equations (2.5.5) & (2.5.3)

$$\mathbf{j} = -\kappa(\mathbf{x}) \operatorname{grad} u \quad (2.5.3)$$

+

$$\operatorname{div} \mathbf{j} = f \quad (2.5.5)$$


$$-\operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega . \quad (2.5.6)$$

▶ *Linear* scalar second order elliptic PDE (for unknown temperature u)

2.6 Boundary conditions

In the examples from Sects. 2.1.1, 2.1.2 we fixed the value of the unknown function $u : \Omega \mapsto \mathbb{R}$ on the boundary $\partial\Omega$: **Dirichlet boundary conditions** in (2.4.21).

Exception: free edge of taut membrane, see Rem. 2.4.24: **Neumann boundary conditions** in (2.4.33).

In this section we resume the discussion of boundary conditions and examine them for stationary heat conduction, see previous section. This has the advantage that for this everyday physical phenomenon boundary conditions have a very clear intuitive meaning.

Boundary conditions on surface/boundary $\partial\Omega$ of Ω :

(i) Temperature u is fixed: with $g : \partial\Omega \mapsto \mathbb{R}$ prescribed

$$u = g \quad \text{on } \partial\Omega . \quad (2.6.1)$$

Dirichlet boundary conditions

(ii) Heat flux \mathbf{j} through $\partial\Omega$ is fixed: with $h : \partial\Omega \mapsto \mathbb{R}$ prescribed ($\mathbf{n} : \partial\Omega \mapsto \mathbb{R}^3$ exterior unit normal vectorfield) on $\partial\Omega$

$$\mathbf{j} \cdot \mathbf{n} = -h \quad \text{on } \partial\Omega . \quad (2.6.2)$$

Neumann boundary conditions

(iii) Heat flux through $\partial\Omega$ depends on (local) temperature: with increasing function $\Psi : \mathbb{R} \mapsto \mathbb{R}$

$$\mathbf{j} \cdot \mathbf{n} = \Psi(u) \quad \text{on } \partial\Omega \quad (2.6.3)$$

radiation boundary conditions

Example 2.6.4 (Convective cooling (simple model)).

Heat is carried away from the surface of the body by a fluid at bulk temperature u_0 . A crude model assumes that the heat flux depends *linearly* on the temperature difference between the surface of Ω and the bulk temperature of the fluid.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

$$\mathbf{j} \cdot \mathbf{n} = q(u - u_0) \quad \text{on } \partial\Omega, \quad \text{where } 0 < q^- \leq q(\mathbf{x}) \leq q^+ < \infty \quad \text{for almost all } \mathbf{x} \in \partial\Omega.$$

When combined with Fourier's law (2.5.3), the convective cooling boundary conditions become

$$\kappa(\mathbf{x}) \mathbf{grad} u(\mathbf{x}) + q(u(\mathbf{x}) - u_0) = 0, \quad \mathbf{x} \in \partial\Omega,$$

and in this form they are known as **Robin boundary conditions**.

Example 2.6.5 (Radiative cooling (simple model)).

A hot body emits electromagnetic radiation (blackbody emission), which drains thermal energy. The radiative energy loss is roughly proportional to the 4th power of the temperature difference between the surface temperature of the body and the ambient temperature.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

$$\mathbf{j} \cdot \mathbf{n} = \alpha |u - u_0| (u - u_0)^3 \quad \text{on } \partial\Omega, \quad \text{with } \alpha > 0$$

→

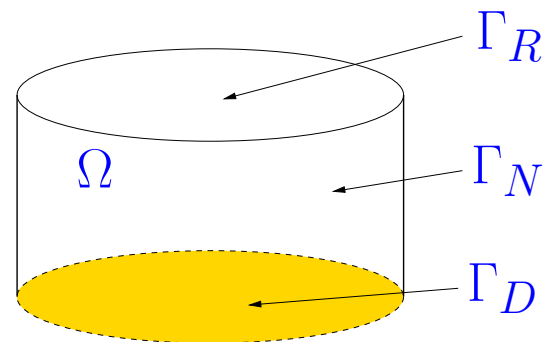
Non-linear boundary condition

◇

Terminology: If $g = 0$ or $h = 0$ → **homogeneous** Dirichlet or Neumann boundary conditions

Remark 2.6.6 (Mixed boundary conditions).

Different boundary conditions can be prescribed on different parts of $\partial\Omega$
(\rightarrow **mixed boundary conditions**, cf. Rem. 2.4.24)



Example 2.6.7 (“Wrapped rock on a stove”).

- Non-homogeneous Dirichlet boundary conditions on $\Gamma_D \subset \partial\Omega$
- Homogeneous Neumann boundary conditions on $\Gamma_N \subset \partial\Omega$
- Convective cooling boundary conditions on $\Gamma_R \subset \partial\Omega$

Partition: $\partial\Omega = \bar{\Gamma}_D \cup \bar{\Gamma}_N \cup \bar{\Gamma}_R$, $\Gamma_D, \Gamma_N, \Gamma_R$ mutually disjoint



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

$-\operatorname{div}(\kappa(\boldsymbol{x}) \operatorname{grad} u) = f$ + boundary conditions \Rightarrow **elliptic boundary value problem (BVP)**

For second order elliptic boundary value problems **exactly one** boundary condition is needed on any part of $\partial\Omega$.

Remark 2.6.9 (Linear BVP).

Observe that the solution mapping $\begin{pmatrix} f \\ g \end{pmatrix} \mapsto u$ for (2.5.6), (2.6.1) is **linear**.

This means that if u_i solves the Dirichlet problem with source function f_i and Dirichlet data g_i , $i = 1, 2$, then $u_1 + u_2$ solves (2.5.6) & (2.6.1) for source $f_1 + f_2$ and boundary values $g_1 + g_2$.



2.7 Characteristics of elliptic boundary value problems

Qualitative insights gained from heat conduction model:

- **continuity**: the temperature u must be continuous (jump in $u \rightarrow \mathbf{j} = \infty$).

- normal component of \mathbf{j} across surfaces inside Ω must be continuous (jump in $\mathbf{j} \cdot \mathbf{n} \rightarrow$ heat source f of infinite intensity).
- **interior smoothness** of u : u smooth where f and D smooth.
- **non-locality**: local alterations in f, g, h affect u everywhere in Ω .
- **quasi-locality**: If local changes in f, g, h confined to $\Omega' \subset \Omega$, their effects decay away from Ω' .
- **maximum principle**: (in the absence of heat sources extremal temperatures are on the boundary)

$$\text{if } f \equiv 0, \text{ then } \quad \inf_{\mathbf{y} \in \partial\Omega} u(\mathbf{y}) \leq \mathbf{u}(\mathbf{x}) \leq \sup_{\mathbf{y} \in \partial\Omega} u(\mathbf{y}) \quad \text{for all } \mathbf{x} \in \Omega$$

Typical features of solutions of elliptic boundary value problems

Example 2.7.1 (Scalar elliptic boundary value problem in one space dimension).

Poisson equation \rightarrow (2.4.20) in 1D:

$$-u'' = f$$



➤ f discontinuous, piecewise $C^0 \Rightarrow u \in C^1$, piecewise C^2

Example 2.7.2 (Smoothness of solution of scalar elliptic boundary value problem).

$$\begin{aligned}
 -\Delta u &= f(\mathbf{x}) \quad \text{in } \Omega :=]0, 1[^2, \quad u = 0 \quad \text{on } \partial\Omega, \\
 f(\mathbf{x}) &:= \text{sign}(\sin(2\pi k_1 x_1) \sin(2\pi k_2 x_2)), \quad \mathbf{x} \in \Omega, \quad k_1, k_2 \in \mathbb{N}.
 \end{aligned}
 \tag{2.7.3}$$

Approximate solution computed by means of linear Lagrangian finite elements + lumping
(→ Ch. 3, details in Sect. 3.2, 3.5.4)

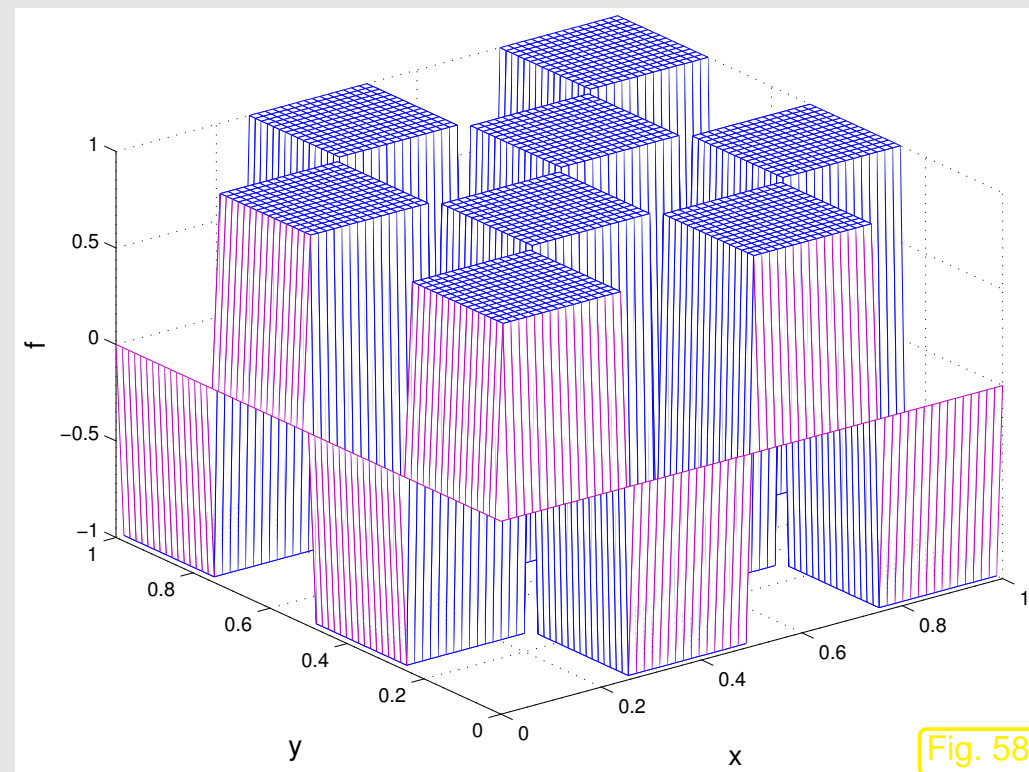


Fig. 58

Source term $f(\mathbf{x})$, $k_1 = k_2 = 2$

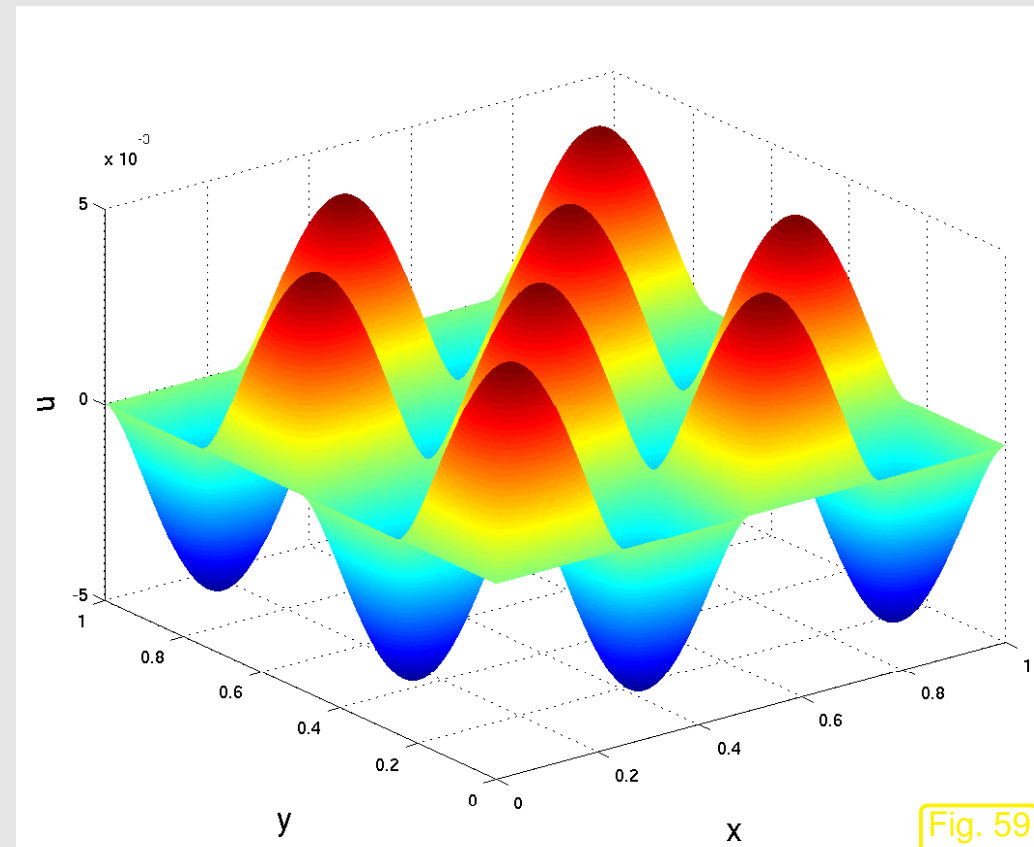


Fig. 59

Solution of (2.7.3)

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

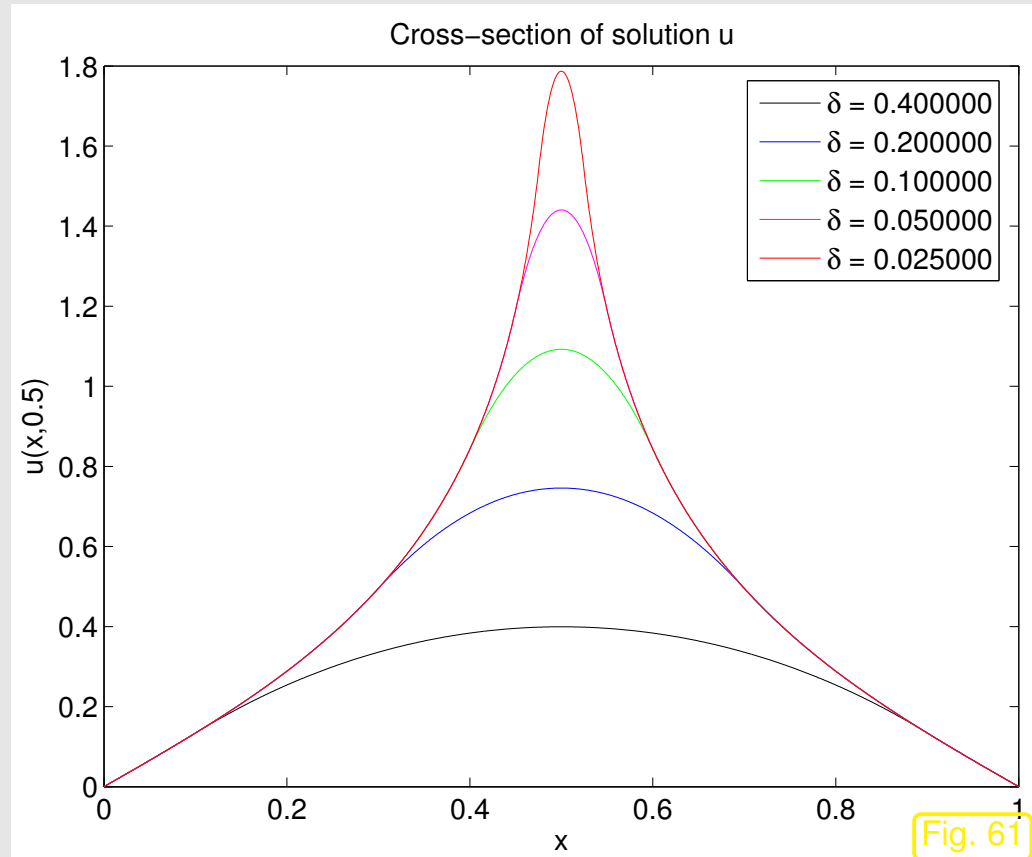
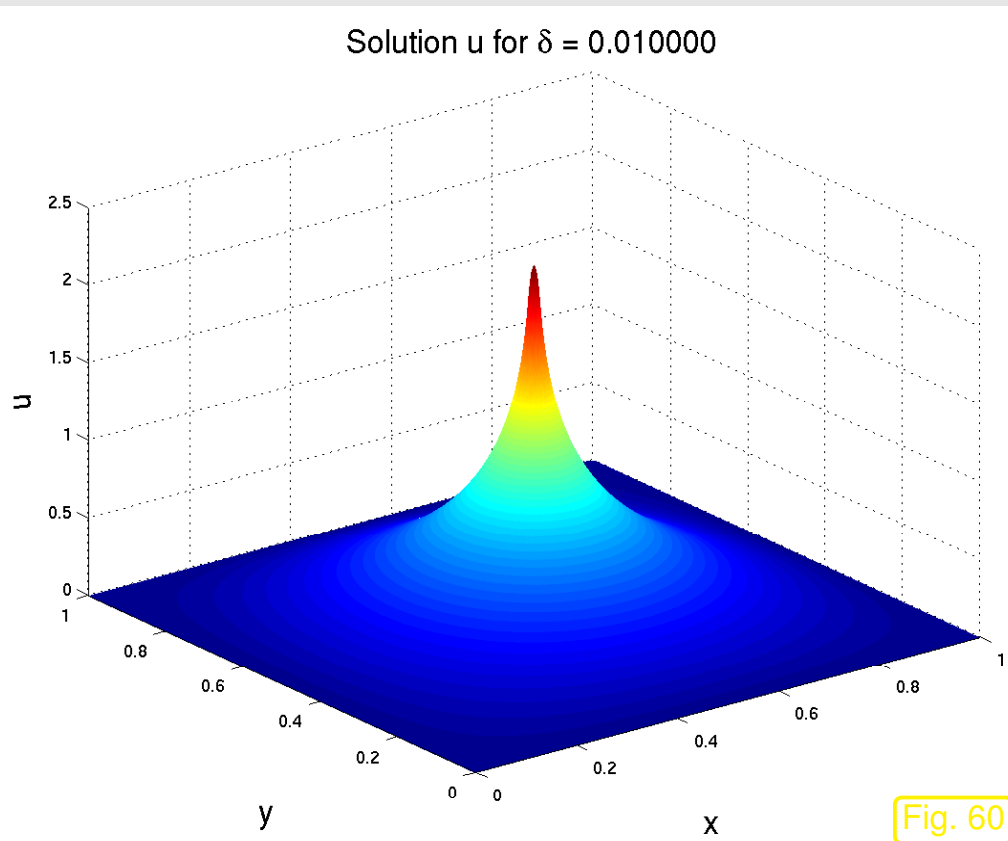
SAM, ETHZ

➤ “Smooth” u despite “rough” f !

Example 2.7.4 (Quasi-locality of solution of scalar elliptic boundary value problem).

$$-\Delta u = f_\delta(\mathbf{x}) \quad \text{in } \Omega :=]0, 1[^2, \quad u = 0 \quad \text{on } \partial\Omega, \quad (2.7.5)$$

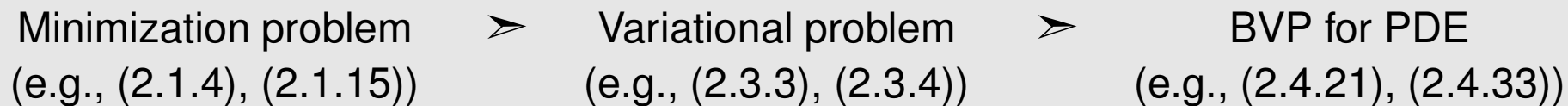
$$f_\delta(\mathbf{x}) = \begin{cases} \delta^{-2} & , \text{ if } \left\| \mathbf{x} - \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \right\|_2 \leq \delta, \\ 0 & \text{elsewhere.} \end{cases}, \quad \delta > 0. \quad (2.7.6)$$



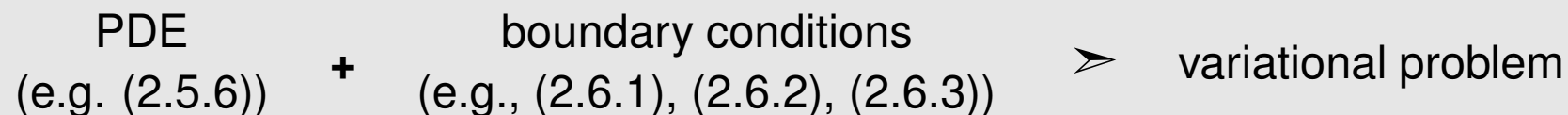


2.8 Second-order elliptic variational problems

In Ch. 1 and Sects. 2.1–2.4 we pursued the derivation:



Now we are proceeding in the opposite direction:



Formal approach:

STEP 1: *test PDE with smooth functions*

(do not test, where the solution is known, e.g., on the boundary)

STEP 2: *integrate over domain*

STEP 3: *perform integration by parts*

(e.g. by using Green's first formula, Thm. 2.4.11)

STEP 4: [optional] *incorporate boundary conditions into boundary terms*

Example 2.8.1 (Variational formulation for heat conduction with Dirichlet boundary conditions).

$$\text{BVP: } -\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) = f \quad \text{in } \Omega, \quad u = g \quad \text{on } \partial\Omega. \quad (2.8.5)$$

STEP 1 & 2:

test with $v \in C_0^\infty(\Omega)$

$$\blacktriangleright -\int_{\Omega} \operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x}. \quad (2.8.6)$$

Note: $v|_{\partial\Omega} = 0$ for test function, because u already fixed on $\partial\Omega$.

STEP 3: use **Green's formula** from Thm. 2.4.11 on $\Omega \subset \mathbb{R}^d$ (multidimensional integration by parts):
Apply (2.4.12) to (2.8.6) with $\mathbf{j} := \kappa(\mathbf{x}) \mathbf{grad} u$:

$$\blacktriangleright \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} - \underbrace{\int_{\partial\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{n} v \, dS}_{=0, \text{ because } v|_{\partial\Omega}=0} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in C_0^\infty(\Omega).$$

This gives the variational formulation after we switch to “maximal admissible function spaces” (Sobolev spaces, see Sect. 2.2, as spaces of functions with finite energy)

Variational form of (2.8.5): seek

$$\begin{array}{l} u \in H^1(\Omega) \\ u = g \text{ on } \partial\Omega \end{array} : \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega). \quad (2.8.7)$$

Example 2.8.8 (Variational formulation: heat conduction with general radiation boundary conditions).

$$\text{BVP: } -\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) = f \quad \text{in } \Omega, \quad -\kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{n} = \Psi(u) \quad \text{on } \partial\Omega. \quad (2.8.11)$$

STEP 1 & 2: $u|_{\partial\Omega}$ not fixed \Rightarrow test with $v \in C^\infty(\bar{\Omega})$

$$\blacktriangleright -\int_{\Omega} \operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in C^\infty(\bar{\Omega}).$$

STEP 3 & 4: apply Green's first formula (2.4.12) and incorporate boundary conditions:

$$\blacktriangleright \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} - \int_{\partial\Omega} \underbrace{\kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{n}}_{= -\Psi(u) \text{ (STEP 4)}} v \, dS = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in C^\infty(\bar{\Omega}).$$

Variational formulation of (2.8.11): seek

$$u \in H^1(\Omega): \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} + \int_{\partial\Omega} \Psi(u) v \, dS = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H^1(\Omega) . \quad (2.8.12)$$



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Theorem 2.8.13. *If $\kappa \in C^1(\Omega)$, classical solutions $u \in C^2(\Omega)$ of the boundary value problems (2.8.5) and (2.8.11) also solve the associated variational problems.*

Proof. Apply Theorem 2.4.11 as in the derivation of the weak formulations.

Example 2.8.14 (Variational formulation for Neumann problem).

2nd-order elliptic (inhomogeneous) **Neumann problem**

$$\text{BVP: } \begin{aligned} -\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) &= f && \text{in } \Omega, \\ \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{n} &= h(\mathbf{x}) && \text{on } \partial\Omega. \end{aligned} \quad (2.8.15)$$

We confront Neumann boundary conditions (2.6.2) (prescribed heat flux) on the whole boundary.

Variational formulation derived as in Ex. 2.8.8, with $\Psi(u) = -h$.

$$u \in H^1(\Omega): \quad \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} - \int_{\partial\Omega} h v \, dS = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H^1(\Omega). \quad (2.8.18)$$

Observation: when we test (2.8.12) with $v \equiv 1$ \blacktriangleright
$$-\int_{\partial\Omega} h \, dS = \int_{\Omega} f \, d\mathbf{x} \quad (2.8.19)$$

This is a **compatibility condition** for the existence of (variational) solutions of the Neumann problem!

Interpretation of (2.8.19) against the backdrop of the stationary heat conduction model:

conservation of energy \rightarrow (2.5.2): Heat generated inside Ω ($\leftrightarrow f$) must be offset by heat flux through $\partial\Omega$ ($\rightarrow h$).

Remark 2.8.20 (Uniqueness of solutions of Neumann problem).

Observation: if compatibility condition (2.8.19) holds true, then

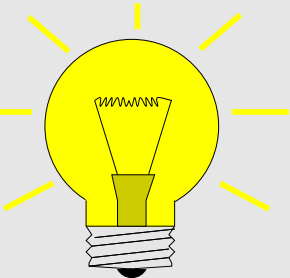
$$v \in H^1(\Omega) \text{ solves (2.8.12)} \iff v + \gamma \text{ solves (2.8.12)} \quad \forall \gamma \in \mathbb{R},$$

we say, “the solution is unique only up to constants”.

Complementary observation: $\mathbf{a}(u, v) := \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x}$ is *not* s.p.d (\rightarrow Def. 2.1.32) on $H^1(\Omega)$.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Idea: Restore uniqueness of solutions by

enforcing average temperature to be zero $\int_{\Omega} u(\mathbf{x}) \, d\mathbf{x} = 0$

This amounts to posing the variational problem (2.8.12) over the **constrained** function space

$$H_*^1(\Omega) := \{v \in H^1(\Omega) : \int_{\Omega} v(\mathbf{x}) \, d\mathbf{x} = 0\} . \tag{2.8.23}$$

The norm on $H_*^1(\Omega)$ is the same as on $H_0^1(\Omega)$, see Def. 2.2.18. Obviously (why ?), the norm property (N1) is satisfied. These arguments also show that \mathbf{a} is s.p.d (\rightarrow Def. 2.1.32) on $H_*^1(\Omega)$, cf. Thm. 2.9.9.

► Variational formulation of Neumann problem:

$$u \in H_*^1(\Omega): \quad \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} + \int_{\partial\Omega} h v \, dS \quad \forall v \in H_*^1(\Omega). \quad (2.8.24)$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



2.9 Essential and natural boundary conditions

Synopsis:

☛ 2nd-order elliptic Dirichlet problem:

$$-\operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega \quad , \quad u = g \quad \text{on } \partial\Omega . \quad (2.4.21)$$

with variational formulation

$$\begin{aligned} u &\in H^1(\Omega) \quad , \\ u &= g \quad \text{on } \partial\Omega \end{aligned} \quad : \quad \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) . \quad (2.3.5)$$

☛ 2nd-order elliptic Neumann problem:

$$-\operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega \quad , \quad (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) \cdot \mathbf{n} = -h \quad \text{on } \partial\Omega . \quad (2.9.4)$$

with variational formulation

$$u \in H_*^1(\Omega): \quad \int_{\Omega} \boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} + \int_{\partial\Omega} h v \, dS \quad \forall v \in H_*^1(\Omega) . \quad (2.8.24)$$

☛ 2nd-order elliptic mixed Neumann-Dirichlet problem, see Rem. 2.4.24:

$$\begin{aligned} -\operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) &= f \quad \text{in } \Omega \quad , \\ (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) \cdot \mathbf{n} &= -h \quad \text{on } \partial\Omega \setminus \Gamma_0 . \end{aligned} \quad \begin{aligned} u &= g \quad \text{on } \Gamma_0 \subset \partial\Omega , \end{aligned} \quad (2.9.5)$$

with variational formulation

$$\begin{aligned} u &\in H^1(\Omega) \quad , \\ u &= g \quad \text{on } \Gamma_0 \end{aligned} \quad : \quad \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} + \int_{\partial\Omega \setminus \Gamma_0} h v \, dS \quad (2.9.6)$$

for all $v \in H^1(\Omega)$ with $v|_{\Gamma_0} = 0$.

In the variational formulations of 2nd-order elliptic BVPs of Sect. 2.8:

Dirichlet boundary conditions are *directly imposed* on trial space and (in homogeneous form) on test space.

Terminology:

essential boundary conditions

Neumann boundary conditions are enforced *only* through the variational equation.

Terminology:

natural boundary conditions

The attribute “natural” has been coined, because Neumann boundary conditions “naturally” emerge when removing constraints on the boundary, as we have seen for the partially free membrane of Rem. 2.4.24.

Remark 2.9.7 (Admissible Dirichlet data).

Requirement for “Dirichlet data” $g : \partial\Omega \mapsto \mathbb{R}$ in (2.4.21):

there is $u \in H^1(\Omega)$ such that $u|_{\partial\Omega} = g$

Analogous to Thm. 2.2.26:

If $g : \partial\Omega \mapsto \mathbb{R}$ is piecewise continuously differentiable (and bounded with bounded piecewise derivatives), then it can be extended to an $u_0 \in H^1(\Omega)$, *if and only if* it is **continuous** on $\partial\Omega$.

Bottom line:

Dirichlet boundary values have to be continuous

This is also stipulated by physical insight, e.g. in the case of the taut membrane model of Sect. 2.1.1: discontinuous displacement on $\partial\Omega$ would entail ripping apart the membrane.

Remark 2.9.8 (Admissible Neumann data).

In the variational problem (2.8.24) Neumann data $h : \partial\Omega \mapsto \mathbb{R}$ enter through the linear form on the right hand side

$$\ell(v) := \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} + \int_{\partial\Omega} h(\mathbf{x})v(\mathbf{x}) \, dS(\mathbf{x}) .$$

Remember the discussion in the beginning of Sect. 2.2, also Rem. 2.3.27: we have to establish that ℓ is continuous on $H_*^1(\Omega)$ defined in (2.8.23). This is sufficient, because the coefficient function κ is uniformly positive and bounded, see (2.5.4). Thus, the energy $\|\cdot\|_a$ associated with the bilinear form


$$a(u, v) = \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x}$$

can be bounded from above and below by $|\cdot|_{H^1(\Omega)}$, cf. the estimate (2.3.25).

Theorem 2.9.9. Second Poincaré-Friedrichs inequality]

If $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, is bounded, then

$$\exists C = C(\Omega) > 0: \quad \|u\|_0 \leq C \operatorname{diam}(\Omega) \|\mathbf{grad} u\|_0 \quad \forall u \in H_*^1(\Omega).$$

 notation: $C = C(\Omega)$ indicates that the constant C may depend on the shape of the domain Ω .

Proof. (for $d = 1$, $\Omega = [0, 1]$ only, technically difficult in higher dimensions, see [6, Thm. 1.6.6])

As in the proof of Thm. 2.2.25, we employ a density argument and assume that u is sufficiently smooth, $u \in C^1([0, 1])$.

By the fundamental theorem of calculus (2.4.3)

$$u(x) = u(y) + \int_y^x \frac{du}{dx}(\tau) d\tau, \quad 0 \leq x, y \leq 1.$$

$$\blacktriangleright \quad u(x) = \int_0^1 u(x) dy = \underbrace{\int_0^1 u(y) dy}_{=0} + \int_0^1 \int_y^x \frac{du}{dx}(\tau) d\tau dy.$$

Then use the Cauchy-Schwarz inequality (2.2.24)

$$u(x)^2 \leq \int_0^1 \int_y^x 1 \, d\tau \, dy \int_0^1 \int_y^x \left| \frac{du}{dx}(\tau) \right|^2 \, d\tau \, dy \leq \int_0^1 \left| \frac{du}{dx}(\tau) \right|^2 \, d\tau .$$

Integrate over Ω yields the estimate

$$\|u\|_0^2 = \int_0^1 u^2(x) \, dx \leq \int_0^1 \left| \frac{du}{dx}(\tau) \right|^2 \, d\tau = |u|_{H^1(\Omega)}^2 . \quad (\square)$$

By (2.2.24), Thm. 2.9.9 implies the continuity of the first term in ℓ .

Continuity of the boundary contribution to ℓ hinges on a **trace theorem**

Theorem 2.9.10 (Multiplicative trace inequality).

$$\exists C = C(\Omega) > 0: \quad \|u\|_{L^2(\partial\Omega)}^2 \leq C \|u\|_{L^2(\Omega)} \cdot \|u\|_{H^1(\Omega)} \quad \forall u \in H^1(\Omega) .$$

Proof. (for $d = 1$, $\Omega = [0, 1]$ only, technically difficult in higher dimensions)

As in the proof of Thms. 2.2.25, 2.9.9, we employ a density argument and assume that u is sufficiently smooth, $u \in C^1([0, 1])$.

By the fundamental theorem of calculus (2.4.3):

$$u(1)^2 = \int_0^1 \frac{dw}{d\xi}(x) dx, \quad \text{with } w(\xi) := \xi u^2(\xi),$$

$$\blacktriangleright \quad u(1)^2 = \int_0^1 u^2(x) + 2u(x) \frac{du}{dx}(x)x dx.$$

Then use the Cauchy-Schwarz inequality (2.2.24)

$$u(1)^2 \leq \int_0^1 u^2(x) dx + 2 \int_0^1 |x||u(x)| \left| \frac{du}{dx}(x) \right| dx \leq \|u\|_0^2 + 2 \|u\|_0 \left\| \frac{du}{dx} \right\|_0.$$

A similar estimate holds for $u(0)^2$. □

Now we can combine

- the Cauchy-Schwarz inequality (2.2.24) on $\partial\Omega$,
- the 2nd Poincaré-Friedrichs inequality of Thm. 2.9.9,
- the multiplicative trace inequality of Thm. 2.9.10:

$$\begin{aligned}
 \int_{\partial\Omega} hv \, dS &\stackrel{(2.2.24)}{\leq} \|h\|_{L^2(\partial\Omega)} \|v\|_{L^2(\partial\Omega)} \stackrel{\text{Thm. 2.9.10}}{\leq} \|h\|_{L^2(\partial\Omega)} \|v\|_{H^1(\Omega)} \\
 &\stackrel{\text{Thm. 2.9.9}}{\leq} \|h\|_{L^2(\partial\Omega)} |v|_{H^1(\Omega)} \quad \forall v \in H_*^1(\Omega) .
 \end{aligned}$$



$h \in L^2(\partial\Omega)$ provides valid Neumann data for the 2nd order elliptic BVP (2.9.4).

In, particular Neumann data h can be *discontinuous*.



Learning outcomes

After having studied this chapter you should be able to

- convert a quadratic minimization problem into a linear variational problem
- use the formal calculus of variations to find the variational problem induced by a minimization problem posed on a space of functions in two or three dimensions.
- know the norms of the Sobolev spaces $L^2(\Omega)$, $H^1(\Omega)$, and $H_0^1(\Omega)$ and how to use them in the statement of variational problems.
- state the continuity featured by piecewise smooth functions in a Sobolev space.

- appreciate the importance of the continuity in the energy norm of right hand side functionals of variational problems.
- extract a PDE and boundary conditions from a variational problem using integration by parts.
- recast a boundary value problem for a 2nd-order PDE in variational form (using suitable Sobolev spaces).
- tell which boundary conditions make sense for a given 2nd-order PDE.
- distinguish essential and natural boundary conditions for a PDE in variational form.

3

Finite Element Methods (FEM)

In this chapter:

- Problem : linear scalar second-order elliptic boundary value problem → Ch. 2
- Perspective : **variational** interpretation in Sobolev spaces → Sect. 2.8
- Objective : algorithm for the computation of an **approximate numerical solution**

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Preface

Sect. 1.5.1 introduced the fundamental ideas of the **Galerkin discretization** of variational problems, or, equivalently, of minimization problems, posed over function spaces. A key ingredient are suitably chosen finite-dimensional trial and test spaces, equipped with ordered bases.

In Sect. 1.5.1.2 the abstract approach was discussed for two-point boundary value problems and the concrete case of **piecewise linear** trial and test spaces, built upon a partition (mesh/grid) of the interval (domain). In this context the locally supported tent functions lent themselves as natural basis functions.

This chapter is devoted to extending the linear finite element method in 1D to

- 2nd-order linear variational problems on bounded spatial domains Ω in two and three dimensions,
- piecewise polynomial trial/test functions of higher degree.

The leap from $d = 1$ to $d = 2$ will encounter additional difficulties and many new aspects will emerge. This chapter will elaborate on them and present policies how to tackle them.

Throughout, we will restrict ourselves to **linear 2nd-order elliptic variational problems** on spatial domains $\Omega \in \mathbb{R}^d$, $d = 2, 3$, with the properties listed in Rem. 2.1.1.

☛ 2nd-order elliptic Dirichlet problem:

$$\begin{aligned} u &\in H^1(\Omega), \\ u &= g \text{ on } \partial\Omega \end{aligned} \quad ; \quad \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega), \quad (2.3.5)$$

with *continuous* (\rightarrow Rem. 2.9.7) Dirichlet data $g \in C^0(\partial\Omega)$.

☛ 2nd-order elliptic Neumann problems:

$$u \in H_*^1(\Omega): \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} + \int_{\partial\Omega} h v \, dS \quad \forall v \in H_*^1(\Omega), \quad (2.8.24)$$

with *piecewise continuous* (\rightarrow Rem. 2.9.8) Neumann data $h \in C_{\text{pw}}^0(\partial\Omega)$ that satisfy the **compatibility condition** (2.8.19).

A simpler version with homogeneous Neumann data and reaction term:

$$u \in H^1(\Omega): \int_{\Omega} \boldsymbol{\alpha}(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v + c(\mathbf{x}) u v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H^1(\Omega), \quad (3.0.1)$$

with reaction coefficient $c: \Omega \mapsto \mathbb{R}^+$, $c \in C_{\text{pw}}^0(\Omega)$. Note that no compatibility conditions is required in this case.

Rem. 1.5.6 still applies: all functions (coefficient $\boldsymbol{\alpha}$, source function f , Dirichlet data g) may be given only in procedural form. Recall the discussion of the consequences in Rem. 1.5.53 and Sect. 1.5.1.2

3.1 Galerkin discretization

Recall the concept of “discretization”, see Sect. 1.5:

Not a moot point: any computer can only handle a finite amount of information (reals)

Variational boundary value
problem

DISCRETIZATION

System of a finite number of
equations for (real) unknowns

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Targetted: **linear variational problem** (1.4.7)

$$u \in V_0: \mathbf{a}(u, v) = \ell(v) \quad \forall v \in V_0, \quad (3.1.1)$$

• $V_0 \hat{=}$ vector space (Hilbert space) (usually a Sobolev space \rightarrow Sect. 2.2) with norm $\|\cdot\|_V$,

• $\mathbf{a}(\cdot, \cdot) \hat{=}$ bilinear form, continuous on V_0 , which means

$$\exists C > 0: |\mathbf{a}(u, v)| \leq C \|u\|_V \|v\|_V \quad \forall u, v \in V. \quad (3.1.2)$$

• $\ell \hat{=}$ continuous linear form in the sense of, *cf.* (2.2.3),

$$\exists C > 0: |\ell(v)| \leq C \|v\|_V \quad \forall v \in V_0. \quad (3.1.3)$$

The importance of this continuity is discussed in the beginning of Sect. 2.2, see also Rem. 2.3.27.

If \mathbf{a} is symmetric and positive definite (\rightarrow Def. 2.1.32), we may choose $\|\cdot\|_V := \|\cdot\|_{\mathbf{a}}$, “energy norm”, see Def. 2.1.35. Continuity of \mathbf{a} w.r.t. $\|\cdot\|_{\mathbf{a}}$ is clear.

Recall from Sect. 1.5.1:

Idea of **Galerkin discretization**

Replace V_0 in (3.1.1) with a **finite dimensional subspace**.
($V_{0,N} \subset V_0$ called Galerkin (or discrete) trial space/test space)

Twofold nature of symbol “ N ”:

- N = formal index, tagging “discrete entities” (\rightarrow “finite amount of information”)
- $N = \dim V_{N,0} \hat{=}$ dimension of Galerkin trial/test space



Discrete variational problem, cf. (1.5.10),

$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N} . \quad (3.1.4)$$

Galerkin solution

Theorem 3.1.5 (Existence and uniqueness of solutions of discrete variational problems).

If the bilinear form $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ is symmetric and positive definite (\rightarrow Def. 2.1.32) and the linear form $\ell : V_0 \mapsto \mathbb{R}$ is **continuous** in the sense of

$$\exists C_\ell > 0: |\ell(u)| \leq C_\ell \|u\|_a \quad \forall u \in V_0, \quad (2.2.3)$$

then the discrete variational problem has a unique **Galerkin solution** $u_N \in V_{0,N}$ that satisfies the **stability estimate** (\rightarrow Sect. 2.3.2)

$$\|u_N\|_a \leq C_\ell. \quad (3.1.6)$$

Proof. Uniqueness of u_N is clear:

$$\begin{aligned} \mathbf{a}(u_N, v_N) &= \ell(v_N) \quad \forall v_N \in V_{0,N} \\ \mathbf{a}(w_N, v_N) &= \ell(v_N) \quad \forall v_N \in V_{0,N} \end{aligned} \quad \Rightarrow \quad \mathbf{a}(u_N - w_N, v_N) = 0 \quad \forall v_N \in V_{N,0}$$

$$\begin{aligned} v_N := u_N - w_N \in V_{0,N} \\ \implies \|u_N - w_N\|_a = 0 \quad \xrightarrow{\mathbf{a} \text{ s.p.d.}} \quad u_N - w_N = 0. \end{aligned}$$

The discrete linear variational problem (3.1.4) is set in the *finite-dimensional* space $V_{0,N}$. Thus, uniqueness of solutions is equivalent to existence of solutions (\rightarrow linear algebra).

If you do not like this abstract argument, wait and see the equivalence of (3.1.4) with a linear system

of equations. It will turn out that under the assumptions of the theorem, the resulting system matrix will be symmetric and positive definite in the sense of [21, Def. 2.7.9].

The estimate (3.1.6) is immediate from setting $v_N := u_N$ in (3.1.4)

$$|\mathbf{a}(u_N, u_N)| = |\ell(u_N)| \leq C_\ell (\mathbf{a}(u_N, u_N))^{1/2} . \quad \square$$

Recall from Sect. 1.5.1:

2nd step of Galerkin discretization:

Introduce (ordered) **basis** \mathfrak{B}_N of $V_{0,N}$:

$$\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\} \subset V_N \quad , \quad V_N = \text{Span} \{ \mathfrak{B}_N \} \quad , \quad N := \dim(V_N) .$$


► **Unique** basis representations:

$$\begin{aligned} u_N &= \mu_1 b_N^1 + \dots + \mu_N b_N^N , & \mu_i &\in \mathbb{R} \\ v_N &= \nu_1 b_N^1 + \dots + \nu_N b_N^N , & \nu_i &\in \mathbb{R} \end{aligned} \quad \text{plug into (3.1.4).}$$

Of course, there are infinitely many ways to choose the basis \mathfrak{B}_N . Below we will study the impact of different choices.

What follows repeats the derivation of (1.5.26) and, in particular, (1.5.64).

$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N}. \quad (3.1.4)$$



$$\left[\begin{array}{l} u_N = \mu_1 b_N^1 + \cdots + \mu_N b_N^N, \mu_i \in \mathbb{R} \\ v_N = \nu_1 b_N^1 + \cdots + \nu_N b_N^N, \nu_i \in \mathbb{R} \end{array} \right]$$

$$\sum_{k=1}^N \sum_{j=1}^N \mu_k \nu_j \mathbf{a}(b_N^k, b_N^j) = \sum_{j=1}^N \nu_j \ell(b_N^j) \quad \forall \nu_1, \dots, \nu_N \in \mathbb{R},$$



$$\sum_{j=1}^N \nu_j \left(\sum_{k=1}^N \mu_k \mathbf{a}(b_N^k, b_N^j) - \ell(b_N^j) \right) = 0 \quad \forall \nu_1, \dots, \nu_N \in \mathbb{R},$$



$$\sum_{k=1}^N \mu_k \mathbf{a}(b_N^k, b_N^j) = \ell(b_N^j) \quad \text{for } j = 1, \dots, N .$$



$$[\vec{\mu} = (\mu_1, \dots, \mu_N)^\top \in \mathbb{R}^N]$$

$$\mathbf{A} = \left(\mathbf{a}(b_N^k, b_N^j) \right)_{j,k=1}^N \in \mathbb{R}^{N,N} ,$$

$$\vec{\varphi} = \left(\ell(b_N^j) \right)_{j=1}^N .$$

$$\mathbf{A}\vec{\mu} = \vec{\varphi} , \text{ with}$$

A linear system of equations

Linear discrete variational problem

$$u_N \in V_{0,N}: \mathbf{a}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N}$$

Choosing basis \mathfrak{B}_N

Linear system

of equations

$$\mathbf{A}\vec{\mu} = \vec{\varphi}$$

Galerkin matrix: $\mathbf{A} = \left(\mathbf{a}(b_N^k, b_N^j) \right)_{j,k=1}^N \in \mathbb{R}^{N,N} ,$

Right hand side vector: $\vec{\varphi} = \left(\ell(b_N^j) \right)_{j=1}^N \in \mathbb{R}^N ,$

Coefficient vector: $\vec{\mu} = (\mu_1, \dots, \mu_N)^\top \in \mathbb{R}^N ,$

Recovery of solution: $u_N = \sum_{k=1}^N \mu_k b_N^k .$

(Legacy) terminology for FEM:	Galerkin matrix	=	stiffness matrix
	Right hand side vector	=	load vector
	Galerkin matrix for $(u, v) \mapsto \int_{\Omega} uv \, dx$	=	mass matrix

Corollary 3.1.7. (3.1.4) has unique solution $\Leftrightarrow \mathbf{A}$ nonsingular

Remark 3.1.8 (Impact of choice of basis).

Choice of \mathcal{B}_N in theory does **not** affect $u_N \Rightarrow$ No impact on discretization error !

But: Key properties (e.g., conditioning) of matrix \mathbf{A} crucially depend on basis \mathcal{B}_N !

Lemma 3.1.9. Consider (3.1.4) and two bases of $V_{0,N}$,

$$\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\} \quad , \quad \underline{\mathfrak{B}}_N := \{\underline{b}_N^1, \dots, \underline{b}_N^N\} \quad ,$$

related by

$$\underline{b}_N^j = \sum_{k=1}^N s_{jk} b_N^k \quad \text{with} \quad \mathbf{S} = (s_{jk})_{j,k=1}^N \in \mathbb{K}^{N,N} \text{ regular.} \quad (3.1.10)$$

► Galerkin matrices $\mathbf{A}, \underline{\mathbf{A}} \in \mathbb{K}^{N,N}$, right hand side vectors $\vec{\varphi}, \underline{\vec{\varphi}} \in \mathbb{K}^N$, and coefficient vectors $\vec{\mu}, \underline{\vec{\mu}} \in \mathbb{R}^N$, respectively, satisfy

$$\underline{\mathbf{A}} = \mathbf{S} \mathbf{A} \mathbf{S}^T \quad , \quad \underline{\vec{\varphi}} = \mathbf{S} \vec{\varphi} \quad , \quad \underline{\vec{\mu}} = \mathbf{S}^{-T} \vec{\mu} \quad . \quad (3.1.11)$$

Proof. Make use of the **bilinearity** of \mathbf{a} (\rightarrow Def. 1.3.23) and (3.1.10):

$$\underline{\mathbf{A}}_{lm} = \mathbf{a}(\underline{b}_N^m, \underline{b}_N^l) = \sum_{k=1}^N \sum_{j=1}^N s_{mk} \mathbf{a}(b_N^k, b_N^j) s_{lj} = \sum_{k=1}^N \underbrace{\left(\sum_{j=1}^N s_{lj} \mathbf{A}_{jk} \right)}_{(\mathbf{S} \mathbf{A})_{lk}} s_{mk} = (\mathbf{S} \mathbf{A} \mathbf{S}^T)_{lm} \quad ,$$

Reminder of linear algebra:

Definition 3.1.12 (Congruent matrices).

Two matrices $\mathbf{A} \in \mathbb{K}^{N,N}$, $\mathbf{B} \in \mathbb{K}^{N,N}$, $N \in \mathbb{N}$, are called *congruent*, if there is a regular matrix $\mathbf{S} \in \mathbb{K}^{N,N}$ such that $\mathbf{B} = \mathbf{SAS}^H$.

Equivalence relation on square matrices

Lemma 3.1.13.

Matrix property invariant under congruence

\Leftrightarrow

Property of Galerkin matrix invariant under change of basis \mathfrak{B}_N

Matrix properties invariant under congruence:

- regularity \rightarrow [21, Def. 2.0.1]
- symmetry
- positive definiteness \rightarrow [21, Def. 2.7.9]

Not invariant are

- sparsity and bandstructure, *cf.* linear finite elements (\rightarrow Sect. 1.5.1.2) and spectral Galerkin (\rightarrow Sect. 1.5.1.1)
- conditioning, *cf.* Ex. 1.5.68

These properties have fundamental consequences for numerical solution of the linear system of equations (required storage, computational effort, and impact of roundoff errors).



3.2 Case study: Triangular linear FEM in two dimensions

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

This section elaborates how to extend the linear finite element Galerkin discretization of Sect. 1.5.1.2 to two dimensions. Familiarity with the 1D setting is essential for understanding the current section.

Initial focus: well-posed 2nd-order linear variational problem posed on $H^1(\Omega)$ (\rightarrow Def. 2.2.18)

Example: Neuman problem with homogeneous Neumann data and reaction term

$$u \in H^1(\Omega): \int_{\Omega} \alpha(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v + c(\mathbf{x}) u v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} \forall v \in H^1(\Omega), \quad (3.0.1)$$

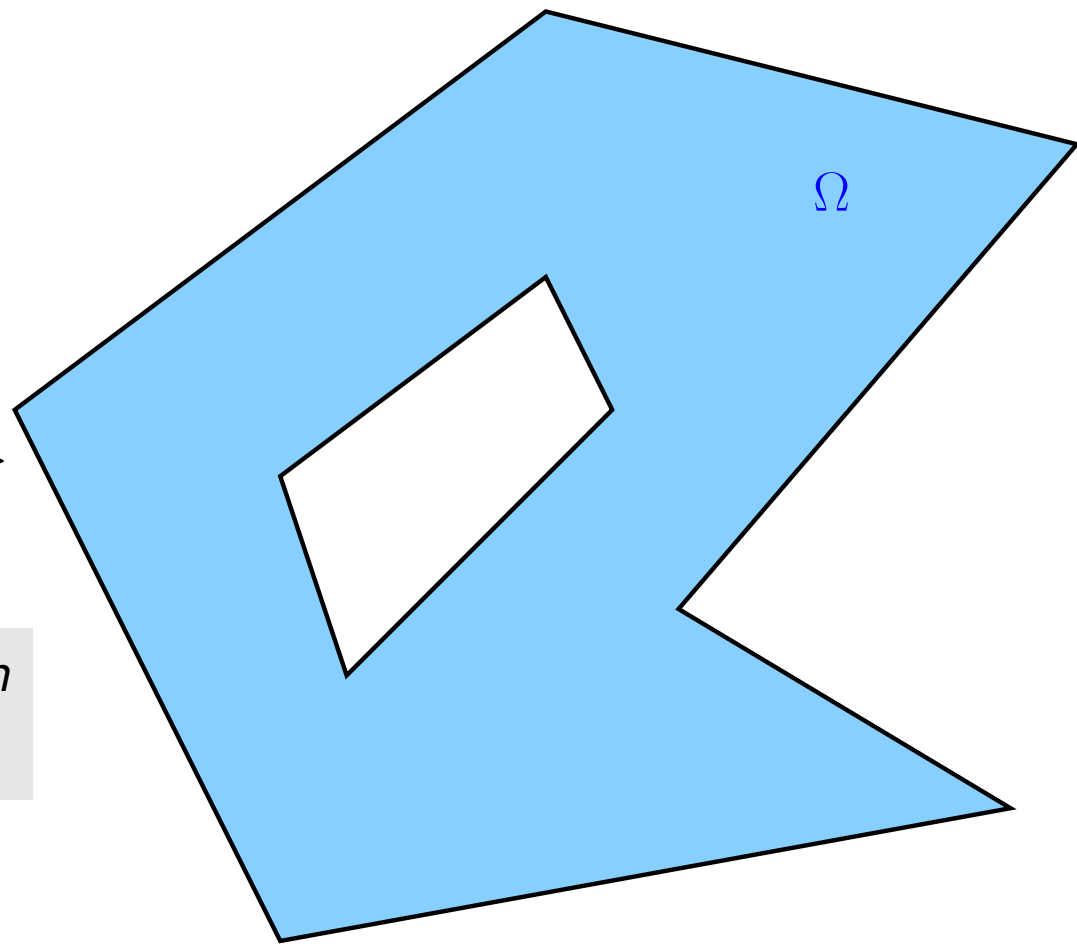
$\Updownarrow \leftarrow$ see Sect. 2.4

$$\text{BVP:} \quad \begin{aligned} -\operatorname{div}(\alpha(\mathbf{x}) \mathbf{grad} u) + c(\mathbf{x}) u &= f && \text{in } \Omega, \\ \mathbf{grad} u \cdot \mathbf{n} &= 0 && \text{on } \partial\Omega. \end{aligned}$$

Assumptions on **domain** $\Omega \subset \mathbb{R}^2$,
see Rem. 2.1.1:

Ω is a **polygon**

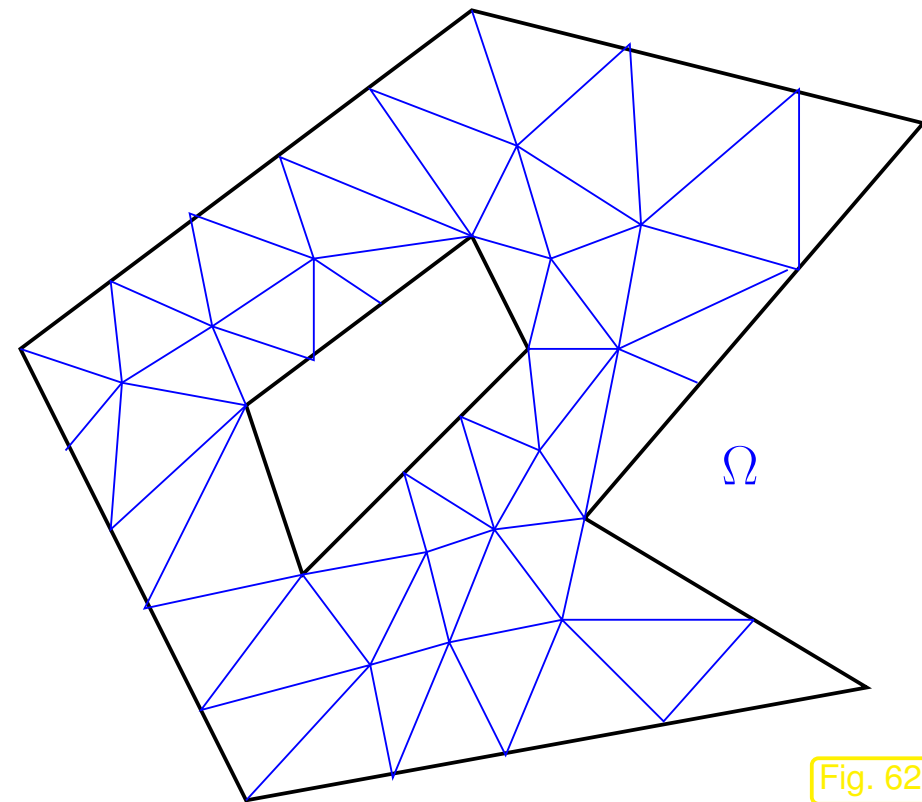
polygon with 10 corners \triangleright



By default, the domain Ω is assumed to be an *open*
set, that is, $x \in \Omega$ implies $x \notin \partial\Omega$!

3.2.1 Triangulations

What is the 2D counterpart of mesh/grid \mathcal{M} from Sect. (1.5.1.2) ?




Triangulation \mathcal{M} of Ω :

- (i) $\mathcal{M} = \{K_i\}_{i=1}^M$, $M \in \mathbb{N}$, $K_i \hat{=} \text{open triangle}$
- (ii) disjoint interiors: $i \neq j \Rightarrow K_i \cap K_j = \emptyset$
- (iii) tiling property: $\bigcup_{i=1}^M \overline{K_i} = \overline{\Omega}$
- (iv) intersection $\overline{K_i} \cap \overline{K_j}$, $i \neq j$,
is
 - either \emptyset
 - or an edge of both triangles
 - or a vertex of both triangles

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

 notation: $\overline{\cdot} \hat{=}$ a subset of \mathbb{R}^d together with its boundary (“closure”)

Parlance: vertices of triangles = **nodes** of mesh (= set $\mathcal{V}(\mathcal{M})$)

A mesh that does not comply with the property (iv)
from above.

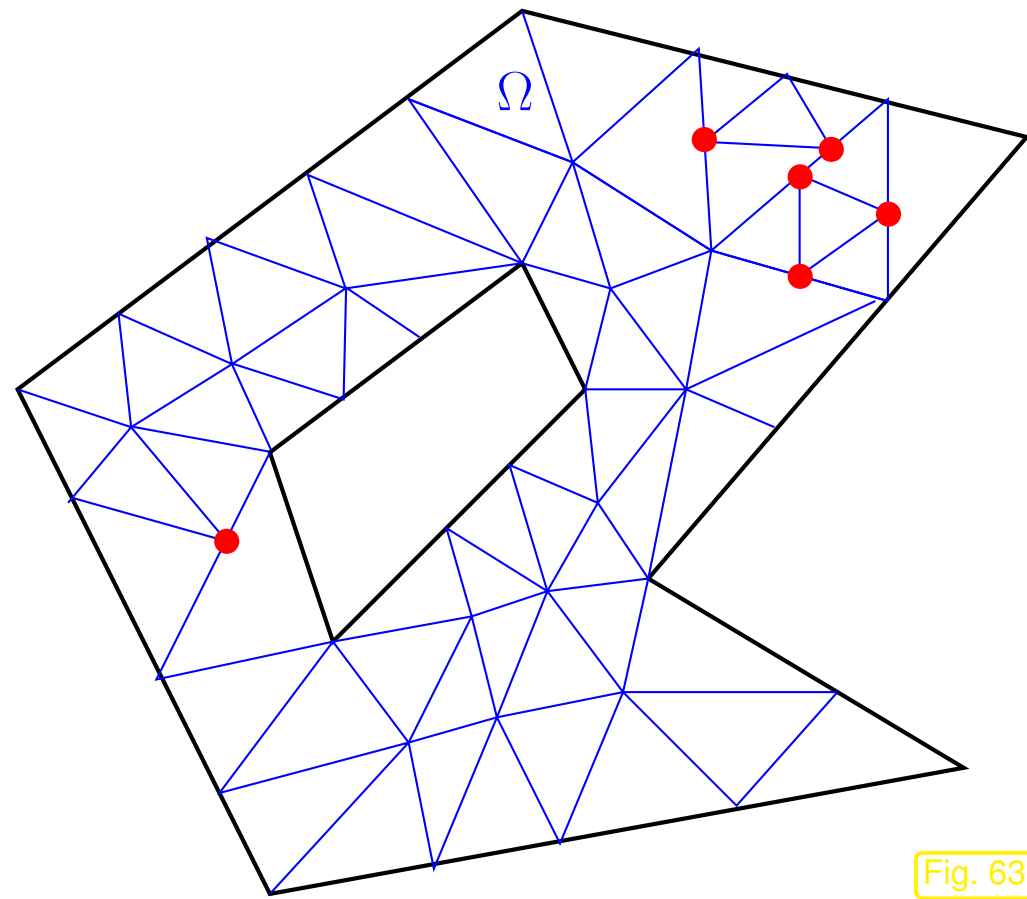


Fig. 63

3.2.2 Linear finite element space

Next goal: generalize the spline space $\mathcal{S}_1^0(\mathcal{M}) \subset H^1([a, b])$ of piecewise linear functions on a 1D grid \mathcal{M} , see Fig. 25, that was used as Galerkin trial/test space in 1D in Sect. 1.5.1.2

$$V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M}) := \left\{ v \in C^0([0, 1]): v|_{[x_{i-1}, x_i]} \text{ linear, } i = 1, \dots, M, v(a) = v(b) = 0 \right\} .$$

$d = 1$

$d = 2$

Grid/mesh **cells**: intervals $]x_{i-1}, x_i[, i = 1, \dots, M$

triangles $K_i, i = 1, \dots, M$

Linear functions: $x \in \mathbb{R} \mapsto \alpha + \beta \cdot x, \alpha, \beta \in \mathbb{R}$

$\mathbf{x} \in \mathbb{R}^2 \mapsto \alpha + \boldsymbol{\beta} \cdot \mathbf{x}, \alpha \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}^2$



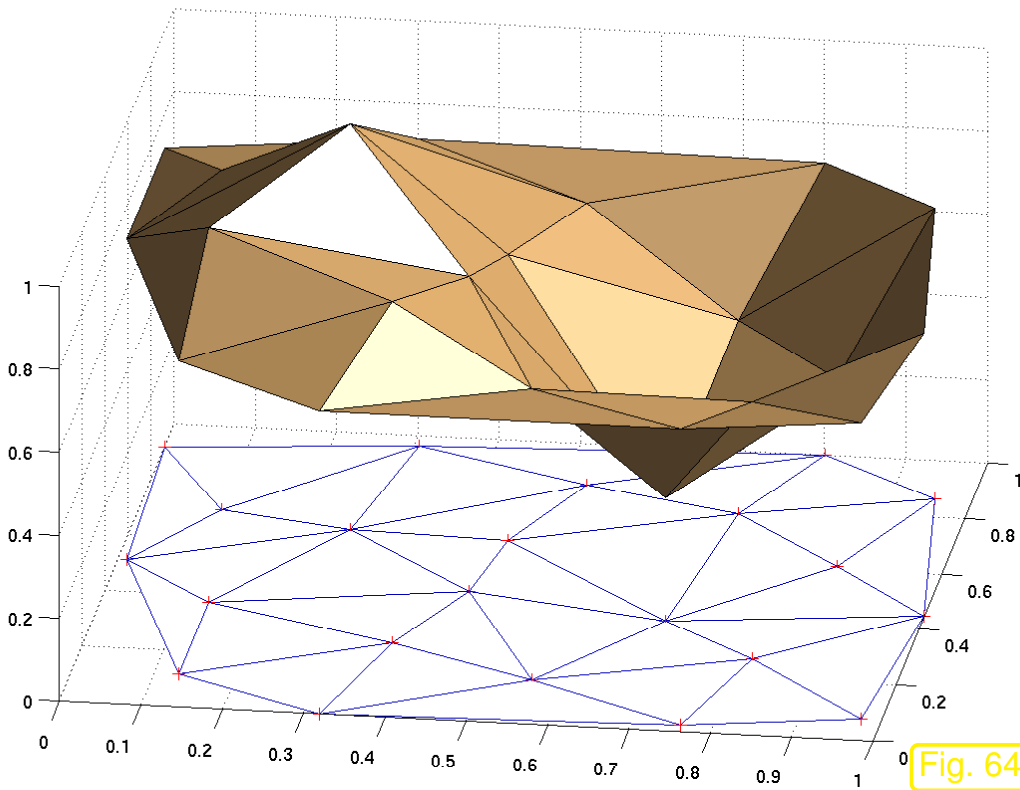
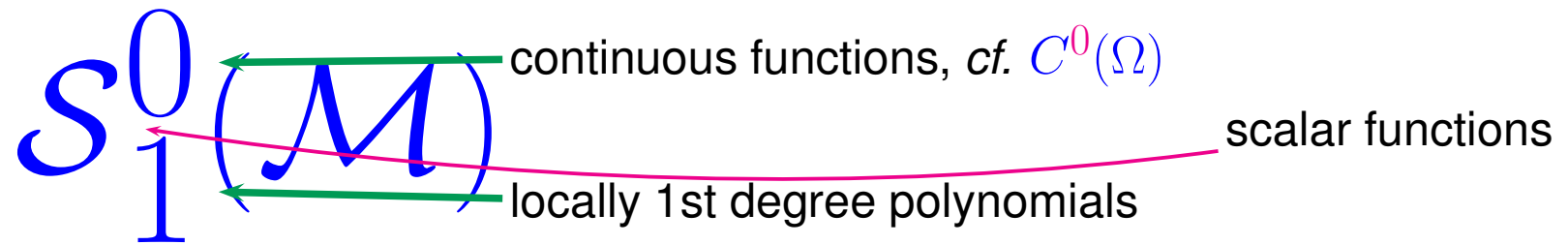
$$V_{0,N} = \mathcal{S}_1^0(\mathcal{M}) := \left\{ v \in C^0(\bar{\Omega}): \forall K \in \mathcal{M}: \begin{array}{l} v|_K(\mathbf{x}) = \alpha_K + \boldsymbol{\beta}_K \cdot \mathbf{x}, \\ \alpha_K \in \mathbb{R}, \boldsymbol{\beta}_K \in \mathbb{R}^2, \mathbf{x} \in K \end{array} \right\} \subset H^1(\Omega)$$

see Thm. 2.2.26

Functions of the form $x \mapsto \alpha_K + \beta_K \cdot x$, $\alpha_K \in \mathbb{R}$, $\beta_K \in \mathbb{R}^2$ are called **(affine) linear**.



notation:



◁ continuous piecewise affine linear function $\in S_1^0(\mathcal{M})$ on a triangular mesh \mathcal{M}

Fig. 64

Remark 3.2.2 (Piecewise gradient).

Thm. 2.2.26 $\Rightarrow \mathcal{S}_1^0(\mathcal{M}) \subset H^1(\Omega)$

\Rightarrow for $u_N \in \mathcal{S}_1^0(\mathcal{M})$ the gradient $\mathbf{grad} u_N$ can be computed on each triangle as **piecewise constant** function, cf. Ex. 2.2.29.

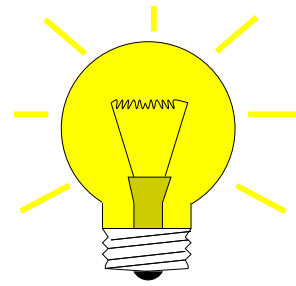
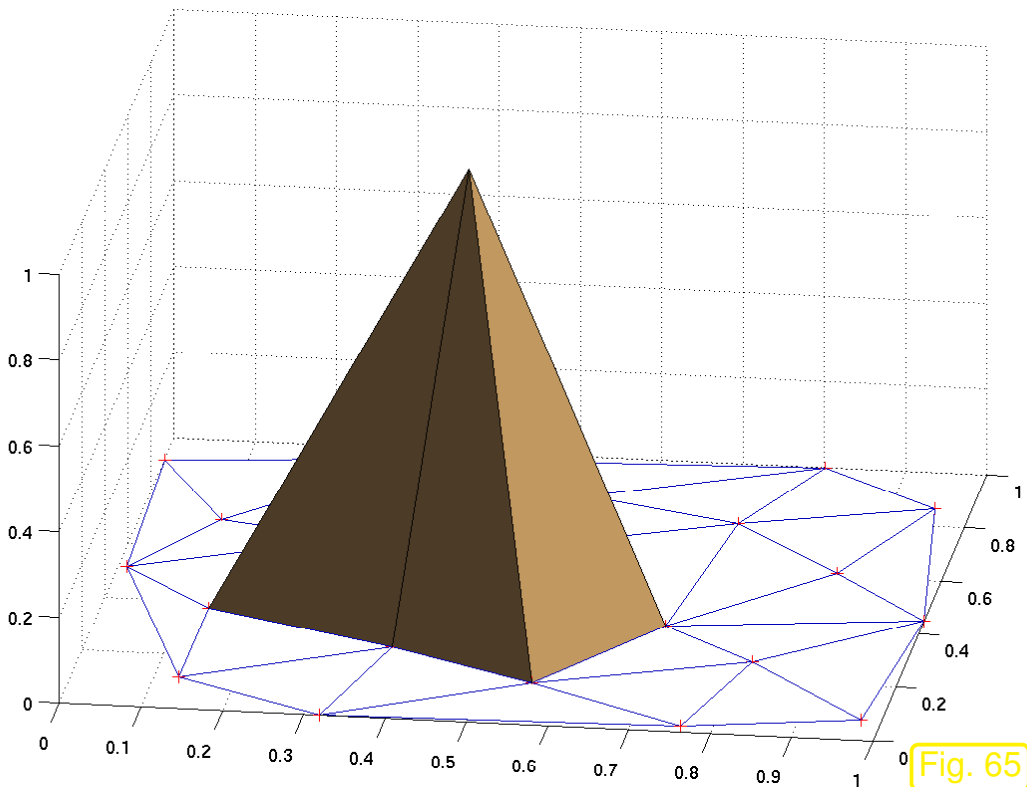
(On $K \in \mathcal{M}$: $\mathbf{grad}(\alpha_K + \beta_K \cdot \mathbf{x}) = \beta_K$)



3.2.3 Nodal basis functions

Next goal: generalization of “tent functions”, see (1.5.76).

Recall condition (1.5.77), which *defines* a tent function in the space $\mathcal{S}_1^0(\mathcal{M})$. This approach carries over to 2D.



Idea: *define* (?) basis function $b_N^x, x \in \mathcal{V}(\mathcal{M})$, by

$$b_N^x(y) = \begin{cases} 1 & , \text{ if } y = x , \\ 0 & , \text{ if } y \in \mathcal{V}(\mathcal{M}) \setminus \{x\} \end{cases} \quad (3.2.3)$$

Is this possible ?

Reasoning: there is exactly one plane through three non-collinear points in \mathbb{R}^3 . The graph of a linear function $\mathbb{R}^2 \mapsto \mathbb{R}$ is a plane.

➤ On a triangle K with vertices a^1, a^2, a^3 : (affine) linear $q : K \mapsto \mathbb{R}$ uniquely determined by values $q(a^i)$.

▶ $v_N \in \mathcal{S}_1^0(\mathcal{M})$ uniquely determined by $\{v_N(x), x \text{ node of } \mathcal{M}\}$!

$$\dim \mathcal{S}_1^0(\mathcal{M}) = \#\mathcal{V}(\mathcal{M})$$

$(\mathcal{V}(\mathcal{M})) = \text{set of nodes (= vertices of triangles) of } \mathcal{M}$

Note: it is the *condition (iv)* on a valid triangulation has made possible the construction of the basis function b_N^x for each $x \in \mathcal{V}(\mathcal{M})$; not simple basis functions could be associated with the red vertices in Fig. 63.

Now we have found the perfect 2D counterpart of the tent function basis (\rightarrow Fig. 26, (1.5.77)) of the linear finite element space in 1D:

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Writing $\mathcal{V}(\mathcal{M}) = \{x^1, \dots, x^N\}$, the **nodal basis** $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ of $\mathcal{S}_1^0(\mathcal{M})$ is defined by the conditions

$$b_N^i(x^j) = \begin{cases} 1 & , \text{ if } i = j , \\ 0 & \text{ else,} \end{cases} \quad i, j \in \{1, \dots, N\} .$$

(3.2.4)

Ordering (\leftrightarrow numbering) of nodes assumed !

Piecewise linear nodal basis function
 (“hat function”/ “tent function”)

$$u_N = \sum_{i=1}^N \mu_i b_N^i \in \mathcal{S}_1^0(\mathcal{M})$$

coefficient μ_j = “nodal value” of u_N at j -th node of \mathcal{M}

$$u_N(\mathbf{x}^j) = \mu_j$$

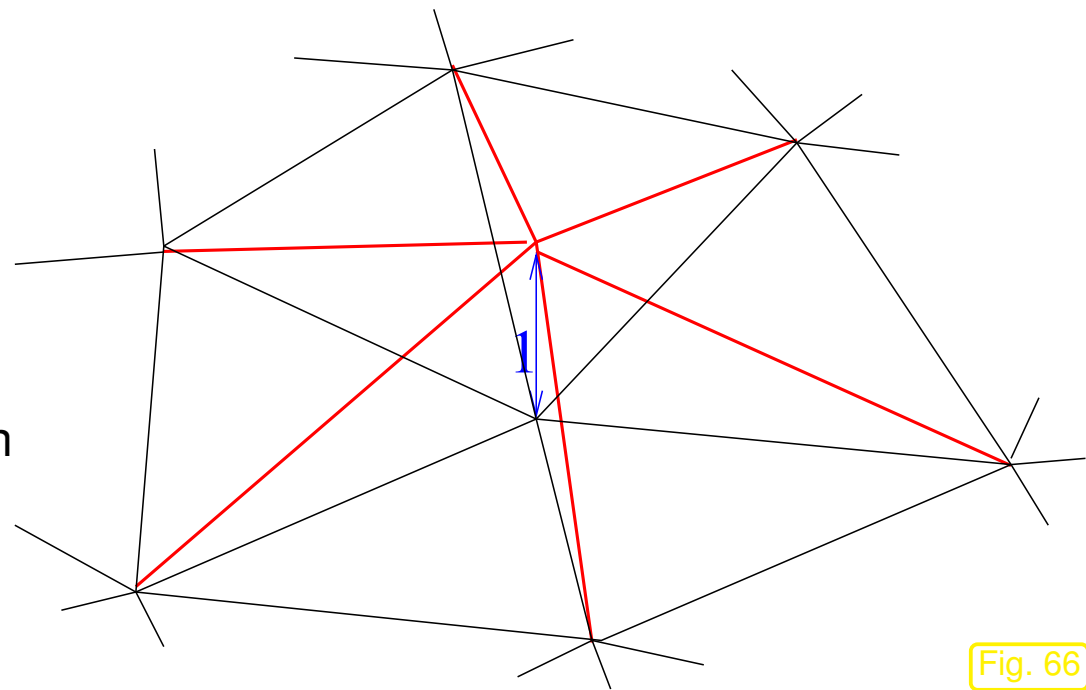


Fig. 66

R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

Remark 3.2.5 (Linear finite element space for homogeneous Dirichlet problem).

Recall that the Dirichlet problem with homogeneous boundary conditions $u|_{\partial\Omega} = 0$ is posed on the Sobolev space $H_0^1(\Omega)$ (\rightarrow Def. 2.2.15), see (2.3.5), Ex. 2.8.1.

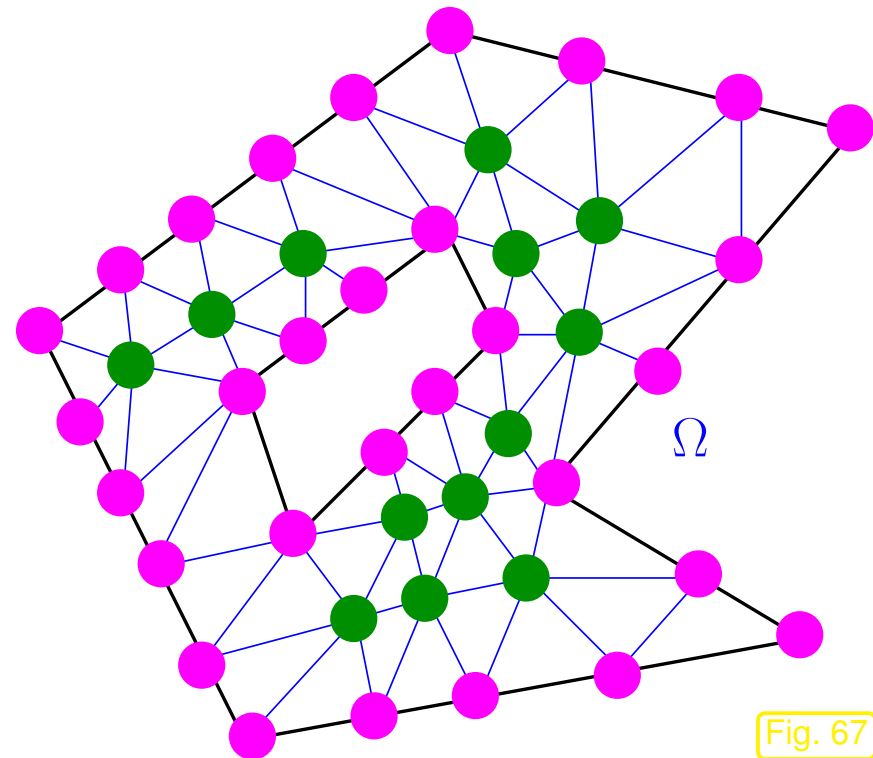
Galerkin space for homogeneous Dirichlet b.c.: $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M}) := \mathcal{S}_1^0(\mathcal{M}) \cap H_0^1(\Omega)$

Notation:

$$\mathcal{S}_{1,0}^0(\mathcal{M}) \xrightarrow{\text{zero on } \partial\Omega, \text{ cf. } H_0^1(\Omega)}$$

▶ $\mathcal{S}_{1,0}^0(\mathcal{M}) = \text{Span} \left\{ b_N^j : \mathbf{x}^j \in \Omega \text{ (interior node !)} \right\}$

▶ $\dim \mathcal{S}_{1,0}^0(\mathcal{M}) = \#\{ \mathbf{x} \in \mathcal{V}(\mathcal{M}) : \mathbf{x} \notin \partial\Omega \}$



◁ “Location” of nodal basis functions:
(mesh $\mathcal{M} \rightarrow$ Fig. 150)

•, • \rightarrow nodal basis functions of $\mathcal{S}_1^0(\mathcal{M})$

• \rightarrow nodal basis functions of $\mathcal{S}_{1,0}^0(\mathcal{M})$

Bottom line: the Galerkin trial/test space contained in $H_0^1(\Omega)$ is obtained by dropping all “tent functions” that do not vanish on $\partial\Omega$ from the basis.



3.2.4 Sparse Galerkin matrix

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Now: $\mathbf{a} \hat{=}$ any (symmetric) bilinear form occurring in a linear 2nd-order variational problem, most general form

$$\mathbf{a}(u, v) := \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} v + c(\mathbf{x}) u v \, d\mathbf{x} + \int_{\partial\Omega} h v \, dS, \quad u, v \in H^1(\Omega). \quad (3.2.6)$$

$b_N^j \hat{=}$ nodal basis function associated with vertex \mathbf{x}^j of triangulation \mathcal{M} of Ω , see Sect. 3.2.3.

Note: \mathbf{a} symmetric \Rightarrow symmetric Galerkin matrix

Now we study the **sparsity** (\rightarrow [21, Sect. 2.6]) of the Galerkin matrix $\mathbf{A} := \left(\mathbf{a}(b_N^j, b_N^i) \right)_{i,j=1}^N \in \mathbb{R}^{N,N}$, $N := \dim \mathcal{S}_1^0(\mathcal{M}) = \#\mathcal{V}(\mathcal{M})$, see Sect. 3.1.

The considerations are fairly parallel to those that made us understand that the Galerkin matrix for the 1D case was *tridiagonal*, see (1.5.84).

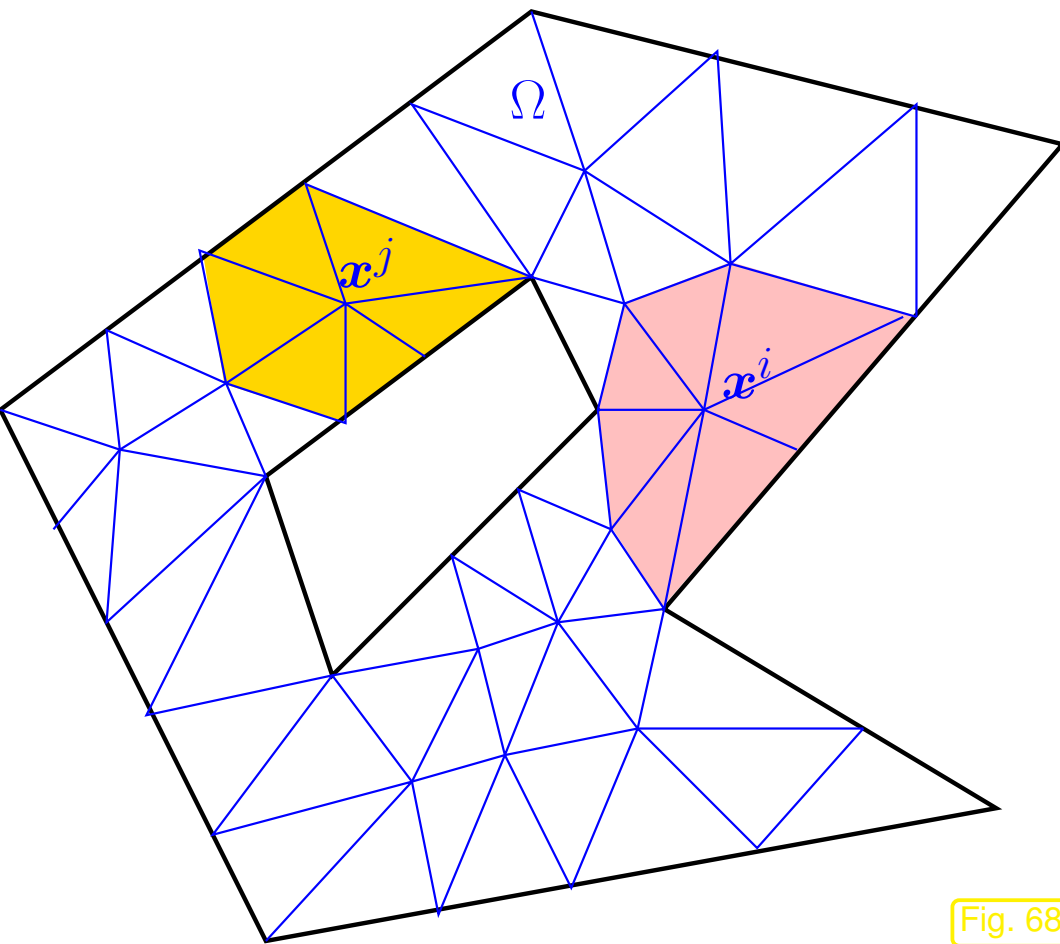


Fig. 68

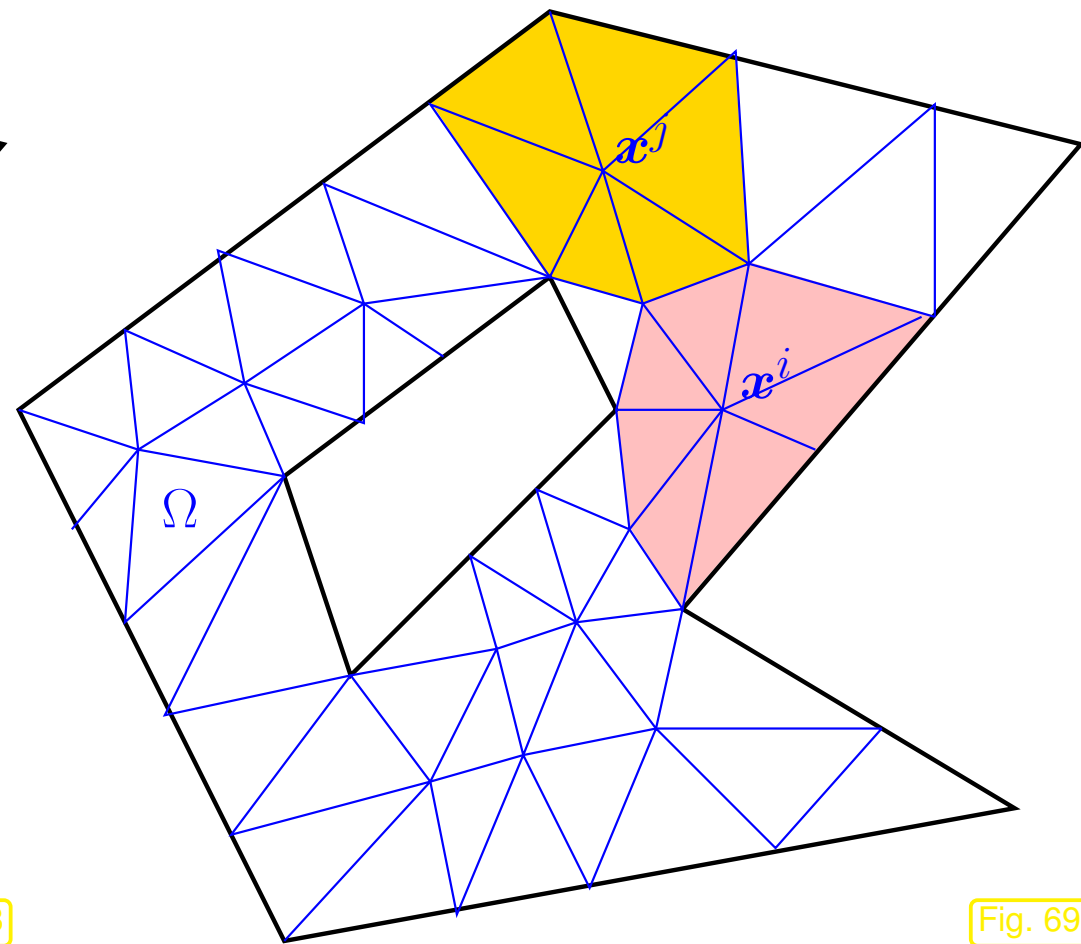


Fig. 69

Nodes $x^i, x^j \in \mathcal{V}(\mathcal{M})$
not connected by an edge $\Leftrightarrow \text{Vol}(\text{supp}(b_N^i) \cap \text{supp}(b_N^j)) = 0 \Rightarrow (\mathbf{A})_{ij} = 0.$

Lemma 3.2.7 (Sparsity of Galerkin matrix).

$$\exists C = C(\text{topology of } \Omega): \#\{(i, j) \in \{1, \dots, N\}^2: (\mathbf{A})_{ij} \neq 0\} \leq 7 \cdot N + C .$$

Proof. Euler's formula (http://en.wikipedia.org/wiki/Euler_characteristic)

$$\#\mathcal{M} - \#\mathcal{E}(\mathcal{M}) + \#\mathcal{V}(\mathcal{M}) = \chi_\Omega, \quad \chi_\Omega = \text{Euler characteristic of } \Omega .$$

Note that χ_Ω is a topological invariant (alternating sum of Betti numbers).

By combinatorial considerations (traverse edges and count triangles):

$$2 \cdot \#\mathcal{E}_I(\mathcal{M}) + \#\mathcal{E}_B(\mathcal{M}) = 3 \cdot \#\mathcal{M} ,$$

where $\mathcal{E}_I(\mathcal{M})$, $\mathcal{E}_B(\mathcal{M})$ stand for the sets of interior and boundary edges of \mathcal{M} , respectively.

$$\blacktriangleright \quad \#\mathcal{E}_I(\mathcal{M}) + 2\#\mathcal{E}_B(\mathcal{M}) = 3(\#\mathcal{V}(\mathcal{M}) - \chi_\Omega) .$$

Then use

$$N = \#\mathcal{V}(\mathcal{M}) \quad , \quad \text{nnz}(\mathbf{A}) \leq N + 2 \cdot \#\mathcal{E}(\mathcal{M}) \leq 7 \cdot \#\mathcal{V}(\mathcal{M}) - 6\chi_\Omega . \quad \square$$

Recall from [21, Def. 2.6.1]:

Notion 3.2.8 (Sparse matrix). $\mathbf{A} \in \mathbb{K}^{m,n}$, $m, n \in \mathbb{N}$, is *sparse*, if

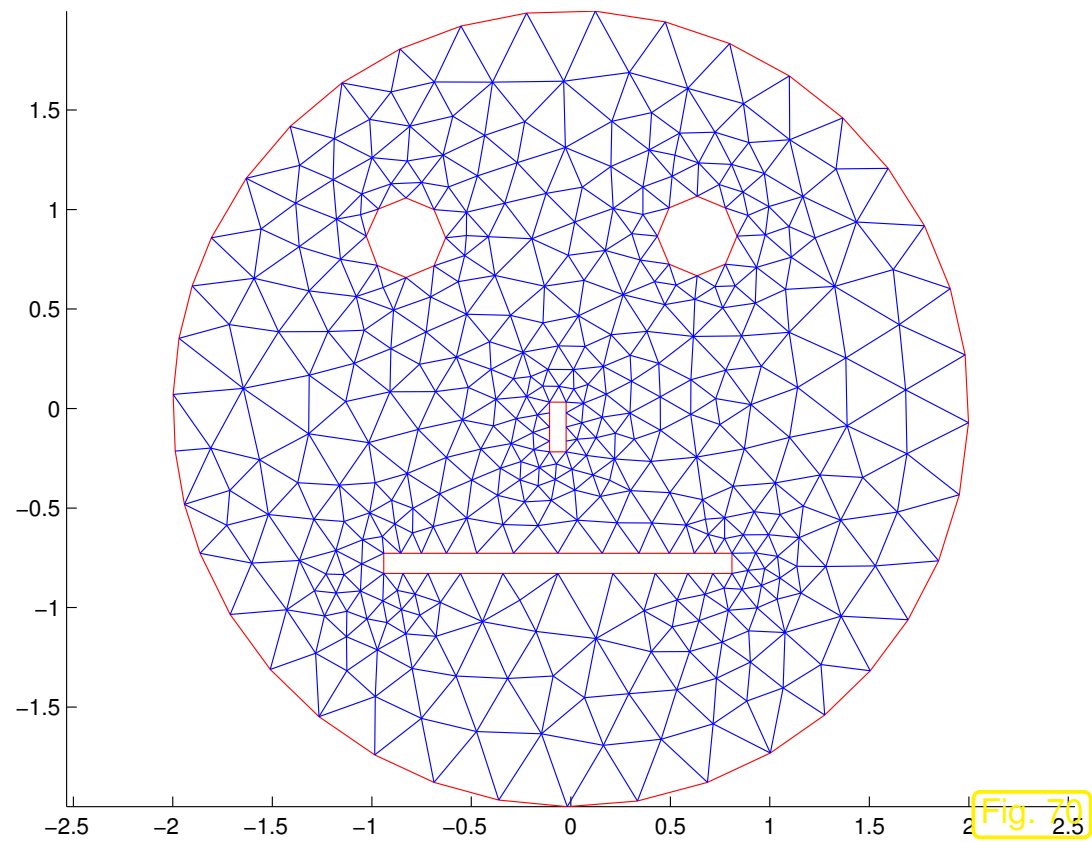
$$\text{nnz}(\mathbf{A}) := \#\{(i, j) \in \{1, \dots, m\} \times \{1, \dots, n\} : a_{ij} \neq 0\} \ll mn .$$

Sloppy parlance: matrix *sparse* \Leftrightarrow “almost all” entries $= 0$ / “only a few percent of” entries $\neq 0$

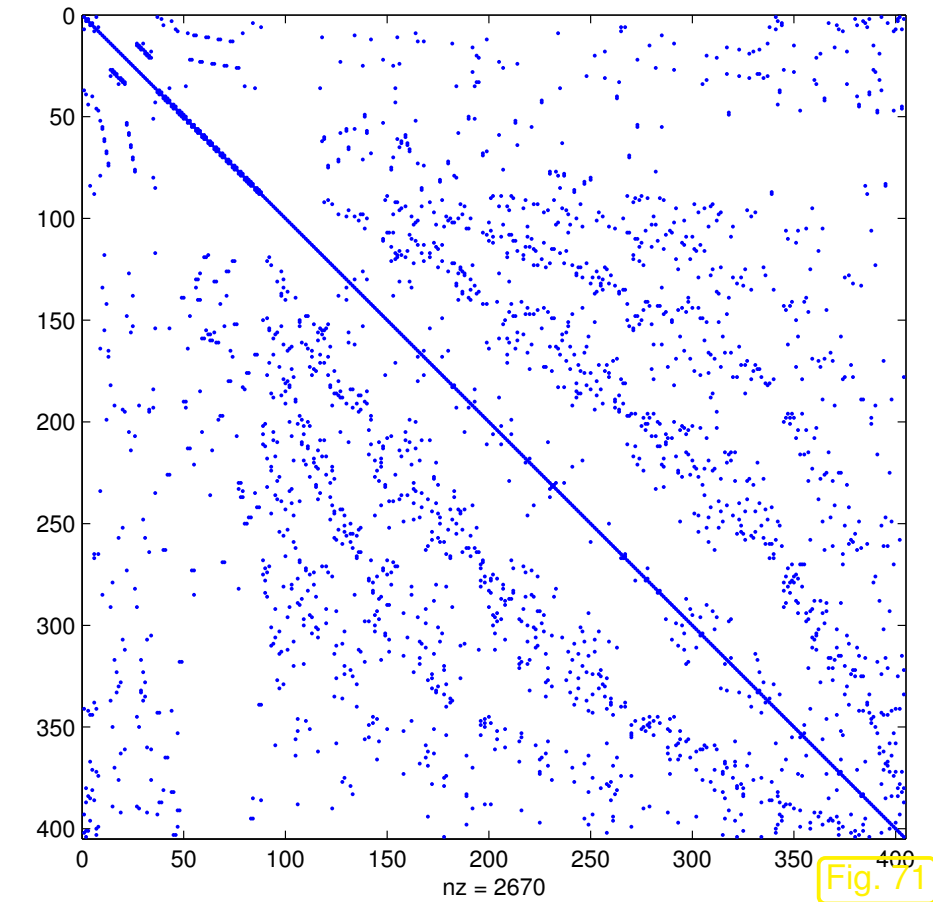
Galerkin discretization of a 2nd-order linear variational problems
utilizing the *nodal basis* of $\mathcal{S}_1^0(\mathcal{M})/\mathcal{S}_{1,0}^0(\mathcal{M})$
leads to sparse linear systems of equations.

Example 3.2.9 (Sparse Galerkin matrices).

\mathcal{M} = triangular mesh, $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$, homogeneous Dirichlet boundary conditions, linear 2nd-order scalar elliptic differential operator.



Triangular mesh \mathcal{M}



Resulting **sparsity pattern** of Galerkin matrix

Recall: visualization of sparsity pattern by means of MATLAB `spy`-command.



3.2.5 Computation of Galerkin matrix

For sake of simplicity consider

$$\mathbf{a}(u, v) := \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x}, \quad u, v \in H_0^1(\Omega).$$

and Galerkin discretization based on

- triangular mesh, see Sect. 3.2.1,
- discrete trial/test space $\mathcal{S}_{1,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$,
- *nodal basis* $\mathfrak{B}_N = \{b_N^j\}$ according to (3.2.3).

$$\blacktriangleright \quad (\mathbf{A})_{i,j} = \mathbf{a}(b_N^j, b_N^i) = \int_{\Omega} \mathbf{grad} b_N^j \cdot \mathbf{grad} b_N^i \, d\mathbf{x}$$

Sect. 3.2.4: we need only study the cases, where $\mathbf{x}^i, \mathbf{x}^j \in \mathcal{V}(\mathcal{M})$

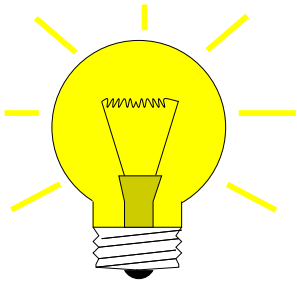
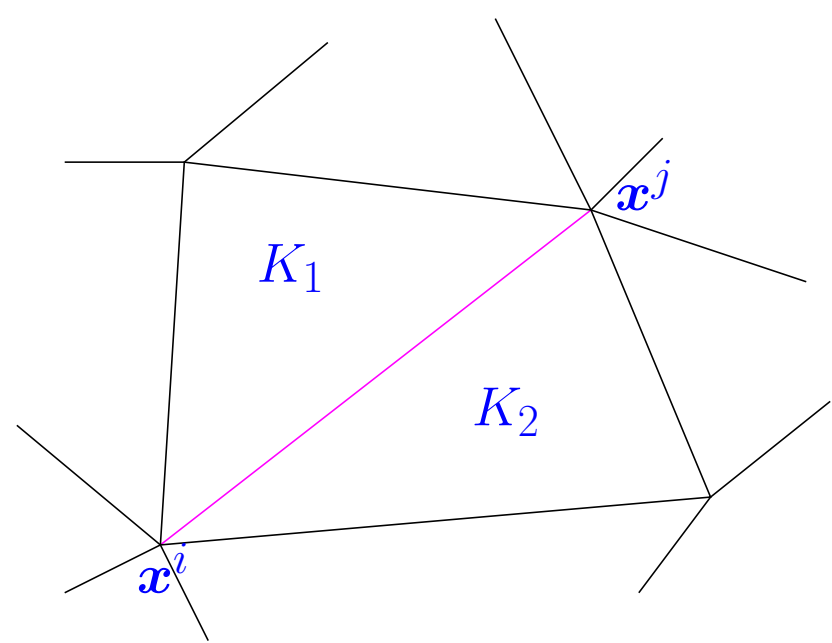
1. are connected by an edge of the triangulation,
2. coincide.

Idea:

“Assembly”

(add up *cell contributions*)

$$(\mathbf{A})_{ij} = \int_{K_1} \text{grad } b_{N|K_1}^j \cdot \text{grad } b_{N|K_1}^i \, dx + \int_{K_2} \text{grad } b_{N|K_2}^j \cdot \text{grad } b_{N|K_2}^i \, dx$$



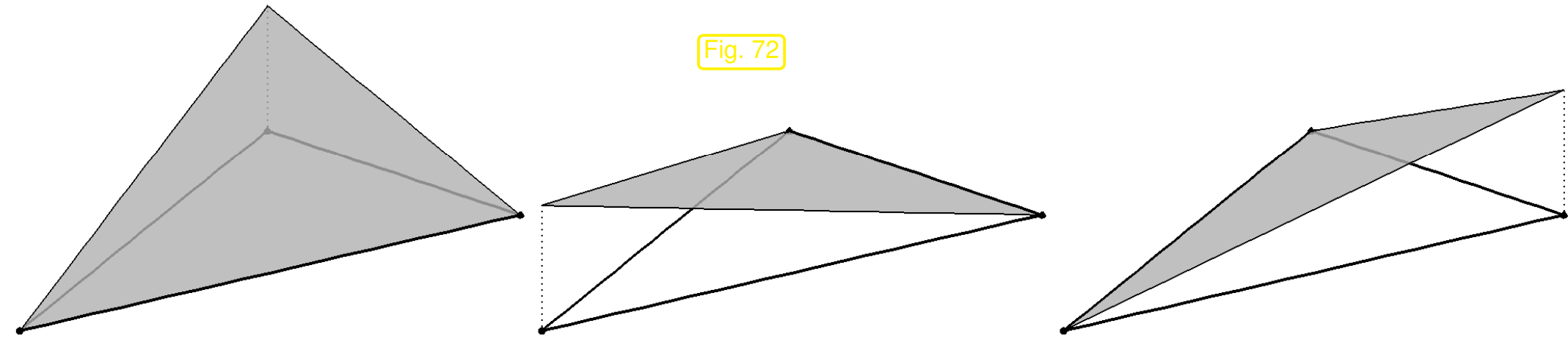
Zero in on single triangle $K \in \mathcal{M}$:

$$a_K(b_N^j, b_N^i) := \int_K \text{grad } b_{N|K}^j \cdot \text{grad } b_{N|K}^i \, dx \quad , \quad \mathbf{x}^i, \mathbf{x}^j \text{ vertices of } K \quad . \quad (3.2.10)$$

Use analytic representation for $b_{N|K}^i$:

if $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$ vertices of K , $\lambda_i := b_{N|K}^i$, $\mathbf{a}^i = \mathbf{x}^j$
 ($i \leftrightarrow$ local vertex number, $j \leftrightarrow$ global node number)

Fig. 72



Restrictions $\lambda_1, \lambda_2, \lambda_3$ of p.w. linear nodal basis functions of $\mathcal{S}_1^0(\mathcal{M})$ to triangle K

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

The functions $\lambda_1, \lambda_2, \lambda_3$ on the triangle K are also known as **barycentric coordinate functions**.

They provide the nonzero restrictions of tent functions to triangles, see Fig. 88.

$$\lambda_1(\mathbf{x}) = \frac{1}{2|K|} (\mathbf{x} - \mathbf{a}^2) \cdot \begin{pmatrix} a_2^2 - a_2^3 \\ a_1^3 - a_1^2 \end{pmatrix} = -\frac{|e_1|}{2|K|} (\mathbf{x} - \mathbf{a}^2) \cdot \mathbf{n}^1,$$

$$\lambda_2(\mathbf{x}) = \frac{1}{2|K|} (\mathbf{x} - \mathbf{a}^3) \cdot \begin{pmatrix} a_2^3 - a_2^1 \\ a_1^1 - a_1^3 \end{pmatrix} = -\frac{|e_2|}{2|K|} (\mathbf{x} - \mathbf{a}^3) \cdot \mathbf{n}^2,$$

$$\lambda_3(\mathbf{x}) = \frac{1}{2|K|} (\mathbf{x} - \mathbf{a}^1) \cdot \begin{pmatrix} a_2^1 - a_2^2 \\ a_1^2 - a_1^1 \end{pmatrix} = -\frac{|e_3|}{2|K|} (\mathbf{x} - \mathbf{a}^1) \cdot \mathbf{n}^3.$$

(e_i = edge opposite vertex \mathbf{a}^i , see Figure for numbering scheme \triangleright)

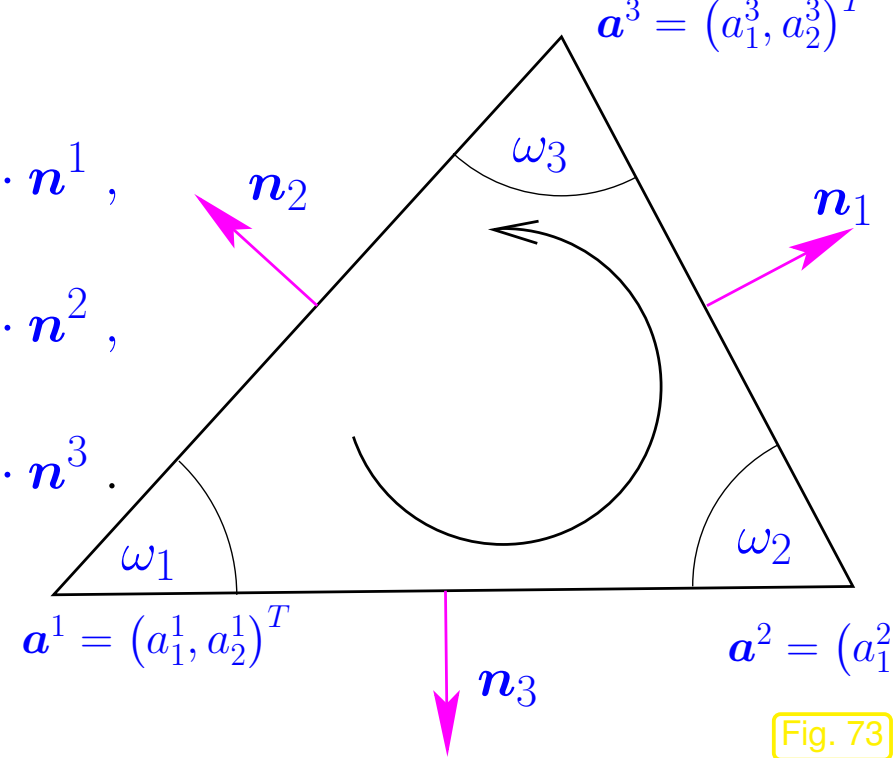


Fig. 73

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

From the distance formula for a point w.r.t to a line given in **Hesse normal form**:

$$(\mathbf{a}^i - \mathbf{a}^j) \cdot \mathbf{n}_i = \text{dist}(\mathbf{a}^i; e_i) = h_i \quad (h_i \hat{=} \text{height}) \quad \text{and} \quad 2|K| = |e_i|h_i \quad \Rightarrow \quad \lambda_i(\mathbf{a}^i) = 1.$$

This shows that the λ_i really provide the restrictions of p.w. linear nodal basis functions (tent functions) of $\mathcal{S}_1^0(\mathcal{M})$ to triangle K , because they are clearly (affine) linear and comply with (3.2.3).

$$\mathbf{grad} \lambda_1 = \frac{1}{2|K|} \begin{pmatrix} a_2^2 - a_2^3 \\ a_1^2 - a_1^3 \end{pmatrix}, \quad \mathbf{grad} \lambda_2 = \frac{1}{2|K|} \begin{pmatrix} a_2^3 - a_2^1 \\ a_1^1 - a_1^3 \end{pmatrix}, \quad \mathbf{grad} \lambda_3 = \frac{1}{2|K|} \begin{pmatrix} a_2^1 - a_2^2 \\ a_1^2 - a_1^1 \end{pmatrix}.$$

$$\left(\int_K \mathbf{grad} \lambda_i \cdot \mathbf{grad} \lambda_j \, d\mathbf{x} \right)_{i,j=1}^3 = \text{element (stiffness) matrix } \mathbf{A}_K = \frac{1}{2} \begin{pmatrix} \cot \omega_3 + \cot \omega_2 & -\cot \omega_3 & -\cot \omega_2 \\ -\cot \omega_3 & \cot \omega_3 + \cot \omega_1 & -\cot \omega_1 \\ -\cot \omega_2 & -\cot \omega_1 & \cot \omega_2 + \cot \omega_1 \end{pmatrix}. \quad (3.2.11)$$

The local numbering and naming conventions are displayed in Fig. 111.

Derivation of (3.2.11), see also [22, Lemma 3.47]: obviously, because the gradients $\mathbf{grad} \lambda_i$ are constant on K ,

$$a(\lambda_i, \lambda_j) = \int_K \mathbf{grad} \lambda_i \cdot \mathbf{grad} \lambda_j \, d\mathbf{x} = \frac{1}{4|K|} |e_i| |e_j| \mathbf{n}_i \cdot \mathbf{n}_j.$$

Then use:

- $\mathbf{n}_i \cdot \mathbf{n}_j = \cos(\pi - \omega_k) = -\cos \omega_k, \quad (i \neq j)$
- $|K| = \frac{1}{2} |e_i| |e_j| \sin \omega_k, \quad (i \neq j).$

Case $i = j$ employs a trick: $\sum_{i=1}^3 \lambda_i = 1 \Rightarrow \sum_{i=1}^3 \mathbf{a}(\lambda_i, \lambda_j) = 0.$

Remark 3.2.12 (Scaling of entries of element matrix for $-\Delta$).

(3.2.11): \mathbf{A}_K does not depend on the “size” of triangle K !
(more precisely, element matrices are equal for *similar* triangles)

This can be seen by the following reasoning:

- Obviously translation and rotation of K does not change. \mathbf{A}_K
- *Scaling* of K by a factor $\rho > 0$ has the following effect that
 - the area $|K|$ is scaled by ρ^2 ,

- the gradients $\mathbf{grad} \lambda_i$ are scaled by ρ^{-1} (the barycentric coordinate functions λ_i become steeper when the triangle shrinks in size.).

Both effects just offset in \mathbf{a}_K from (3.2.10) such that \mathbf{A}_K remains invariant under scaling.



“Assembly” of $(\mathbf{A})_{ij}$ starts from the sum

$$(\mathbf{A})_{ij} = \int_{K_1} \mathbf{grad} b_{N|K_1}^j \cdot \mathbf{grad} b_{N|K_1}^i \, d\mathbf{x} + \int_{K_2} \mathbf{grad} b_{N|K_2}^j \cdot \mathbf{grad} b_{N|K_2}^i \, d\mathbf{x} .$$

- $(\mathbf{A})_{ij}$ can be obtained by summing respective $(*)$ entries of the elements matrices of the elements adjacent to the edge connecting \mathbf{x}^i and \mathbf{x}^j

$(*)$: watch correspondence of local and global vertex numbers !

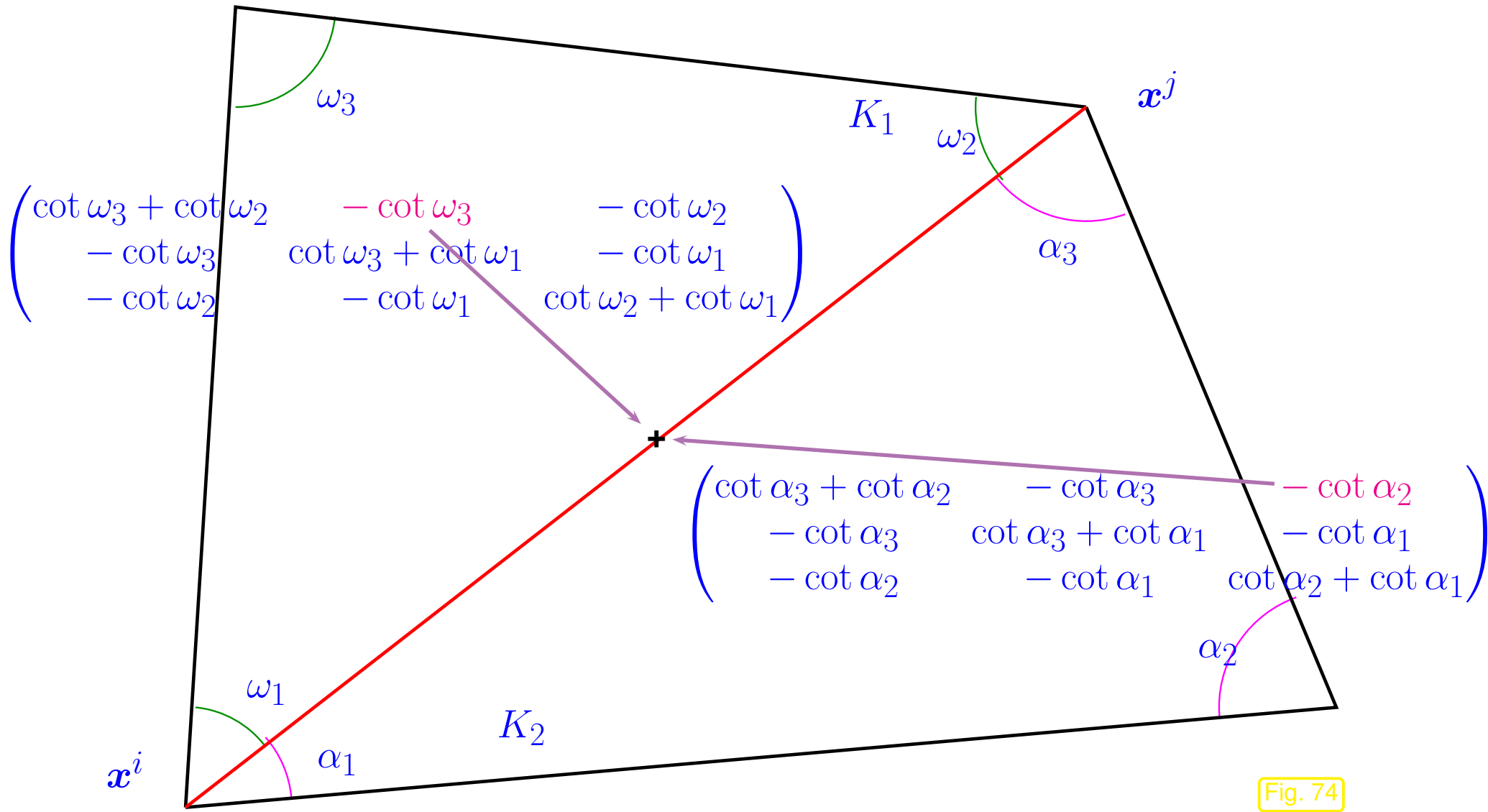


Fig. 74

$(\mathbf{A})_{ij}$ by summing entries of two element matrices

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

“Assembly” of diagonal entry $(\mathbf{A})_{ii}$: summing corresponding diagonal entries of element matrices belonging to triangles adjacent to node x^i .

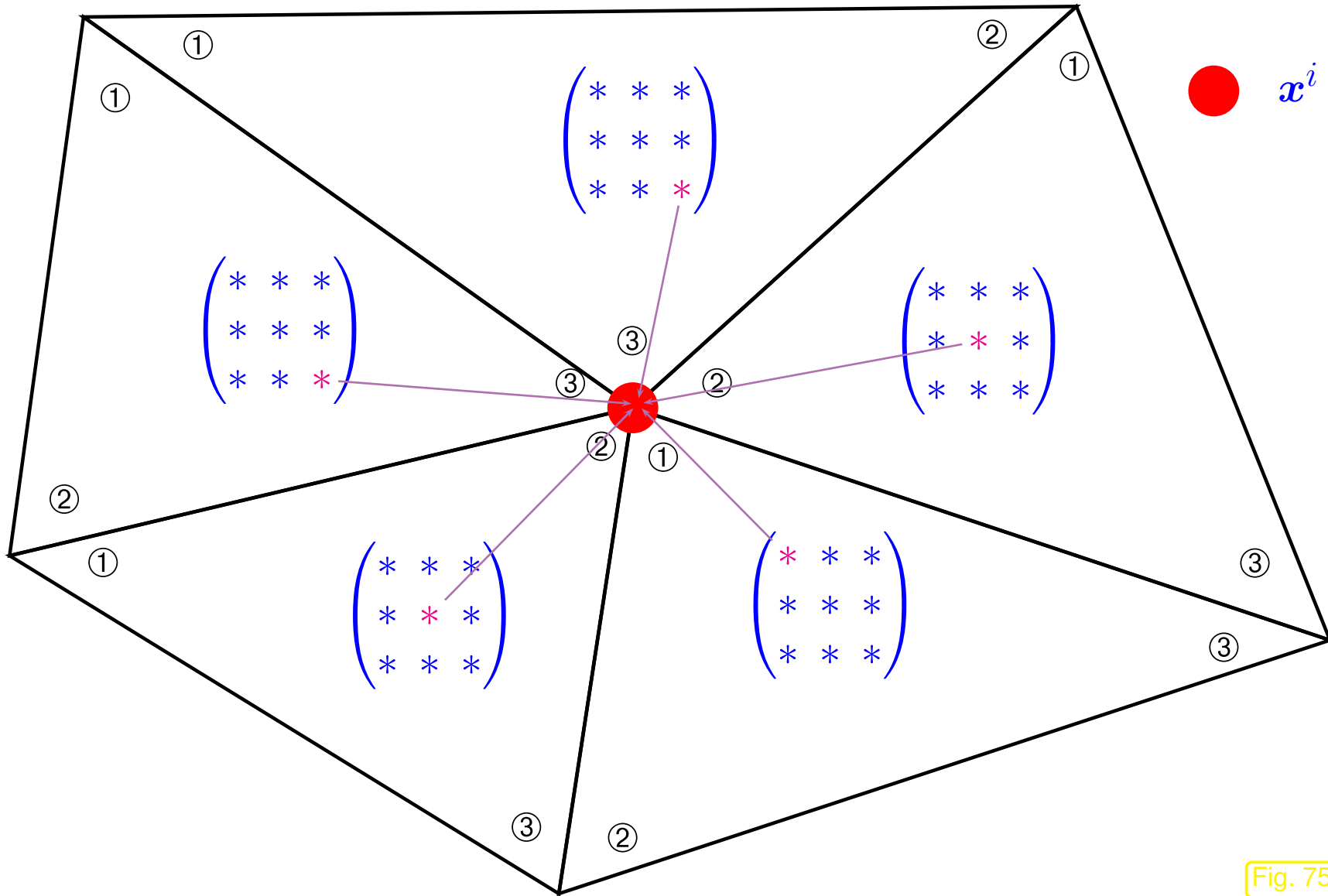


Fig. 75

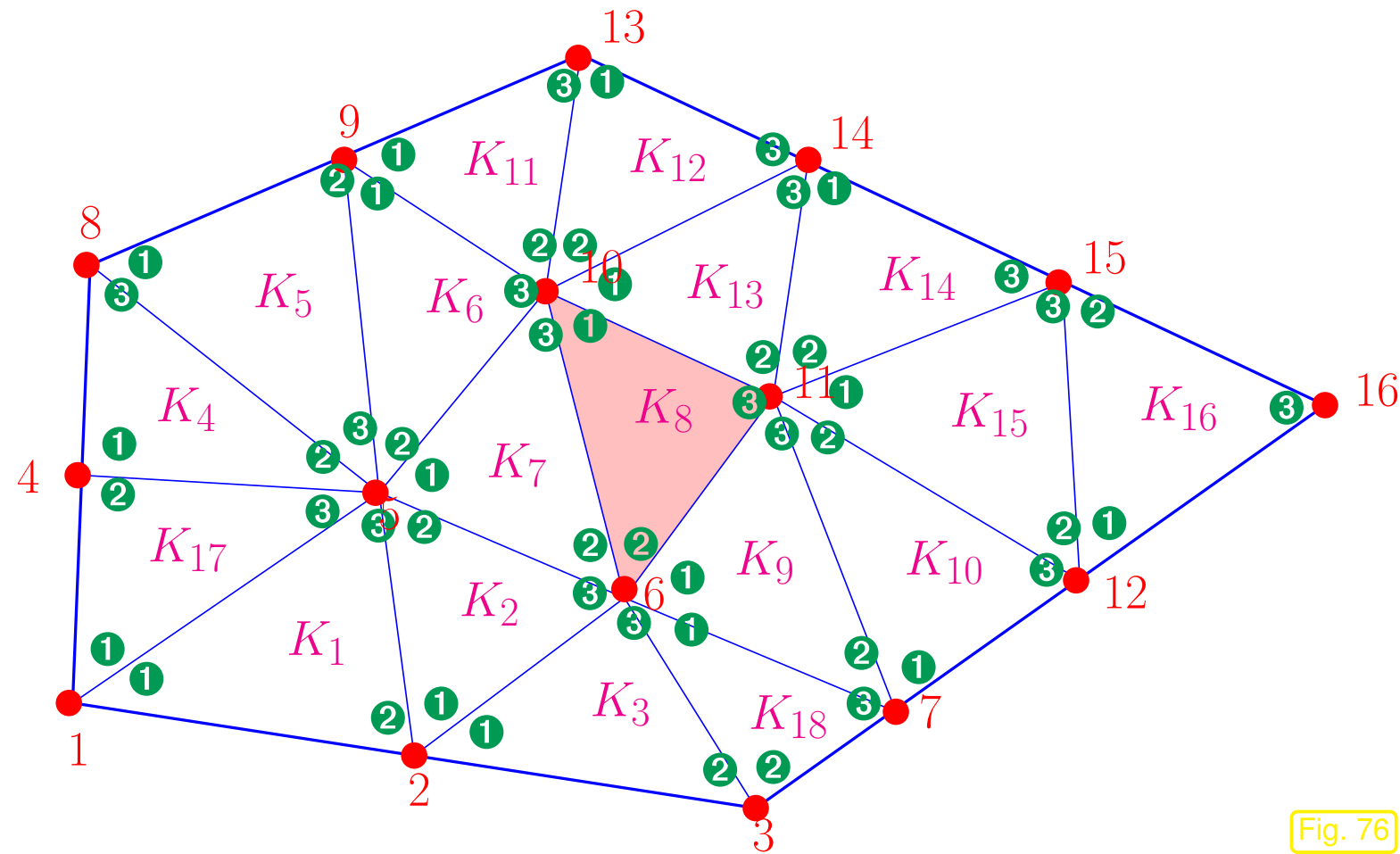
$(A)_{ii}$ by summing diagonal entries of element matrices of adjacent triangles

Remark 3.2.13 (Assembly algorithm for linear Lagrangian finite elements).

- Assume: • numbering of nodal basis functions \leftrightarrow numbering of mesh vertices $\in \mathcal{V}(\mathcal{M})$
- numbering of triangles (cells) of mesh $\mathcal{M} = \{K_1, \dots, K_M\}$, $M := \#\mathcal{M}$

Data structure: $\text{idx} \in \mathbb{N}^{\#\mathcal{M},3}$: local \rightarrow global index mapping array

$$\begin{aligned} \text{idx}(k,l) &= \text{global number of vertex } l \text{ of } k\text{-th cell} \\ \blacktriangle \quad x^{\text{idx}(k,l)} &= a^l \quad \text{when } a^1, a^2, a^3 \text{ are the vertices of } K_k. \end{aligned} \tag{3.2.14}$$



idx:	1	2	5
	2	5	6
	2	3	6
	4	5	8
	8	9	5
	9	5	10
	5	6	10
	10	6	11
	6	7	11
	7	11	12
	9	10	13
	13	10	14
	10	11	14
	14	11	15
	11	12	15
	12	15	16
	1	4	5
	6	3	7

Fig. 76

In Fig. 76, for cell K_8 :

A_K contributes to $A([10 \ 6 \ 11], [10 \ 6 \ 11])$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Code 3.2.15: Assembly of finite element Galerkin matrix for linear finite elements $\mathcal{S}_1^0(\mathcal{M})$

```

1 A = zeros(N,N); % N = #V(M), number of vertices
2 for i=1:M % M = #M, number of cells
3     Ak = getElementMatrix(i); % Compute 3x3 element matrix, see (3.2.11)
4     % idx(i,:) is a vector of global vertex numbers of cell i, see (3.2.14)
5     A(idx(i,:),idx(i,:)) = A(idx(i,:),idx(i,:)) + Ak; %
6 end

```

A note on line 5 of Code 3.2.14: nn MATLAB sub-matrices can be accessed by supplying integer vectors instead of the usual integer indices for matrix elements.

Note:  Homogeneous Dirichlet boundary conditions not taken into account in Code 3.2.14
 Regard Code 3.2.14 as “MATLAB pseudo-code”: in actual implementation \mathbb{A} must be initialized as sparse matrix, see Rem. 3.5.18.

Computational effort = $O(\#\mathcal{M})$

3.2.6 Computation of right hand side vector

We consider the linear form (right hand side of linear variational problem), see (2.3.5), (3.0.1):

$$\ell(v) := \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x} , \quad v \in H^1(\Omega) , \quad f \in L^2(\Omega) .$$

Recall formula for right hand side vector

$$(\vec{\varphi})_j = \ell(b_N^j) = \int_{\Omega} f(\mathbf{x}) b_N^j(\mathbf{x}) \, d\mathbf{x} , \quad j = 1, \dots, N . \quad (3.2.16)$$

Idea: **“Assembly”**

$$(\vec{\varphi})_j = \sum_{l=1}^{N_j} \int_{K_l} f(\mathbf{x}) b_{N|K_l}^j(\mathbf{x}) d\mathbf{x},$$

where $K_1, \dots, K_{N_j} \hat{=} \text{triangles}$ adjacent to node \mathbf{x}^j .

(Integration confined to $\text{supp}(b_N^j)$!)

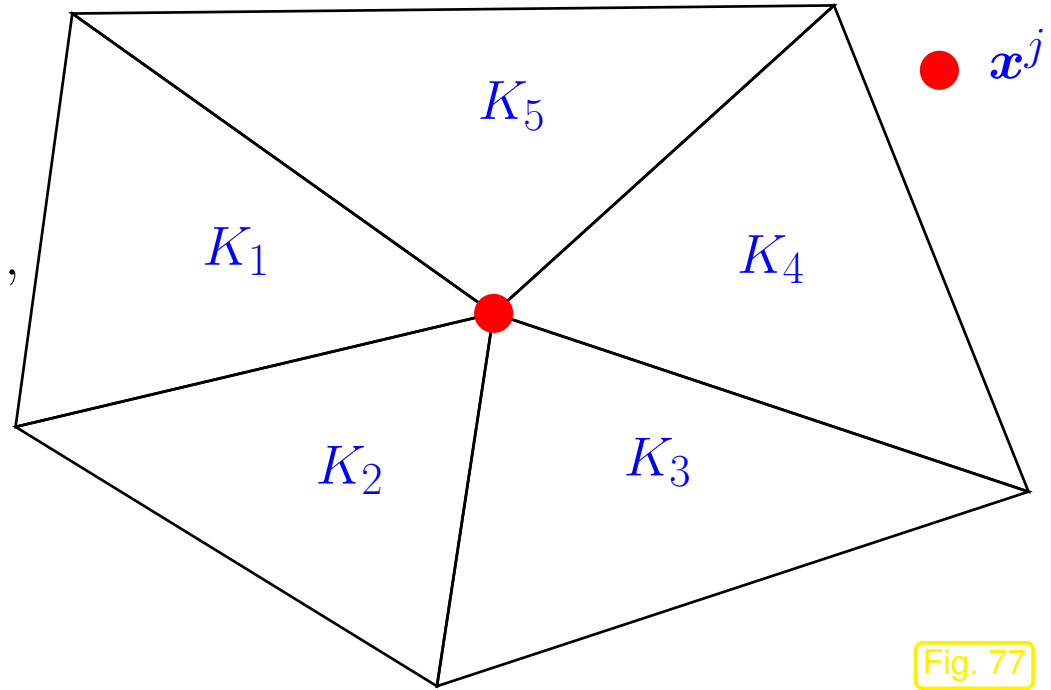


Fig. 77

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

► Zero in on single triangle $K \in \mathcal{M}$:

$$\ell_K(b_N^j) := \int_K f(\mathbf{x}) b_{N|K}^j(\mathbf{x}) d\mathbf{x}, \quad \mathbf{x}^j \text{ vertex of } K. \tag{3.2.17}$$

Rem. 1.5.6: $f : \Omega \mapsto \mathbb{R}$ given in **procedural form**

function `y = f(x)`

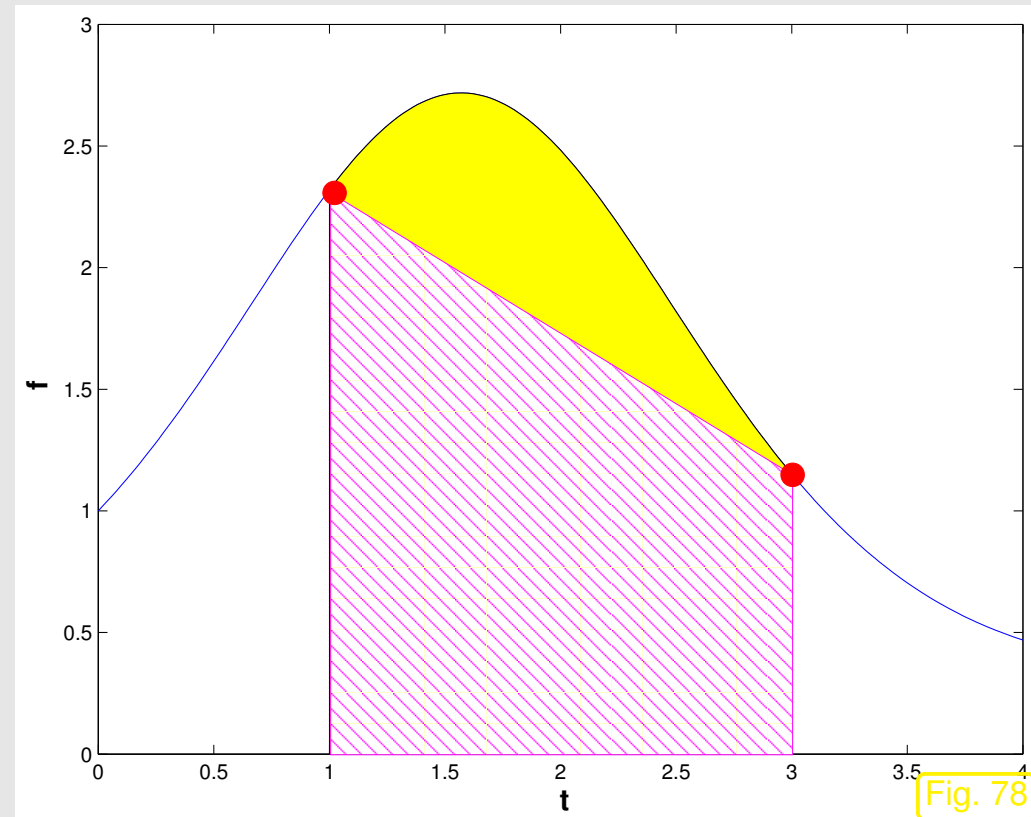
► Mandatory: use of **numerical quadrature** for approximate evaluation of $\ell_K(b_N^j)$, cf. (1.5.85).

1D setting of Sect. 1.5.1.2: use of composite quadrature rules based on low Gauss/Newton-Cotes quadrature formulas on the cells $[x_{j-1}, x_j]$ of the grid, e.g. composite trapezoidal rule (1.5.85).

What is the 2D counterpart of the composite trapezoidal rule ?

Recall:

trapezoidal rule [21, Eq. 10.2.7] integrates linear interpolant of integrand based on endpoint values



for triangle K with vertices $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$

$$\int_K f(\mathbf{x}) \, d\mathbf{x} \approx \frac{|K|}{3} (f(\mathbf{a}^1) + f(\mathbf{a}^2) + f(\mathbf{a}^3)) . \quad (3.2.18)$$

$\hat{=}$ integration of linear interpolant $\sum_{i=1}^3 f(\mathbf{a}^i) \lambda_i$ of f .

► **element (load) vector:**
$$\vec{\varphi}_K := \left(\ell_K(b_N^{j(i)}) \right)_{i=1}^3 = \frac{|K|}{3} \begin{pmatrix} f(\mathbf{a}^1) \\ f(\mathbf{a}^2) \\ f(\mathbf{a}^3) \end{pmatrix} ,$$

where $\mathbf{x}^{j(i)} = \mathbf{a}^i, i = 1, 2, 3$ (global node number \leftrightarrow local vertex number).

As above in Fig. 75: “Assembly” of $(\vec{\varphi})_j$ by summing up contributions from element vectors of triangles adjacent to \mathbf{x}^j .

$$(\vec{\varphi})_j = \sum_{l=1}^{N_j} \ell_{K_l}(b_N^j|_{K_l}) = \sum_{l=1}^{N_j} (\vec{\varphi}_K)_{i(l,j)} = f(\mathbf{x}^j) \cdot \frac{1}{3} \sum_{l=1}^{N_j} |K_l| , \quad (3.2.19)$$

where $i(l, j)$ is the **local** vertex index of the node \mathbf{x}^j (**global** index j) in the triangle K_l .

Note: with index array idx from Rem. 3.2.13: $idx(l, i(l, j)) = j$

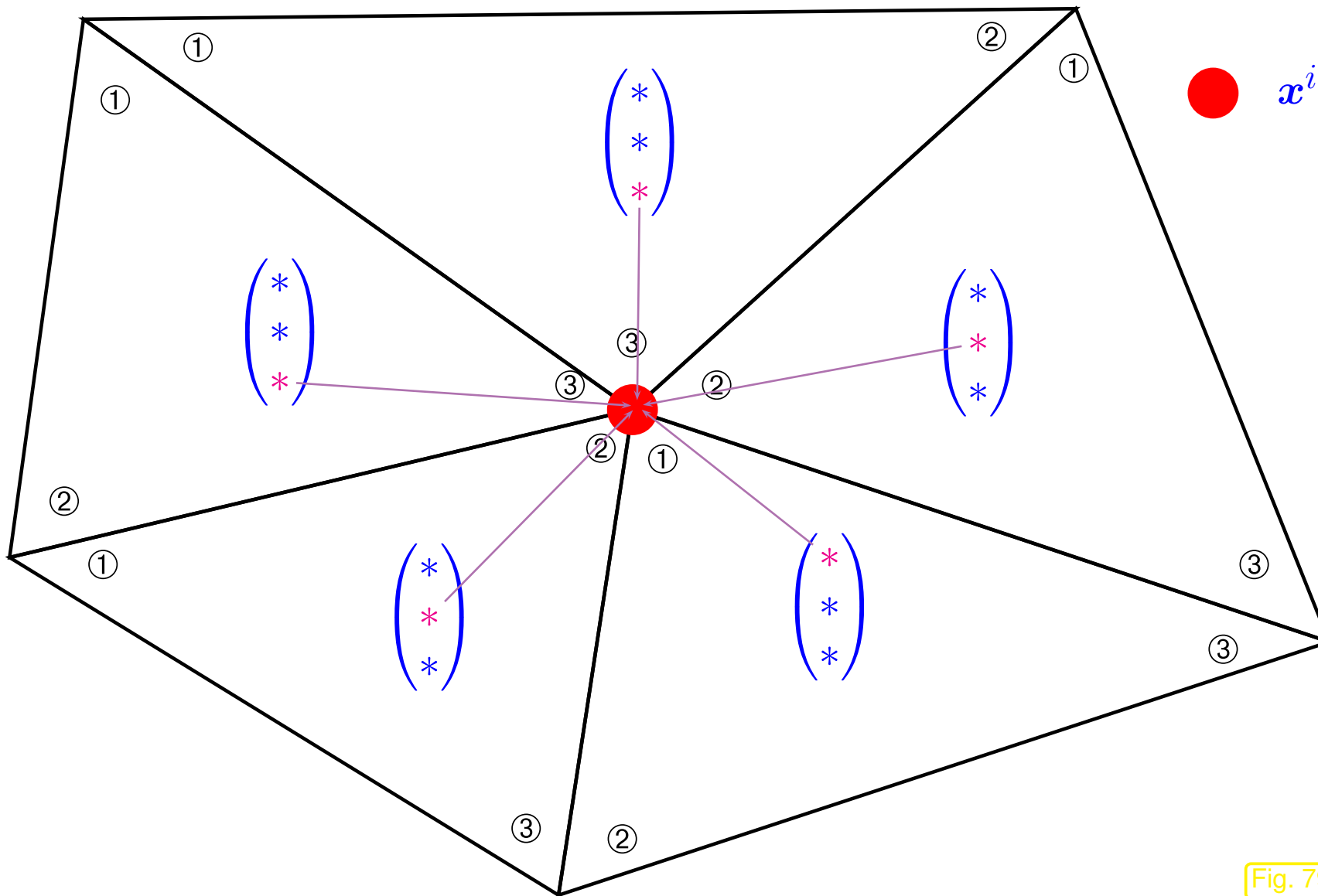


Fig. 79

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 3.2.20 (Assembly of right hand side vector for linear finite elements). \rightarrow Rem. 3.2.13

Code 3.2.21: Assembly of right hand side vector for linear finite elements, see (3.2.19)

```
1 phi = zeros(N,1); % N = #V(M), number of vertices
2 for i=1:M % M = #M, number of cells
3     phiK = getElementVector(i); % Compute element (load) vector  $\in \mathbb{R}^3$ 
4     % update sub-vector corresponding to the vertices of current cell,
5     phi(idx(i,:)) = phi(idx(i,:)) + phiK; % idx according to (3.2.14)
6 end
```



3.3 Building blocks of general FEM

The previous section explored the details of a simple finite element discretization of 2nd-order elliptic variational problems. Yet, it already introduced *key features and components* that distinguish the finite element approach to the discretization of linear boundary value problems for partial differential equations:

- a focus on the **variational formulation** of a boundary value problem \rightarrow Sect. 2.8,
- a partitioning of the computational domain Ω by means of a **mesh** \mathcal{M} (\rightarrow Sect. 3.2.1)
- the use of Galerkin trial and test spaces based on **piecewise polynomials** w.r.t. \mathcal{M} (\rightarrow Sect. 3.2.2),
- the use of **locally supported** basis functions for the assembly of the resulting linear system of equations (\rightarrow Sect. 3.2.3).

In this section a more abstract point of view is adopted and the components of a finite element method for scalar 2nd-order elliptic boundary value problems will be discussed in greater generality. However, prior perusal of Sect. 3.2 is strongly recommended.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

3.3.1 Meshes

First main ingredient of FEM: **triangulation/mesh** of Ω \rightarrow Sect. 3.2.1

Definition 3.3.1. A *mesh* (or *triangulation*) of $\Omega \subset \mathbb{R}^d$ is a finite collection $\{K_i\}_{i=1}^M$, $M \in \mathbb{N}$, of open non-degenerate (curvilinear) polygons ($d = 2$)/polyhedra ($d = 3$) such that

(A) $\bar{\Omega} = \bigcup \{\bar{K}_i, i = 1, \dots, M\}$,

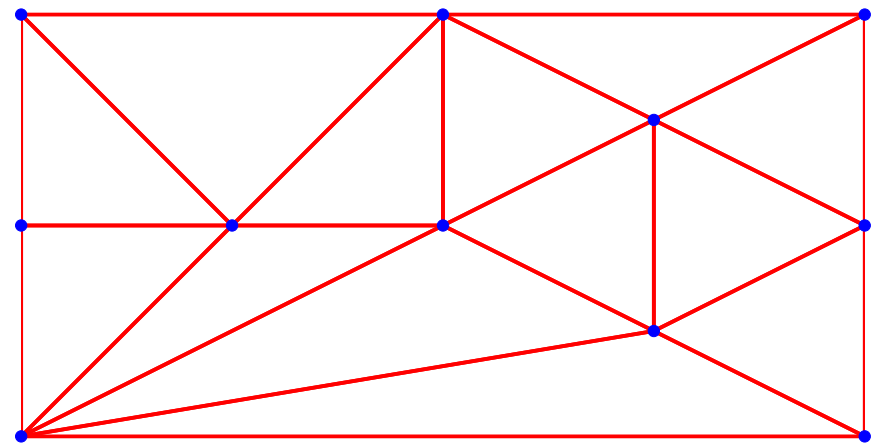
(B) $K_i \cap K_j = \emptyset \Leftrightarrow i \neq j$,

(C) for all $i, j \in \{1, \dots, M\}$, $i \neq j$, the intersection $\bar{K}_i \cap \bar{K}_j$ is either empty or a vertex, edge, or face of both K_i and K_j .

► “vertex”, “edge”, “face” of polygon/polyhedron: \rightarrow geometric intuition

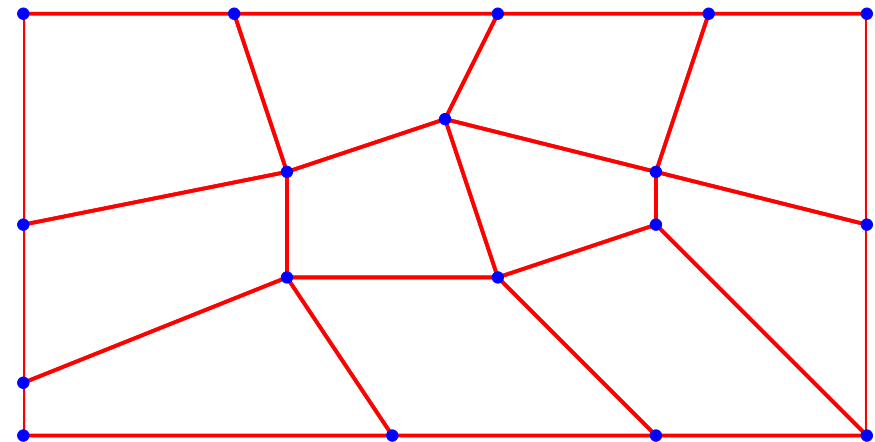
Terminology: Given mesh $\mathcal{M} := \{K_i\}_{i=1}^M$: K_i called **cell** or **element**.
Vertices of a mesh \rightarrow **nodes** (set $\mathcal{V}(\mathcal{M})$)

Types of meshes:



Triangular mesh in 2D

Fig. 80



Quadrilateral mesh in 2D

Fig. 81

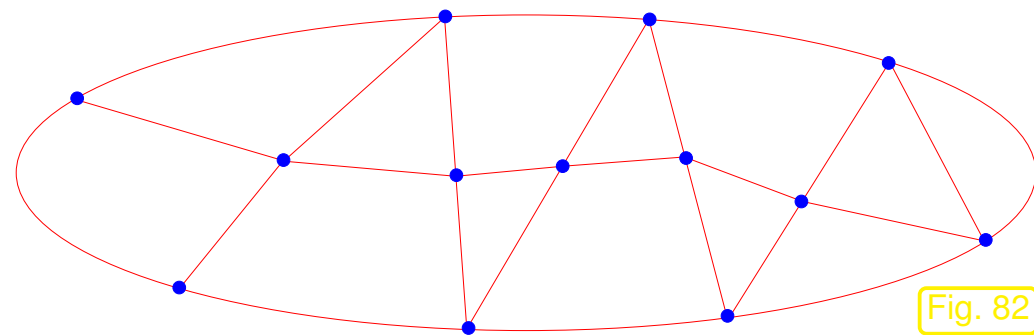
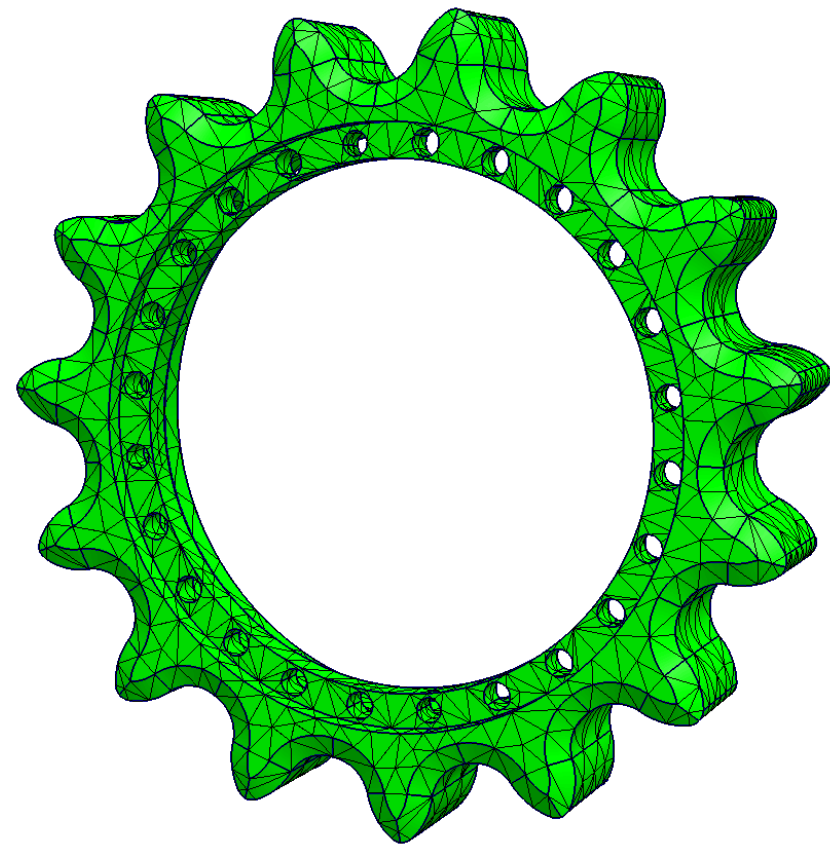
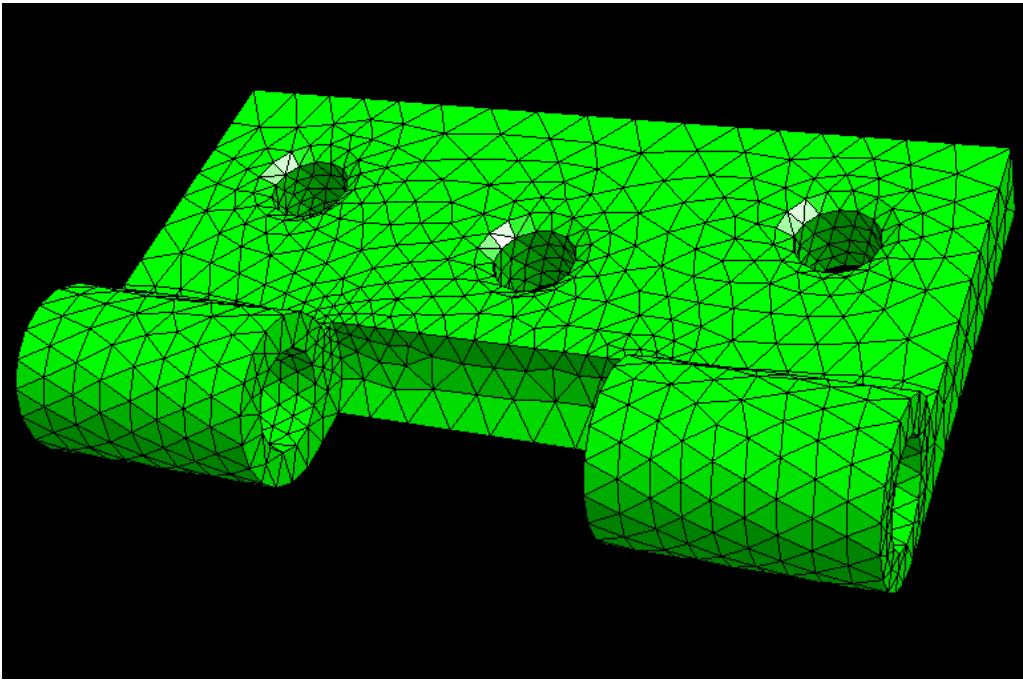


Fig. 82

◁ 2D **hybrid** mesh comprising

- triangles
- quadrilaterals
- curvilinear cells (at $\partial\Omega$)

(Curved) tetrahedral meshes in 3D (created with NETGEN):



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Tensor product mesh = **grid**



in 2D: $a = x_0 < x_1 < \dots < x_n = b$,
 $c = y_0 < y_1 < \dots < y_m = d$.

▶ $\mathcal{M} = \{]x_{i-1}, x_i[\times]y_{j-1}, y_j[: \quad (3.3.2)$
 $1 \leq i \leq n, 1 \leq j \leq m \}$.

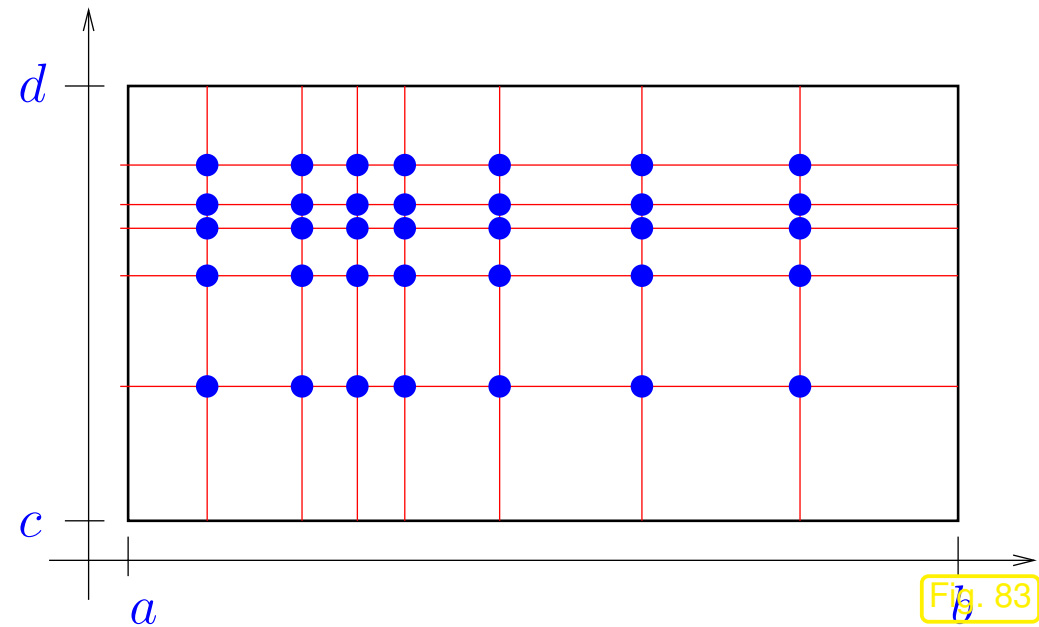


Fig. 83

If (C) does not hold

▶ Triangular **non-conforming** mesh
(with **hanging nodes**)

$\bar{K}_i \cap \bar{K}_j$ is only part of an edge/face for at most one of the adjacent cells.

(However, conforming if degenerate quadrilaterals admitted)

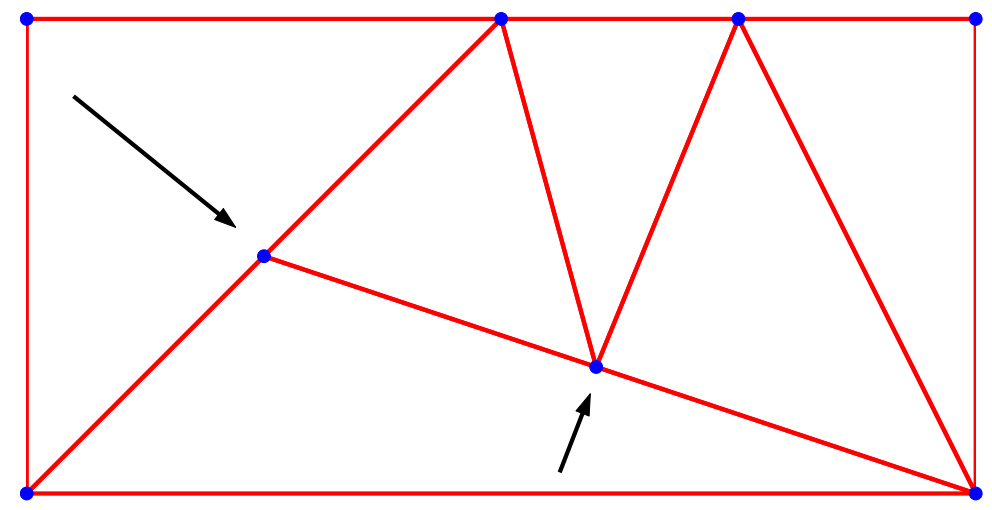


Fig. 84

Terminology:

Simplicial mesh

=

triangular mesh in 2D
tetrahedral mesh in 3D

3.3.2 Polynomials

Second main ingredient of FEM:

In FEM: Galerkin trial/test space comprise *locally polynomial* functions on Ω

Clear: polynomials of **degree** $\leq p$, $p \in \mathbb{N}_0$, in 1D (**univariate** polynomials), see (1.5.27)

$$\mathcal{P}_p(\mathbb{R}) := \{x \mapsto c_0 + c_1x + c_2x^2 + \dots + c_px^p\} .$$

In higher dimensions this concept allows various generalizations, one given in the following definition, one given in Def. 3.3.7.

Definition 3.3.3 (Multivariate polynomials).

Space of ***multivariate*** (*d-variate*) **polynomials** of (total) **degree** $p \in \mathbb{N}_0$:

$$\mathcal{P}_p(\mathbb{R}^d) := \{\mathbf{x} \in \mathbb{R}^d \mapsto \sum_{\alpha \in \mathbb{N}_0^d, |\alpha| \leq p} c_\alpha \mathbf{x}^\alpha, c_\alpha \in \mathbb{R}\} .$$

Def. 3.3.3 relies on **multi-index notation**:

$$\alpha = (\alpha_1, \dots, \alpha_d): \quad \mathbf{x}^\alpha := x_1^{\alpha_1} \cdot \dots \cdot x_d^{\alpha_d}, \quad (3.3.4)$$

$$|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \cdots + \alpha_d .$$

(3.3.5)

Special case:

$$d = 2: \mathcal{P}_p(\mathbb{R}^2) = \left\{ \sum_{\substack{\alpha_1, \alpha_2 \geq 0 \\ \alpha_1 + \alpha_2 \leq p}} c_{\alpha_1, \alpha_2} x_1^{\alpha_1} x_2^{\alpha_2}, c_{\alpha_1, \alpha_2} \in \mathbb{R} \right\} .$$

Example:

$$\mathcal{P}_2(\mathbb{R}^2) = \text{Span} \left\{ 1, x_1, x_2, x_1^2, x_2^2, x_1 x_2 \right\}$$

Lemma 3.3.6 (Dimension of spaces of polynomials).

$$\dim \mathcal{P}_p(\mathbb{R}^d) = \binom{d+p}{p} \text{ for all } p \in \mathbb{N}_0, d \in \mathbb{N}$$

Proof. Distribute p “powers” to the d independent variables or discard them $\triangleright d + 1$ bins.

Combinatorial model: number of different linear arrangements of p identical items and d separators

$$= \binom{d+p}{p} .$$

□

$$\dim \mathcal{P}_p(\mathbb{R}^d) = O(p^d)$$

Definition 3.3.7 (Tensor product polynomials).

Space of *tensor product polynomials* of degree $p \in \mathbb{N}$ in each coordinate direction

$$\mathcal{Q}_p(\mathbb{R}^d) := \{ \mathbf{x} \mapsto p_1(x_1) \cdots p_d(x_d), p_i \in \mathcal{P}_p(\mathbb{R}), i = 1, \dots, d \} .$$

Example:

$$\mathcal{Q}_2(\mathbb{R}^2) = \text{Span} \left\{ 1, x_1, x_2, x_1x_2, x_1^2, x_1^2x_2, x_1^2x_2^2, x_1x_2^2, x_2^2 \right\}$$

Lemma 3.3.8 (Dimension of spaces of tensor product polynomials).

$$\dim \mathcal{Q}_p(\mathbb{R}^d) = (p+1)^d \quad \text{for all } p \in \mathbb{N}_0, d \in \mathbb{N}$$

Terminology: $\mathcal{P}_p(\mathbb{R}^d)/\mathcal{Q}_p(\mathbb{R}^d) = \text{complete spaces of polynomials/tensor product polynomials}$

3.3.3 Basis functions

Third main ingredient of FEM: **locally supported** basis functions
(see Sect. 3.1 for role of bases in Galerkin discretization)

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Basis functions b_N^1, \dots, b_N^N for a finite element trial/test space $V_{0,N}$ built on a mesh \mathcal{M} satisfy:

- (a) $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ is basis of $V_{0,N}$ $\Rightarrow N = \dim V_{0,N}$,
- (b) each b_N^i is **associated** with a single cell/edge/face/vertex of \mathcal{M} ,
- (c) $\text{supp}(b_N^i) = \bigcup \{\bar{K} : K \in \mathcal{M}, \mathbf{p} \subset \bar{K}\}$, if b_N^i associated with cell/edge/face/vertex \mathbf{p} .

Finite element terminology: $b_N^i =$ global shape functions/global basis functions

Mesh \mathcal{M} + global shape functions \rightarrow complete description of finite element space

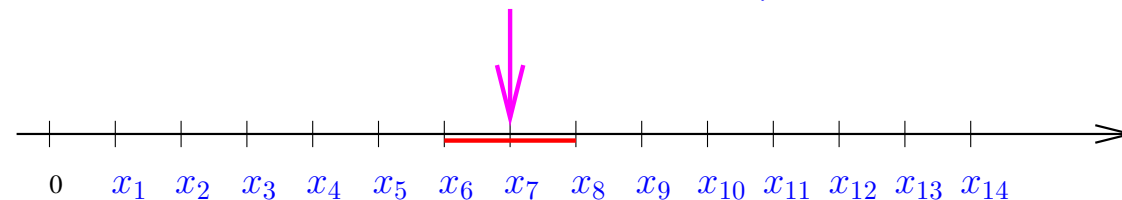
Example 3.3.9 (Supports of global shape functions in 1D). \rightarrow Sect. 1.5.1.2

- $\Omega =]a, b[\hat{=}$ interval
- Equidistant mesh

Support (\rightarrow Def. 1.5.83) of global shape function associated with x_7

$$\mathcal{M} := \{]x_{j-1}, x_j[, j = 1, \dots, M \},$$

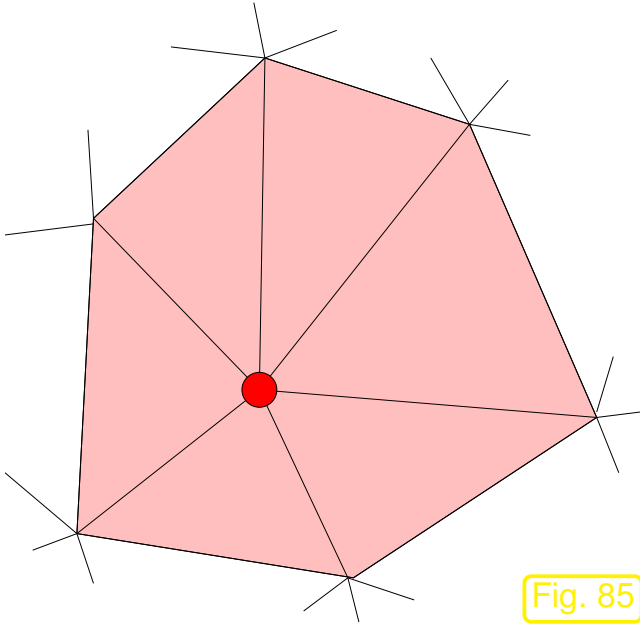
$$x_j := a + hj, h := (b - a)/M, M \in \mathbb{N}.$$



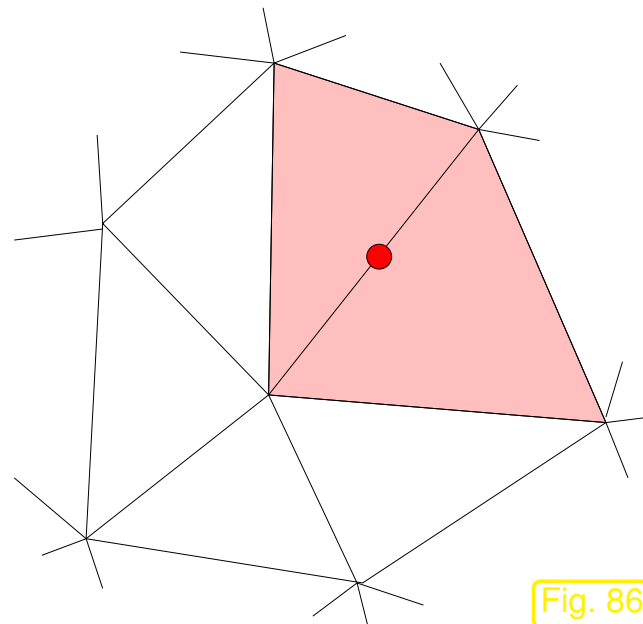
R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

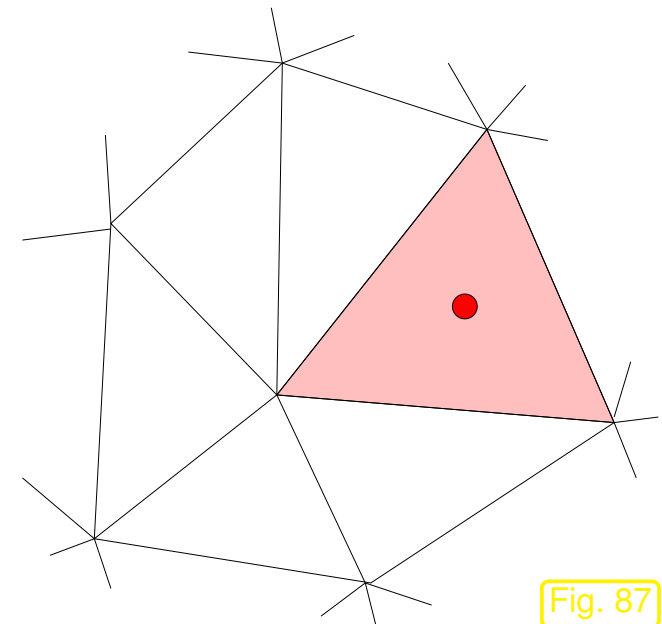
Example 3.3.10 (Supports of global shape functions on triangular mesh).



Support of node-associated basis function, *cf.* Fig. 88



Support of edge-associated basis function



Support of cell-associated basis function

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs



SAM, ETHZ

Requirement **(c)** implies that

global finite element basis functions are **locally supported**.

What is the rationale for this requirement ?

Consider a generic bilinear form \mathbf{a} arising from a linear scalar 2nd-order elliptic BVP, see (3.2.6): it involves integration over $\Omega/\partial\Omega$ of products of (derivatives of) basis functions. Thus the integrand for $\mathbf{a}(b_N^j, b_N^i)$ vanishes outside the overlap of the supports of b_N^j and b_N^i .

► Galerkin matrix $\mathbf{A} \in \mathbb{R}^{N,N}$ with $(\mathbf{A})_{ij} := \mathbf{a}(b_N^j, b_N^i)$, $i, j = 1, \dots, N$ satisfies

$a_{ij} \neq 0$ only if b_N^i and b_N^j associated with
vertices/faces/edges(cells) adjacent to common
cell



Finite element stiffness matrices are **sparse** (\rightarrow Notion 3.2.8)

Global shape functions $\xrightarrow{\text{Restriction to element}}$ local shape functions (3.3.11)

Definition 3.3.12 (Local shape functions).

Given finite element function space on mesh \mathcal{M} with global shape functions $b_N^i, i = 1, \dots, N$:

$$\{b_N^j|_K, K \subset \text{supp}(b_N^j)\} = \text{set of local shape functions on } K \in \mathcal{M}.$$

Local shape functions $b_K^1, \dots, b_K^Q, Q = Q(K) \in \mathbb{N}$ also associated with vertices/edges/faces/interior of K

Example 3.3.13 (Local shape functions for $\mathcal{S}_1^0(\mathcal{M})$ in 2D). \rightarrow Sect. 3.2.3

Global basis function for $\mathcal{S}_1^0(\mathcal{M})$

On “unit triangle” K with vertices

$$\mathbf{a}^1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{a}^2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{a}^3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Local shape functions:

$$b_K^1(\mathbf{x}) = 1 - x_1 - x_2,$$

$$b_K^2(\mathbf{x}) = x_1,$$

$$b_K^3(\mathbf{x}) = x_2.$$

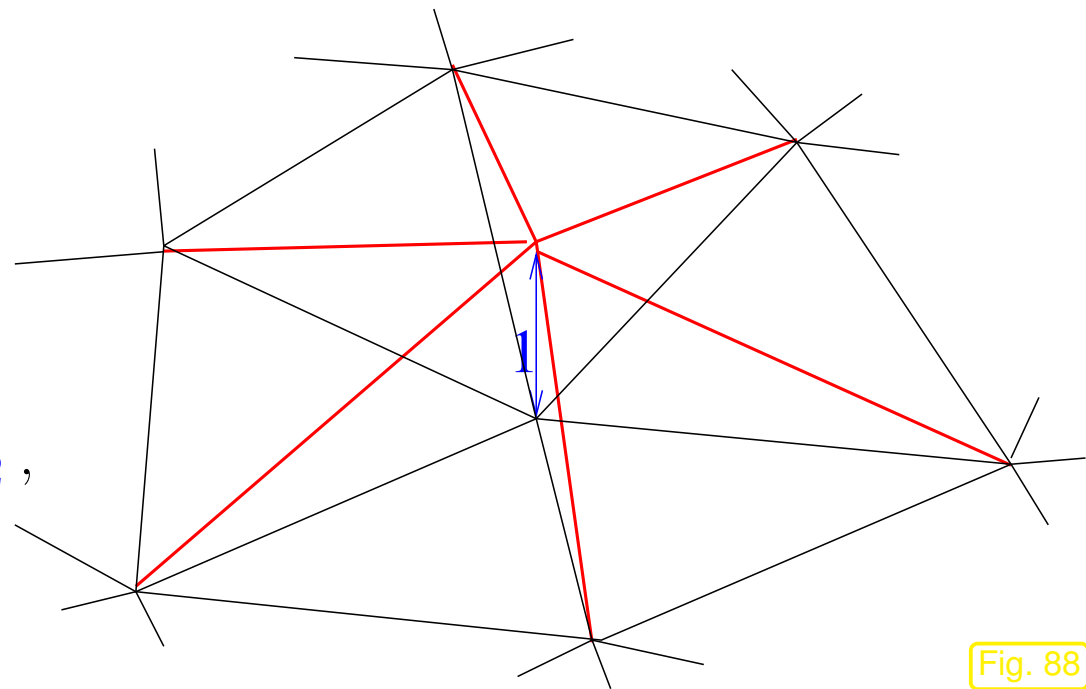


Fig. 88

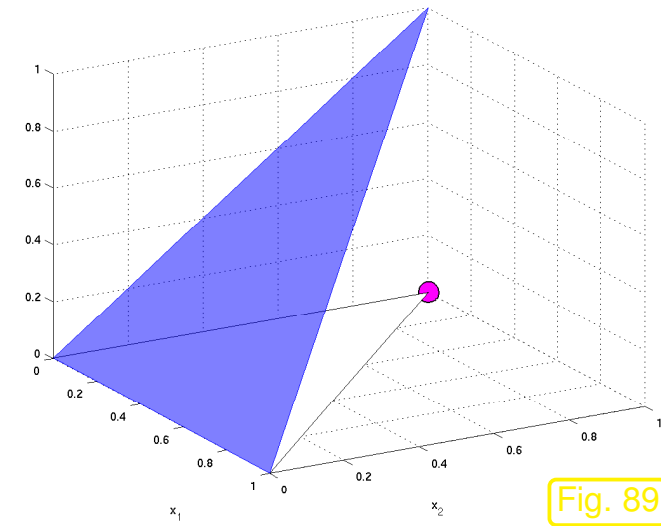
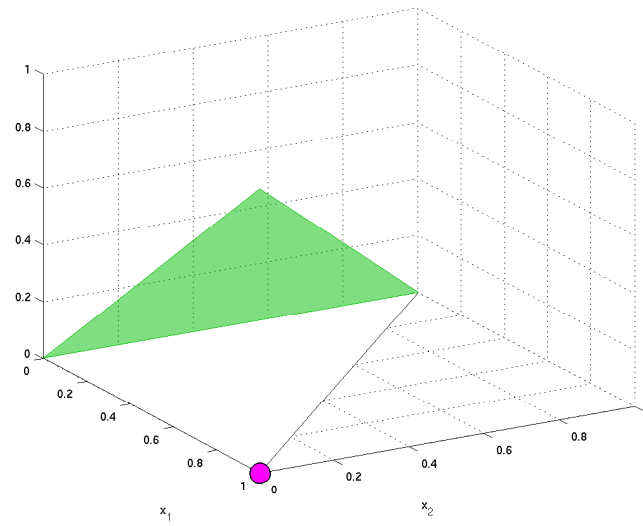
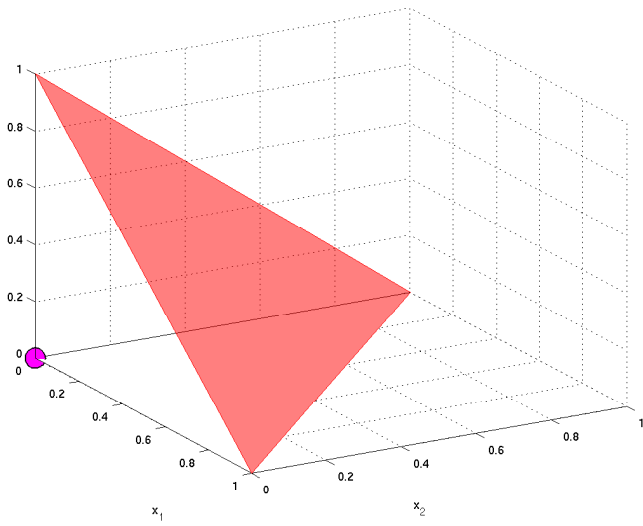


Fig. 89

These are the barycentric coordinate functions λ_1 , λ_2 , λ_3 introduced in Sect. 3.2.5



3.4 Lagrangian FEM

Taken for granted: finite element mesh \mathcal{M} according to Def. 3.3.1.

Goal: construction of finite element spaces and global shape functions of higher polynomials degrees.

Lagrangian finite element spaces provide spaces $V_{0,N}$ of \mathcal{M} -piecewise polynomials that fulfill

$$V_{N,0} \subset C^0(\Omega) \xrightarrow{\text{Thm. 2.2.26}} \boxed{V_{N,0} \subset H^1(\Omega)} .$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Notation:

(Lagrangian FE spaces)

$$S_p^0(\mathcal{M}) \begin{cases} \text{continuous functions, cf. } C^0(\Omega) \\ \text{locally polynomials of degree } p, \text{ e.g. } \mathcal{P}_p(\mathbb{R}^d) \end{cases}$$

3.4.1 Simplicial Lagrangian FEM

\mathcal{M} : Simplicial mesh, consisting of triangles in 2D, tetrahedra in 3D.

Now we generalize $\mathcal{S}_1^0(\mathcal{M})/\mathcal{S}_{1,0}^0(\mathcal{M})$ from Sect. 3.2 to higher polynomial degree $p \in \mathbb{N}_0$.

Definition 3.4.1 (Simplicial Lagrangian finite element spaces).

Space of *p -th degree Lagrangian finite element functions* on simplicial mesh \mathcal{M}

$$\mathcal{S}_p^0(\mathcal{M}) := \{v \in C^0(\bar{\Omega}) : v|_K \in \mathcal{P}_p(K) \quad \forall K \in \mathcal{M}\} .$$

Def. 3.4.1 merely describes the space of trial/test functions used in a Lagrangian finite element method on a Simplicial mesh. A crucial ingredient is still missing (\rightarrow Sect. 3.3.3): the global shape functions still need to be specified. This is done by generalizing (3.2.3) based on sets of special *interpolation nodes*.

Example 3.4.2 (Triangular quadratic Lagrangian finite elements).

interpolation nodes

$$\mathcal{N} := \mathcal{V}(\mathcal{M}) \cup \{\text{midpoints of edges}\},$$

$$\mathcal{N} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}.$$

Nodal basis functions b_N^j , $j = 1, \dots, N$ defined by, cf. (3.2.3)

$$b_N^j(\mathbf{p}_i) = \begin{cases} 1 & , \text{ if } i = j, \\ 0 & \text{ else.} \end{cases} \quad (3.4.3)$$

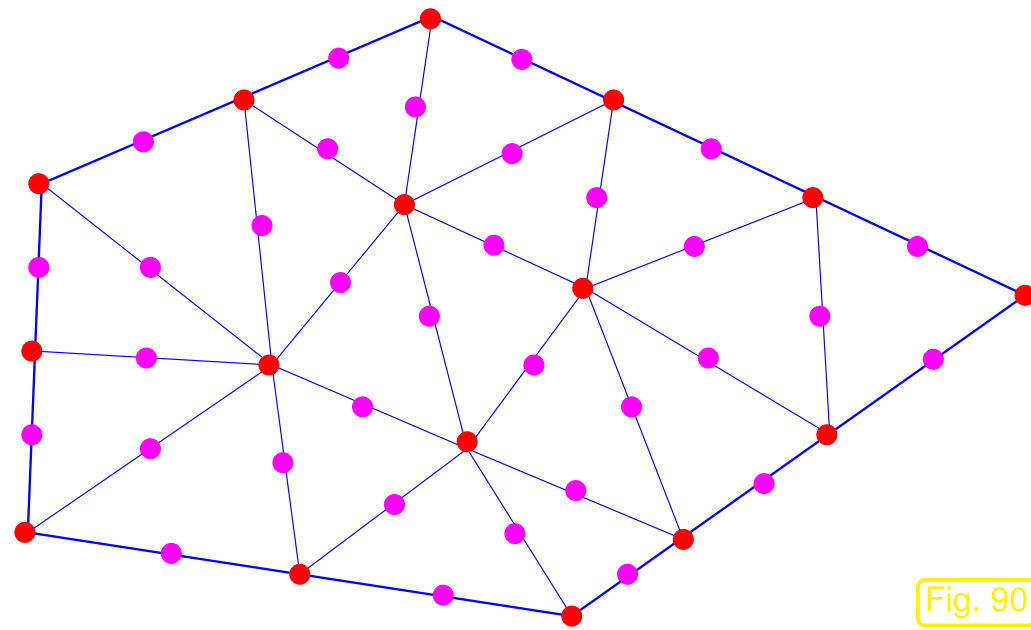


Fig. 90

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

A “definition” like (3.4.3) is cheap, but it may be pointless, in case no such functions b_N^j exist. To establish their existence, we first study the case of a single triangle K .

We have to show that there is a basis of $\mathcal{P}_2(\mathbb{R}^2)$ that satisfies (3.4.3) in the case of a mesh consisting of a single triangle $\mathcal{M} = \{K\}$.

A first simple consistency check: does the number of interpolation nodes $\#\mathcal{N}$ for $\mathcal{M} = \{K\}$ agree with $\dim \mathcal{P}_2(\mathbb{R}^2) = 6$? Yes, it does!

Local shape functions (barycentric coordinate representation)

$$\begin{aligned}
 b_K^1 &= (2\lambda_1 - 1)\lambda_1, \\
 b_K^2 &= (2\lambda_2 - 1)\lambda_2, \\
 b_K^3 &= (2\lambda_3 - 1)\lambda_3, \\
 b_K^4 &= 4\lambda_1\lambda_2, \\
 b_K^5 &= 4\lambda_2\lambda_3, \\
 b_K^6 &= 4\lambda_1\lambda_3.
 \end{aligned}
 \tag{3.4.4}$$

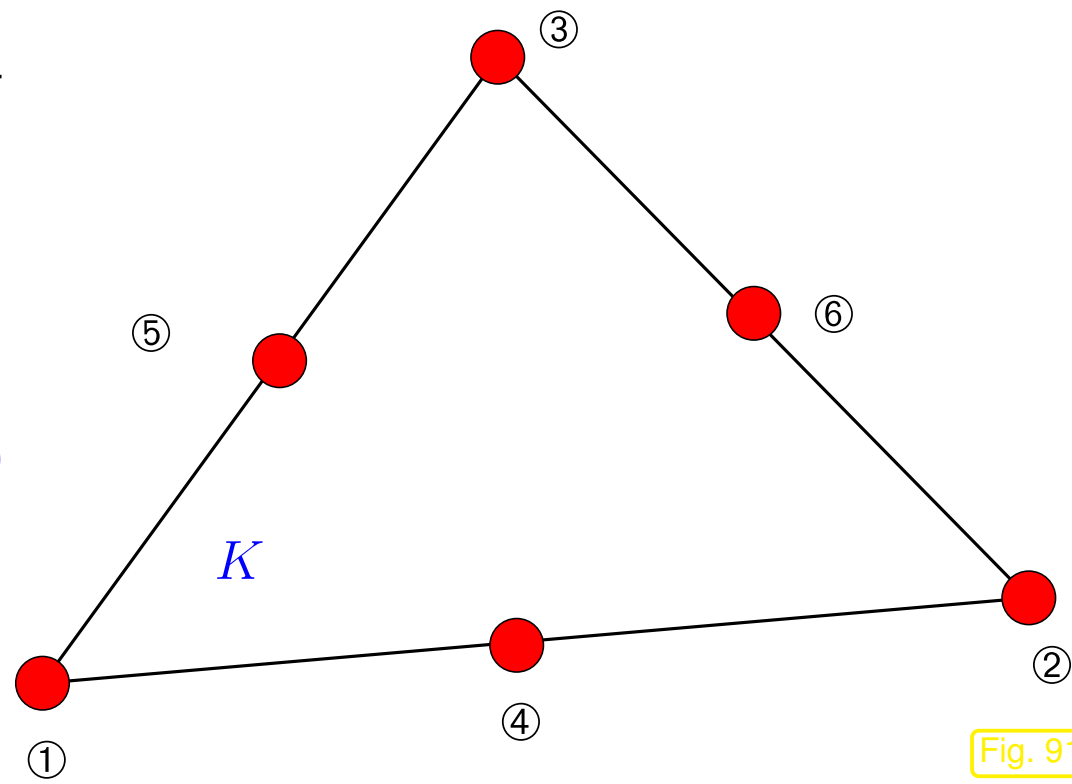


Fig. 91

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

To see the validity of the formulas (3.4.4), note that

- $\lambda_i(\mathbf{a}^i) = 1$ and $\lambda_i(\mathbf{a}^j) = 0$, if $i \neq j$, where $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$ are the vertices of the triangle K ,
- $\lambda_1(\mathbf{m}^{12}) = \lambda_1(\mathbf{m}^{13}) = \frac{1}{2}$, where $\mathbf{m}^{ij} = \frac{1}{2}(\mathbf{a}^i + \mathbf{a}^j)$ denotes the midpoint of the edge connecting \mathbf{a}^i and \mathbf{a}^j ,
- each barycentric coordinate function λ_i is affine linear such that $\lambda_i\lambda_j \in \mathcal{P}_2(\mathbb{R}^2)$.

Selected local shape functions:

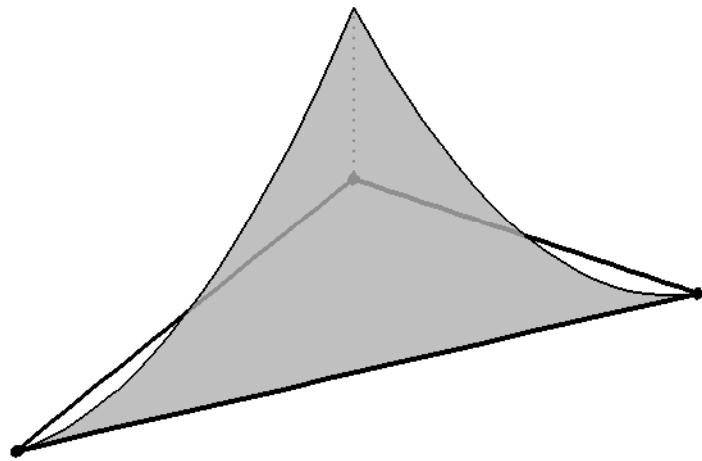
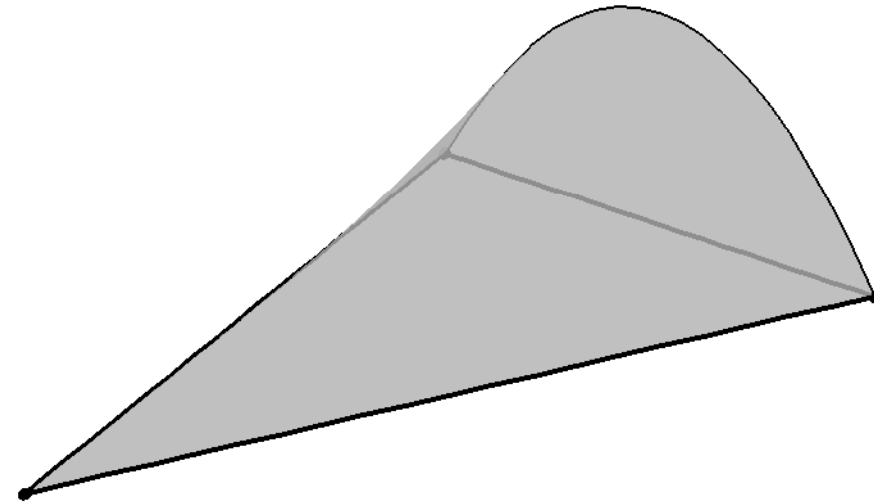
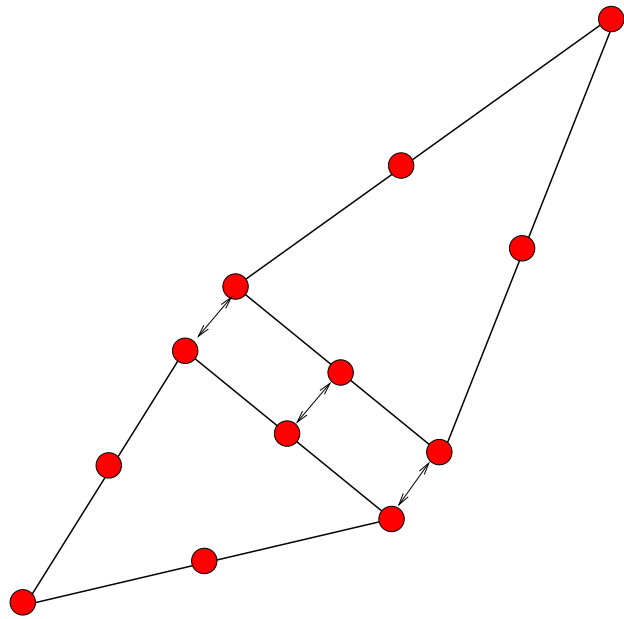


Fig. 92



So far we have seen that *local shape functions* can be found that satisfy (3.4.3).

Issue: can the local shape functions from (3.4.4) be “stitched together” across interelement edges such that they yield a *continuous* global basis function? (Remember that Thm. 2.2.26 demands global continuity in order to obtain a subspace of $H^1(\Omega)$.)



The restriction of a quadratic polynomial to an edge is an *univariate* quadratic polynomial.

Fixing its value in three points, the midpoint of the edge and the endpoints, *uniquely* fixes this polynomial.

The local shape functions associated with the same interpolation node “from left and right” agree on the edge.

➤ continuity !

Fig. 93

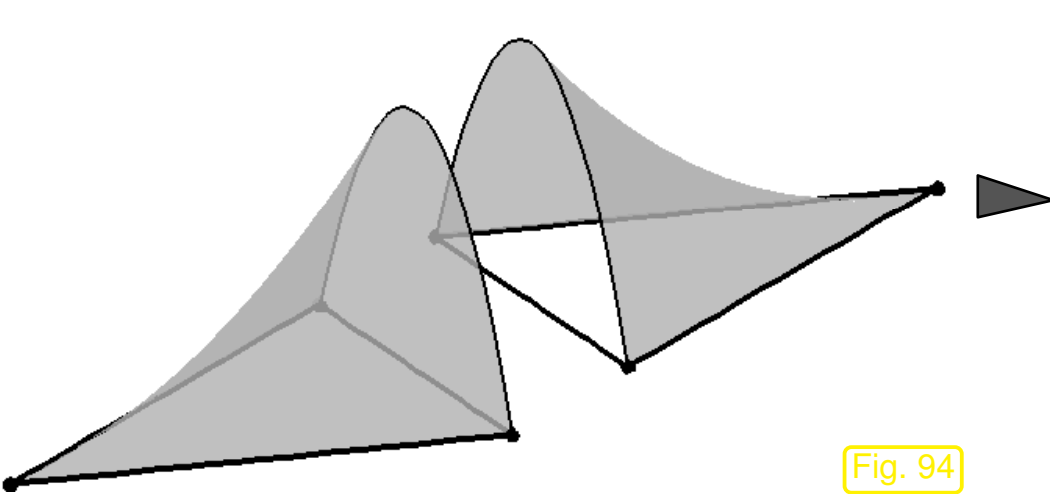


Fig. 94

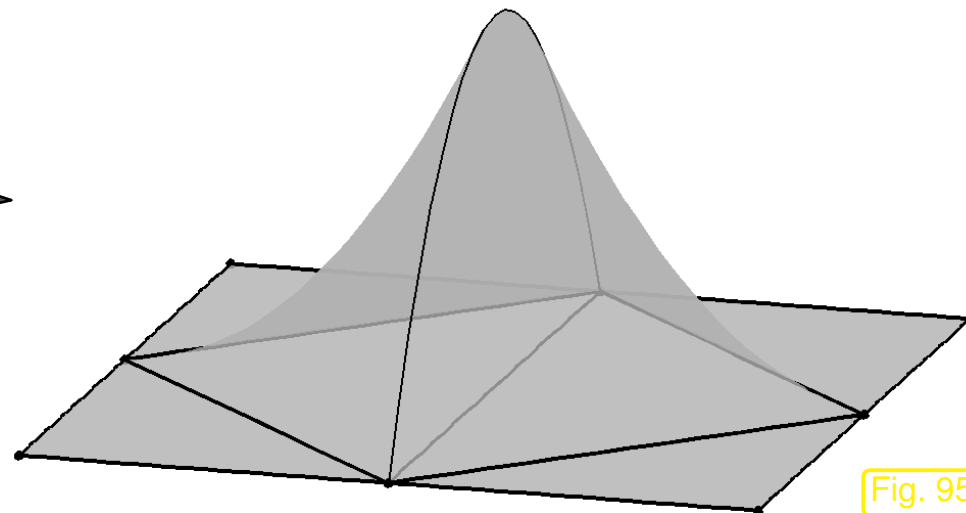
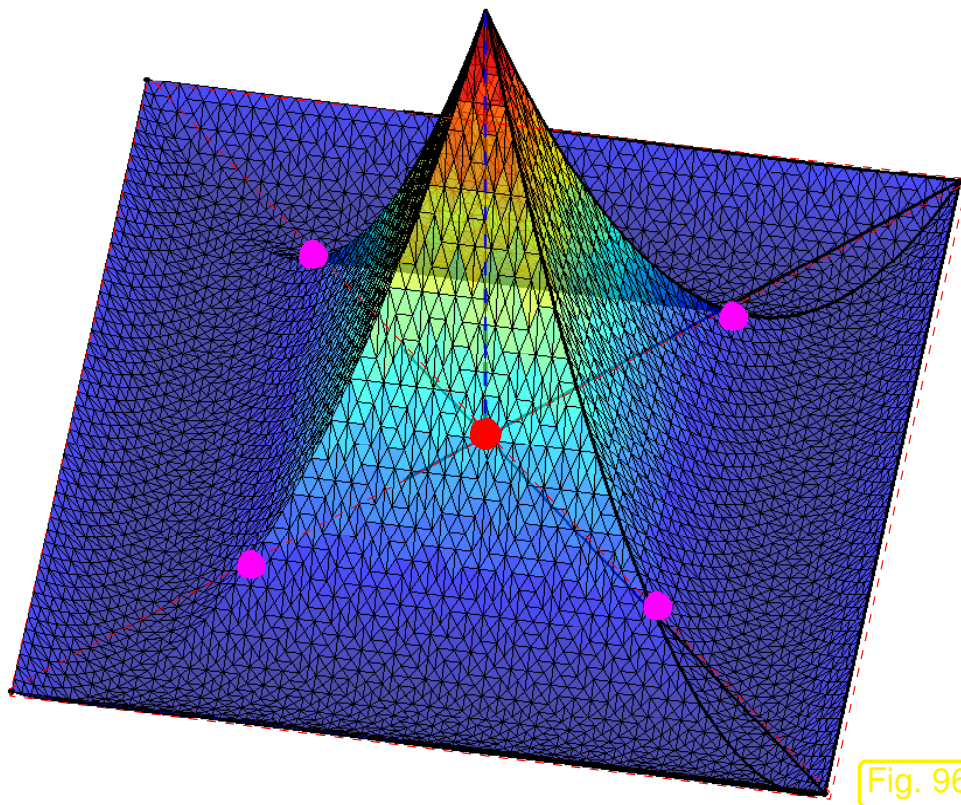


Fig. 95



◁ Global basis function for $\mathcal{S}_2^0(\mathcal{M})$ associated with a vertex

(3.4.3): this function attains value $= 1$ at a vertex (●) and vanishes at the midpoints (●) of the edges of adjacent triangles, as well as at any other vertex.

Fig. 96



Example 3.4.5 (Interpolation nodes for cubic and quartic Lagrangian FE in 2D).

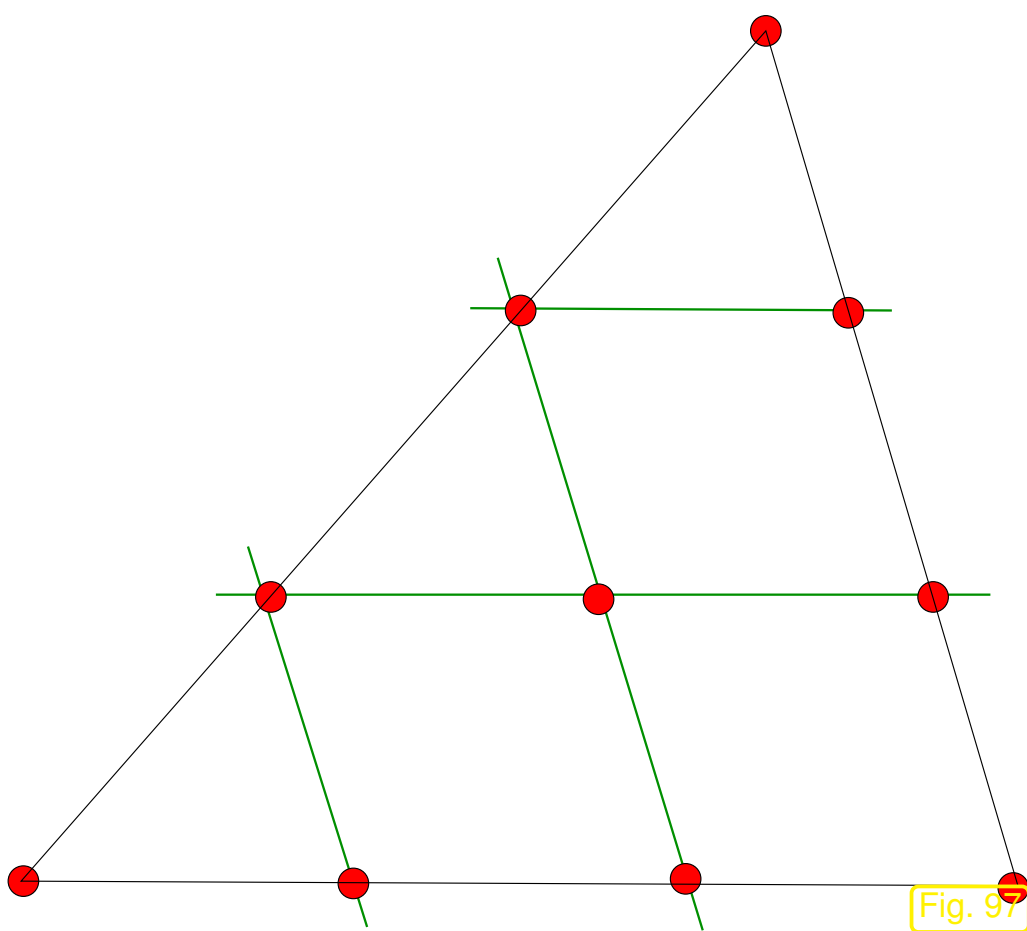


Fig. 97

(local) interpolation nodes for $S_3^0(\mathcal{M})$

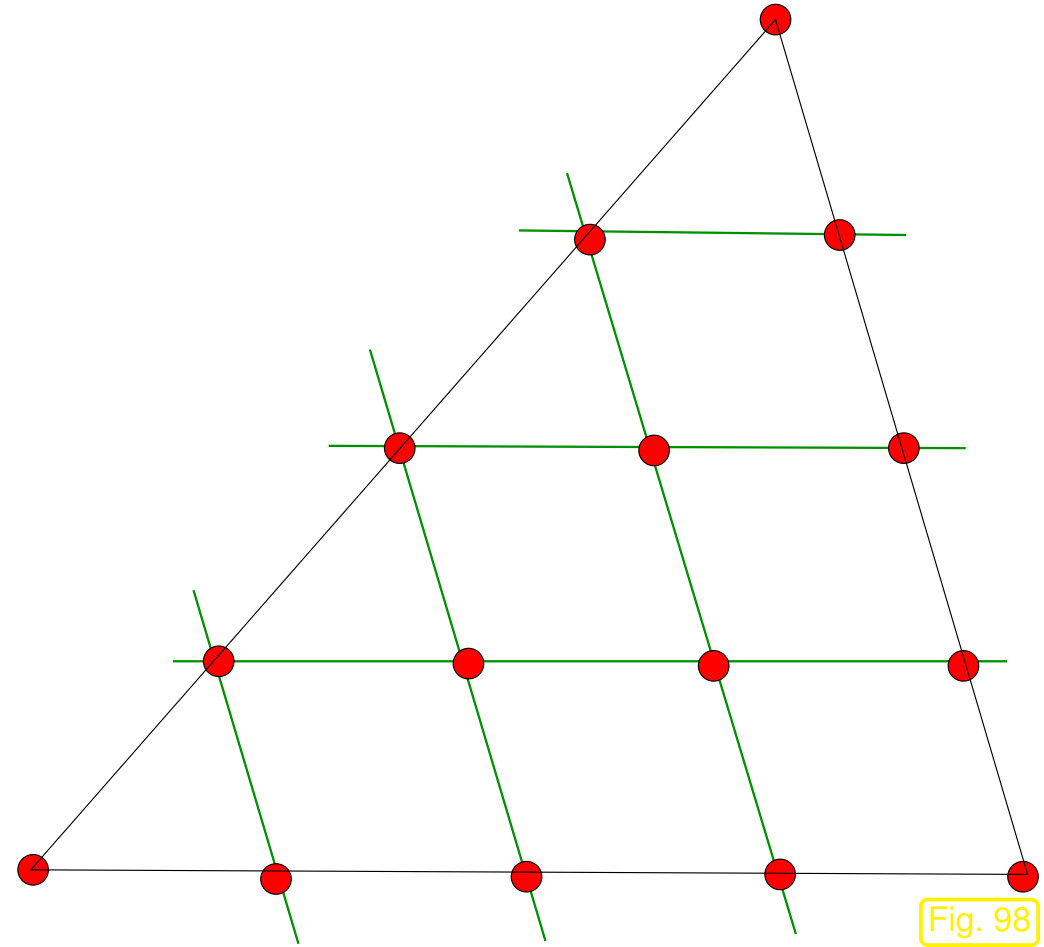


Fig. 98

(local) interpolation nodes for $S_4^0(\mathcal{M})$



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

3.4.2 Tensor-product Lagrangian FEM

Now we consider tensor product meshes (grids), see (3.3.2), Fig. 83, for a 2D example.

Example 3.4.6 (Bilinear Lagrangian finite elements).

Sought: generalization of 1D piecewise linear finite element functions from Sect. 1.5.1.2, see Fig. 25, to 2D tensor product grid \mathcal{M} .

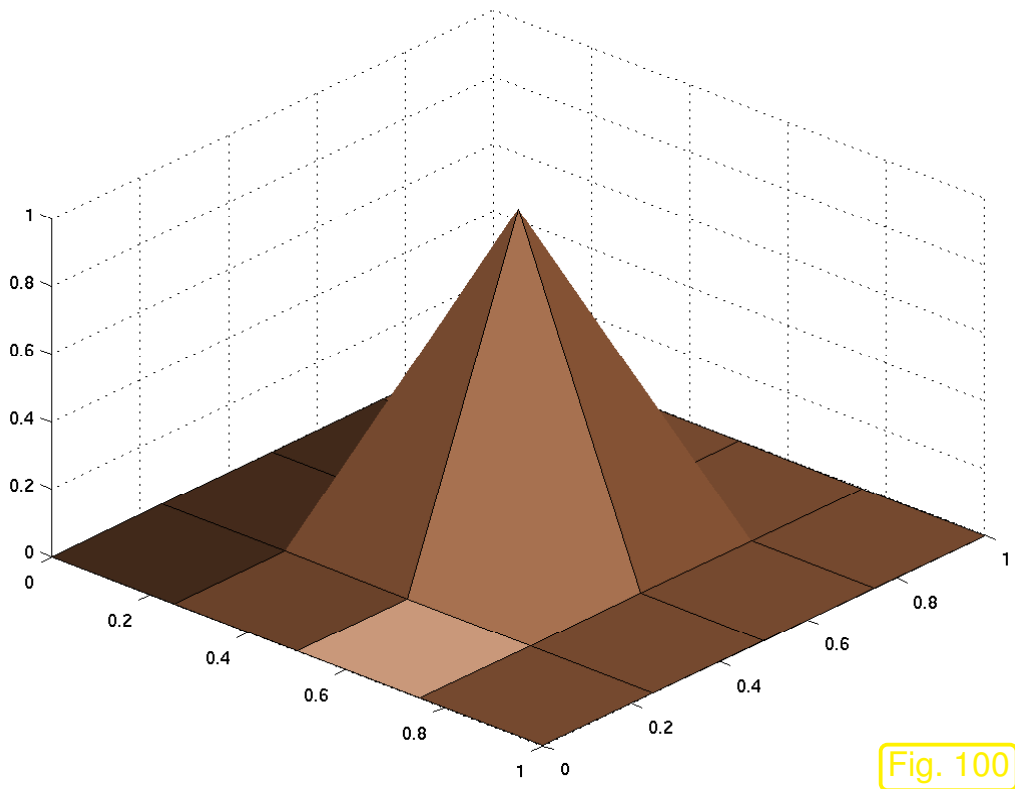
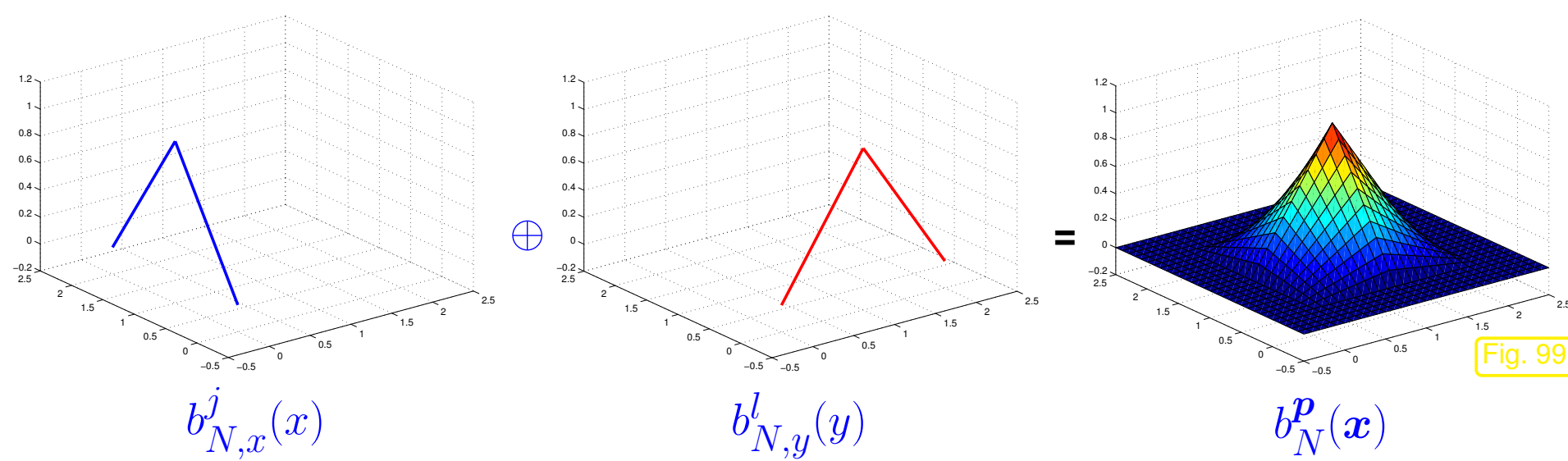
Tensor product structure of \mathcal{M} ➤ tensor product construction of FE space

This is best elucidated by a tensor product construction of basis functions:

$$\begin{aligned} b_{N,x}^j(x) &: \text{1D tent function on } \mathcal{M}_x = \{[x_{j-1}, x_j], j = 1, \dots, n\} \\ b_{N,y}^l(y) &: \text{1D tent function on } \mathcal{M}_y = \{[y_{j-1}, y_j], j = 1, \dots, n\} \end{aligned}$$

2D tensor product “tent function” associated with node \mathbf{p} :

$$b_N^{\mathbf{p}}(\mathbf{x}) = b_{N,x}^j(x_1) \cdot b_{N,y}^l(x_2), \quad \text{where } \mathbf{p} = (x_j, y_l)^T. \quad (3.4.7)$$



◁ 2D tensor product tent function

No pyramid !

Basis functions *associated* (\rightarrow Sect. 3.3.3, condition (c)) with nodes of \mathcal{M} ,

Tensor product construction ➤ **bilinear** local shape functions, e.g. on $K =]0, 1[^2$

$$\begin{aligned}
 b_K^1(\mathbf{x}) &= (1 - x_1)(1 - x_2) , \\
 b_K^2(\mathbf{x}) &= x_1(1 - x_2) , \\
 b_K^3(\mathbf{x}) &= x_1x_2 , \\
 b_K^4(\mathbf{x}) &= (1 - x_1)x_2 .
 \end{aligned}
 \tag{3.4.8}$$

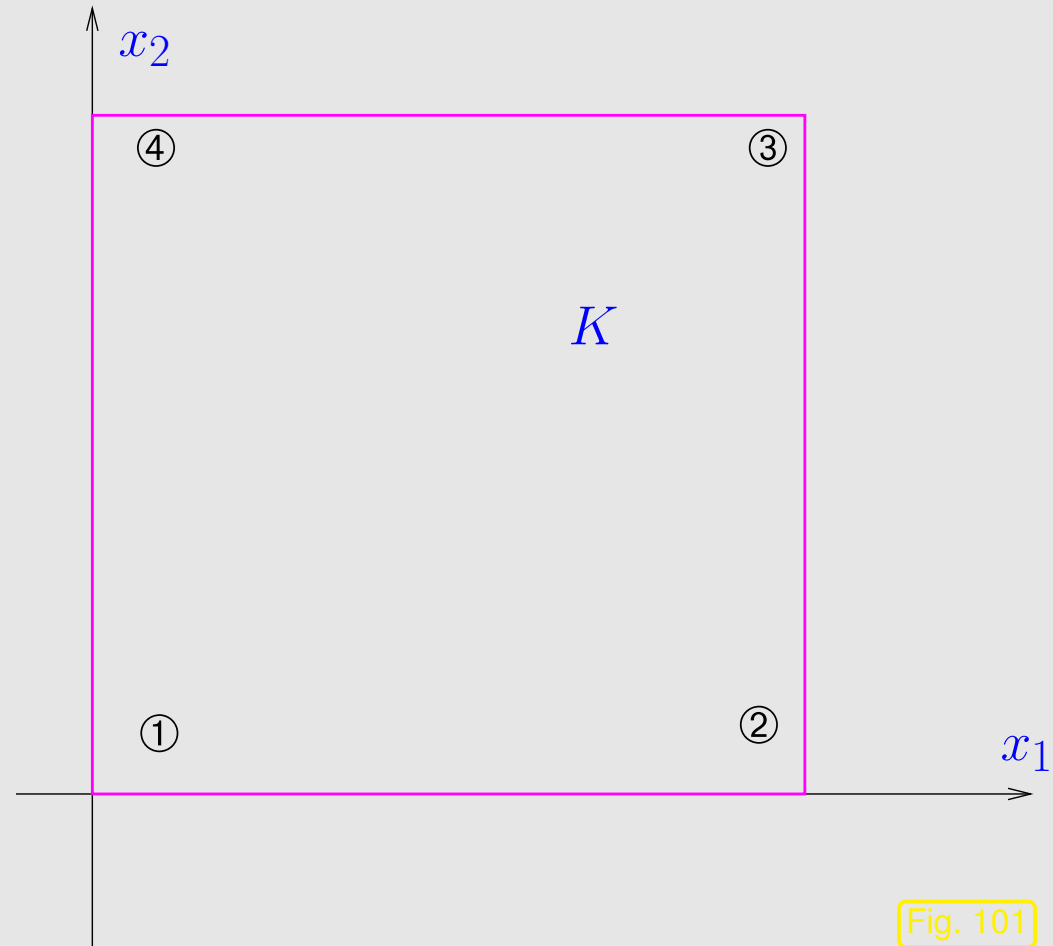


Fig. 101

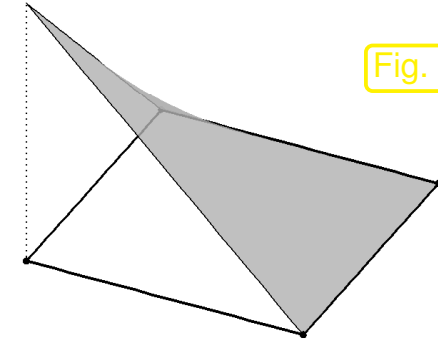
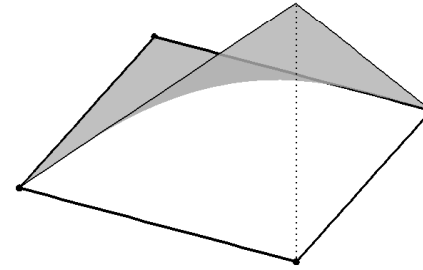
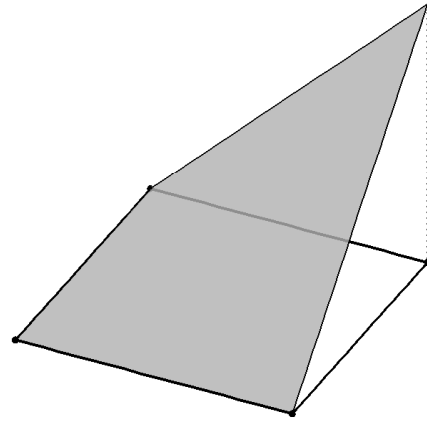
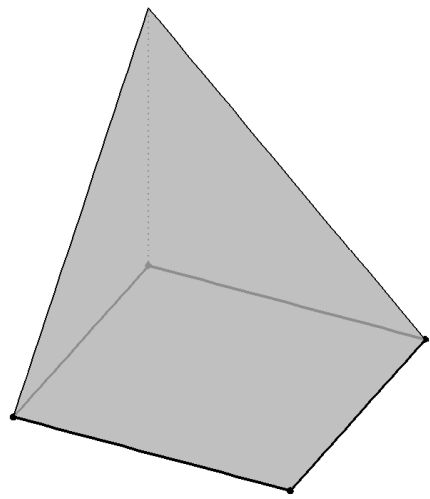


Fig. 102

Bilinear local shape functions on unit square K

▶ $\text{Span} \{ b_K^1, b_K^2, b_K^3, b_K^4 \} = \mathcal{Q}_1(\mathbb{R}^2) .$



Bilinear Lagrangian finite element space on 2D tensor product mesh \mathcal{M} :

$$\mathcal{S}_1^0(\mathcal{M}) := \{ v \in C^0(\Omega) : v|_K \in \mathcal{Q}_1(\mathbb{R}^2) \forall K \in \mathcal{M} \} . \quad (3.4.9)$$



The following is a natural generalization of (3.4.9) to higher degree local tensor product polynomials, see Def. 3.3.7:

Definition 3.4.10 (Tensor product Lagrangian finite element spaces).

Space of p -th degree Lagrangian finite element functions on tensor product mesh \mathcal{M}

$$\mathcal{S}_p^0(\mathcal{M}) := \{v \in C^0(\bar{\Omega}) : v|_K \in \mathcal{Q}_p(K) \forall K \in \mathcal{M}\} .$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

Terminology: $\mathcal{S}_1^0(\mathcal{M}) =$ multilinear finite elements ($p = 1, d = 2 =$ bilinear finite elements)

SAM, ETHZ

Remaining issue: definition of global basis functions (global shape functions)

Policy: use of **interpolation nodes** as in Sect. 3.4.1, see Ex. 3.4.2.

Example 3.4.11 (Quadratic tensor product Lagrangian finite elements).

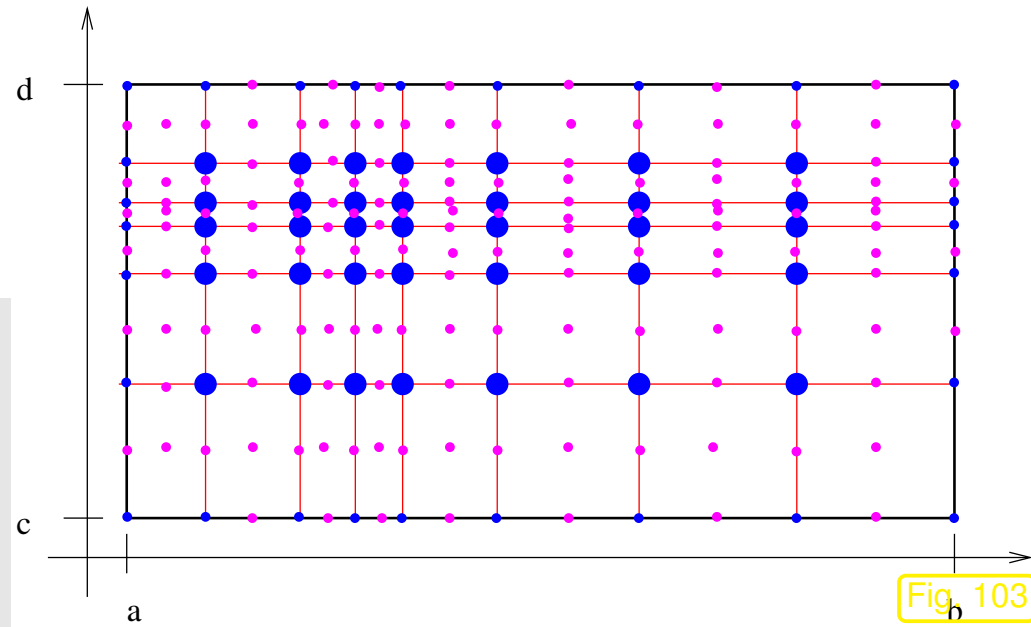
Consider case $p = 2, d = 2$ of Def. 3.4.10:

Interpolation nodes for $\mathcal{S}_2^0(\mathcal{M})$

$$\mathcal{N} = \mathcal{V}(\mathcal{M}) \cup \{\text{midpoints of edges}\} .$$

Note: number of interpolation nodes belonging to one cell is

$$9 = \dim \mathcal{Q}_2(\mathbb{R}^2) .$$



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

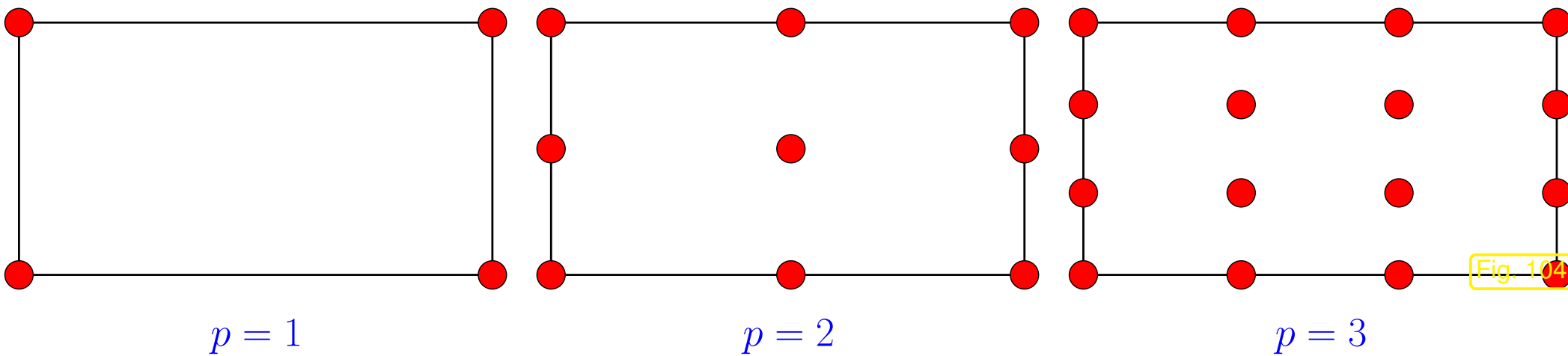
SAM, ETHZ

► Global basis functions defined analogously to (3.4.3).



Choice of interpolation nodes for tensor product Lagrangian finite elements:

Fig. 104



Remark 3.4.12 (Imposing homogeneous Dirichlet boundary conditions).

What is a global basis for $\mathcal{S}_p^0(\mathcal{M}) \cap H_0^1(\Omega)$, where \mathcal{M} is either a simplicial mesh or a tensor product mesh?

We proceed analogous to Rem. 3.2.5: recall that global basis functions are defined via interpolation nodes \mathbf{p}^j , $j = 1, \dots, N$, see (3.4.3).

$$\mathcal{S}_{p,0}^0(\mathcal{M}) := \mathcal{S}_p^0(\mathcal{M}) \cap H_0^1(\Omega) = \text{Span} \left\{ b_N^j : \mathbf{p}^j \in \Omega \text{ (interior node)} \right\}. \quad (3.4.13)$$



Remark 3.4.14 ((Bi)-linear Lagrangian finite elements on hybrid meshes).

\mathcal{M} : 2D hybrid mesh comprising triangles & rectangles

Idea: use

- linear functions (\rightarrow Def. 3.3.3, $p = 1$) on triangular cells,
- bi-linear functions (\rightarrow Def. 3.4.10, $p = 1$) on rectangles.

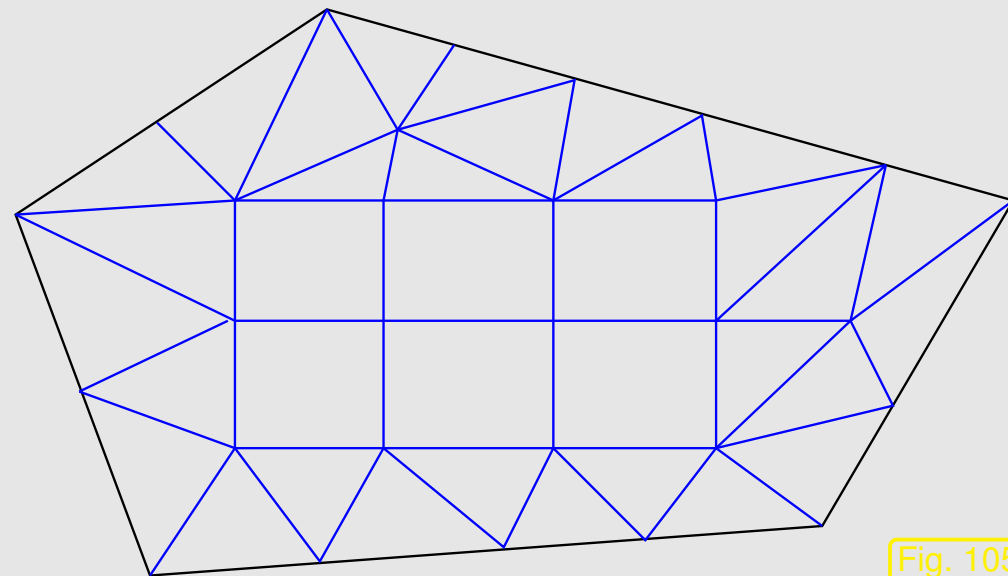
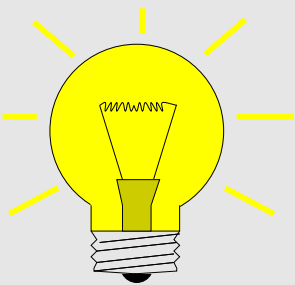


Fig. 105



$$\mathcal{S}_1^0(\mathcal{M}) = \left\{ v \in H^1(\Omega) : v|_K \in \begin{cases} \mathcal{P}_1(\mathbb{R}^2) & , \text{ if } K \in \mathcal{M} \text{ is triangle,} \\ \mathcal{Q}_1(\mathbb{R}^2) & , \text{ if } K \in \mathcal{M} \text{ is rectangle} \end{cases} \right\}. \quad (3.4.15)$$

Two issues arise:

1. Does the prescription (3.4.15) yield a large enough space? (Note that $v \in H^1(\Omega) \Rightarrow \mathcal{S}_1^0(\mathcal{M}) \subset C^0(\Omega)$, see Thm. 2.2.26, but continuity might enforce too many constraints.)

2. Does the space from (3.4.15) allow for locally supported basis functions associated with nodes of the mesh?

We will give a positive answer to both question by constructing the basis functions:

Define global shape functions b_N^j according to (3.2.4)

This makes sense, because

- linear/bi-linear functions on K are uniquely determined by their values in the vertices,
- the restrictions to an edge of K of the local linear and bi-linear shape functions are both *linear* univariate functions, see Figs. 72, 102.

► Fixing vertex values for $v_N \in \mathcal{S}_1^0(\mathcal{M})$ uniquely determines v on all edges of \mathcal{M} already, thus, *ensuring global continuity*, which is necessary due to Thm. 2.2.26.

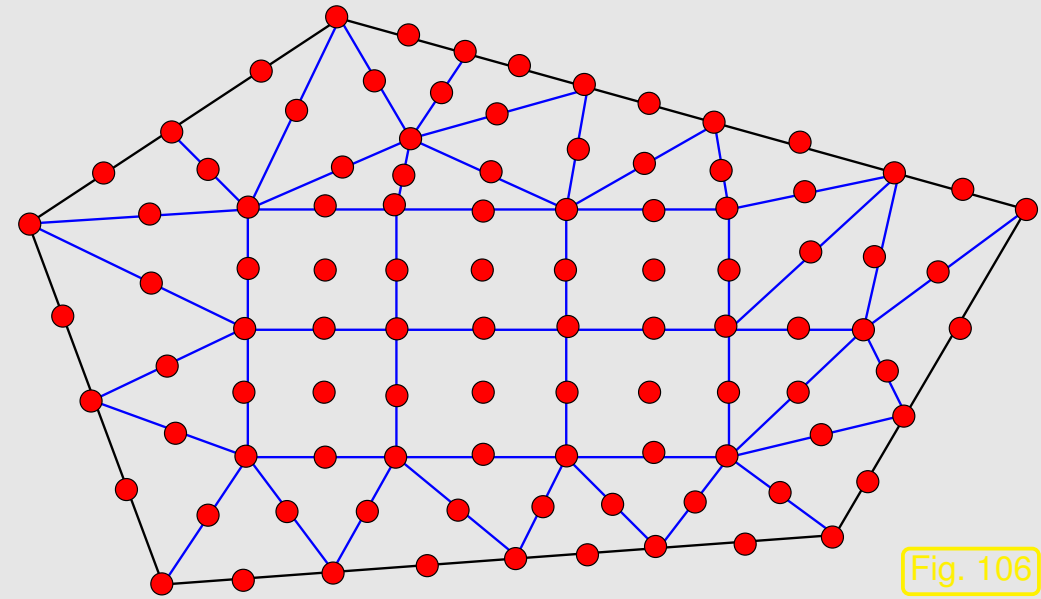
Remark 3.4.16 (Lagrangian finite elements on hybrid meshes).

\mathcal{M} : 2D hybrid mesh comprising triangles & rectangles

☞ Matching interpolation nodes on edges of triangles and rectangles

▶ Glueing of local shape functions on triangles and rectangles possible

global interpolation nodes for $p = 2$ ▶
 ▲



R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

3.5 Implementation of FEM

This section discusses algorithmic details of Galerkin finite element discretization of 2nd-order elliptic variational problems for spatial dimension $d = 2, 3$ on bounded polygonal/polyhedral domains $\Omega \subset \mathbb{R}^d$.

The presentation matches the **LehrFEM finite element MATLAB library**, parts of which will be made available for participants of the course. A detailed documentation is available from [7].

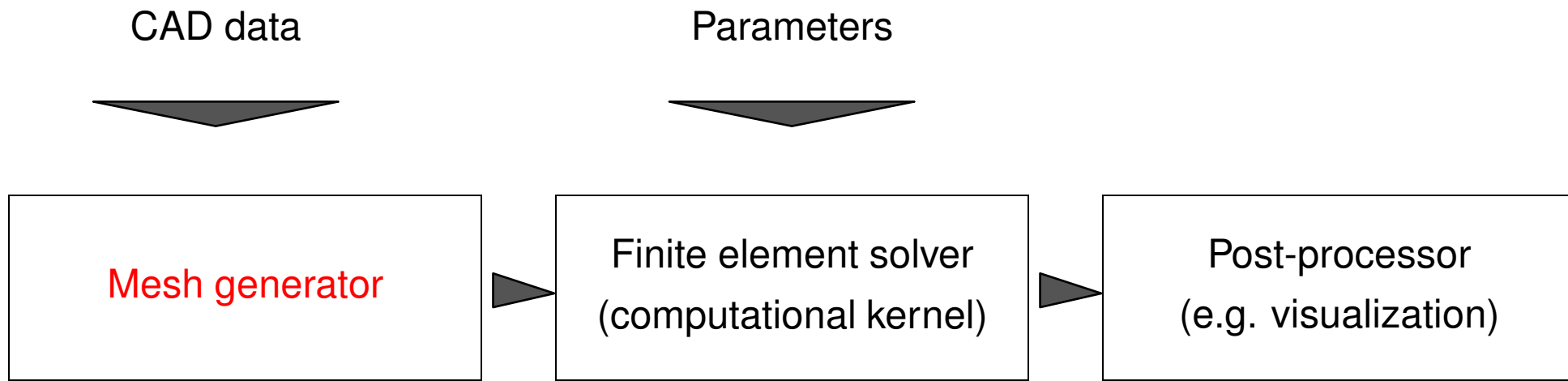
The guiding principle behind the implementation of finite element codes is

to rely on *local* computations as much as possible!

This is made possible by the *local supports* of the global basis functions, see Sect. 3.3.3, Ex. 3.3.10.

3.5.1 Mesh file format

Data flow in (most) finite element software packages:



Here “▶” designates passing of information, which is usually done by writing and reading files to and from hard disk. This requires particular file formats.

Example 3.5.1 (Triangular mesh: file format).

File format for storing triangular mesh (of polygonal domain):

```

# Two-dimensional simplicial mesh

1   $\xi_1$    $\eta_1$           # Coordinates of first node
2   $\xi_2$    $\eta_2$           # Coordinates of second node
       $\vdots$ 
N   $\xi_N$    $\eta_N$           # Coordinates of N-th node
1   $n_1^1$    $n_2^1$    $n_3^1$    $X_1$       # Indices of nodes of first triangle
2   $n_1^2$    $n_2^2$    $n_3^2$    $X_2$       # Indices of nodes of second triangle
       $\vdots$ 
M   $n_1^M$    $n_2^M$    $n_3^M$    $X_M$  # Indices of nodes of M-th triangle

```

(3.5.2)

$X_i, i = 1, \dots, M \rightarrow$ extra information (e.g. material properties in triangle $\#i$).

Optional: additional information about edges (on $\partial\Omega$):

$$\begin{aligned}
 K \in \mathbb{N} & \quad \# \text{ Number of edges on } \partial\Omega \\
 n_1^1 \ n_2^1 \ Y_1 & \quad \# \text{ Indices of endpoints of first edge} \\
 n_1^2 \ n_2^2 \ Y_2 & \quad \# \text{ Indices of endpoints of second edge} \\
 & \quad \vdots \\
 n_1^K \ n_2^K \ Y_K & \quad \# \text{ Indices of endpoints of } K\text{-th edge}
 \end{aligned} \tag{3.5.3}$$

$Y_k, k = 1, \dots, K \rightarrow$ extra information

◇ R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Example 3.5.4 (Mesh file format for MATLAB code “LehrFEM”).

Vertex coordinate file:

```

% List of vertices
1 +0.0000000e+00 -1.0000000e+00
2 +1.0000000e+00 +0.0000000e+00
3 +0.0000000e+00 +1.0000000e+00
4 -1.0000000e+00 +0.0000000e+00
5 +0.0000000e+00 +0.0000000e+00

```

Cell information file:

```

% List of elements
1      1      2      5
2      2      3      5
3      3      4      5
4      4      1      5

```

Loading a mesh

```
m = load_Mesh('Coord_Circ.dat', ...
              'Elem_Circ.dat');
plot_Mesh(m, 'apts');
```

Option flags:

'a' : with axes

'p' : vertex labels on

't' : cell labels on

's' : caption/title on

For details see [7, Sect. 1.3.1], [7, Sect. 1.3.2].

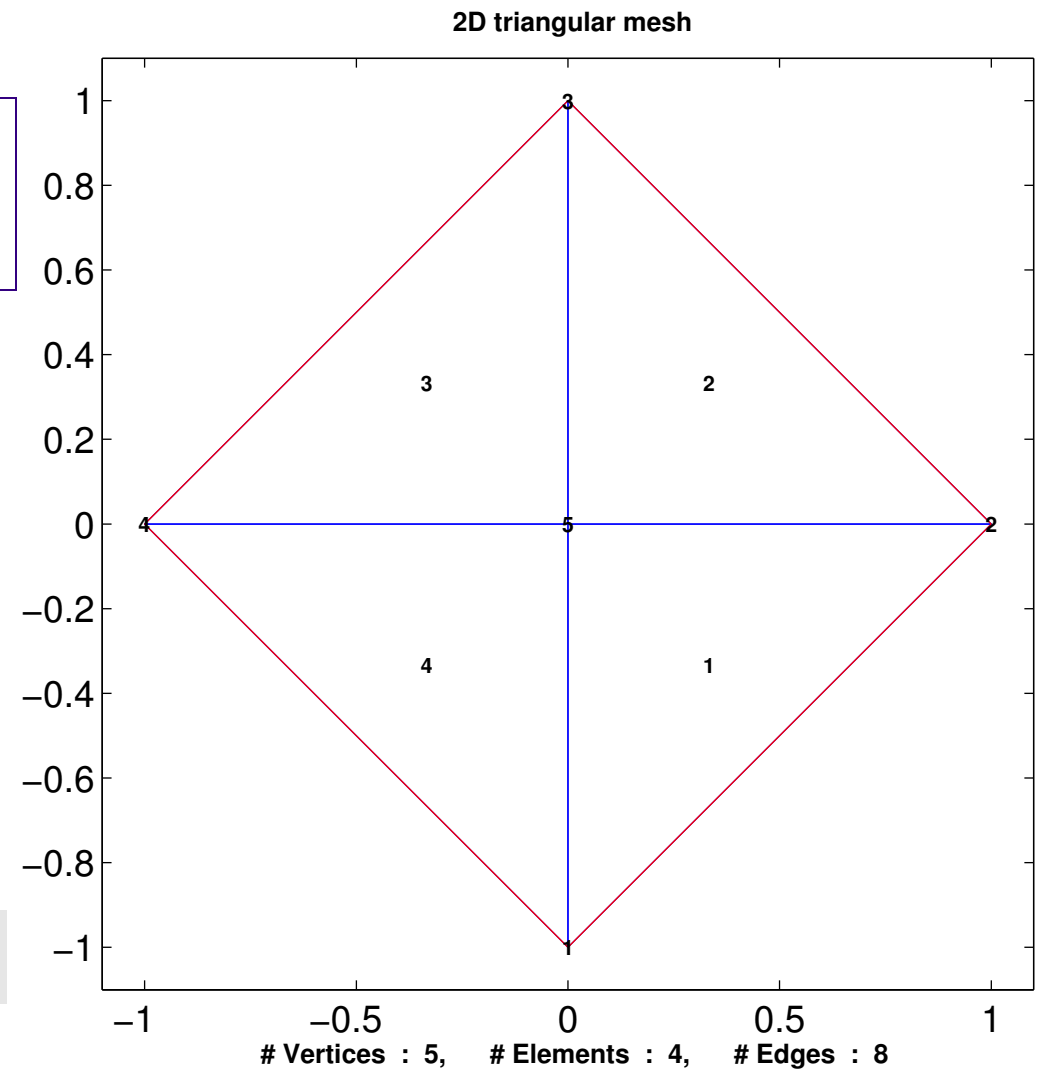


Fig. 107

How to create a mesh ?

→ **Mesh generation** (beyond scope of this course)

- Free software:
- DistMesh (MATLAB, used in “LehrFEM”, see [7, Sect. 1.2])
 - NETGEN (industrial strength open source mesh generator)
 - Triangle (easy to use 2D mesh generator)
 - TETGEN (Tetrahedral mesh generation)
 - GMSH a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Example 3.5.5 (Mesh generation in LehrFEM).

Algorithm & details → [27], more explanations in [7, Sect. 1.2].

MATLAB-CODE: mesh generation for circular domain

```

BBOX = [-1 -1; 1 1];
H0 = 0.1;
DHD = @(x) sqrt(x(:,1).^2+x(:,2).^2)-1;
HHANDLE = @(x) ones(size(x,1),1);
Mesh = init_Mesh(BBOX,H0,DHD,...
                HHANDLE,[],1);
save_Mesh(Mesh,'Coordinates.dat',...
          'Elements.dat');

```

Bounding box

Largest reasonable edge length

Signed distance function $\varphi(\mathbf{x})$:
(distance from $\partial\Omega$, $\varphi(\mathbf{x}) < 0 \Leftrightarrow$
 $\mathbf{x} \in \Omega$)Element size function
(determines local edge length)R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

3.5.2 Mesh data structures [7, Sect. 1.1]

Issue: internal representation of mesh (\rightarrow Def. 3.3.1) in computer code

mesh data structure must provide:

1. offer unique identification of cells/(faces)/(edges)/vertices
2. represent **mesh topology** (= incidence relationships of cells/faces/edges/vertices)
3. describe **mesh geometry** (= location/shape of cells/faces/edges/vertices)
4. allow sequential access to edges/faces of a cell
(→ traversal of local shape functions/degrees of freedom)
5. make possible traversal of cells of the mesh (→ **global numbering**)

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Focus: **array oriented data layout** (→ MATLAB, FORTRAN)

Notation:

\mathcal{M} = mesh (set of elements), $\mathcal{V}(\mathcal{M})$ = set of nodes (vertices) in \mathcal{M} , $\mathcal{E}(\mathcal{M})$ = set of edges in \mathcal{M}

Case: d -dimensional simplicial triangulation \mathcal{M} , *minimal* data structure (cf. Sect. 3.5.1)

→ Coordinates of vertices $\mathcal{V}(\mathcal{M})$: $\#\mathcal{V}(\mathcal{M}) \times d$ -array Coordinates of reals

→ Vertex indices for cells: $\#\mathcal{M} \times (d + 1)$ -array Elements of integers.

Example 3.5.6 (Arrays storing 2D triangular mesh).

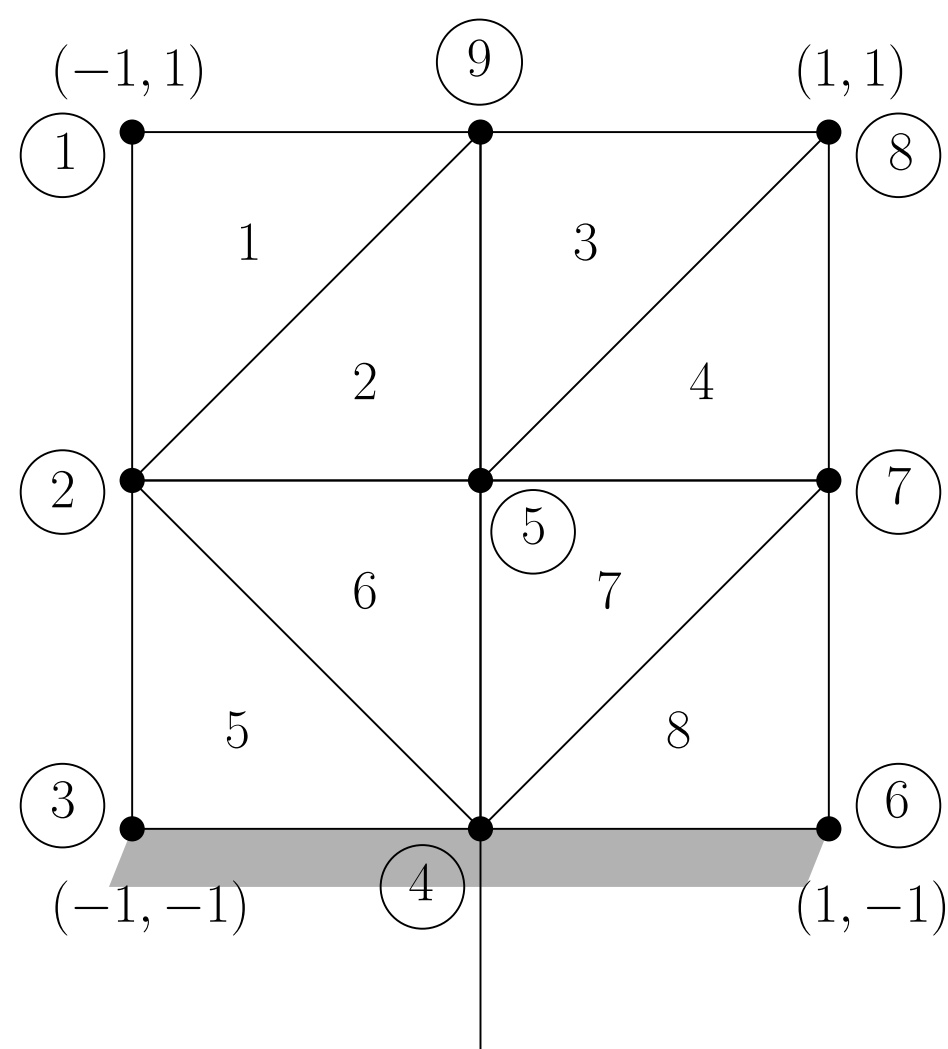


Fig. 108

i	(x_1^i, x_2^i)	
1	-1	1
2	-1	0
3	-1	-1
4	0	-1
5	0	0
6	1	-1
7	1	0
8	1	1
9	0	1

Array Coordinates

K_j	Vertex indices		
1	1	2	9
2	2	5	9
3	5	8	9
4	5	7	8
5	3	4	2
6	4	5	2
7	4	7	5
8	4	6	7

Array Elements

Recall `idx`-array discussed in the context of assembly for linear finite elements in Sect. 3.2.5, see (3.2.14). This corresponds to the `elements` array in the mesh data structure.

► `Coordinates & Elements` already offers complete description of the mesh topology and geometry !

This information is enough for efficient assembly of finite element Galerkin matrices/right hand side vectors for (bi-)linear Lagrangian finite elements, see Rem. 3.2.13, 3.2.20.

Note: Global shape functions associated with edges/faces ➤ extra information required !

Optional extra information:

→ Edge connecting vertices: $\#\mathcal{V}(\mathcal{M}) \times \#\mathcal{V}(\mathcal{M})$ symmetric sparse integer matrix $I_{\mathcal{E}}$

$$(I_{\mathcal{E}})_{ij} := \begin{cases} 0 & , \text{ if vertex } \#i \text{ not linked to } \#j \\ e_{ij} & , \text{ if edge connecting } \#i \text{ and } \#j \end{cases}$$

here e_{ij} is the unique edge number $\in \{1, 2, \dots, \#\mathcal{E}(\mathcal{M})\}$

→ End points of the edges: $\#\mathcal{E}(\mathcal{M}) \times 2$ array of integer (= vertex indices of end points).

→ Cell adjacent to edges: $\#\mathcal{E}(\mathcal{M}) \times 2$ array of integers (=cell indices)
(one cell index =0 if edge is on $\partial\Omega$)



Example 3.5.7 (Extended MATLAB mesh data structure). → [7, Sect. 1.1]

```
mesh = add_Edge2Elem(add_Edges(init_Mesh(BBOX, H0, DHD, HHANDLE, [], 1)))
```

(init_Mesh → Ex. 3.5.5)

mesh =

- Coordinates: [5x2 double] ← vertex coordinates, see Ex. 3.5.4
- Elements: [4x3 double] ← vertex indices of triangles, see Ex. 3.5.4
- Edges: [8x2 double] ← indices of endpoints in Coordinates array
- Vert2Edge: [5x5 double] ← $\#\mathcal{V}(\mathcal{M}) \times \#\mathcal{V}(\mathcal{M})$ sparse integer matrix:
entry $(i, j) =$ edge index, if $\neq 0$
- Edge2Elem: [8x2 double] ← $\#\mathcal{E}(\mathcal{M}) \times 2$ integer array:
indices of adjacent cells in Elements array
- EdgeLoc: [8x2 double] ← $\#\mathcal{E}(\mathcal{M}) \times 2$ integer array: local indices of edges w.r.t. adjacent cells

Notation: $\mathcal{E}(\mathcal{M}) \hat{=}$ edges of 2D mesh

How to number ↔ order local shape functions ?
 global shape functions

Elements, Edges arrays ➤ ordering of vertices of cells/endpoints of edges

Arrays (of vertices,cells,edges) ➤ array indices ➤ numbering of global shape functions

Remark 3.5.8. Second option: C++/JAVA-style object oriented data layout

Nodes, cells of \mathcal{M} \longleftrightarrow **dynamically allocated** objects (instances of classes Node, Cell)

```
class Node {
private:
    double x,y;
    ID id;
public:
    Node(double x,double y,ID id=0);
    Point getCoords(void) const;
    ID getId(void) const;
};
```

```
class Cell {
private:
    const vector<Node*> vertices;
    ID id;
public:
    Cell(const vector<Node*> &vertices,ID id=0);
    int NoNodes(void) const;
    const Node &getNode(int) const;
    ID getId(void) const;
};
```

```
class BdFace {  
  private:  
    const vector<Node*> vertices;  
    BdCond bdcond;  
  public:  
    BdFace(const vector<Node*> &vertices);  
    int NoNodes(void) const;  
    const Node &getNode(int) const;  
    BdCond getBdCond(void) const;  
};
```

```
class Mesh {  
  private:  
    list<Node> nodes;  
    list<Cell> cells;  
    list<BdFace> bdfaces;  
  public:  
    Mesh(istream &file);  
    virtual Mesh(void);  
    const list<Node> &Nodes(void) const;  
    const list<Cell> &Cells(void) const;  
    const list<BdFace> &BdFaces(void) const;  
};
```

ID `getId()` → provides **unique identifier** for each node/cell.

Distinguish: — **local objects** (→ classes `Node`, `Cell`, `BdFace`)
— **global objects** (“mesh management” class `Mesh`, see below)



3.5.3 Assembly [7, Sect. 5]

“Assembly” = term used for computing entries of stiffness matrix/right hand side vector (load vector) in a finite element context.

From the dictionary: “Assemble” = to fit together all the separate parts of sth.

Aspects of assembly for linear Lagrangian finite elements ($V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$) were discussed in Sects. 3.2.5, 3.2.6. (Refresh yourself on these sections in case you cannot remember the main ideas behind building the Galerkin matrix and right hand side vector.)

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

We consider a discrete variational problem ($V_{0,N}$ = FE space, $\dim V_{0,N} = N \in \mathbb{N}$, see (3.1.4))

$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N} . \quad (3.1.4)$$

To be computed (see also Sect. 3.2.5, Sect. 3.2.6):

- Galerkin matrix (stiffness matrix): $\mathbf{A} = \left(\mathbf{a}(b_N^j, b_N^i) \right)_{i,j=1}^N \in \mathbb{R}^{N,N}$
- r.h.s. vector (load vector): $\vec{\varphi} := \left(\ell(b_N^i) \right)_{i=1}^N \in \mathbb{R}^N$

both can be written in terms of **local cell contributions**, since usually

$$\mathbf{a}(u, v) = \sum_{K \in \mathcal{M}} \mathbf{a}_K(u|_K, v|_K) \quad , \quad \ell(v) = \sum_{K \in \mathcal{M}} \ell_K(v|_K) . \quad (3.5.9)$$

Example: bilinear forms/linear forms arising from 2nd-order elliptic BVPs, e.g, (2.9.4), (2.9.5), (2.9.6), can be localized in straightforward fashion by restricting integration to mesh cells (\rightarrow Rem. 3.2.2): for $u, v \in H^1(\Omega)$

$$\mathbf{a}(u, v) := \int_{\Omega} \alpha(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = \sum_{K \in \mathcal{M}} \underbrace{\int_K \alpha(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x}}_{=: \mathbf{a}_K(u|_K, v|_K)} , \quad (3.5.10)$$

$$\ell(v) := \int_{\Omega} f v \, d\mathbf{x} = \sum_{K \in \mathcal{M}} \underbrace{\int_K f v \, d\mathbf{x}}_{=: \ell_K(v|_K)} . \quad (3.5.11)$$

Recall (3.3.11): Restrictions of global shape functions to cells = local shape functions

Definition 3.5.12 (Element (stiffness) matrix and element (load) vector).

Given local shape functions $\{b_K^1, \dots, b_K^Q\}$, $Q \in \mathbb{N}$, we call

$$\text{element (stiffness) matrix } \mathbf{A}_K := \left(\mathbf{a}_K(b_K^j, b_K^i) \right)_{i,j=1}^Q \in \mathbb{R}^{Q,Q},$$

$$\text{element (load) vector } \vec{\varphi}_K := \left(\ell_K(b_K^i) \right)_{i=1}^Q \in \mathbb{R}^Q.$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Note: Here Q , the number of local shape functions on element $K \in \mathcal{M}$, is independent of K . In general, we could also have $Q = Q_K$ when we blend several element types in one mesh, see Rem. 3.4.14.

Type of FE space	Q
degree p Lagrangian FE on <i>triangular</i> mesh	$\dim \mathcal{P}_p(\mathbb{R}^2) = \frac{1}{2}(p+1)(p+2)$
degree p Lagrangian FE on <i>tetrahedral</i> mesh	$\dim \mathcal{P}_p(\mathbb{R}^3) = \frac{1}{6}(p+1)(p+2)(p+3)$
degree p Lagrangian FE on <i>tensor product</i> mesh in 2D	$\dim \mathcal{Q}_p(\mathbb{R}^2) = (p+1)^2$

Again scrutinize Figs. 74, 75 and the accompanying remarks in Sect. 3.2.5. We learn that in the special setting of this section

- the entries of the finite element Galerkin matrix can be obtained by summing *corresponding* entries of *some* element matrices,
- this corresponding entry of an element matrices is determined by the unique association of a local basis function to a global basis function.

These insights are formalized in the next theorem.

Theorem 3.5.13. *The stiffness matrix and load vector can be obtained from their cell counterparts by*

$$\mathbf{A} = \sum_K \mathbf{T}_K^\top \mathbf{A}_K \mathbf{T}_K, \quad \vec{\varphi} = \sum_K \mathbf{T}_K^\top \vec{\varphi}_K, \quad (3.5.14)$$

with the *index mapping matrices* (“T-matrices”) $\mathbf{T}_K \in \mathbb{R}^{Q,N}$, defined by

$$(\mathbf{T}_K)_{ij} := \begin{cases} 1 & , \text{ if } (b_N^j)|_K = b_K^i, \\ 0 & , \text{ otherwise.} \end{cases} \quad 1 \leq i \leq Q, 1 \leq j \leq N. \quad (3.5.15)$$

Note: Every T-matrix has exactly one non-vanishing entry per row.

Proof. (of Thm. 3.5.13)

$$(\mathbf{A})_{ij} = \mathbf{a}(b_N^j, b_N^i) = \sum_{K \in \mathcal{M}} \mathbf{a}_K(b_{N|K}^j, b_{N|K}^i) = \sum_{\substack{K \in \mathcal{M}, \text{supp}(b_N^j) \cap K \neq \emptyset, \\ \text{supp}(b_N^i) \cap K \neq \emptyset}} \mathbf{a}_K(b_K^{l(j)}, b_K^{l(i)}) = \sum_{\substack{K \in \mathcal{M}, \text{supp}(b_N^j) \cap K \neq \emptyset, \\ \text{supp}(b_N^i) \cap K \neq \emptyset}} (\mathbf{A}_K)_{l(i), l(j)}$$

$l(i) \in \{1, \dots, Q\}$, $1 \leq i \leq N \hat{=}$ index of the local shape function corresponding to the global shape function b_N^i on K .

➤ By (3.5.15), the indices $l(i)$ encode the T-matrix according to

$$(\mathbf{T}_K)_{l(i),i} = 1, \quad i = 1, \dots, N,$$

where all other entries of \mathbf{T}_K are understood to vanish.

$$\Rightarrow (\mathbf{A})_{ij} = \sum_{\substack{K \in \mathcal{M}, \text{supp}(b_N^j) \cap K \neq \emptyset, \\ \text{supp}(b_N^i) \cap K \neq \emptyset}} \sum_{l=1}^Q \sum_{n=1}^Q (\mathbf{T}_K)_{li} (\mathbf{A}_K)_{ln} (\mathbf{T}_K)_{nj}. \quad \square$$

Example 3.5.16 (Assembly for linear Lagrangian finite elements on triangular mesh).

Using the local/global numbering indicated beside

$$\rightarrow \mathbf{T}_{K^*} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Local→global index mapping:

$$\begin{aligned} \text{In } K^* : \quad l(9) &= 3, \\ l(7) &= 2, \\ l(2) &= 1. \end{aligned}$$

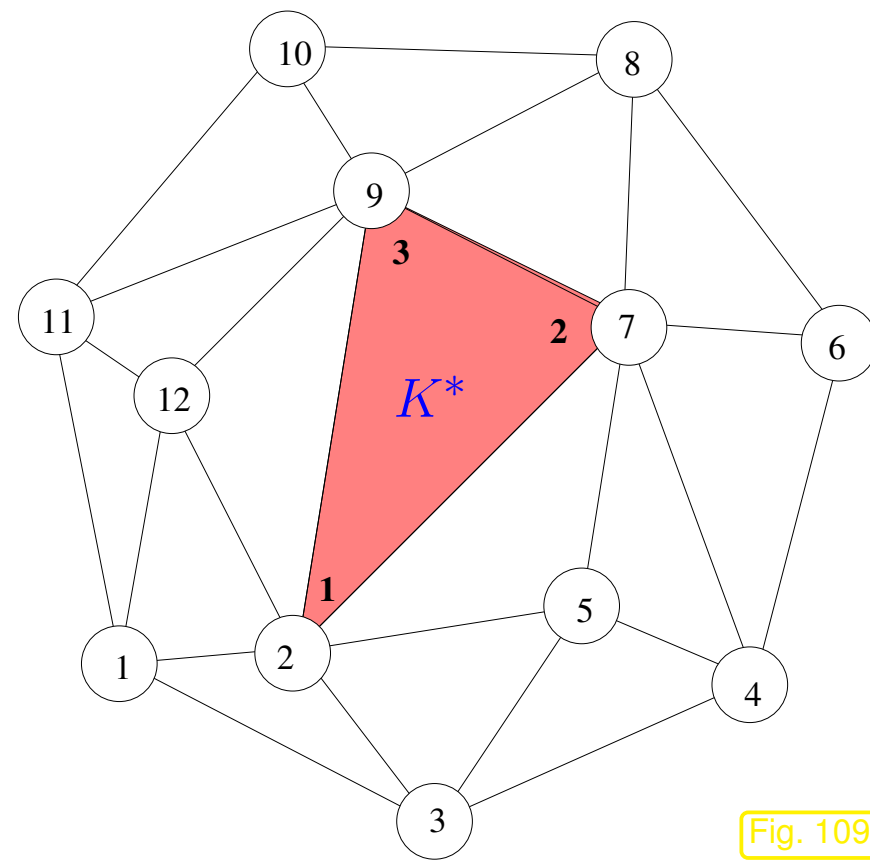




Fig. 109

Cell oriented assembly \leftrightarrow (3.5.14) $\leftrightarrow \mathbf{A} = \sum_K \mathbf{T}_K^\top \mathbf{A}_K \mathbf{T}_K$

$$\mathbf{A} = \sum_K \mathbf{T}_K^\top \mathbf{A}_K \mathbf{T}_K := \left\{ \begin{array}{l} \text{foreach } K \in \mathcal{M} \text{ do} \\ \quad \text{local operations on } K \text{ (} \rightarrow \mathbf{A}_K \text{) and } \mathbf{A} = \mathbf{A} + \mathbf{T}_K^\top \mathbf{A}_K \mathbf{T}_K \\ \text{enddo} \end{array} \right\}$$


Notion: **local operations** $\hat{=}$  required only data from fixed “neighbourhood” of K
 computational effort “ $O(1)$ ”: independent of $\#\mathcal{M}$

Computational cost(Assembly of Galerkin matrix \mathbf{A}) = $O(\#\mathcal{M})$

Cell oriented assembly in LehrFEM [7, Sect. 5.1]

```
function A = assemble(Mesh)
for k = Mesh.Elements'
  idx = ①
  Aloc = ②
  A(idx, idx) = A(idx, idx) + Aloc;
end
```

① row vector of index numbers of global shape functions $b_N^{i_1}, \dots, b_N^{i_Q} \in V_N$ corresponding to local shape functions b_K^1, \dots, b_K^Q :

 $\text{idx} = (i_1, \dots, i_Q)$
 (encodes index mapping matrix \mathbf{T}_K)

② $Q \times Q$ element stiffness matrix

This code generalizes Code 3.2.14 for triangular linear Lagrangian finite elements, where the local→global index mapping could be inferred from the mesh data directly through the `idx`-array (\leftrightarrow `Elements`-array).

For Lagrangian FEM of fixed degree p (\rightarrow Sect. 3.4):

the total computational effort is of the order $O(\#\mathcal{M}) = O(N)$, $N := \dim \mathcal{S}_p^0(\mathcal{M})$.

Example 3.5.17 (Assembly for quadratic Lagrangian FE in MATLAB code).

Setting: FE space $\mathcal{S}_2^0(\mathcal{M})$ on triangular mesh \mathcal{M} of polygon $\Omega \subset \mathbb{R}^2$, see Ex. 3.4.2

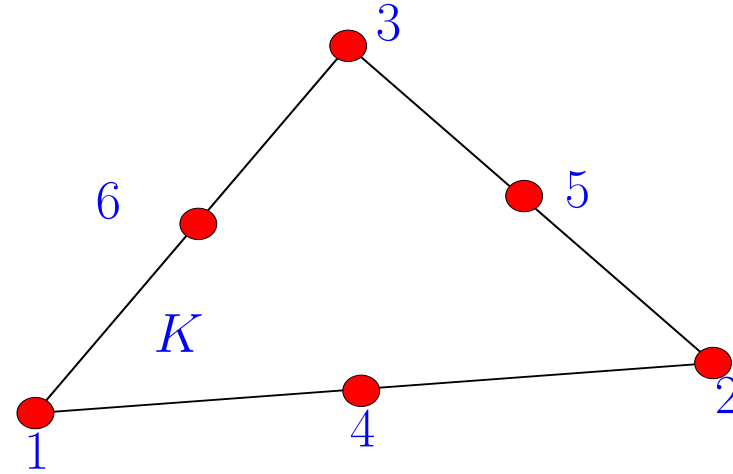
Recall: 6 local shape functions: 3 vertex-associated, 3 edge-associated \rightarrow (3.4.4)

Convention: vertex-associated global shape functions $\rightarrow b_N^1, \dots, b_N^{\#\mathcal{V}(\mathcal{M})}$
 edge-associated global shape functions $\rightarrow b_N^{\#\mathcal{V}(\mathcal{M})+1}, \dots, b_N^{\#\mathcal{V}(\mathcal{M})+\#\mathcal{E}(\mathcal{M})}$

Local numbering

$(1 \leftrightarrow \mathbf{a}^1, 2 \leftrightarrow \mathbf{a}^2, 3 \leftrightarrow \mathbf{a}^3)$

\rightarrow



①

```

function A = assemMat_QFE (Mesh, EHandle, varargin)

nV = size(Mesh.Coordinates,1);
nE = size(Mesh.Elements,1)

I ② = zeros(36*nE,1); J = I; a = I; offset = 0;
for k =1:nE
    vidx = Mesh.Elements(k,:)
    idx ③ = [vidx,...
            Mesh.Vert2Edge(vidx(1),vidx(2))+nV,...
            Mesh.Vert2Edge(vidx(2),vidx(3))+nV,...
            Mesh.Vert2Edge(vidx(3),vidx(1))+nV];
    Aloc ④ = transpose(EHandle(Mesh.Coordinates(vidx,:),...
                               Mesh.ElemFlag(k), varargin{:}));
    ⑤

    Qsq = prod(size(Aloc)); range = offset + 1:Qsq;
    t = idx(ones(length(idx),1),:)' ; I(range) = t(:);
    t = idx(ones(1,length(idx)),:); J(range) = t(:);
    a(range) = Aloc(:);
    offset = offset + Qsq;
end
A ⑥ = sparse(I, J, a);

```

- ①: `EHandle` (function handle) \rightarrow provides element stiffness matrix $\mathbf{A}_K \in \mathbb{R}^{6,6}$
- ②: `I`, `J`, `a` $\hat{=}$ linear arrays storing $(i, j, (\mathbf{A})_{ij})$ for stiffness matrix \mathbf{A} .
Initialized with 0 for the sake of efficiency \rightarrow Ex. 3.5.18
- ③: `idx` $\hat{=}$ index mapping vector, see ① above.
(`Mesh.Vert2Edge` $\hat{=}$ $\#\mathcal{V}(\mathcal{M}) \times \#\mathcal{V}(\mathcal{M})$ -sparse matrix providing the number of the connecting edge for each pair of vertices, see Ex. 3.5.7)
- ④: `Aloc` = $\mathbf{A}_K \in \mathbb{R}^{6,6}$ (element stiffness matrix \rightarrow Def. 3.5.12)
- ⑤: `Mesh.ElemFlag(k)` marks groups of elements (e.g. to select local coefficient function $\alpha(\mathbf{x})$ in (2.8.7))
- ⑥: Build *sparse* MATLAB-matrix (\rightarrow Def. 3.2.8) from index-entry arrays, see manual entry for MATLAB function `sparse` and [21, Sect. 2.6.2].

Remark 3.5.18 (Efficient implementation of assembly). → [21, Sect. 2.6.2]

tic-toc-timing (min of 4v runs), MATLAB V7, Intel Pentium 4 Mobile CPU 1.80GHz, Linux
Computation of element stiffness matrices skipped !

• *Sparse assembly:*

```
A(idx,idx) = A(idx,idx) + Aloc;
```

• *Array assembly I: "growing arrays"*

```
I = []; J = []; a = [];
...
t = idx(:,ones(length(idx),1))';
I = [I;t(:)];
t = idx(:,ones(1,length(idx)));
J = [J;t(:)];
a = [a; Aloc(:)];
```

• *Array assembly III*

→ see code fragment above

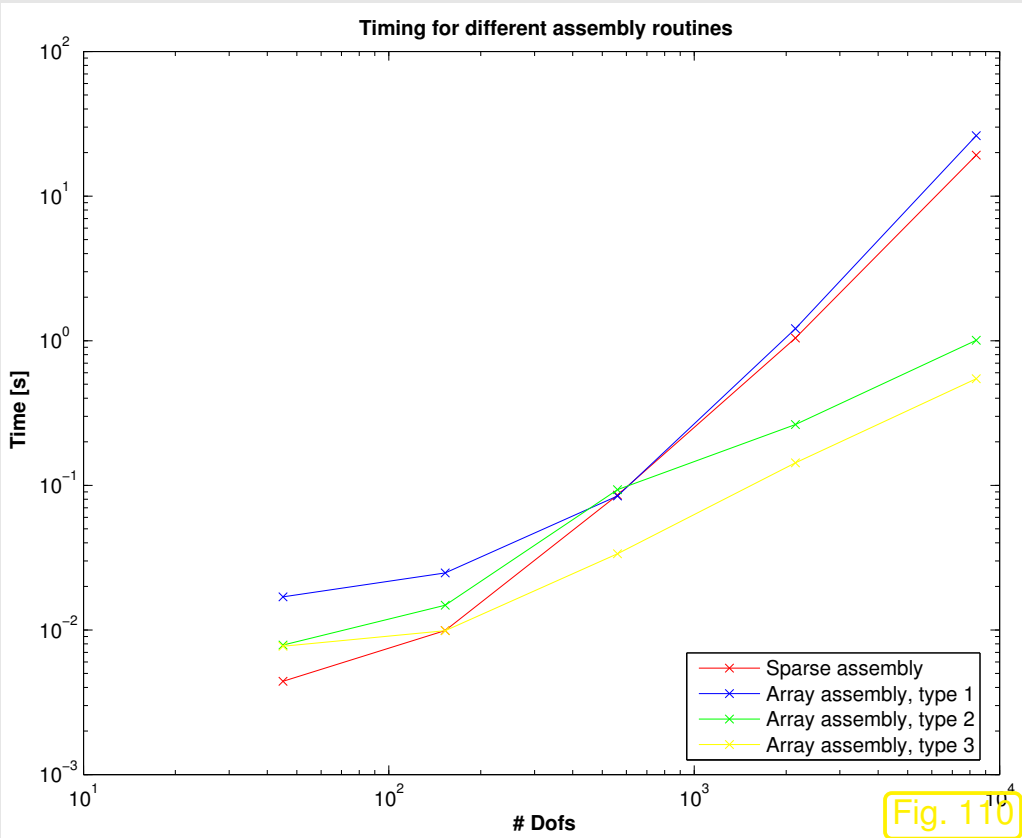


Fig. 110

More detailed discussion → [35] and [21, Sect. 2.6.2].



3.5.4 Local computations and quadrature

We have seen that the (global) Galerkin matrix and right hand side vector are conveniently generated by “assembling” entries of element (stiffness) matrices and element (load) vectors.

Now we study the computation of these local quantities, see also Sect. 3.2.5, 3.2.6.

First option:

analytic evaluations

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

We discuss bilinear form related to $-\Delta$, triangular Lagrangian finite elements of degree p , Sect. 3.4.1, Def. 3.4.1:

$$K \text{ triangle: } \mathbf{a}_K(u, v) := \int_K \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} \quad \blacktriangleright \quad \text{element stiffness matrix .}$$

Use **barycentric coordinate representations** of local shape functions, in 2D

$$b_K^i = \sum_{\alpha \in \mathbb{N}_0^3, |\alpha| \leq p} \kappa_\alpha \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3}, \quad \kappa_\alpha \in \mathbb{R}, \quad (3.5.19)$$

where λ_i are the affine linear barycentric coordinate functions (linear shape functions), see Fig. 72.

For the barycentric coordinate representation of the quadratic local shape functions see (3.4.4), for a justification of (3.5.19) consult Rem. 3.6.9.

$$\Rightarrow \mathbf{grad} b_K^i = \sum_{\alpha \in \mathbb{N}_0^3, |\alpha| \leq p} \kappa_\alpha \left(\begin{aligned} &\alpha_1 \lambda_1^{\alpha_1-1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3} \mathbf{grad} \lambda_1 + \alpha_2 \lambda_1^{\alpha_1} \lambda_2^{\alpha_2-1} \lambda_3^{\alpha_3} \mathbf{grad} \lambda_2 + \\ &\alpha_3 \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3-1} \mathbf{grad} \lambda_3 \end{aligned} \right). \quad (3.5.20)$$



To evaluate $\int_K \lambda_1^{\beta_1} \lambda_2^{\beta_2} \lambda_3^{\beta_3} \mathbf{grad} \lambda_i \cdot \mathbf{grad} \lambda_j \, d\mathbf{x}$, $i, j \in \{1, 2, 3\}$, $\beta_k \in \mathbb{N}$. (3.5.21)

If $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$ vertices of K (counterclockwise ordering):

$$\lambda_1(\mathbf{x}) = \frac{1}{2|K|} \left(\mathbf{x} - \begin{pmatrix} a_1^2 \\ a_2^2 \end{pmatrix} \right) \cdot \begin{pmatrix} a_2^3 - a_2^1 \\ a_1^3 - a_1^1 \end{pmatrix},$$

$$\lambda_2(\mathbf{x}) = \frac{1}{2|K|} \left(\mathbf{x} - \begin{pmatrix} a_1^3 \\ a_2^3 \end{pmatrix} \right) \cdot \begin{pmatrix} a_2^1 - a_2^2 \\ a_1^1 - a_1^2 \end{pmatrix},$$

$$\lambda_3(\mathbf{x}) = \frac{1}{2|K|} \left(\mathbf{x} - \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix} \right) \cdot \begin{pmatrix} a_2^2 - a_2^3 \\ a_1^2 - a_1^3 \end{pmatrix}.$$

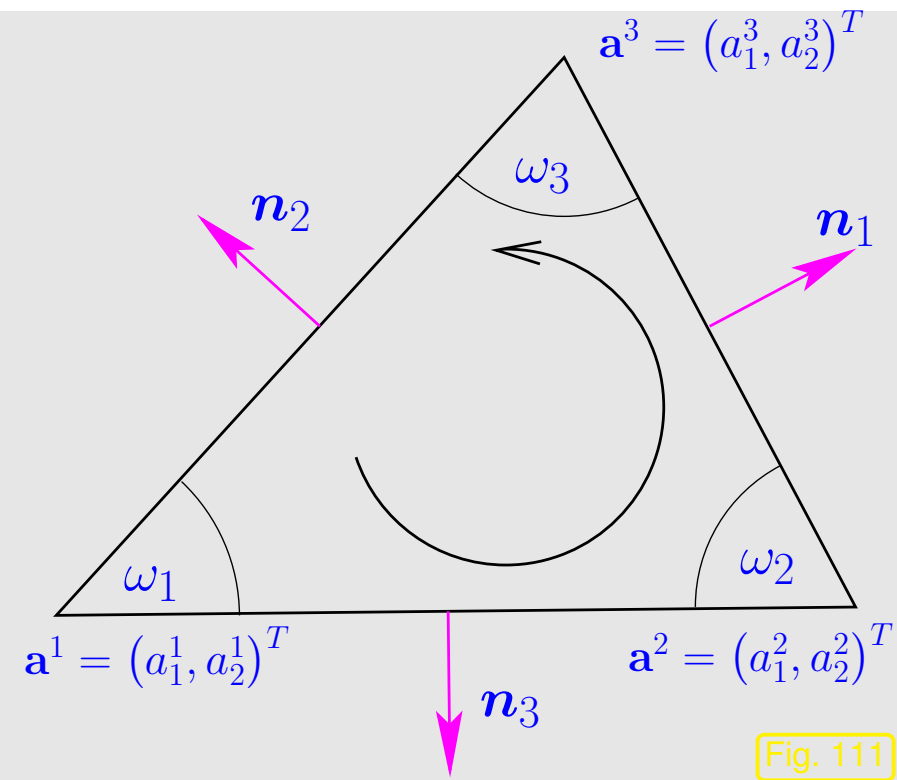


Fig. 111

$$\mathbf{grad} \lambda_1 = \frac{1}{2|K|} \begin{pmatrix} a_2^3 - a_2^1 \\ a_1^3 - a_1^1 \end{pmatrix}, \quad \mathbf{grad} \lambda_2 = \frac{1}{2|K|} \begin{pmatrix} a_2^1 - a_2^2 \\ a_1^1 - a_1^2 \end{pmatrix}, \quad \mathbf{grad} \lambda_3 = \frac{1}{2|K|} \begin{pmatrix} a_2^2 - a_2^3 \\ a_1^2 - a_1^3 \end{pmatrix}. \quad (3.5.22)$$

Lemma 3.5.23 (Integration of powers of barycentric coordinate functions).

For any non-degenerate d -simplex K and $\alpha_j \in \mathbb{N}$, $j = 1, \dots, d + 1$,

$$\int_K \lambda_1^{\alpha_1} \cdots \lambda_{d+1}^{\alpha_{d+1}} d\mathbf{x} = d!|K| \frac{\alpha_1! \alpha_2! \cdots \alpha_{d+1}!}{(\alpha_1 + \alpha_2 + \cdots + \alpha_{d+1} + d)!} \quad \forall \boldsymbol{\alpha} \in \mathbb{N}_0^{d+1}. \quad (3.5.24)$$

Proof for $d = 2$

Step #1: transformation $K \rightarrow$ “unit triangle” $\widehat{K} := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$,

$$\begin{aligned} \Rightarrow \int_K \lambda_1^{\beta_1} \lambda_2^{\beta_2} \lambda_3^{\beta_3} d\mathbf{x} &= 2|K| \int_0^1 \int_0^{1-\xi_1} \xi_1^{\beta_1} \xi_2^{\beta_2} (1 - \xi_1 - \xi_2)^{\beta_3} d\xi_2 d\xi_1 \\ &\stackrel{(*)}{=} 2|K| \int_0^1 \xi_1^{\beta_1} \int_0^1 (1 - \xi_1)^{\beta_2 + \beta_3 + 1} s^{\beta_2} (1 - s)^{\beta_3} ds d\xi_1 \\ &= 2|K| \int_0^1 \xi_1^{\beta_1} (1 - \xi_1)^{\beta_2 + \beta_3 + 1} d\xi_1 \cdot B(\beta_2 + 1, \beta_3 + 1) \end{aligned}$$

$$= 2|K| B(\beta_1 + 1, \beta_2 + \beta_3 + 2) \cdot B(\beta_2 + 1, \beta_3 + 1) ,$$

(*) $\hat{=}$ substitution $s(1 - \xi_1) = \xi_2$, $B(\cdot, \cdot) \hat{=}$ Euler's beta function

$$B(\alpha, \beta) := \int_0^1 t^{\alpha-1} (1-t)^{\beta-1} dt , \quad 0 < \alpha, \beta < \infty .$$

Using $\Gamma(\alpha + \beta) B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)$, $\Gamma \hat{=}$ Gamma function, $\Gamma(n) = (n-1)!$,

$$\Rightarrow \int_K \lambda_1^{\beta_1} \lambda_2^{\beta_2} \lambda_3^{\beta_3} d\mathbf{x} = 2|K| \cdot \frac{\Gamma(\beta_1 + 1)\Gamma(\beta_2 + 1)\Gamma(\beta_3 + 1)}{\Gamma(\beta_1 + \beta_2 + \beta_3 + 3)} \quad \square .$$

Remark. Alternative: **symbolic computing** (MAPLE, Mathematica) for local computations

Second option: **cell-based quadrature**

At this point turn the pages back to (1.5.87) and remember the use of numerical quadrature for computing the Galerkin matrix for the linear finite element method in 1D.

Reminder: numerical quadrature mandatory in the presence of coefficients/source terms in *procedural form* \rightarrow Rem. 1.5.6.

Local quadrature formula, *cf.* (3.2.18)

$$\int_{\Omega} f(\mathbf{x}) \, d\mathbf{x} \approx \sum_{K \in \mathcal{M}} |K| \sum_{l=1}^P \omega_l^K f(\zeta_l^K), \quad \zeta_l^K \in K, \omega_l^K \in \mathbb{R}, \quad P \in \mathbb{N}. \quad (3.5.25)$$

Terminology:

$$\omega_l^K \rightarrow \text{weights}, \quad \zeta_l^K \rightarrow \text{quadrature nodes}$$

(3.5.25) = P -point local quadrature rule

- Mandatory
- for computation of load vector (f complicated/only available in procedural form, Rem. 1.5.6,)
 - for computation of stiffness matrix, if $\alpha = \alpha(\mathbf{x})$ does not permit analytic integration.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Example for local quadrature rule: 2D trapezoidal rule from (3.2.18)

Guideline [21, Sect. 10.2]: only quadrature rules with positive weights are numerically stable.

How to gauge the quality of parametric local quadrature rules ? \rightarrow [21, Sect. 10.3]

Quality of a parametric local quadrature rule on $K \sim$ maximal degree of polynomials (multivariate \rightarrow Def. 3.3.3, or tensor product \rightarrow Def. 3.3.7) on K integrated exactly by the corresponding quadrature rule on K .

Parlance: Quadrature rule exact for $\mathcal{P}_p(\mathbb{R}^d) \Rightarrow$ quadrature rule of order $p + 1$
degree of exactness p

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs


SAM, ETHZ

How are quadrature rules specified for the many different cells of a finite element mesh ?

Remark 3.5.26 (Affine transformation of triangles).

Definition 3.5.27 (Affine (linear) transformation).

Mapping $\Phi : \mathbb{R}^d \mapsto \mathbb{R}^d$ is *affine (linear)*, if $\Phi(\mathbf{x}) = \mathbf{F}\mathbf{x} + \boldsymbol{\tau}$ with some $\mathbf{F} \in \mathbb{R}^{d,d}$, $\boldsymbol{\tau} \in \mathbb{R}^d$.

 notation: ‘unit triangle’ $\hat{K} := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$

Lemma 3.5.28 (Affine transformation of triangles).

For any non-degenerate triangle $K \subset \mathbb{R}^2$ ($|K| > 0$) there is a unique affine transformation Φ_K , $\Phi_K(\hat{\mathbf{x}}) = \mathbf{F}_K \hat{\mathbf{x}} + \boldsymbol{\tau}_K$ (\rightarrow Def. 3.5.27), with $K = \Phi(\hat{K})$.

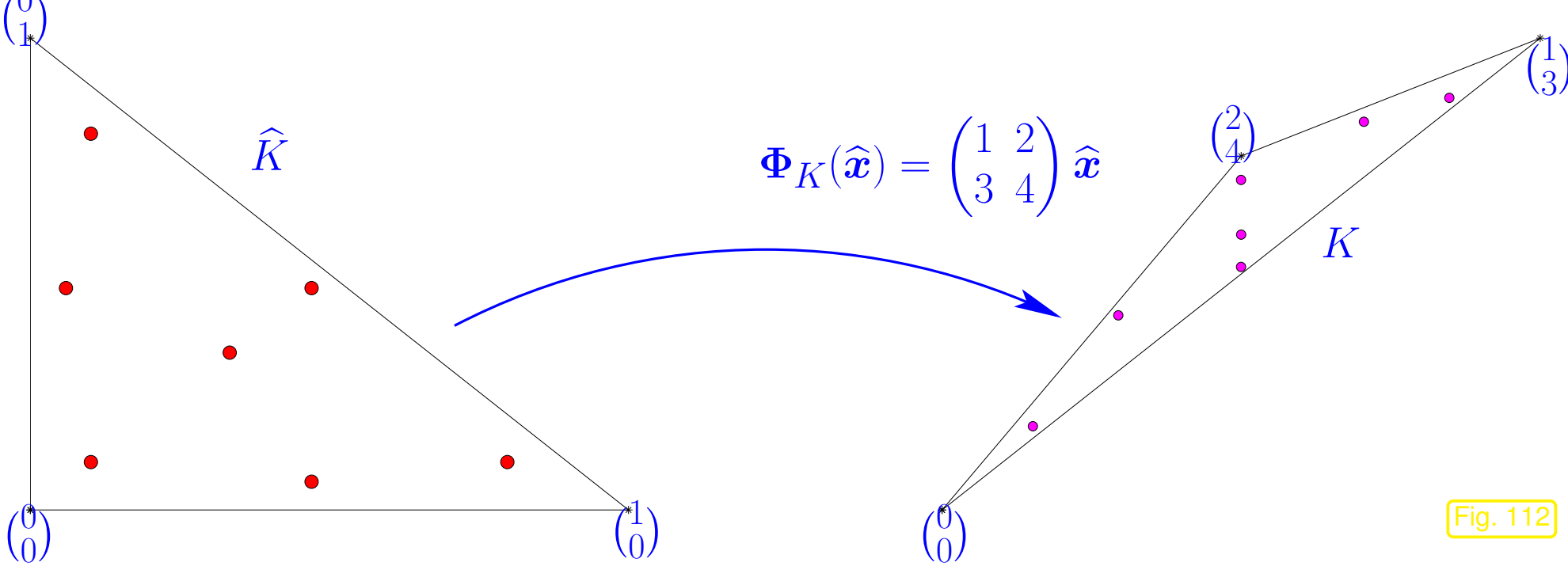


Fig. 112

Formula:

$$K = \text{convex} \left\{ \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix}, \begin{pmatrix} a_1^2 \\ a_2^2 \end{pmatrix}, \begin{pmatrix} a_1^3 \\ a_2^3 \end{pmatrix} \right\} \Rightarrow \Phi_K(\hat{\mathbf{x}}) = \begin{pmatrix} a_1^2 - a_1^1 & a_1^3 - a_1^1 \\ a_2^2 - a_2^1 & a_2^3 - a_2^1 \end{pmatrix} \hat{\mathbf{x}} + \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix}. \tag{3.5.29}$$

Note that

$$|K| = \frac{1}{2} |\det \mathbf{F}_K|.$$

Remark 3.5.30 (Transformation of local quadrature rules on triangles).

$\Phi_K(\hat{\mathbf{x}}) := \mathbf{F}_K \hat{\mathbf{x}} + \boldsymbol{\tau}_K \hat{=} \text{affine transformation}$ (\rightarrow Def. 3.5.27) mapping \hat{K} to triangle K , see Lemma 3.5.28.

By transformation formula for integrals [32, Satz 8.5.2]

$$\int_K f(\mathbf{x}) \, d\mathbf{x} = \int_{\hat{K}} f(\Phi_K(\hat{\mathbf{x}})) |\det \mathbf{F}_K| \, d\hat{\mathbf{x}} . \tag{3.5.31}$$

P -point quadrature formula on \hat{K} \blacktriangleright P -point quadrature formula on K

$$\int_{\hat{K}} f(\hat{\mathbf{x}}) \, d\hat{\mathbf{x}} \approx |\hat{K}| \sum_{l=1}^P \hat{\omega}_l f(\hat{\boldsymbol{\zeta}}_l) \quad \blacktriangleright \quad \int_{\Omega} f(\mathbf{x}) \, d\mathbf{x} \approx \sum_{K \in \mathcal{M}} |K| \sum_{l=1}^P \omega_l^K f(\boldsymbol{\zeta}_l^K) \tag{3.5.32}$$

with $\omega_l^K = \hat{\omega}_l, \boldsymbol{\zeta}_l^K = \Phi_K(\hat{\boldsymbol{\zeta}}_l)$.

- Only quadrature formula (3.5.25) on unit triangle \hat{K} needs to be specified!
(The same applies to tetrahedra, where affine mappings for $d = 3$ are used.)

Since the space $\mathcal{P}_p(\mathbb{R}^d)$ is *invariant* under affine mappings,

$$q \in \mathcal{P}_p(\mathbb{R}^d) \Rightarrow \hat{\mathbf{x}} \mapsto q(\Phi(\hat{\mathbf{x}})) \in \mathcal{P}_p(\mathbb{R}^d) \quad \text{for any affine transformation } \Phi, \quad (3.5.33)$$

the orders of the quadrature rules on the left and right hand side of (3.5.31) agree.



Example 3.5.34 (Useful quadrature rules on triangles). → [7, Sect. 3.3.2]

Specification of quadrature rule for “unit triangle” $\hat{K} := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$.

Quadrature rules described by pairs $(\hat{\omega}_1, \hat{\zeta}_1), \dots, (\hat{\omega}_P, \hat{\zeta}_P)$, $P \in \mathbb{N}$.

- Quadrature rule of order 2 (exact for $\mathcal{P}_1(\hat{K})$)

$$\left\{ \left(\frac{1}{3}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right), \left(\frac{1}{3}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right), \left(\frac{1}{3}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \right\} . \quad (3.5.35)$$

- Quadrature rule of order 3 (exact for $\mathcal{P}_2(\hat{K})$)

$$\left\{ \left(\frac{1}{3}, \begin{pmatrix} 1/2 \\ 0 \end{pmatrix} \right), \left(\frac{1}{3}, \begin{pmatrix} 0 \\ 1/2 \end{pmatrix} \right), \left(\frac{1}{3}, \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \right) \right\} . \quad (3.5.36)$$

- One-point quadrature rule of order 2 (exact for $\mathcal{P}_1(\hat{K})$)

$$\left\{ \left(1, \begin{pmatrix} 1/3 \\ 1/3 \end{pmatrix} \right) \right\} . \quad (3.5.37)$$

- Quadrature rule of order 6 (exact for $\mathcal{P}_5(\hat{K})$)

$$\left\{ \left(\frac{9}{40}, \begin{pmatrix} 1/3 \\ 1/3 \end{pmatrix} \right), \left(\frac{155 + \sqrt{15}}{1200}, \begin{pmatrix} 6 + \sqrt{15}/21 \\ 6 + \sqrt{15}/21 \end{pmatrix} \right), \left(\frac{155 + \sqrt{15}}{1200}, \begin{pmatrix} 9 - 2\sqrt{15}/21 \\ 6 + \sqrt{15}/21 \end{pmatrix} \right), \right. \\ \left. \left(\frac{155 + \sqrt{15}}{1200}, \begin{pmatrix} 6 + \sqrt{15}/21 \\ 9 - 2\sqrt{15}/21 \end{pmatrix} \right), \left(\frac{155 - \sqrt{15}}{1200}, \begin{pmatrix} 6 - \sqrt{15}/21 \\ 9 + 2\sqrt{15}/21 \end{pmatrix} \right), \right. \\ \left. \left(\frac{155 - \sqrt{15}}{1200}, \begin{pmatrix} 6 - \sqrt{15}/21 \\ 6 + \sqrt{15}/21 \end{pmatrix} \right), \left(\frac{155 - \sqrt{15}}{1200}, \begin{pmatrix} 9 + 2\sqrt{15}/21 \\ 6 + \sqrt{15}/21 \end{pmatrix} \right) \right\} , \quad (3.5.38)$$

In [13]: quadrature rules up to order $p = 21$ with $P \leq 1/6p(p + 1) + 5$

Remark 3.5.39 (Numerical quadrature in LehrFEM). \rightarrow [7, Sect. 3]

Routines return P -point quadrature formulas for

$$\hat{K} = \begin{cases} \text{unit triangle} & \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} & \text{for triangular cell,} \\ \text{unit square} & \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} & \text{for rectangular cell,} \end{cases}$$

in MATLAB *structure* `QuadRule` with fields

`QuadRule.w`: weights $\hat{\omega}_l$ of quadrature rule on \hat{K} ,

`QuadRule.x`: coordinates of nodes $\hat{\zeta}_l \in \hat{K}$ of quadrature rule on \hat{K}

For triangles: `QuadRule = PnOq()`, $\hat{=}$ n -point quadrature of order q

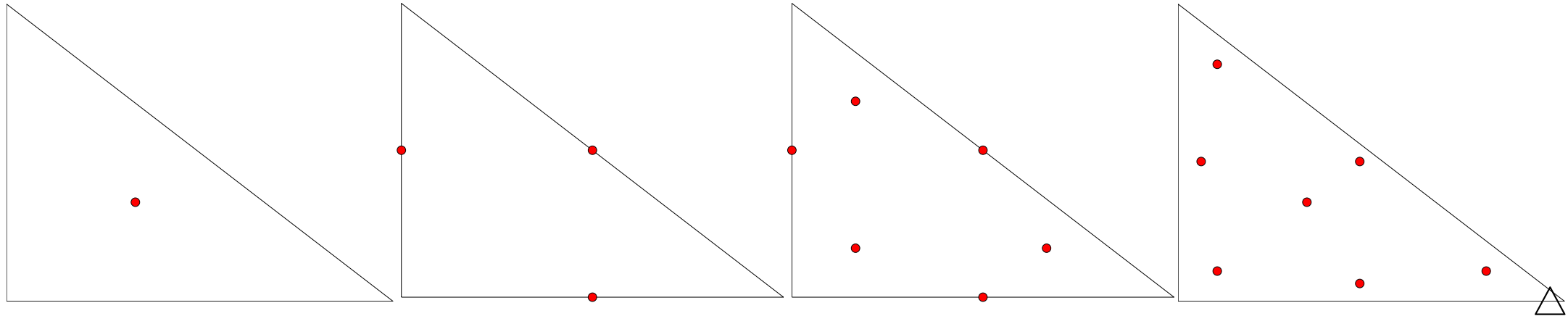
Location of quadrature nodes $\hat{\zeta}_l$ in unit triangle \hat{K} :

Quadrature rule P1O2

Quadrature rule P3O3

Quadrature rule P6O4

Quadrature rule P7O6



Example 3.5.40 (Local quadrature rules on quadrilaterals).

If K quadrilateral $\Rightarrow \hat{K} := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$ (unit square).

On \hat{K} : **tensor product construction:**

If $\{(\omega_1, \zeta_1), \dots, (\omega_P, \zeta_P)\}$, $P \in \mathbb{N}$, quadrature rule on the interval $]0, 1[$, exact for $\mathcal{P}_p]0, 1[$, then

$$\left\{ \begin{array}{ccc} (\omega_1^2, (\zeta_1)) & \cdots & (\omega_1 \omega_P, (\zeta_1)) \\ \vdots & & \vdots \\ (\omega_1 \omega_P, (\zeta_1^P)) & \cdots & (\omega_P^2, (\zeta_P)) \end{array} \right\}$$

provides a quadrature rule on the unit square \hat{K} , exact for $\mathcal{Q}_p(\hat{K})$.

Quadrature rules on $]0, 1[$ (\rightarrow [21, Ch. 10]):

- classical **Newton-Cotes formulas** (equidistant quadrature nodes).
- **Gauss-Legendre quadrature rules**, exact for $\mathcal{P}_{2P}(]0, 1[)$ using only P nodes.
- **Gauss-Lobatto quadrature rules**: P nodes including $\{0, 1\}$, exact for $\mathcal{P}_{2P-1}(]0, 1[)$.

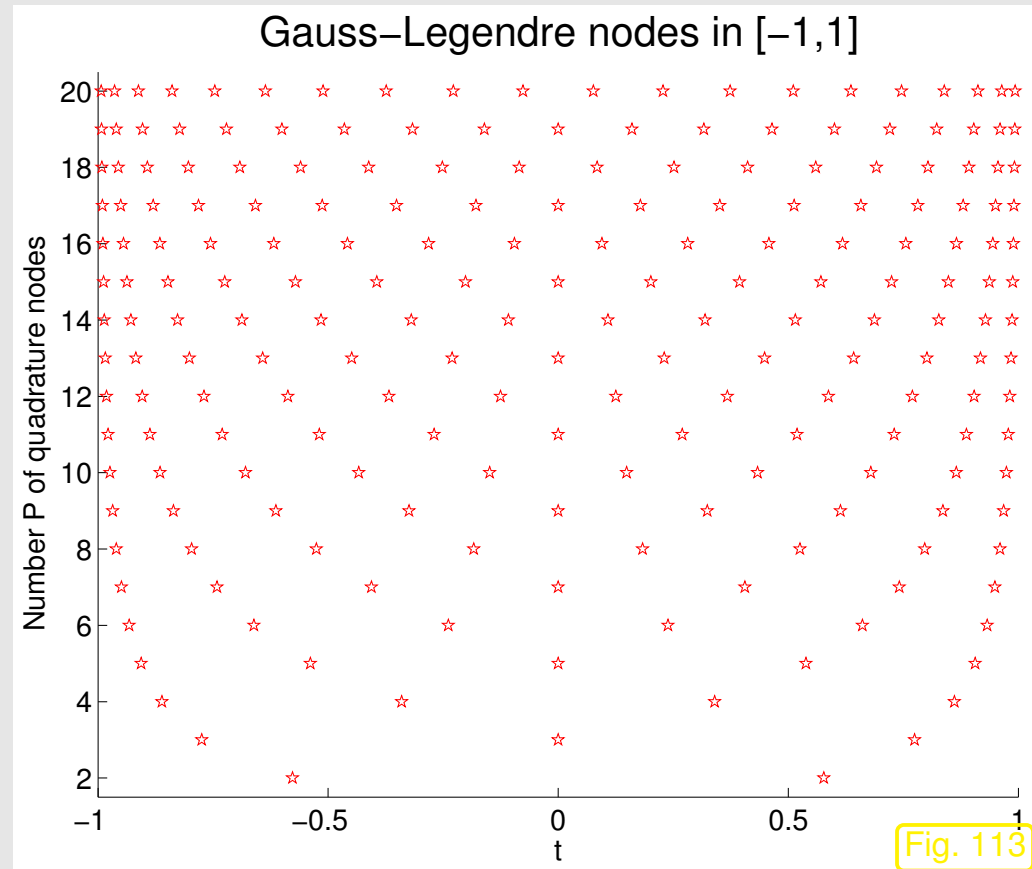


Fig. 113

3.5.5 Incorporation of essential boundary conditions

Recall variational formulation of *non-homogeneous* Dirichlet boundary value problem from Ex. 2.8.1:

$$\begin{aligned} u \in H^1(\Omega) \\ u = g \text{ on } \partial\Omega \end{aligned} : \quad \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) . \quad (2.8.7)$$

$$\Downarrow$$
$$-\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) = f \quad \text{in } \Omega \quad , \quad u = g \quad \text{on } \partial\Omega \quad ,$$

with (admissible \rightarrow Rem. 2.9.7) Dirichlet data $g \in C^0(\partial\Omega)$.

Recall from Sect. 2.9: Dirichlet b.c. = essential boundary conditions
(built into trial space)

Remember offset function technique, see (1.3.32) and Sect. 2.1.3:

$$\begin{aligned} (2.8.7) \quad \Leftrightarrow \quad u = u_0 + w \quad , \\ w \in H_0^1(\Omega) : \quad \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} w \cdot \mathbf{grad} v \, d\mathbf{x} \\ = \int_{\Omega} -\kappa(\mathbf{x}) \mathbf{grad} u_0 \cdot \mathbf{grad} v + f v \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) \quad , \end{aligned} \quad (3.5.41)$$

where

$$u_0 = g \text{ on } \partial\Omega$$

Adapt this to finite element Galerkin discretization by generalizing the 1D example Rem. 1.5.90 to $d = 2, 3$:

Remember: we already know finite element subspaces $V_{0,N} := \mathcal{S}_{p,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$, see Rem. 3.4.12.

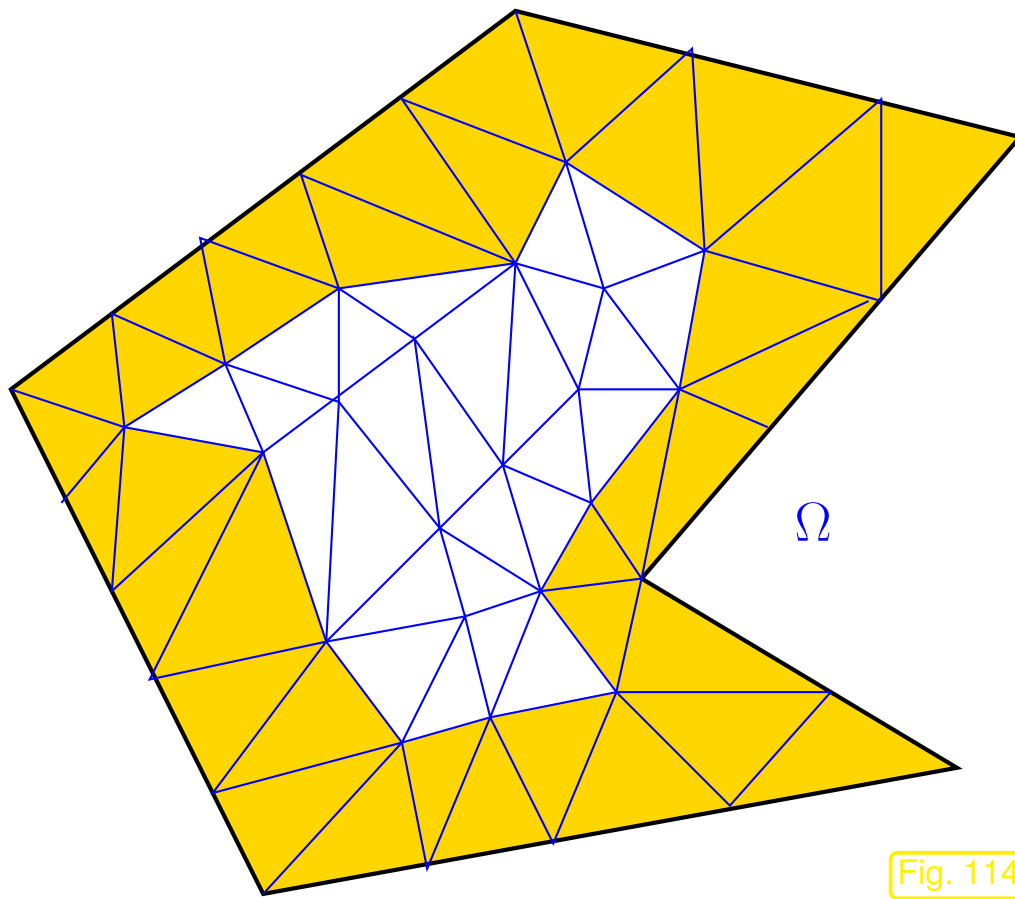
Idea from Rem. 1.5.90:

use offset function $u_0 \in V_N := \mathcal{S}_p^0(\mathcal{M})$
locally supported near the boundary:

$$\text{supp}(u_0) \subset \bigcup \{K \in \mathcal{M} : \overline{K} \cap \partial\Omega \neq \emptyset\}. \quad (3.5.42)$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



◁ Maximal support of u_0 on triangular mesh.

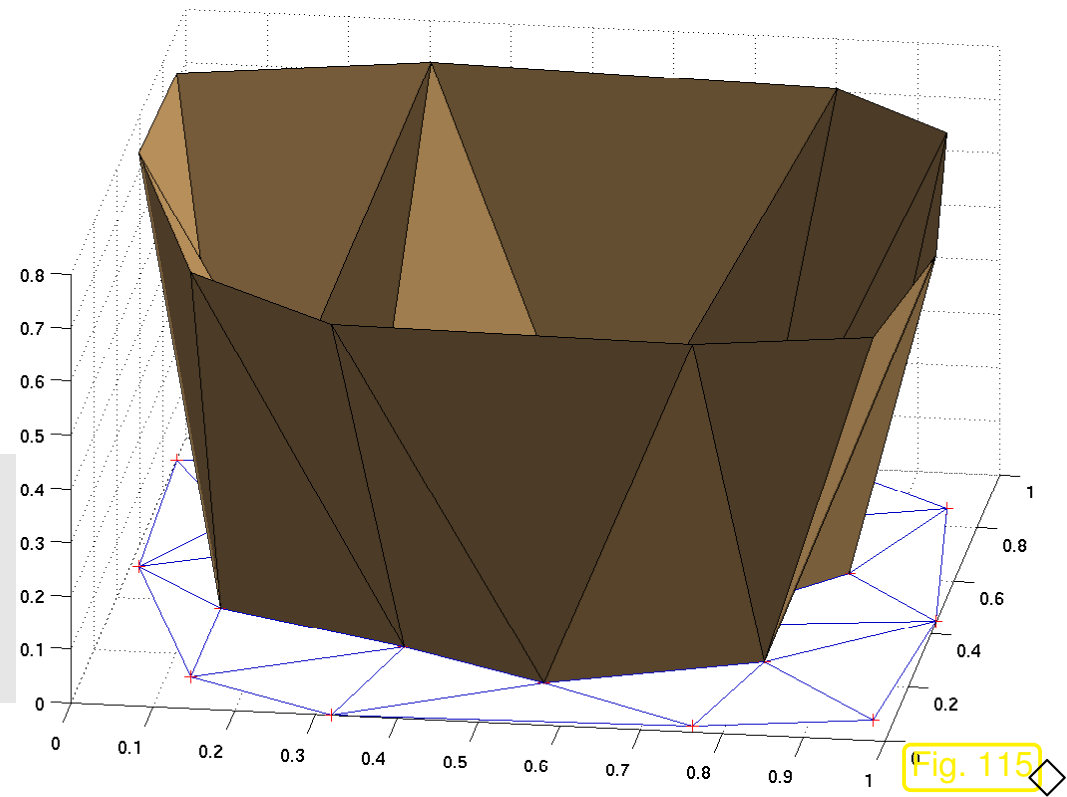
Fig. 114

Example 3.5.43 (offset functions for linear Lagrangian FE).

For Dirichlet data $g \in C^0(\partial\Omega)$

$$u_0 = \sum_{x \in \mathcal{V}(\mathcal{M}) \cap \partial\Omega} g(x) b_N^x \quad (3.5.44)$$

$b_N^x \hat{=}$ tent function associated with node $x \in \mathcal{V}(\mathcal{M})$, cf. Sect. 3.2.3. (3.5.44) generalizes (1.5.91) to 2D.



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 3.5.45 (Approximate Dirichlet boundary conditions).

Be aware that for the choice (3.5.44)

$$u_0 \neq g \quad \text{on } \partial\Omega .$$

Rather, u_0 is a *piecewise linear interpolant* of the Dirichlet data $g \in C^0(\partial\Omega)$. Therefore, another *approximation* comes into play when enforcing Dirichlet boundary conditions by means of piecewise

Example 3.5.46 (Implementation of non-homogeneous Dirichlet b.c. for linear FE).

Consider (2.8.7) and assume the following ordering of the nodal basis functions, see Fig. 67

$$\begin{aligned} \mathfrak{B}_0 &:= \{b_N^1, \dots, b_N^N\} && \hat{=} \text{nodal basis of } \mathcal{S}_{1,0}^0(\mathcal{M}), \\ & && \text{(tent functions associated with interior nodes)} \\ \mathfrak{B} &:= \mathfrak{B}_0 \cup \{b_N^{N+1}, \dots, b_N^M\} && \hat{=} \text{nodal basis of } \mathcal{S}_1^0(\mathcal{M}) \\ & && \text{(extra basis functions associated with nodes } \in \partial\Omega\text{)}. \end{aligned}$$

Note: $M = \#\mathcal{V}(\mathcal{M})$, $N = \#\{\mathbf{x} \in \mathcal{V}(\mathcal{M}), \mathbf{x} \notin \partial\Omega\}$ (no. of interior nodes)

$$\begin{aligned} \mathbf{A}_0 &\in \mathbb{R}^{N,N} && \hat{=} \text{Galerkin matrix for discrete trial/test space } \mathcal{S}_{1,0}^0(\mathcal{M}), \\ \mathbf{A} &\in \mathbb{R}^{M,M} && \hat{=} \text{Galerkin matrix for discrete trial/test space } \mathcal{S}_1^0(\mathcal{M}). \end{aligned}$$

$$\blacktriangleright \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_{0\partial} \\ \mathbf{A}_{0\partial}^T & \mathbf{A}_{\partial\partial} \end{pmatrix}, \quad \mathbf{A}_{0\partial} \in \mathbb{R}^{N, M-N}, \quad \mathbf{A}_{\partial\partial} \in \mathbb{R}^{M-N, M-N}. \quad (3.5.47)$$

If $u_0 \in \mathcal{S}_1^0(\mathcal{M})$ is chosen according to (3.5.44), then

$$u_0 \in \text{Span} \{b_N^{N+1}, \dots, b_N^M\} \Leftrightarrow u_0 = \sum_{j=N+1}^M \gamma_{j-N} b_N^j,$$

which means that the coefficient vector \vec{v} of the finite element approximation $w_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$ of $w \in H_0^1(\Omega)$ from (3.5.41) solves the linear system of equations

$$\boxed{\mathbf{A}_0 \vec{v} = \vec{\varphi} - \mathbf{A}_{0\partial} \vec{\gamma}}. \quad (3.5.48)$$

➤ Non-homogeneous Dirichlet boundary data are taken into account through a **modified right hand side vector**.

Alternative consideration leading to (3.5.48):

❶ First ignore essential boundary conditions and assemble the linear system of equations arising

from the discretization of \mathbf{a} on the (larger) FE space $\mathcal{S}_1^0(\mathcal{M})$:

$$\begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_{0\partial} \\ \mathbf{A}_{0\partial}^T & \mathbf{A}_{\partial\partial} \end{pmatrix} \begin{pmatrix} \vec{\mu}_0 \\ \vec{\mu}_\partial \end{pmatrix} = \begin{pmatrix} \vec{\varphi} \\ \vec{\varphi}_\partial \end{pmatrix}. \quad (3.5.49)$$

Here, $\vec{\mu}_0 \hat{=}$ coefficients for *interior* basis functions b_N^1, \dots, b_N^N

$\vec{\mu}_\partial \hat{=}$ coefficient for basis functions b_N^{N+1}, \dots, b_N^M for basis functions associated with nodes $\in \partial\Omega$.

② We realize that the coefficient vector of (3.5.49) is that of a FE approximation of u



$\vec{\mu}_\partial$ known = values of g at boundary nodes: $\vec{\mu}_\partial = \vec{\gamma}$

③ Moving known quantities in (3.5.49) to the right hand side yields (3.5.48).



Example 3.5.50 (Non-homogeneous Dirichlet boundary conditions in LehrFEM).

Code 3.5.51: Solving 2nd-order Dirichlet BVP with linear FE in LehrFEM

```
1 % Initialize constants and functions
2 F_HANDLE = @f_LShap;           % Right hand side source term
3 GD_HANDLE = @g_D_LShap;       % Dirichlet boundary data
4 % Load mesh
```

```
5 Mesh = load_Mesh('meshvert.dat', 'meshel.dat');
6 Mesh.ElemFlag = ones(size(Mesh.Elements,1),1);
7 Mesh = add_Edges(Mesh);
8 Loc = get_BdEdges(Mesh); % Obtain indices of edges on  $\partial\Omega$ 
9 Mesh.BdFlags = zeros(size(Mesh.Edges,1),1);
10 Mesh.BdFlags(Loc) = 1; % Set flag '1' for edges on the boundary
11
12 % Assemble Galerkin (stiffness) matrix and right hand side (load) vector for
   linear Lagrangian FE
13 A = assemMat_LFE(Mesh, @STIMA_Lapl_LFE);
14 % Note: the arguments 0,1,2 are passed on to F_HANDLE through
15 % varargin and they are specific to this particular example.
16 phi = assemLoad_LFE(Mesh, P706(), F_HANDLE, 0, 1, 2);
17
18 % Incorporate Dirichlet boundary data for vertices adjacent to edges
19 % carrying flag 1. U contains nodal values for Dirichlet
20 % boundary data, FreeDofs contains numbers of interior nodes.
21 % See Ex. 3.5.46 for further explanations.
22 [U, FreeDofs] = assemDir_LFE(Mesh, [1], GD_HANDLE);
23 phi = phi - A*U; %
24
25 % Solve the linear system with matrix  $A_0$  from (3.5.49)
26 U(FreeDofs) = A(FreeDofs, FreeDofs) \ phi(FreeDofs);
27
28 % Plot solution
```

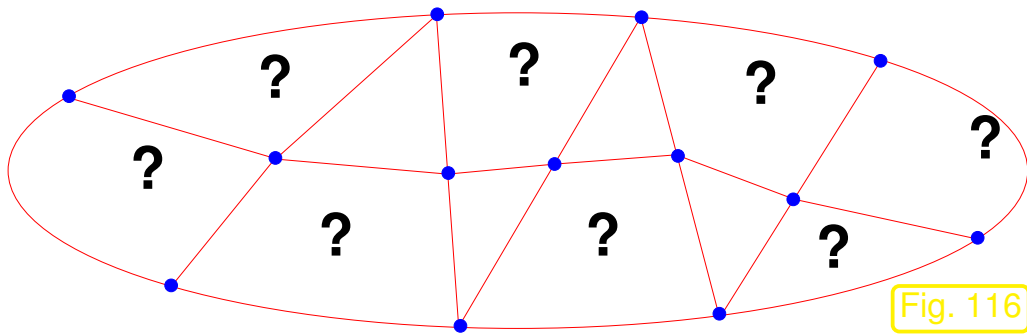
```
29 plot_LFE(U, Mesh); colorbar;  
30 plotLine_LFE(U, Mesh, [0 0] , [1 1]);
```

Line 23: see (3.5.48), but take note of some special conventions:

- U corresponds to the vector $\begin{pmatrix} 0 \\ \vec{\mu}_\partial \end{pmatrix}$.
- A represents the Galerkin matrix on the (larger) FE space $\mathcal{S}_1^0(\mathcal{M})$.



3.6 Parametric finite elements



◁ 2D hybrid mesh \mathcal{M} with (curvilinear) triangles and quadrilaterals

How to build $\mathcal{S}_1^0(\mathcal{M})$?

3.6.1 Affine equivalence

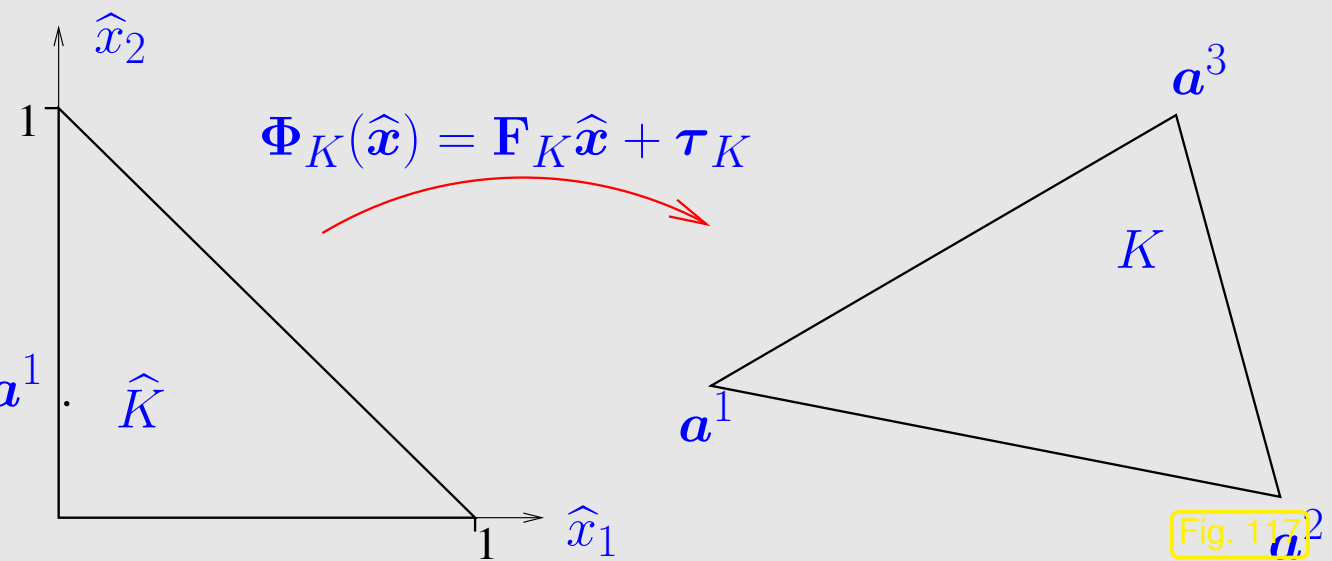
Recall Lemma 3.5.28: affine transformation of triangles (3.5.29)

▶ All cells of a triangular mesh are affine images of “unit triangle” \hat{K}

“Unit triangle”: $\widehat{K} = \left\langle \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle$

For $K = \text{convex} \{ \mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3 \}$:

$$\mathbf{F}_K = \begin{pmatrix} a_1^2 - a_1^1 & a_1^3 - a_1^1 \\ a_2^2 - a_2^1 & a_2^3 - a_2^1 \end{pmatrix}, \quad \boldsymbol{\tau}_K = \mathbf{a}^1.$$



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 3.6.1 (Pullback of functions).

In a natural way, a transformation of domains induces a transformation of the functions defined on them:

Definition 3.6.2 (Pullback).

Given domains $\Omega, \widehat{\Omega} \subset \mathbb{R}^d$ and a bijective mapping $\Phi : \widehat{\Omega} \mapsto \Omega$, the **pullback** $\Phi^*u : \widehat{\Omega} \mapsto \mathbb{R}$ of a function $u : \Omega \mapsto \mathbb{R}$ is a function on $\widehat{\Omega}$ defined by

$$(\Phi^*u)(\widehat{\mathbf{x}}) := u(\Phi(\widehat{\mathbf{x}})), \quad \widehat{\mathbf{x}} \in \widehat{\Omega}.$$

- Implicitly, we used the pullback of integrands when defining quadrature rules through transformation, see (3.5.31).
- Obviously, the pullback Φ^* induces a *linear mapping* between spaces of functions on Ω and $\hat{\Omega}$, respectively.

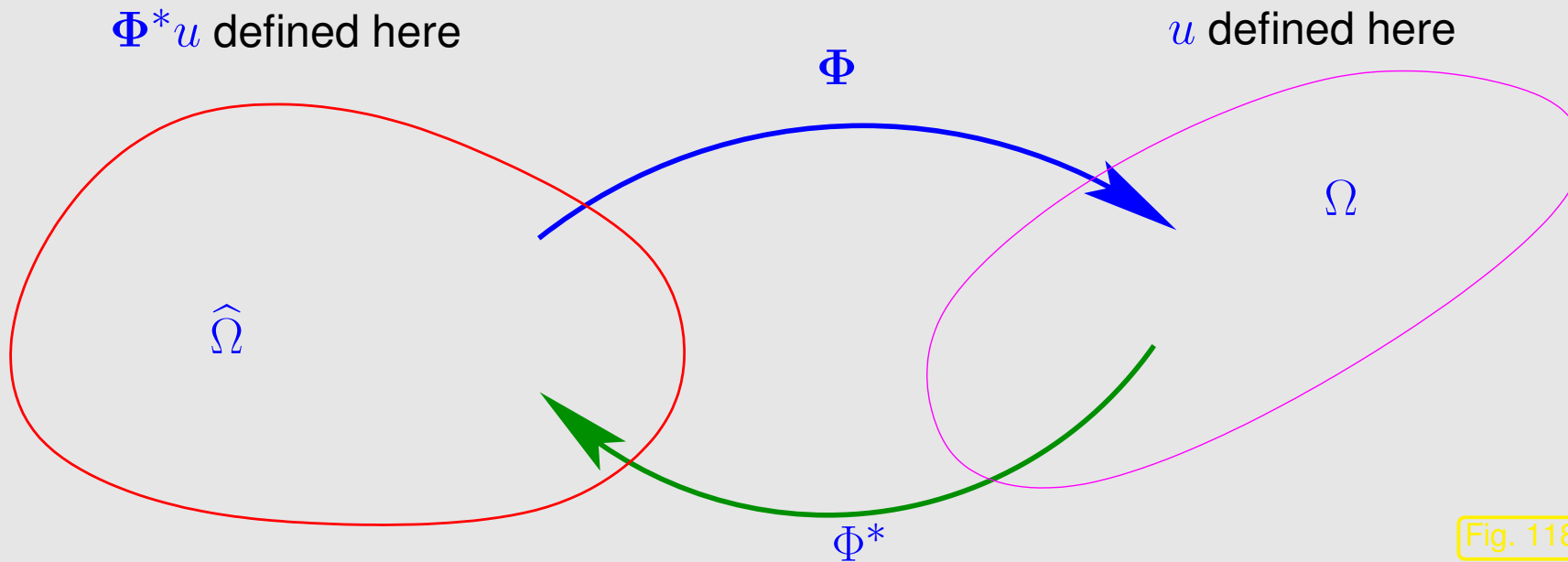


Fig. 118



Lemma 3.6.3 (Preservation of polynomials under affine pullback).

If $\Phi : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an affine (linear) transformation (\rightarrow Def. 3.5.27), then

$$\Phi^*(\mathcal{P}_p(\mathbb{R}^d)) = \mathcal{P}_p(\mathbb{R}^d) \quad \text{and} \quad \Phi^*(\mathcal{Q}_p(\mathbb{R}^d)) = \mathcal{Q}_p(\mathbb{R}^d) .$$

In fact, Lemma 3.5.28 reveals another reason for the preference for polynomials in building discrete Galerkin spaces.

Proof. (of Lemma 3.5.28)

Since the pullback is linear, we only need to study its action on the (monomial) basis $\mathbf{x} \mapsto \mathbf{x}^\alpha$, $\alpha \in \mathbb{N}_0^d$ of $\mathcal{P}_p(\mathbb{R}^d)$, see Def. 3.3.3 and the explanations on multi-index notation (3.3.4).

Then resort to induction w.r.t. degree p .

$$\Phi_K^*(\mathbf{x}^\alpha) = \Phi_K^*(x_1) \cdot \underbrace{\Phi_K^*\left(\underbrace{\mathbf{x}^{\alpha'}}_{\in \mathcal{P}_{p-1}(\mathbb{R}^d)}\right)}_{\in \mathcal{P}_1(\mathbb{R}^d)} = \underbrace{\left(\sum_{l=1}^d (\mathbf{F})_{1l} \hat{x}_l + \tau_1\right)}_{\in \mathcal{P}_1(\mathbb{R}^d)} \cdot \underbrace{\Phi_K^*(\mathbf{x}^{\alpha'})}_{\in \mathcal{P}_{p-1}(\mathbb{R}^d)} \in \mathcal{P}_p(\mathbb{R}^d) ,$$

with $\alpha' := (\alpha_1 - 1, \alpha_2, \dots, \alpha_d)$, where we assumed $\alpha_1 > 0$. Here, we have used the induction hypothesis to conclude $\Phi_K^*(\mathbf{x}^{\alpha'}) \in \mathcal{P}_{p-1}(\mathbb{R}^d)$. \square

A simple observation:

Consider $\mathcal{S}_1^0(\mathcal{M})$, triangle $K \in \mathcal{M}$, unit triangle \hat{K} , affine mapping $\Phi_K : \hat{K} \mapsto K$

- b_K^1, b_K^2, b_K^3 (standard) local shape functions on K ,
 - $\hat{b}^1, \hat{b}^2, \hat{b}^3$ (standard) local shape functions on \hat{K} ,
- Ex. 3.3.13

$$\hat{b}^i = \Phi_K^* b_K^i \iff \hat{b}^i(\hat{\mathbf{x}}) = b_K^i(\mathbf{x}), \quad \mathbf{x} = \Phi_K(\hat{\mathbf{x}}) \quad (3.6.4)$$

Of course, we assume that Φ_K respects the local numbering of the vertices of \hat{K} and K .

The proof of (3.6.4) is straightforward: both $\Phi_K^* b_K^i$ (by Lemma 3.6.3) and \hat{b}^i are (affine) linear functions that attain the same values at the vertices of \hat{K} . Hence, they have to agree.

Note: (3.6.4) holds true for *all* simplicial Lagrangian finite element spaces

Proof. (of (3.6.4)) Recall the definition of global shape functions and also local shape functions for $\mathcal{S}_p^0(\mathcal{M})$, $p \in \mathbb{N}$, by means of the conditions (3.4.3) at interpolation nodes, see Ex. 3.4.2 for $p = 2$. \square

Note: we already used the definition of basis functions through basis functions on the “reference cell” $[0, 1]$ and affine pullback in 1D, see Rem. 1.5.48

Now write $\mathbf{p}_K^i \hat{=} (\text{local})$ interpolation nodes on triangle K ,
 $\hat{\mathbf{p}}^i \hat{=} (\text{local})$ interpolation nodes on unit triangle \hat{K} .

Observe: Assuming a matching numbering $\mathbf{p}_K^i = \Phi_K(\hat{\mathbf{p}}^i)$. where $\Phi_K : \hat{K} \mapsto K$ is the unique affine transformation mapping \hat{K} onto K , see (3.5.29).

This is clear for $p = 2$, because affine transformations take midpoints of edges to midpoints of edges. The same applies to the interpolation nodes for higher degree Lagrangian finite elements defined in Ex. 3.4.5.

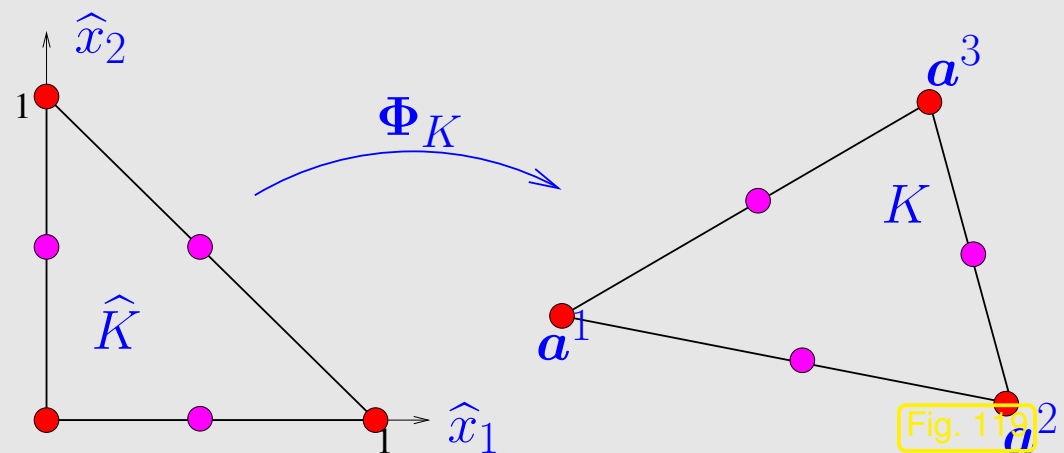


Fig. 1.19

The local shape functions $b_K^i \in \mathcal{P}_p(\mathbb{R}^d)$, $\widehat{b}^i \in \mathcal{P}_p(\mathbb{R}^d)$, $i = 1, \dots, Q$, are *uniquely defined* by the interpolation conditions

$$b_K^i(\mathbf{p}_K^j) = \delta_{ij} \quad , \quad \widehat{b}^i(\widehat{\mathbf{p}}^j) = \delta_{ij} . \quad (3.6.5)$$

Together with $\mathbf{p}_K^i = \Phi_K(\widehat{\mathbf{p}}^i)$ this shows that $\Phi_K^* b_K^i$ satisfies the interpolation conditions (3.6.5) on \widehat{K} and, thus, has to agree with \widehat{b}^i . \square

Terminology: finite element spaces satisfying (3.6.4) are called **affine equivalent**

Remark 3.6.6 (Evaluation of local shape functions at quadrature points).

We consider Lagrangian finite element spaces on a simplicial mesh \mathcal{M} .

Recall from Sect. 3.5.4: definition (3.5.32) of local quadrature formulas via “unit simplex”.

In particular: quadrature nodes on K : $\zeta_l^K = \Phi_K(\widehat{\zeta}^l)$

$$b_K^i(\zeta_l^K) \stackrel{\text{Def. 3.6.2}}{=} \Phi_K^*(b_K^i)(\widehat{\zeta}^l) \stackrel{(3.6.4)}{=} \widehat{b}^i(\widehat{\zeta}^l) \quad \text{independent of } K ! . \quad (3.6.7)$$

$$\int_K F(b_K^i, b_K^j) d\mathbf{x} \approx |K| \sum_{l=1}^P \omega_l F(\widehat{b}^i(\zeta_l), \widehat{b}^j(\zeta_l)), \quad (3.6.8)$$

for any function $F : \mathbb{R}^2 \mapsto \mathbb{R}$.

➤ Precompute $\widehat{b}^i(\zeta_l)$, $i = 1, \dots, Q$, $l = 1, \dots, P$ and store the values in a table!



Remark 3.6.9 (Barycentric representation of local shape functions).

We consider Lagrangian finite element spaces on a simplicial mesh \mathcal{M} .

(3.4.4): formulas for local shape functions for $\mathcal{S}_2^0(\mathcal{M})$ ($d = 2$) in terms of barycentric coordinate functions λ_i , $i = 1, 2, 3$. Is this coincidence? **NO!** Does (3.5.19) hold for any (simplicial) Lagrangian finite element space?

YES!

$$\begin{aligned}
 b_K^i(\mathbf{x}) &\stackrel{(3.6.4)}{=} (\Phi_K^{-1})^* \left(\widehat{\mathbf{x}} \mapsto \widehat{b}^i(\widehat{x}_1, \widehat{x}_2) \right) \\
 &= \widehat{b}^i((\Phi_K^{-1})^*(\widehat{\lambda}_2)(\mathbf{x}), (\Phi_K^{-1})^*(\widehat{\lambda}_3)(\mathbf{x})) = \widehat{b}^i(\lambda_2(\mathbf{x}), \lambda_3(\mathbf{x}))
 \end{aligned}$$

where $\lambda_2(\widehat{\mathbf{x}}) = \widehat{x}_1$, $\lambda_3(\widehat{\mathbf{x}}) = \widehat{x}_2$, $\lambda_1(\widehat{\mathbf{x}}) = 1 - \widehat{x}_1 - \widehat{x}_2 \hat{=}$ barycentric coordinate functions on \widehat{K} , see Ex. 3.3.13,

$\lambda_i \hat{=}$ barycentric coordinate functions on triangle K , see Fig. 72,

$\Phi_K \hat{=}$ affine transformation (\rightarrow Def. 3.5.27), $\Phi_K(\widehat{K}) = K$, see (3.5.29).

➤ By the chain rule:

$$\mathbf{grad} b_K^i(\mathbf{x}) = \frac{\partial \widehat{b}^i}{\partial \widehat{x}_1}(\widehat{\mathbf{x}}) \mathbf{grad} \lambda_2 + \frac{\partial \widehat{b}^i}{\partial \widehat{x}_2}(\widehat{\mathbf{x}}) \mathbf{grad} \lambda_3, \quad \mathbf{x} = \Phi_K(\widehat{\mathbf{x}}).$$

This formula is convenient, because $\mathbf{grad} \lambda_i \equiv \text{const}$, see (3.5.22).

This facilitates the computation of element (stiffness) matrices for 2nd-order elliptic problems in variational form: when using a quadrature formula according to (3.5.32)

$$\int_K (\boldsymbol{\alpha}(\mathbf{x}) \mathbf{grad} b_K^i) \cdot \mathbf{grad} b_K^j \, d\mathbf{x}$$

$$\approx |K| \sum_{l=1}^{P_K} \omega_l \left(\begin{pmatrix} \frac{\partial \widehat{b}^i}{\partial \widehat{x}_1}(\widehat{\zeta}_l) \\ \frac{\partial \widehat{b}^i}{\partial \widehat{x}_2}(\widehat{\zeta}_l) \end{pmatrix}^T \begin{pmatrix} \mathbf{grad} \lambda_1 \cdot \mathbf{grad} \lambda_1 & \mathbf{grad} \lambda_1 \cdot \mathbf{grad} \lambda_2 \\ \mathbf{grad} \lambda_1 \cdot \mathbf{grad} \lambda_2 & \mathbf{grad} \lambda_2 \cdot \mathbf{grad} \lambda_2 \end{pmatrix} \begin{pmatrix} \frac{\partial \widehat{b}^j}{\partial \widehat{x}_1}(\widehat{\zeta}_l) \\ \frac{\partial \widehat{b}^j}{\partial \widehat{x}_2}(\widehat{\zeta}_l) \end{pmatrix} \right)$$

This is very interesting, because

- the values $\frac{\partial \hat{b}^i}{\partial \hat{x}_1}(\hat{\zeta}_l)$ can be *precomputed*,
- simple expressions for $\text{grad } \lambda_i \cdot \text{grad } \lambda_j$ are available, see Sect. 3.2.5.

3.6.2 Example: Quadrilateral Lagrangian finite elements

So far, see Sect. 3.3.3 and (3.3.11), we have adopted the perspective

global shape functions $\xrightarrow{\text{Restriction to element}}$ local shape functions

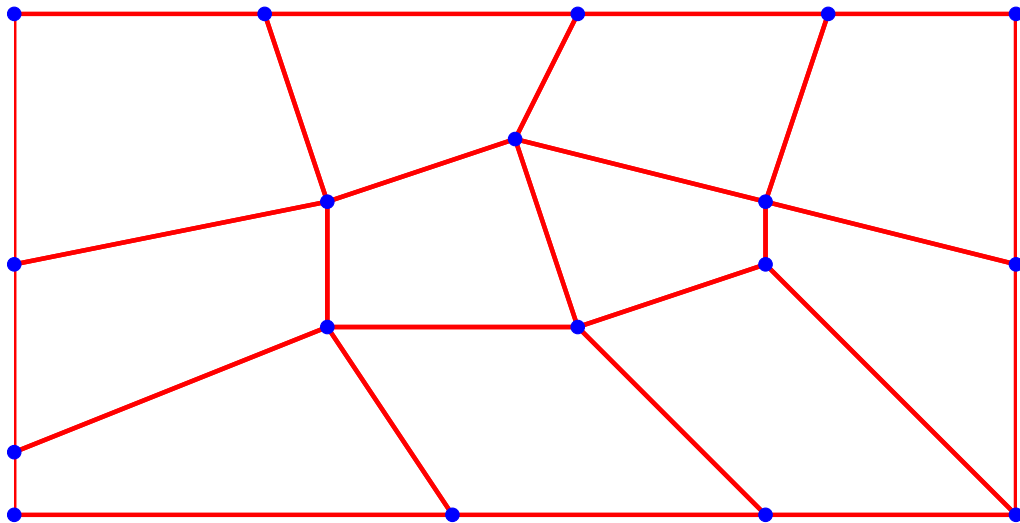
Now we reverse this construction

local shape functions $\xrightarrow{\text{"glueing"}}$ global shape functions (3.6.10)

In fact, when building the global basis functions for quadratic Lagrangian finite elements we already proceeded this way, see Ex. 3.4.2. Fig. 95 lucidly conveys what is meant by “glueing”.

Be aware that the possibility to achieve a continuous global basis function by glueing together local shape function on adjacent cells, entails a judicious choice of the local shape functions.

This section will demonstrate how the policy (3.6.10) together with the formula (3.6.4) will enable us to extend Lagrangian finite element beyond the meshes discussed in Sect. 3.4.



◁ quadrilateral mesh \mathcal{M} in 2D

What is “ $S_1^0(\mathcal{M})$ ”?

Clear: If K is a rectangle, \hat{K} the unit square, then there is a unique affine transformation Φ_K (\rightarrow Def. 3.5.27) with $K = \Phi_K(\hat{K})$.

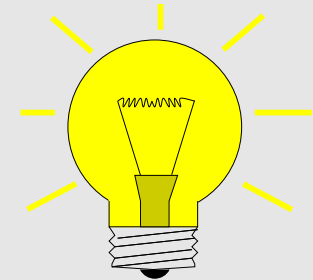
In this case (3.6.4) holds for the local shape functions of bilinear Lagrangian finite elements from Ex. 3.4.6 (and all tensor product Lagrangian finite elements introduced in Sect. 3.4.2)

Idea: \bullet local shape functions $\xrightarrow{\text{"glueing"}} \bullet$ global shape functions

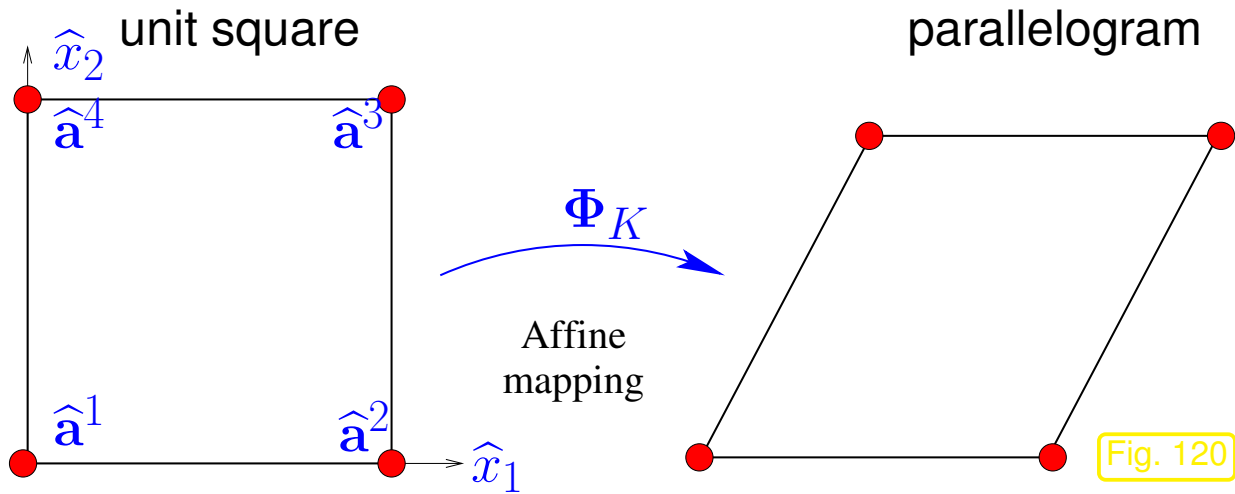
\bullet Build local shape functions by "inverse pullback"

$$b_K^i = (\Phi_K^{-1})^* \hat{b}^i, \tag{3.6.11}$$

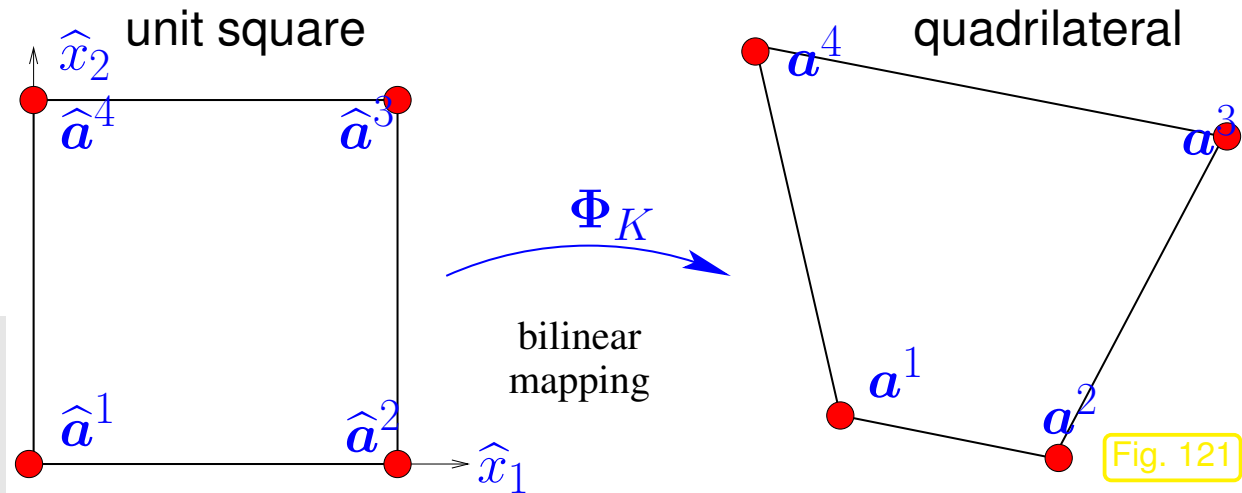
where $\{\hat{b}^i\}_{i=1}^Q \hat{=} \text{set of shape functions on reference element } \hat{K}$.



➤ What is Φ_K for a general quadrilateral ?



Affine transformations fail to produce general quadrilaterals from a square. They only give parallelograms.



It takes *bilinear transformations* to obtain a generic quadrilateral from the unit square.

Bilinear transformation of unit square to quadrilateral with vertices $\mathbf{a}^i, i = 1, 2, 3, 4$:

$$\Phi_K(\hat{\mathbf{x}}) = (1 - \hat{x}_1)(1 - \hat{x}_2) \mathbf{a}^1 + \hat{x}_1(1 - \hat{x}_2) \mathbf{a}^2 + \hat{x}_1\hat{x}_2 \mathbf{a}^3 + (1 - \hat{x}_1)\hat{x}_2 \mathbf{a}^4. \quad (3.6.12)$$

$$\Phi_K(\hat{\mathbf{x}}) = \begin{pmatrix} \alpha_1 + \beta_1 \hat{x}_1 + \gamma_1 \hat{x}_2 + \delta_1 \hat{x}_1 \hat{x}_2 \\ \alpha_2 + \beta_2 \hat{x}_1 + \gamma_2 \hat{x}_2 + \delta_2 \hat{x}_1 \hat{x}_2 \end{pmatrix}, \quad \alpha_i, \beta_i, \gamma_i, \delta_i \in \mathbb{R}.$$

The mapping property $\Phi_K(\hat{\mathbf{a}}^i) = \mathbf{a}^i$ is evident. In order to see $\Phi_K(\hat{K}) = K$ ($\hat{K} \hat{=}$ unit square) for (3.6.12), verify that Φ_K maps all parallels to the coordinate axes to straight lines.

Moreover, a simple computation establishes:

If \hat{K} is the unit square, $\Phi_K : \hat{K} \mapsto K$ a bilinear transformation, and \hat{b}^i the bilinear local shape functions (3.4.8) on \hat{K} ,

then $(\Phi_K^{-1})^* \hat{b}^i$ are linear on the edges of K .

“Glueing” of local shape functions possible

Explanation:

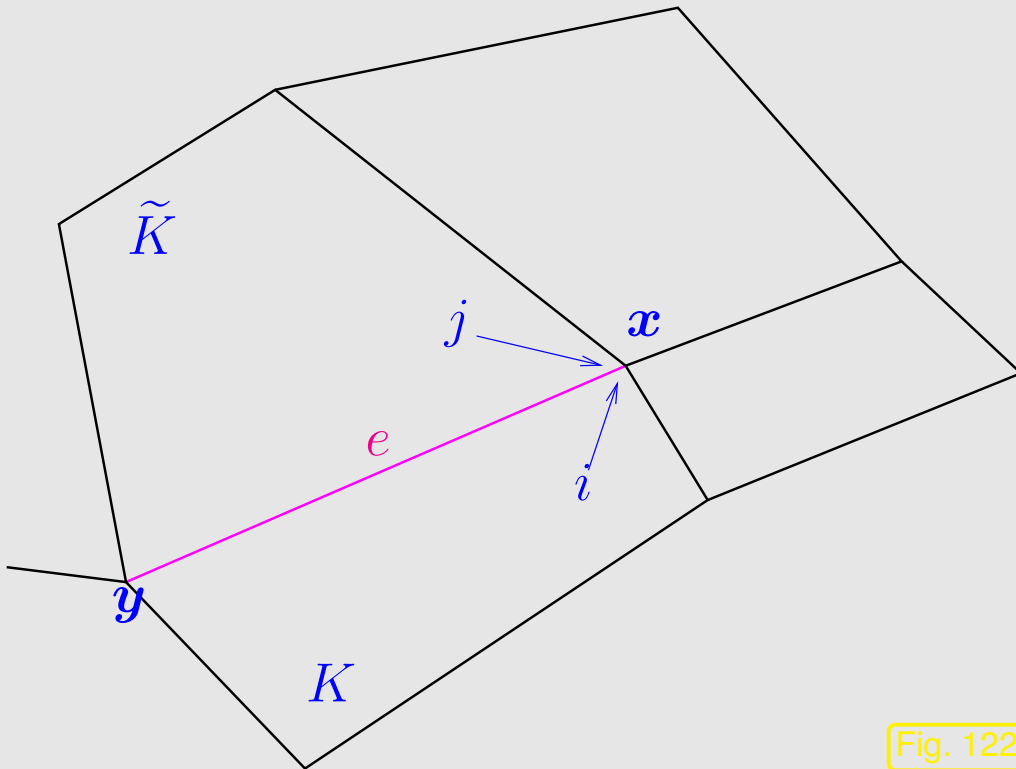


Fig. 122

- ❶ Pick a vertex $\mathbf{x} \in \mathcal{V}(\mathcal{M})$ and consider an adjacent quadrilateral K , on which there is a local shape function b_K^i such that $b_K^i(\mathbf{x}) = 1$ and b_K^i vanishes on all other vertices of K . This local shape function is obtained by inverse pullback of the \widehat{b}^i associated with $\Phi_K^{-1}(\mathbf{x})$.
- ❷ The same construction can be carried out for another quadrilateral \widetilde{K} that shares the vertex \mathbf{x} and an edge e with K . On that quadrilateral we find the local shape function $b_{\widetilde{K}}^j$

- ❸ Both $b_{K|e}^i$ and $b_{\widetilde{K}|e}^j$ are linear and attain the same values, that is 0 and 1 at the endpoints \mathbf{x} and \mathbf{y} of e , respectively.



$$b_{K|e}^i = b_{\widetilde{K}|e}^j$$



Continuity of global shape function (defined by interpolation conditions at nodes)

Remark 3.6.13 (Non-polynomial “bilinear” local shape functions).

Note that the components of Φ_K^{-1} are *not polynomial* even if Φ_K is a bilinear transformation (3.6.12).

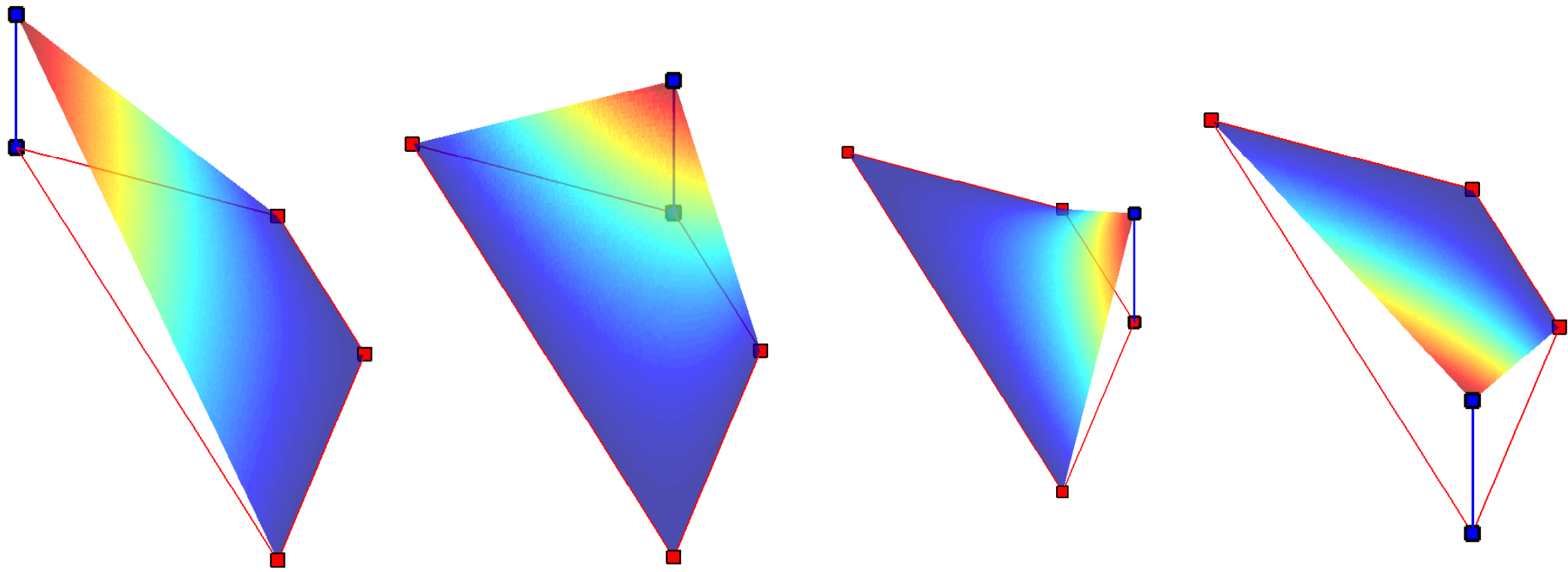


The local shape functions b_K^i defined by (3.6.11), where Φ_K is a bilinear transformation and \widehat{b}^i are the bilinear local shape functions on the unit square, are **not polynomial** in general.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Visualization of local shape functions on trapezoidal cell $K := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$:



3.6.3 Transformation techniques

“Bilinear” Lagrangian finite elements = a specimen of **parametric finite elements**

Definition 3.6.14 (Parametric finite elements).

A finite element space on a mesh \mathcal{M} is called *parametric*, if there exists a *reference element* \hat{K} , $Q \in \mathbb{N}$, and functions $\hat{b}^i \in C^0(\overline{\hat{K}})$, $i = 1, \dots, Q$, such that

$$\forall K \in \mathcal{M}: \exists \text{ bijection } \Phi_K : \hat{K} \mapsto K: \hat{b}^i = \Phi_K^* b_K^i, \quad i = 1, \dots, Q,$$

where $\{b_K^1, \dots, b_K^Q\}$ = set of local shape functions on K .

This definition takes the possibility of “glueing” for granted: the concept of a local shape function, see (3.3.11), implies the existence of a global shape function with the right continuity properties.

How to implement parametric finite elements ?

We consider a generic elliptic 2nd-order variational Dirichlet problem

$$\begin{aligned} u &\in H^1(\Omega), \\ u &= g \text{ on } \partial\Omega \end{aligned} \quad ; \quad \int_{\Omega} (\alpha(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega). \quad (2.3.5)$$

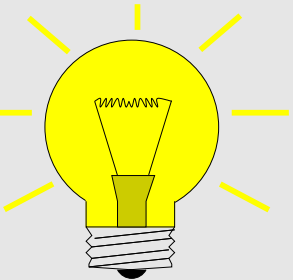
Issue: computation of element (stiffness) matrices and element (load) vectors (\rightarrow Def. 3.5.12).

Challenge: local shape functions b_K^1, \dots, b_K^Q , $K \in \mathcal{M}$, only known implicitly

$$b_K^i = (\Phi_K^{-1})^* \widehat{b}^i$$

\triangleright Known: transformation $\Phi : \widehat{K} \mapsto K$, \widehat{K} reference element, functions $\widehat{b}^1, \dots, \widehat{b}^Q$

$$\widehat{b}^i = \Phi^* b_K^i, \quad i = 1, \dots, Q \quad (\rightarrow \text{pullback, Def. 3.6.2})$$



Use transformation to \widehat{K} to compute element stiffness matrix \mathbf{A}_K ,
element load vector $\vec{\varphi}_K$:

$$\begin{aligned} (\mathbf{A}_K)_{ij} &= \int_K \alpha(\mathbf{x}) \operatorname{grad} b_K^j(\mathbf{x}) \cdot \operatorname{grad} b_K^i(\mathbf{x}) \, d\mathbf{x} \\ &= \int_{\widehat{K}} (\Phi^* \alpha)(\widehat{\mathbf{x}}) \underbrace{(\Phi^* (\operatorname{grad} b_K^j))(\widehat{\mathbf{x}})}_{=?} \cdot \underbrace{(\Phi^* (\operatorname{grad} b_K^i))(\widehat{\mathbf{x}})}_{=?} |\det D\Phi(\widehat{\mathbf{x}})| \, d\widehat{\mathbf{x}}, \end{aligned}$$

$$(\vec{\varphi}_K)_i = \int_K f(\mathbf{x}) b_K^i(\mathbf{x}) \, d\mathbf{x} = \int_{\widehat{K}} (\Phi_K^* f)(\widehat{\mathbf{x}}) \widehat{b}^i(\widehat{\mathbf{x}}) |\det D\Phi(\widehat{\mathbf{x}})| \, d\widehat{\mathbf{x}},$$

by **transformation formula** (for multidimensional integrals, see also (3.5.31)):

$$\int_K f(\mathbf{x}) \, d\mathbf{x} = \int_{\hat{K}} f(\hat{\mathbf{x}}) |\det D\Phi(\hat{\mathbf{x}})| \, d\hat{\mathbf{x}} \quad \text{for } f : K \mapsto \mathbb{R}, \quad (3.6.15)$$

All integrals have been transformed to the reference element \hat{K} , where we apply a quadrature formula (3.5.32).

Needed: values of determinant of Jacobi matrix $D\Phi$ at quadrature nodes $\hat{\zeta}_l$.

Also needed: gradients $\Phi^*(\mathbf{grad} b_K^i)$ at quadrature nodes $\hat{\zeta}_l$!?
(Seems to be a problem as b_K^i may be elusive, cf. Rem. 3.6.13!)

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Lemma 3.6.16 (Transformation formula for gradients).

For differentiable $u : K \mapsto \mathbb{R}$ and any diffeomorphism $\Phi : \hat{K} \mapsto K$ we have

$$(\mathbf{grad}_{\hat{\mathbf{x}}}(\Phi^*u))(\hat{\mathbf{x}}) = (D\Phi(\hat{\mathbf{x}}))^T \underbrace{(\mathbf{grad}_{\mathbf{x}} u)(\Phi(\hat{\mathbf{x}}))}_{=\Phi^*(\mathbf{grad} u)(\mathbf{x})} \quad \forall \hat{\mathbf{x}} \in \hat{K}. \quad (3.6.17)$$

Proof: use **chain rule** for components of the gradient

$$\frac{\partial \Phi^* u}{\partial \hat{x}_i}(\hat{\mathbf{x}}) = \frac{\partial}{\partial \hat{x}_i} u(\Phi(\hat{\mathbf{x}})) = \sum_{j=1}^d \frac{\partial u}{\partial x_j}(\Phi(\hat{\mathbf{x}})) \frac{\partial \Phi_j}{\partial \hat{x}_i}(\hat{\mathbf{x}}) .$$

$$\blacktriangleright \begin{pmatrix} \frac{\partial \Phi^* u}{\partial \hat{x}_1}(\hat{\mathbf{x}}) \\ \vdots \\ \frac{\partial \Phi^* u}{\partial \hat{x}_d}(\hat{\mathbf{x}}) \end{pmatrix} = (\mathbf{grad}_{\hat{\mathbf{x}}} \Phi^* u)(\hat{\mathbf{x}}) = D\Phi(\hat{\mathbf{x}})^T \begin{pmatrix} \frac{\partial u}{\partial x_1}(\Phi(\hat{\mathbf{x}})) \\ \vdots \\ \frac{\partial u}{\partial x_d}(\Phi(\hat{\mathbf{x}})) \end{pmatrix} = D\Phi(\hat{\mathbf{x}})^T (\mathbf{grad}_{\mathbf{x}} u)(\Phi(\hat{\mathbf{x}})) .$$

Here, $D\Phi(\hat{\mathbf{x}}) \in \mathbb{R}^{d,d}$ is the Jacobian of Φ at $\hat{\mathbf{x}} \in \hat{K}$, see [32, Bem. 7.6.1].


R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Using Lemma 3.6.16 we arrive at:

$$(\mathbf{A}_K)_{ij} = \int_{\hat{K}} (\boldsymbol{\alpha}(\Phi(\hat{\mathbf{x}}))(D\Phi)^{-T} \mathbf{grad} \hat{b}^i) \cdot ((D\Phi)^{-T} \mathbf{grad} \hat{b}^j) |\det D\Phi| d\hat{\mathbf{x}} . \quad (3.6.18)$$

Note that the argument $\hat{\mathbf{x}}$ is suppressed for some terms in the integrand.

 notation: $\mathbf{M}^{-T} := (\mathbf{M}^{-1})^T = (\mathbf{M}^T)^{-1}$

Example 3.6.19 (Transformation techniques for bilinear transformations).

$$\Phi(\hat{\boldsymbol{x}}) = \begin{pmatrix} \alpha_1 + \beta_1 \hat{x}_1 + \gamma_1 \hat{x}_2 + \delta_1 \hat{x}_1 \hat{x}_2 \\ \alpha_2 + \beta_2 \hat{x}_1 + \gamma_2 \hat{x}_2 + \delta_2 \hat{x}_1 \hat{x}_2 \end{pmatrix}, \quad \alpha_i, \beta_i, \gamma_i, \delta_i \in \mathbb{R},$$

$$\Rightarrow D\Phi(\hat{\boldsymbol{x}}) = \begin{pmatrix} \beta_1 + \delta_1 \hat{x}_2 & \gamma_1 + \delta_1 \hat{x}_1 \\ \beta_2 + \delta_2 \hat{x}_2 & \gamma_2 + \delta_2 \hat{x}_1 \end{pmatrix},$$

$$\Rightarrow \det(D\Phi(\hat{\boldsymbol{x}})) = \beta_1 \gamma_2 - \beta_2 \gamma_1 + (\beta_1 \delta_2 - \beta_2 \delta_1) \hat{x}_1 + (\delta_1 \gamma_2 - \delta_2 \gamma_1) \hat{x}_2.$$

Both $D\Phi(\hat{\boldsymbol{x}})$ and $\det(D\Phi(\hat{\boldsymbol{x}}))$ are (componentwise) linear in \boldsymbol{x} .

If $\Phi = \Phi_K$ for a generic quadrilateral K as in (3.6.12), then the coefficients $\alpha_i, \beta_i, \gamma_i, \delta_i$ depend on the shape of K in a straightforward fashion:

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \mathbf{a}^1, \quad \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \mathbf{a}^2 - \mathbf{a}^1, \quad \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \mathbf{a}^4 - \mathbf{a}^1, \quad \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} = \mathbf{a}^3 - \mathbf{a}^2 - \mathbf{a}^4 + \mathbf{a}^1.$$



3.6.4 Boundary approximation

Intuition: Approximating a (smooth) curved boundary $\partial\Omega$ by a polygon/polyhedron will introduce a (sort of) **discretization error**.

Parametric finite element constructions provide a tool for avoiding polygonal/polyhedral approximation of boundaries.

Here we discuss this for a very simple case of triangular meshes in 2D (more details \rightarrow [6, Sect, 10.2]).

Idea: **Piecewise polynomial approximation** of boundary (boundary fitting)
($\partial\Omega$ locally considered as function over straight edge of an element)

Example: Piecewise quadratic boundary approximation
(Part of $\partial\Omega$ between \mathbf{a}^1 and \mathbf{a}^2 approximated by parabola)

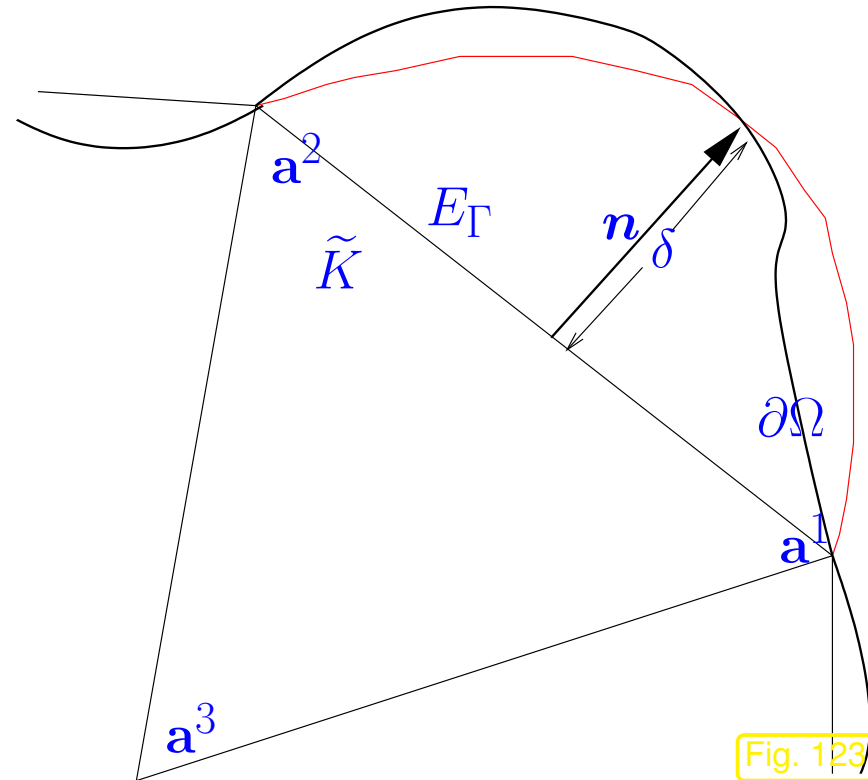


Fig. 123

Mapping $\tilde{K} \rightarrow$ “curved element” K :

$$\tilde{\Phi}_K(\tilde{\mathbf{x}}) := \tilde{\mathbf{x}} + 4\delta \lambda_1(\tilde{\mathbf{x}})\lambda_2(\tilde{\mathbf{x}}) \mathbf{n} . \tag{3.6.20}$$

(λ_i barycentric coordinate functions on \tilde{K} , \mathbf{n} normal to E_Γ , see Fig. 123)

Note: Essential: δ sufficiently small $\implies \Phi$ bijective

The complete transformation $\Phi_K : \hat{K} \mapsto K$ is obtained by joining an affine transformation (\rightarrow

Def. 3.5.27) $\Phi_K^a : \widehat{K} \mapsto \widetilde{K}$, $\Phi_K^a(\widehat{\boldsymbol{x}}) := \mathbf{F}_K \widehat{\boldsymbol{x}} + \boldsymbol{\tau}_K$, and $\widetilde{\Phi}_K$:

$$\Phi_K = \widetilde{\Phi}_K \circ \Phi_K^a .$$

For parabolic boundary fitting:

$$D\widetilde{\Phi}_K = \mathbf{I} + 4\delta \boldsymbol{n} \cdot \mathbf{grad}(\lambda_1 \lambda_2)^\top \in \mathbb{R}^{2,2} \quad , \quad \det(D\widetilde{\Phi}_K) = 1 + 4\delta \boldsymbol{n} \cdot \mathbf{grad}(\lambda_1 \lambda_2) .$$

3.7 Linearization

So far we have discussed the finite elements for *linear* second-order variational boundary value problems only.

However, as we have learned in Ex. 1.5.92, in 1D the Galerkin approach based on linear finite elements was perfectly capable of dealing with *non-linear* two-point boundary value problems. Indeed the abstract discussion of the Galerkin approach in Sect. 1.5.1 was aimed at general and possibly non-linear variational problems, see (1.5.10), (1.5.26).

It goes without saying that the abstract (and formal) discussion of Sect. 1.5.1 remains true for *non-linear* second-order boundary value problems in variational form.

Difficult: Characterization of “spaces of functions with finite energy” (\rightarrow Sobolev spaces, Sect. 2.2) for non-linear variational problems.

(Relief!) In this course we do not worry that much about function spaces.

Recall (\rightarrow Rem. 1.3.21): **Non-linear variational problem**

$$u \in V: \quad a(u; v) = \ell(v) \quad \forall v \in V_0, \quad (1.3.24)$$

- $V_0 \hat{=}$ test space, (real) vector space (usually a function space, “Sobolev-type” space \rightarrow Sect. 2.2)
- $V \hat{=}$ trial space, affine space: usually $V = u_0 + V_0$, with **offset function** $u_0 \in V$,
- $f \hat{=}$ a linear mapping $V_0 \mapsto \mathbb{R}$, a **linear form**,
- $a \hat{=}$ a mapping $V \times V_0 \mapsto \mathbb{R}$, *linear in the second argument*, that is

$$a(u; \alpha v + \beta w) = \alpha a(u; v) + \beta a(u; w) \quad \forall u \in V, v, w \in V_0, \alpha, \beta \in \mathbb{R}. \quad (3.7.1)$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Example 3.7.2 (Heat conduction with radiation boundary conditons).

➤ 2nd-order elliptic boundary value problem, *cf.* (2.5.6) & (2.6.3)

$$\begin{aligned} -\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) &= f && \text{in } \Omega, \\ \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{n}(\mathbf{x}) + \Psi(u) &= 0 && \text{on } \partial\Omega. \end{aligned}$$

▶ Variational formulation from Ex. 2.8.8

$$u \in H^1(\Omega): \quad \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} + \int_{\partial\Omega} \Psi(u) v \, dS = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H^1(\Omega). \quad (2.8.12)$$

If $\Psi : \mathbb{R} \mapsto \mathbb{R}$ is not an affine linear function, then (2.8.12) represents a non-linear variational problem (1.3.24) with

- trial/test space $V = V_0 = H^1(\Omega)$ (\rightarrow Def. 2.2.18),
- right hand side linear form $\ell(v) := \int_{\Omega} f v \, d\mathbf{x}$,
- $\mathbf{a}(u; v) := \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} + \int_{\partial\Omega} \Psi(u) v \, dS$.

Note that the non-linearity enters only through the boundary term.



Pursuing the policy of Galerkin discretization (choice of discrete spaces and corresponding bases, → Sect. 1.5.1) we can convert (1.3.24) into a non-linear system of equations

$$\mathbf{a}(u_0 + \sum_{j=1}^N \mu_j b_N^j; b_N^k) = f(b_N^k) \quad \forall k = 1, \dots, N. \quad (1.5.26)$$

If the left hand side depends smoothly on the unknowns (the coefficients μ_j of $\vec{\mu}$), then the classical Newton method (→ [21, Sect. 4.4]) to solve it iteratively.

Here, we focus on a different approach that reverses the order of the steps:

1. Linearization of problem (“**Newton in function space**”),
2. Galerkin discretization of linearized problems.

“**Newton in function space**”:

Recall idea of **Newton’s method** [21, Sect. 4.4] for the iterative solution of $F(\mathbf{x}) = 0$, $F : D \subset \mathbb{R}^N \mapsto \mathbb{R}^N$ smooth:

Idea: **local linearization:**

Given $\vec{\xi}^{(k)} \in D \succ \vec{\xi}^{(k+1)}$ as zero of affine linear model function

$$F(\vec{\xi}) \approx \tilde{F}(\vec{\xi}) := F(\vec{\xi}^{(k)}) + DF(\vec{\xi}^{(k)})(\vec{\xi} - \vec{\xi}^{(k)}) .$$

▶ **Newton iteration:**

$$\vec{\xi}^{(k+1)} := \vec{\xi}^{(k)} - DF(\vec{\xi}^{(k)})^{-1} F(\vec{\xi}^{(k)}) , \quad [\text{if } DF(\vec{\xi}^{(k)}) \text{ regular}] \quad (3.7.3)$$

◀ ← apply idea to (1.3.24)

Idea: **local linearization:**

Given $u^{(k)} \in V \succ u^{(k+1)}$ from

$$\begin{aligned} w \in V_0: \quad & \mathbf{a}(u^{(k)}; v) + D_u \mathbf{a}(u^{(k)}; v)w = \ell(v) \quad \forall v \in V_0 , \\ & u^{(k+1)} := u^{(k)} + w . \end{aligned} \quad (3.7.4)$$

The meaning of $DF(\vec{\xi}^{(k)})$ in (3.7.3) is clear: it stands for the **Jacobian** of F evaluated at $\vec{\xi}^{(k)}$

But what is the meaning of $D_u \mathbf{a}(u^{(k)}; v)w$ in (3.7.4)?

Remember the “definition” of the Jacobian (for sufficiently smooth F)

$$DF(\vec{\xi})\vec{\mu} = \lim_{t \rightarrow 0} \frac{F(\vec{\xi} + t\vec{\mu}) - F(\vec{\xi})}{t}, \quad \vec{\xi} \in D, \vec{\mu} \in \mathbb{R}^N. \quad (3.7.5)$$

➤ try the “definition”

$$D_u \mathbf{a}(u^{(k)}; v)w = \lim_{t \rightarrow 0} \frac{\mathbf{a}(u + tw; v) - \mathbf{a}(u; v)}{t}, \quad u^{(k)} \in V, v, w \in V_0. \quad (3.7.6)$$

If $(u, v) \mapsto \mathbf{a}(u; v)$ depends smoothly on u , then

$$(v, w) \mapsto D_u \mathbf{a}(u^{(k)}; v)w \quad \text{is a bilinear form } V_0 \times V_0 \mapsto \mathbb{R}.$$

Example 3.7.7 (Derivative of non-linear $u \mapsto \mathbf{a}(u; \cdot)$).

Apply formula (3.7.6) to the non-linear boundary term in (2.8.12), that is, here

$$\mathbf{a}(u; v) := \int_{\partial\Omega} \Psi(u)v \, dS, \quad u, v \in H^1(\Omega).$$

$$\blacktriangleright \mathbf{a}(u + tw; v) - \mathbf{a}(u; v) = \int_{\partial\Omega} (\Psi(u + tw) - \Psi(u))v \, dS, \quad u, v \in H^1(\Omega).$$

Assume $\Psi : \mathbb{R} \mapsto \mathbb{R}$ is smooth with derivative Ψ' and employ *Taylor expansion* for fixed $w \in H^1(\Omega)$ and $t \rightarrow 0$

$$\mathbf{a}(u + tw; v) - \mathbf{a}(u; v) = \int_{\partial\Omega} t\Psi'(u)wv \, dS + O(t^2).$$

$$\blacktriangleright D_u \mathbf{a}(u^{(k)}; v)w = \lim_{t \rightarrow 0} \frac{\mathbf{a}(u + tw; v) - \mathbf{a}(u; v)}{t} = \int_{\partial\Omega} \Psi'(u)wv \, dS.$$

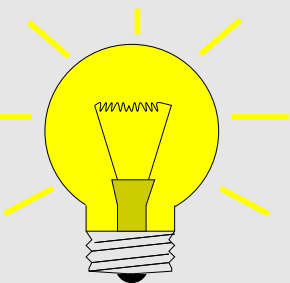
= a **bilinear** form in v, w on $H^1(\Omega) \times H^1(\Omega)$!

This example also demonstrates how to actually compute $D_u \mathbf{a}(u^{(k)}; v)w$!

Idea: Galerkin discretization of the **linear variational** problem from (3.7.4)

$$w \in V_0: \quad \mathbf{c}(w, v) = g(v) \quad \forall v \in V_0,$$

$$\mathbf{c}(w, v) = D_u \mathbf{a}(u^{(k)}; v)w, \quad g(v) := \ell(v) - \mathbf{a}(u^{(k)}; v).$$



► **Newton-Galerkin iteration** for (1.3.24)

Given $u_N^{(k)} \in V_N^{(k)} \rhd u_N^{(k+1)} \in V_N^{(k+1)}$ from

$$\begin{aligned} w_N \in V_{0,N}^{(k+1)} : \quad & D_u \mathbf{a}(u_N^{(k)}; v_N) w_N = \ell(v_N) - \mathbf{a}(u_N^{(k)}; v_N) \quad \forall v_N \in V_{0,N}^{(k+1)}, \\ & u_N^{(k+1)} := \mathbf{P}_N^{(k+1)} u_N^{(k)} + w. \end{aligned} \quad (3.7.8)$$

Newton update

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Note: different Galerkin trial/test spaces $V_N^{(k)}$, $V_{0,N}^{(k)}$ may be used in different steps of the iteration!

(It may enhance efficiency to use Galerkin trial/test spaces of a rather small dimension in the beginning and switch to larger when the iteration is about to converge.)

Warning! If $V_N^{(k)} \neq V_N^{(k+1)}$ you cannot simply add $u_N^{(k)}$ and w

► Linear projection operator $\mathbf{P}_N^{(k+1)} : V_N^{(k)} \mapsto V_N^{(k+1)}$ required in (3.7.8)

Any of the Lagrangian finite element spaces introduced in Sect. 3.4 will supply valid $V_N/V_{0,N}$. Offset functions can be chosen according to the recipes from Sect. 3.5.5.

Important aspect: **termination** of iteration, see [21, Thm. 4.4.3].

Option: termination based on relative size of Newton update, with $w, u_N^{(k+1)}$ from (3.7.8)

$$\mathbf{STOP}, \text{ if } \|w\| \leq \tau \left\| u_N^{(k+1)} \right\|, \quad (3.7.9)$$

where $\|\cdot\|$ is a relevant norm (e.g., energy norm) on $V_N^{(k+1)}$ and $\tau > 0$ a prescribed **relative tolerance**.

Learning outcomes

Skills to be acquired in Chapter 3:

- Familiarity with all aspects of abstract Galerkin discretization of a linear variational problem.
- Knowledge of the role of the main ingredients for a finite element Galerkin discretization: variational problem, mesh, global and local shape functions.
- Understanding of properties of finite element Galerkin matrices, in particular, their sparsity patterns.
- Ability to implement the (approximate, by means of quadrature) computation of element matrices and element right hand side vectors for Lagrangian finite elements and rather general 2nd-order elliptic boundary value problems
- Grasp of rationale and realization of *local assembly* of finite element Galerkin matrices and right hand side vectors.
- Use of LehrFEM finite element MATLAB library to implement a finite element simulation code for a given 2nd-order elliptic boundary value problem.
- Ability to deal with non-zero essential boundary conditions in a finite element context.
- mastery of transformation techniques for computing element matrices and vectors, and dealing with more general shapes of cells.

4

Finite Differences (FD) and Finite Volume Methods (FV)

Now we examine two approaches to the discretization of scalar linear 2nd-order elliptic BVPs that offer an alternative to finite element Galerkin methods discussed in Ch. 3.

What these methods have in common with (low degree) Lagrangian finite element methods is

- that they rely on meshes (\rightarrow Sect. 3.3.1) tiling the computational domain Ω ,
- they lead to *sparse* linear systems of equations.

Remark 4.0.1 (Collocation approach on “complicated” domains).

Sect. 1.5.2.2 taught us **spline collocation methods**. A crucial insight was that collocation methods (see beginning of Sect. 1.5.2 for a presentation of the idea), which target the boundary value problem in ODE/PDE form, have to employ discrete trial spaces comprised of *continuously differentiable* functions, see Rem. 1.5.119.

It is very difficult to construct spaces of piecewise polynomial C^1 -functions on non-tensor product domains for $d = 2, 3$ and find suitable collocation nodes, *cf.* (1.5.112).

Therefore we skip the discussion of collocation methods for 2nd-order elliptic BVPs on $\Omega \subset \mathbb{R}^d$, $d = 2, 3$.



4.1 Finite differences

A finite difference scheme for a 2-point boundary value problem was presented in Sect. 1.5.3, which you are advised to browse again. Its gist was

to replace the derivatives in the *differential equation* with *difference quotients* connecting approximate values of the solutions *at the nodes of a grid/mesh*.

Recall: Finite differences target the “ODE/PDE-formulation” of the boundary value problem.

Our goal: extension to higher dimensions

2D model problem:

Homogeneous Dirichlet BVP for Laplacian:

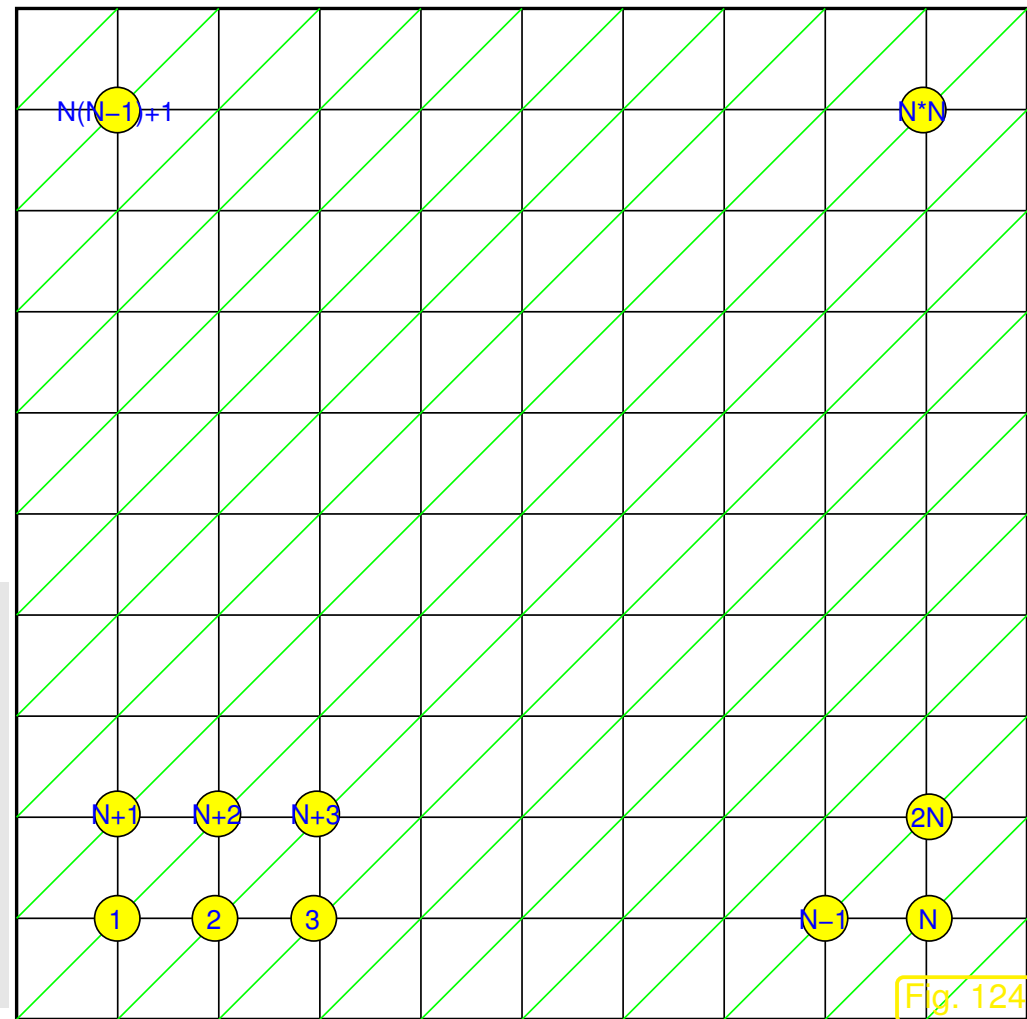
$$-\Delta u = -\frac{\partial^2 u}{\partial x_1^2} - \frac{\partial^2 u}{\partial x_2^2} = f \quad \text{in } \Omega :=]0, 1[^2,$$

$$u = 0 \quad \text{on } \partial\Omega.$$

Discretization based on

\mathcal{M} = (triangular) **tensor-product grid**
 (meshwidth $h = (1 + N)^{-1}$, $N \in \mathbb{N}$)

lexikographic (line-by-line) ordering of nodes of \mathcal{M}



R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

1 finite difference approach to $-\Delta$: approximation of derivatives by symmetric difference quotients

This is nothing new: we did this in (1.5.138).

$$\frac{\partial^2}{\partial x_1^2} u \Big|_{\mathbf{x}=(\xi,\eta)} \approx \frac{u(\xi - h, \eta) - 2u(\xi, \eta) + u(\xi + h, \eta)}{h^2},$$

$$\frac{\partial^2}{\partial x_2^2} u \Big|_{\mathbf{x}=(\xi,\eta)} \approx \frac{u(\xi, \eta - h) - 2u(\xi, \eta) + u(\xi, \eta + h)}{h^2}.$$

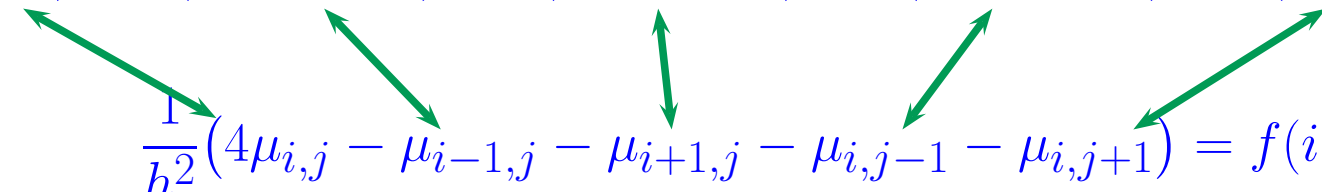
►
$$-\Delta u \Big|_{\mathbf{x}=(\xi,\eta)} \approx \frac{1}{h^2} (4u(\xi, \eta) - u(\xi - h, \eta) - u(\xi + h, \eta) - u(\xi, \eta - h) - u(\xi, \eta + h)).$$

Use this approximation at grid point $\mathbf{p} = (ih, jh)$. This will connect the five point values $u(ih, jh)$, $u((i - 1)h, jh)$, $u((i + 1)h, jh)$, $u(ih, (j - 1)h)$, $u(ih, (j + 1)h)$.

Approximations $\mu_{i,j}$ to the *point values* $u(ih, jh)$
will be the **unknowns** of the finite difference method.

Centering the above difference quotients at grid points yields linear relationships between the unknowns:

$$\frac{1}{h^2} (4u(ih, jh) - u(ih - h, jh) - u(ih + h, jh) - u(ih, jh - h) - u(ih, jh + h)) = f(ih, jh) ,$$

$$\frac{1}{h^2} (4\mu_{i,j} - \mu_{i-1,j} - \mu_{i+1,j} - \mu_{i,j-1} - \mu_{i,j+1}) = f(ih, jh) . \quad (4.1.1)$$


Also this is familiar from the discussion in 1D. Yet, in 1D the association of the point values and of components of the vector $\vec{\mu}$ of unknowns was straightforward and suggested by the linear ordering of the nodes of the grid. In 2D we have much more freedom.

One option on tensor-product grids is the line-by-line ordering (lexikographic ordering) depicted in Fig. 124. This allows a simple indexing scheme:

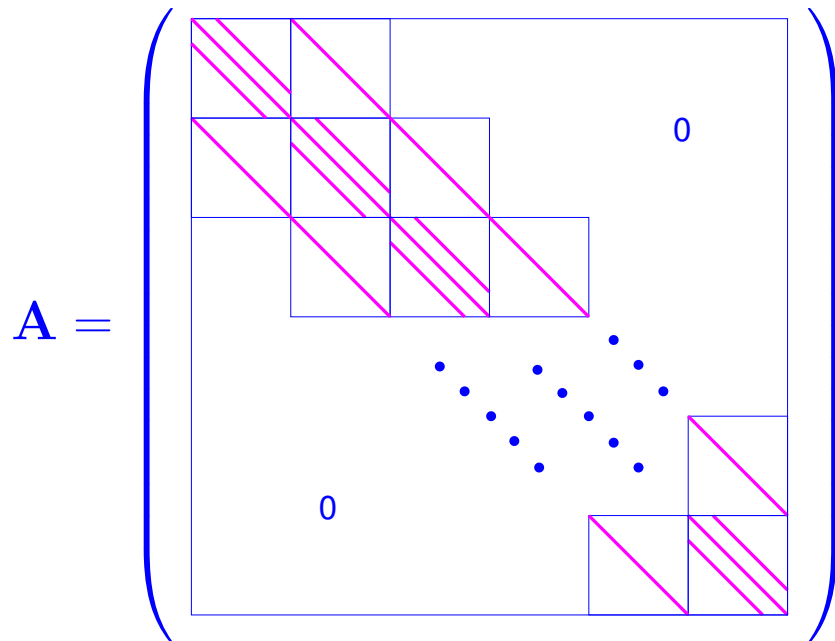
$$u(\mathbf{p}) \leftrightarrow \mu_{i,j} \leftrightarrow \mu_{(j-1)N+i}$$



$$\frac{-\mu_{(j-2)N+i} - \mu_{(j-1)N+i-1} + 4\mu_{(j-1)N+i} - \mu_{(j-1)N+i+1} - \mu_{jN+i}}{h^2} = \underbrace{f(ih, jh)}_{=\varphi_{(j-1)N+i}}. \quad (4.1.2)$$

► linear system of N^2 equations $\mathbf{A}\vec{\mu} = \vec{\varphi}$ with $N^2 \times N^2$ block-tridiagonal **Poisson matrix**

$$\mathbf{A} := \frac{1}{h^2} \begin{pmatrix} \mathbf{T} & -\mathbf{I} & 0 & \cdots & \cdots & 0 \\ -\mathbf{I} & \mathbf{T} & -\mathbf{I} & & & \vdots \\ 0 & -\mathbf{I} & \mathbf{T} & -\mathbf{I} & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & -\mathbf{I} & \mathbf{T} & -\mathbf{I} \\ 0 & \cdots & \cdots & 0 & -\mathbf{I} & \mathbf{T} \end{pmatrix}, \quad \mathbf{T} := \begin{pmatrix} 4 & -1 & 0 & & & 0 \\ -1 & 4 & -1 & & & \vdots \\ 0 & -1 & 4 & -1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & -1 & 4 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 4 \end{pmatrix} \in \mathbb{R}^{N,N} \quad (4.1.3)$$



◁ band structure of Poisson matrix

The MATLAB command

```
A = gallery('poisson', n)
```

creates a sparse $n^2 \times n^2$ Poisson matrix.

Already in Sect. 1.5.3 we saw that the linear system of equations popping out of the finite difference discretization of the linear two-point BVP (1.5.117) was the same as that obtained via the linear finite Galerkin approach on the same mesh.

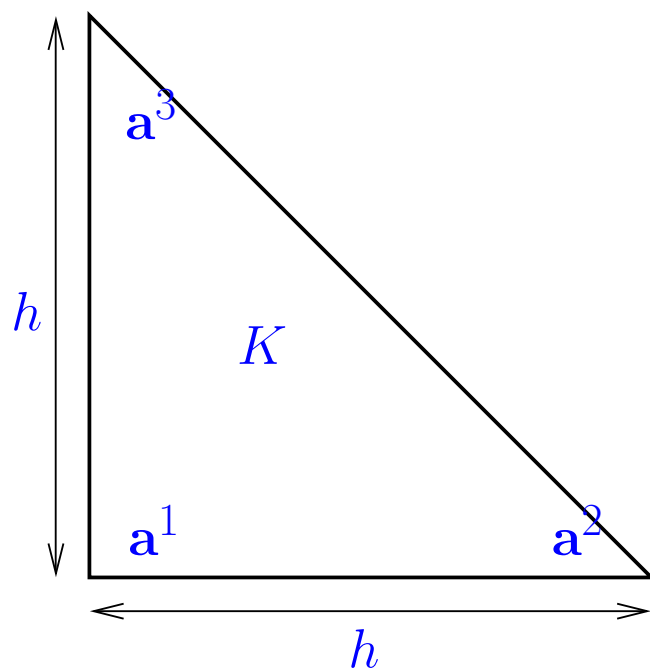
In two dimensions we will also come to this conclusion! So, let us derive the Galerkin matrix and right hand side vector for the 2D model problem on the tensor product mesh depicted in Fig. 124. To begin with we convert it into a **triangular mesh** \mathcal{M} by splitting each square into two equal triangles by inserting a diagonal (green lines in Fig. 124). On this mesh we use **linear Lagrangian finite elements** as in Sect. 3.2.

Then we repeat the considerations of Sect. 3.2.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

- ② Linear Lagrangian finite element Galerkin discretization \rightarrow Sect. 3.2: $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$
(global shape functions $\hat{=}$ “tent functions”, \rightarrow Fig. 88)



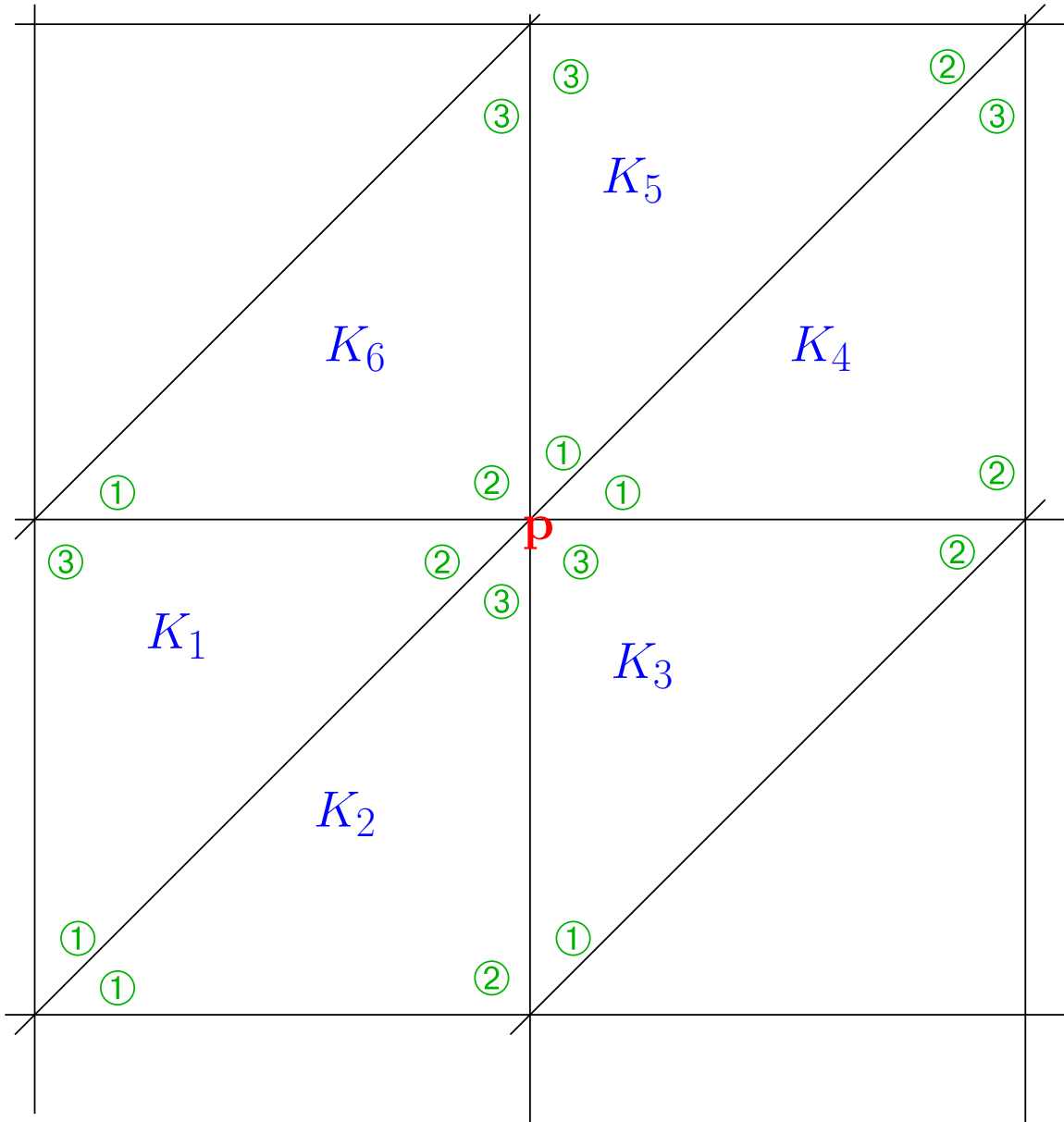
Element stiffness matrix from (3.2.11):

$$\mathbf{A}_K = \frac{1}{2} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} .$$

(\leftarrow numbering of local shape functions)

Element load vector: use **three-point quadrature formula** (3.5.35)

$$\vec{\varphi}_K = \frac{1}{6} h^2 \begin{pmatrix} f(\mathbf{a}^1) \\ f(\mathbf{a}^2) \\ f(\mathbf{a}^3) \end{pmatrix} .$$



Local assembly:

← green: local vertex numbers

Contributions to load vector component associated with node **p**:

From K_1 : $(\vec{\varphi}_{K_1})_2$

From K_2 : $(\vec{\varphi}_{K_2})_3$

From K_3 : $(\vec{\varphi}_{K_3})_3$

From K_4 : $(\vec{\varphi}_{K_4})_1$

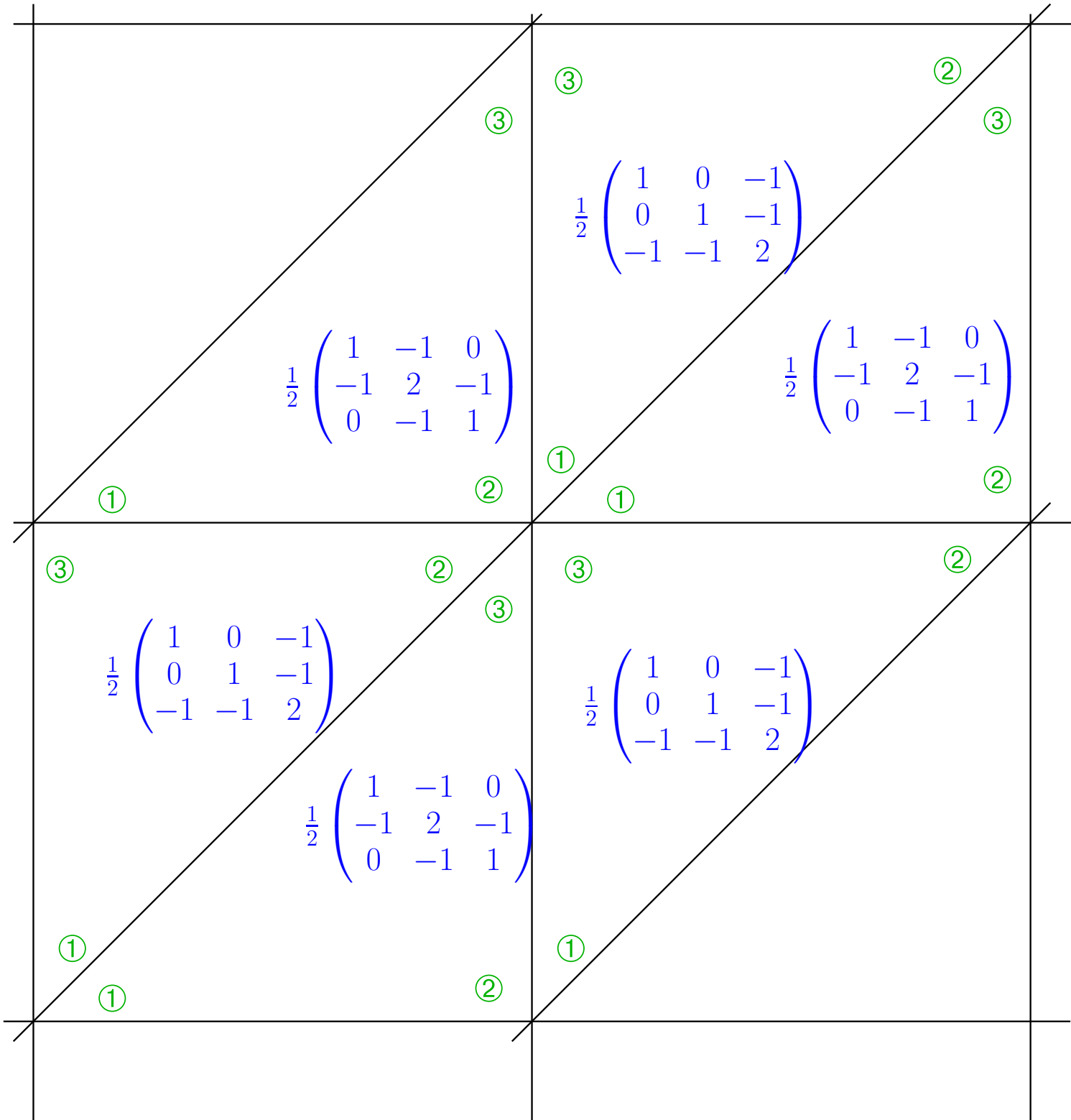
From K_5 : $(\vec{\varphi}_{K_5})_1$

From K_6 : $(\vec{\varphi}_{K_6})_2$



$$\vec{\varphi}_{\mathbf{p}} = h^2 f(\mathbf{p}) .$$

Assembly of finite element Galerkin matrix from element (stiffness) matrices (→ Sect. 3.5.3):



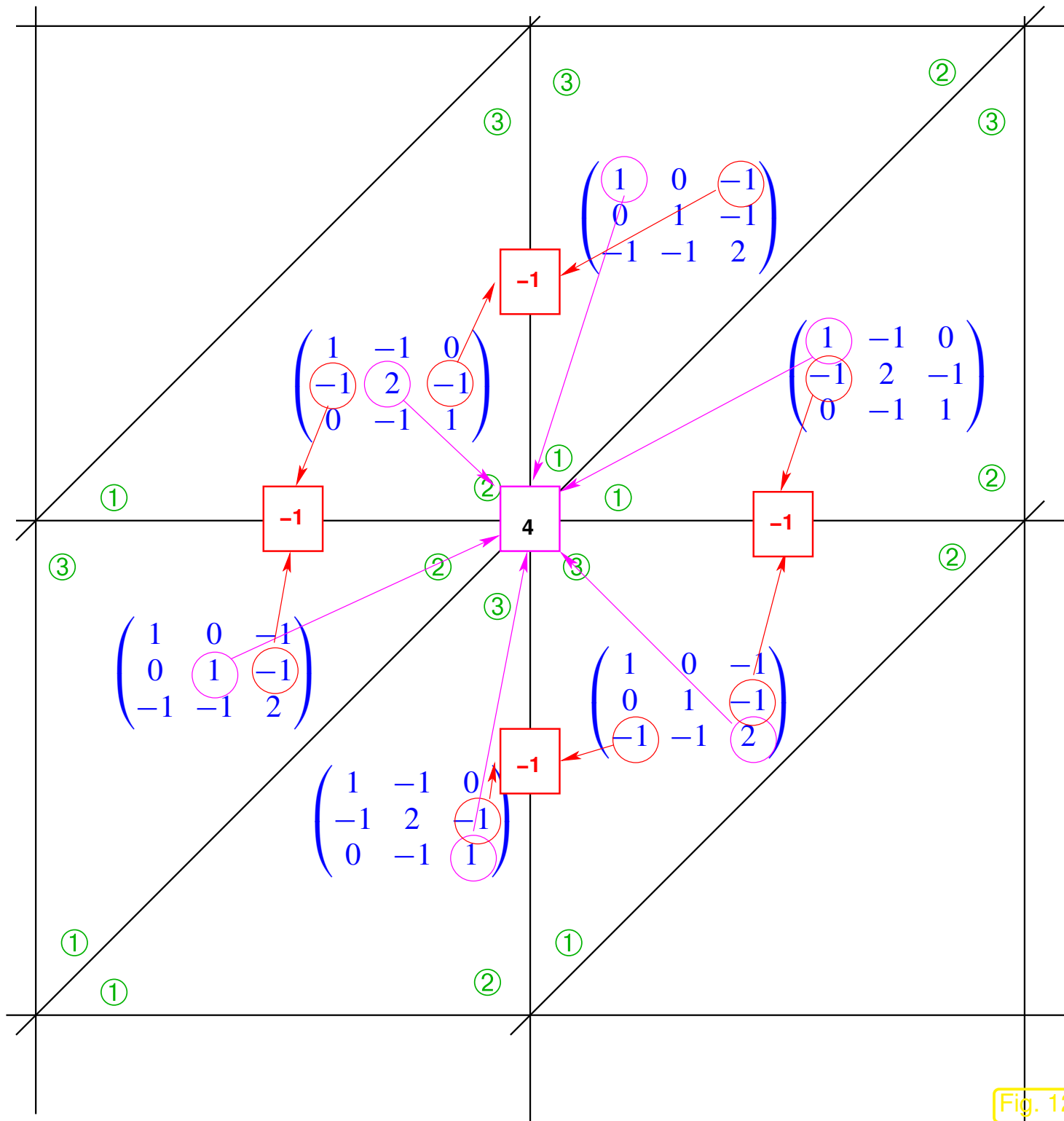


Fig. 126

➤ $N^2 \times N^2$ linear system of equations $h^2 \mathbf{A} \vec{\mu} = h^2 \vec{\varphi}$, $\mathbf{A} \hat{=} \text{Poisson matrix}$ (4.1.3)



(Most) finite difference schemes \leftrightarrow finite element Galerkin schemes
with numerical quadrature
on **structured** meshes

Discussion:

finite differences vs. finite element Galerkin methods
(here focused on 2nd-order linear scalar problems)

- Finite element methods can be used on general triangulations and structured (tensor-product) meshes alike, which delivers superior flexibility in terms of geometry resolution (advantage FEM).
- The correct treatment of all kinds of boundary conditions (\rightarrow Sect. 2.6). naturally emerges from the variational formulations in the finite element method (advantage FEM).
- Finite element methods have built-in “safety rails” because there are clear criteria for choosing viable finite element spaces and once this is done, there is no freedom left to go astray (advantage FEM).

- Finite element methods are harder to understand (advantage FD, but only with students who have not attended this course!)

Then, why are “finite difference methods” ubiquitous in scientific and engineering simulations ?

When people talk “finite differences” they have in mind **structured meshes** (translation invariant, tensor product structure) and use the term as synonym for “discretization on structured meshes”. The popularity of structured meshes is justified:

- structured meshes allow regular data layout and vectorization, which boost the performance of algorithms on high performance computing hardware.

→ course “Parallel Computing for Scientific Simulations”

- translation invariant PDE operators give rise to simple Galerkin matrices that need not be assembled and stored (recall the 5-point-stencil for $-\Delta$) and support very efficient matrix \times vector operations.



Use structured meshes whenever possible!

4.2 Finite volume methods (FVM)

4.2.1 Discrete balance laws

Focus: linear scalar 2nd-order elliptic boundary value problem in 2D (\rightarrow Sect. 2.5), homogeneous Dirichlet boundary conditions (\rightarrow Sect. 2.6), uniformly positive scalar heat conductivity $\kappa = \kappa(\mathbf{x})$

$$-\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) = f \quad \text{in } \Omega \quad , \quad u = 0 \quad \text{on } \partial\Omega .$$

Finite volume methods for 2nd-order elliptic BVP are inspired by the *conservation principle* (2.5.2).

$$\int_{\partial V} \mathbf{j} \cdot \mathbf{n} \, dS = \int_V f \, d\mathbf{x} \quad \text{for all "control volumes" } V . \quad (2.5.2)$$

Physics requires that this holds for all (infinitely many) "control volumes" $V \subset \Omega$.

Since discretization has to lead to a finite number of equations, the idea is to demand that (2.5.2) holds for only a *finite number of special control volumes*.

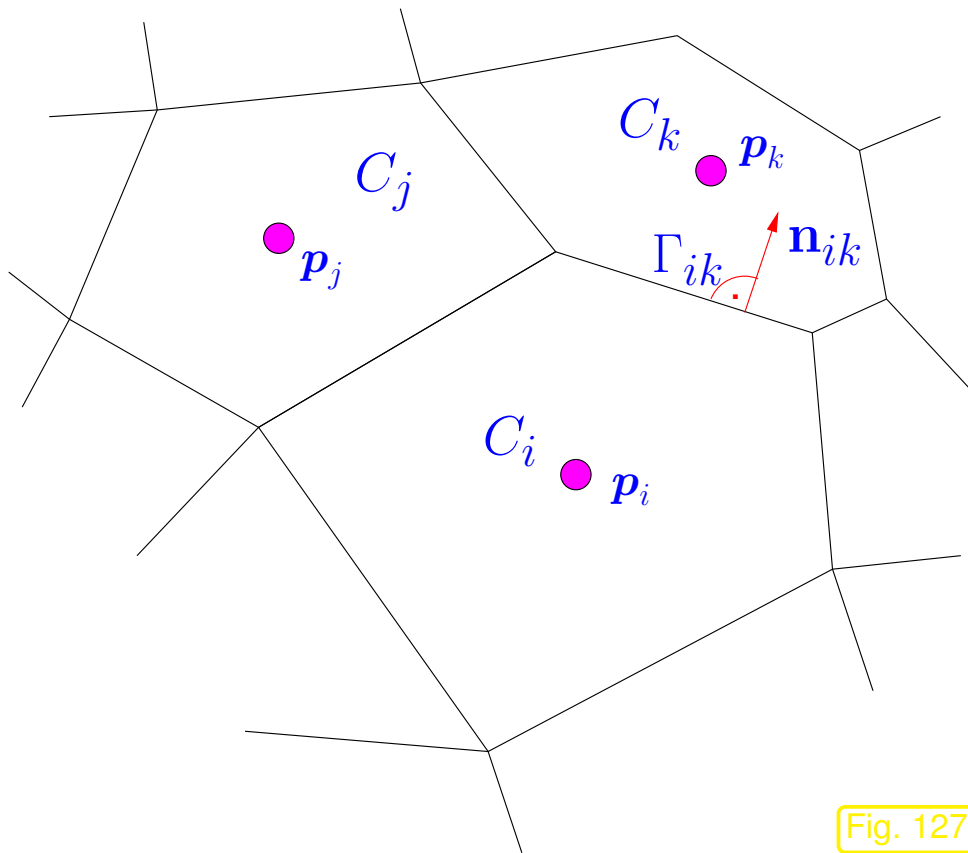


Fig. 127

Concrete choice:

Control volumes =

(polygonal) cells of a mesh $\tilde{\mathcal{M}} = \{C_i\}_i$
covering computational domain Ω .

Associate cell $C_i \leftrightarrow$ nodal value μ_i

Meaning: $\mu_i \approx u(\mathbf{p}_i)$, \mathbf{p}_i = "center" of C_i

Correspondingly, “heat conservation in control volumes” has to be supplemented by a rule that furnishes the heat flux between two adjacent control volumes.

Second ingredient: local **numerical fluxes**

For two adjacent cells C_k, C_i with common edge $\Gamma_{ik} := \bar{C}_i \cap \bar{C}_k$.

$$\text{Numerical flux } J_{ik} = \Psi(\mu_i, \mu_k) \approx \int_{\Gamma_{ik}} \mathbf{j} \cdot \mathbf{n}_{ik} dS$$

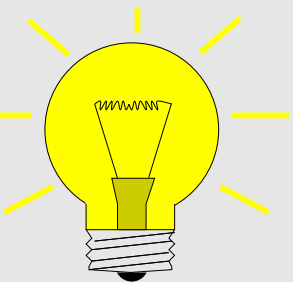
(Ψ = **numerical flux function**, \mathbf{j} = (heat) flux, see (2.5.1), $\mathbf{n}_{ik} \hat{=}$ edge normal.)

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Idea: consider balance law on (finitely many !) control volumes C_i

$$\int_{\partial C_i} \mathbf{j} \cdot \mathbf{n}_i dS = \int_{C_i} f d\mathbf{x} \implies \sum_{k \in \mathcal{U}_i} J_{ik} = \int_{C_i} f d\mathbf{x} .$$



notation: $\mathcal{U}_i := \{j : C_i \text{ and } C_j \text{ share edge, } C_j \in \tilde{\mathcal{M}}\}$, \mathbf{p}_i = node associated with control volume C_i .

System of equations ($\widetilde{M} := \#\widetilde{\mathcal{M}}$ equations, unknowns μ_i):

$$\sum_{k \in \mathcal{U}_i} \Psi(\mu_i, \mu_k) = \int_{C_i} f \, d\mathbf{x} \quad \forall i = 1, \dots, \widetilde{M}. \quad (4.2.1)$$

Further approximation: 1-point quadrature for approximate evaluation of integral over C_i ,

$$\sum_{k \in \mathcal{U}_i} \Psi(\mu_i, \mu_k) = |C_i| f(\mathbf{p}_i) \quad \forall i = 1, \dots, \widetilde{M}. \quad (4.2.2)$$

Note: homogeneous Dirichlet problem \triangleright only “interior” control volumes in (4.2.2)

4.2.2 Dual meshes

Dual meshes are a commonly used technique for the construction of control volumes for FVM, based on conventional FE triangulation \mathcal{M} of Ω (\rightarrow Sect. 3.3.1).

Focus: dual mesh for triangular mesh \mathcal{M} in 2D, Ω polygon

Popular choice: Voronoi dual mesh

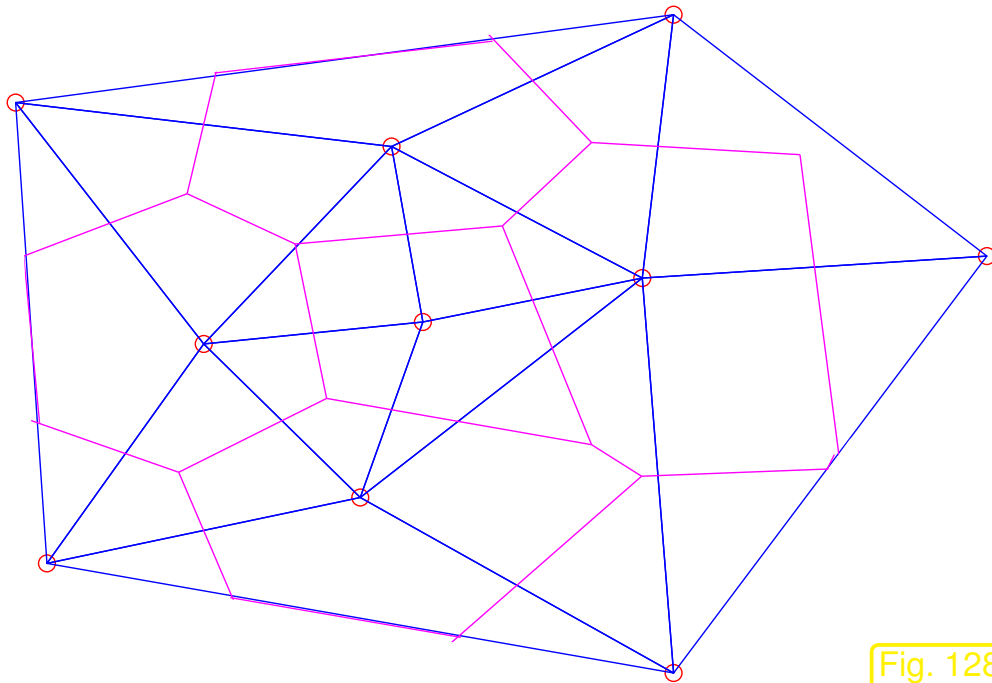


Fig. 128

$$\mathcal{V}(\mathcal{M}) = \{\mathbf{p}_1, \dots, \mathbf{p}_M\} = \text{nodes of } \mathcal{M}$$

Define **Voronoi cells**

$$C_i := \{\mathbf{x} \in \Omega: |\mathbf{x} - \mathbf{p}_i| < |\mathbf{x} - \mathbf{p}_j| \forall j \neq i\} . \quad (4.2.3)$$

Voronoi dual mesh $\tilde{\mathcal{M}} := \{C_i\}_{i=1}^M$

Construction of Voronoi dual cells:

edges \rightarrow perpendicular bisectors

nodes \rightarrow circumcenters of triangles

► straightforward generalization to 3D

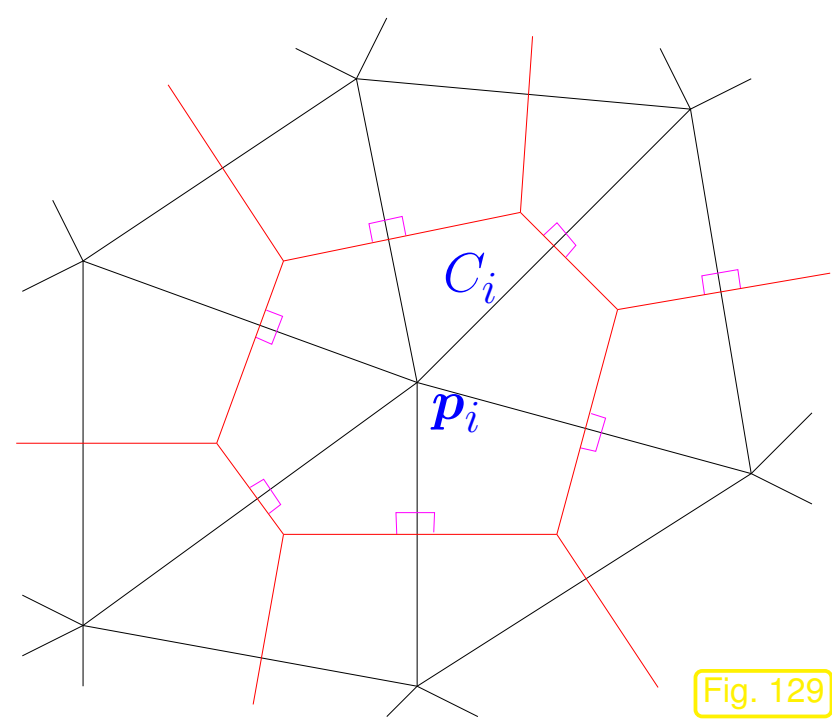


Fig. 129

Remark 4.2.4 (Geometric obstruction to Voronoi dual meshes).

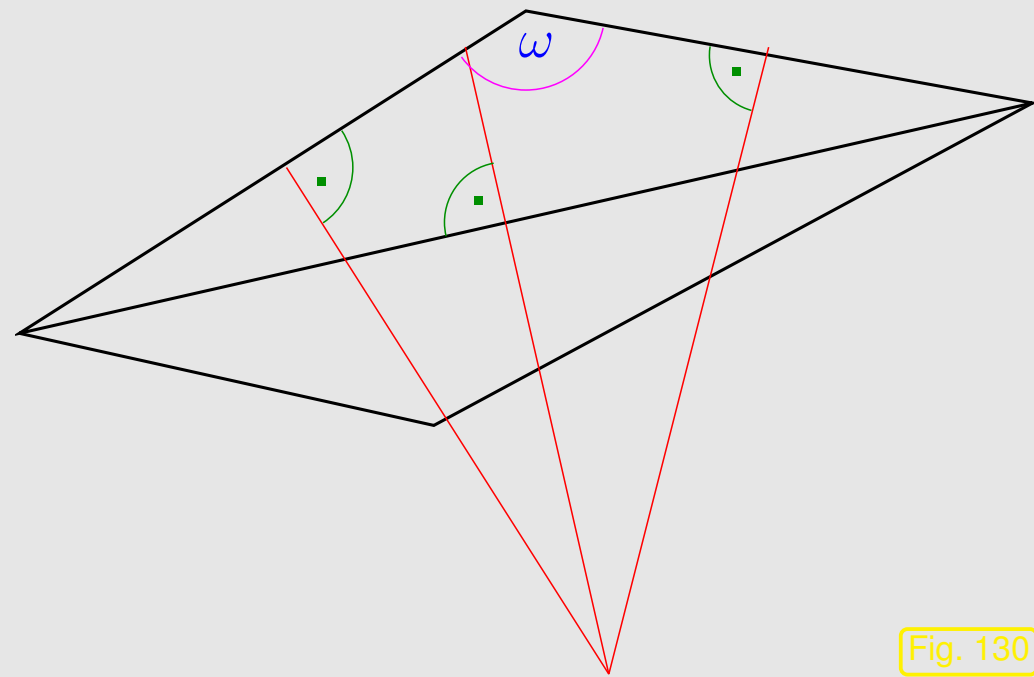


Fig. 130

⇐ **Obtuse angle ω :**

- circumcenter \notin triangle
- $\bar{C}_i \cap \bar{C}_j \neq \emptyset \not\Rightarrow$ nodes i, j connected by edge
- geometric construction breaks down
- connectivity of unknowns hard to determine



Angle condition to ensure $\bar{C}_i \cap \bar{C}_j \neq \emptyset \Leftrightarrow$ nodes i, j connected by edge of \mathcal{M} :

(i) sum of angles facing interior edge $\leq \pi$,

(ii) angles facing boundary edges $\leq \pi/2$ (for non-Dirichlet boundary conditions).

▶ (i), (ii) characterize **Delaunay triangulations**

Popular choice: **Barycentric dual mesh**

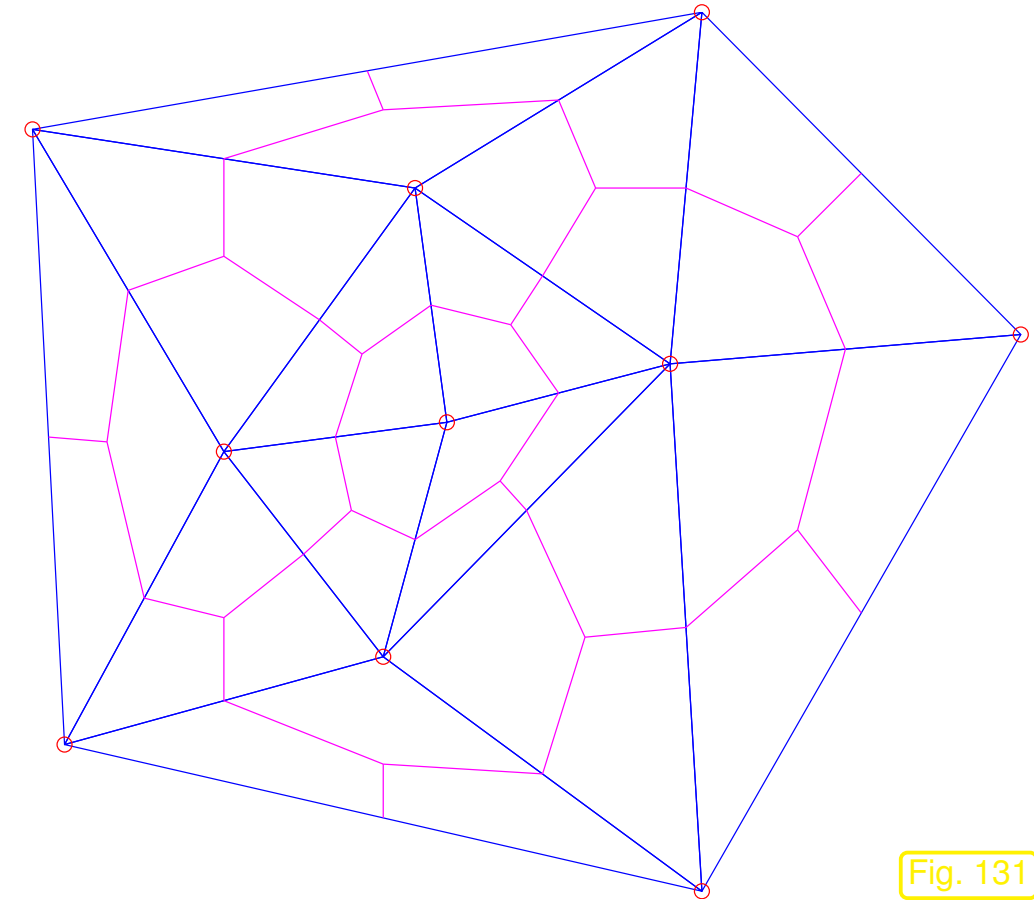


Fig. 131

Dual cells:

edges \rightarrow union of lines connecting
barycenters and midpoints of
edges of \mathcal{M}

nodes \rightarrow barycenters of triangles



No geometric obstructions

4.2.3 Relationship of finite elements and finite volume methods

Hardly surprising, finite volume methods and finite element Galerkin discretizations are closely related. This will be explored in this section for a model problem.

Setting:

- We consider the homogeneous Dirichlet problem for the Laplacian Δ

$$-\Delta u = f \quad \text{in } \Omega \quad , \quad u = 0 \quad \text{on } \partial\Omega . \quad (4.2.5)$$

- Discretization by finite volume method based on a triangular mesh \mathcal{M} and on Voronoi dual cells
→ Fig. 128:

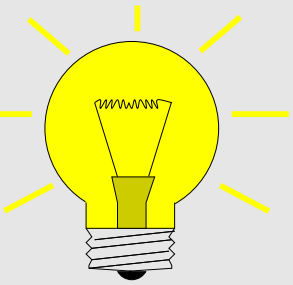
Assumption: \mathcal{M} = Delaunay triangulation of Ω \Leftrightarrow angle condition

Number of control volumes = number of interior nodes of \mathcal{M}

Still missing: specification of numerical flux function $\Psi : \mathbb{R}^2 \mapsto \mathbb{R}$ for each dual edge

Idea: obtain numerical flux from

Fourier's law (2.5.3) applied to a (sufficiently smooth) $u_N : \Omega \mapsto \mathbb{R}$
reconstructed from dual cell values μ_i .



Natural approach, since μ_i is read as approximation of $u(\mathbf{p}_i)$, where the “center” \mathbf{p}^i of the dual cell C_i coincides with an *interior node* $\mathbf{x}^i \in \mathcal{V}(\mathcal{M})$ of the triangular mesh \mathcal{M} :

$$u_N = I_1 \vec{\mu} := \sum_{i=1}^N \mu_i b_N^i, \quad (4.2.6)$$

where $N = \#\mathcal{V}(\mathcal{M})$ = number of dual cells, size of vector $\vec{\mu}$,
 $b_N^i \hat{=}$ nodal basis function (“tent function”) of $\mathcal{S}_{1,0}^0(\mathcal{M})$ belonging to the node inside C_i .

$u_N \hat{=}$ **piecewise linear interpolant** of vertex values μ_i

Note that u_N is not smooth across inner edges of \mathcal{M} . However, we do not care when computing $\mathbf{j} := \kappa(\mathbf{x}) \mathbf{grad} u_N$, because this flux is *only needed at edges of the dual mesh*, which lie inside triangles of \mathcal{M} (with the exception of single points that are irrelevant for the flux integrals).

Illustration of point evaluation

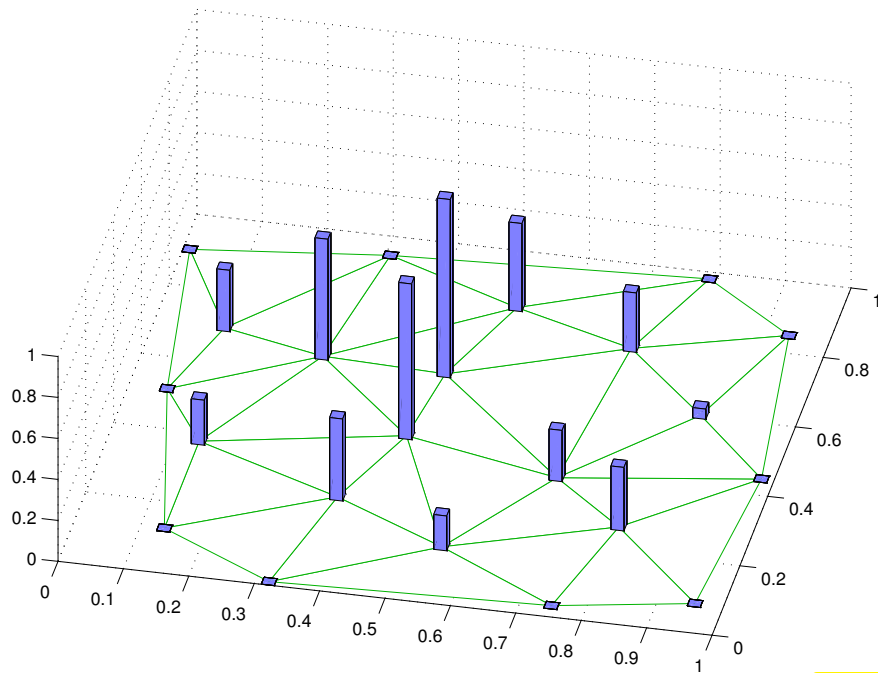


Fig. 132

vertex values μ_i on $\mathcal{V}(\mathcal{M})$

Illustration of point evaluation

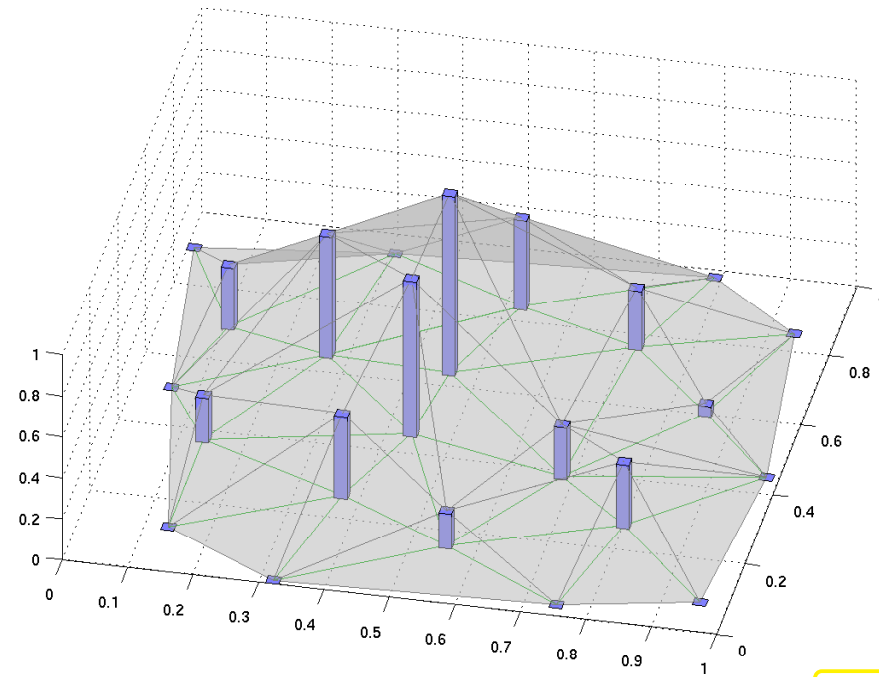


Fig. 133

p.w. linear interpolant $u_N := I_1 \vec{\mu} \in \mathcal{S}_{1,0}^0(\mathcal{M})$

Choice of **numerical flux**:

$$J_{ik} := - \int_{\Gamma_{ik}} \text{grad } I_1 \vec{\mu} \cdot \mathbf{n}_{ik} \, dS \quad (4.2.7)$$

(4.2.7) \Rightarrow (4.2.2) \Leftrightarrow one row of finite volume discretization matrix from

$$\begin{aligned} \sum_{k \in \mathcal{U}_i} \int_{\Gamma_{ik}} \mathbf{grad} \, |_1 \vec{\mu} \cdot \mathbf{n}_{ik} \, dS &= \underbrace{\mu_i \int_{\partial C_i} \mathbf{grad} \, b_N^i \, dS}_{= \text{matrix entry } (\mathbf{A})_{ii}} + \sum_{j \in \mathcal{U}_i} \mu_j \underbrace{\left(\sum_{k \in \mathcal{U}_i} \int_{\Gamma_{ik}} \mathbf{grad} \, b_N^j \cdot \mathbf{n}_{ik} \, dS \right)}_{= \text{matrix entry } (\mathbf{A})_{ij}} \\ &= \int_{C_i} f(\mathbf{x}) \, d\mathbf{x} . \\ \Rightarrow \quad (\mathbf{A})_{ij} &= \int_{\partial C_i} \mathbf{grad} \, b_N^j \cdot \mathbf{n}_i \, dS , \quad i, j \in \{1, \dots, N\} . \end{aligned} \quad (4.2.9)$$

$\mathbf{n}_i \hat{=}$ exterior unit normal vector to ∂C_i .

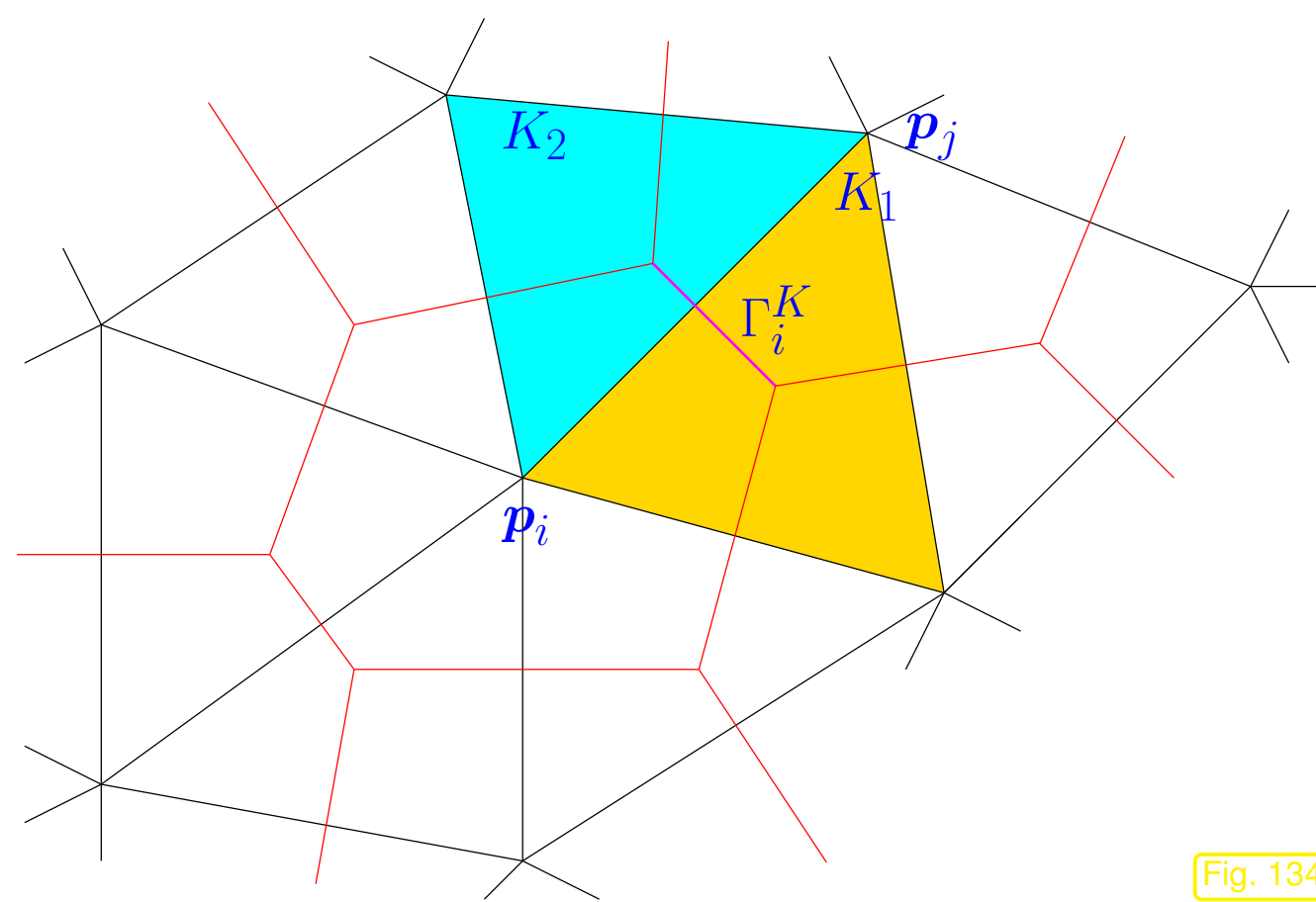


Fig. 134

Part of the boundary of the control volume C_i :

$$\Gamma_i^K := \partial C_i \cap K .$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Now, consider $i \neq j \leftrightarrow$ off-diagonal entries of \mathbf{A} :

First, we recall that the intersection of the support of the “tent function” b_N^j with ∂C_i is located inside $K_1 \cup K_2$, see Fig. 134.

$$\blacktriangleright (\mathbf{A})_{ij} = \int_{\Gamma_i^{K_1}} \mathbf{grad} b_N^j \cdot \mathbf{n}_i \, dS + \int_{\Gamma_i^{K_2}} \mathbf{grad} b_N^j \cdot \mathbf{n}_i \, dS .$$

Next observe that $\text{grad } b_N^j$ is piecewise constant, which implies

$$\text{div grad } b_N^j = 0 \quad \text{in } K_1 \quad , \quad \text{div grad } b_N^j = 0 \quad \text{in } K_2 . \quad (4.2.10)$$

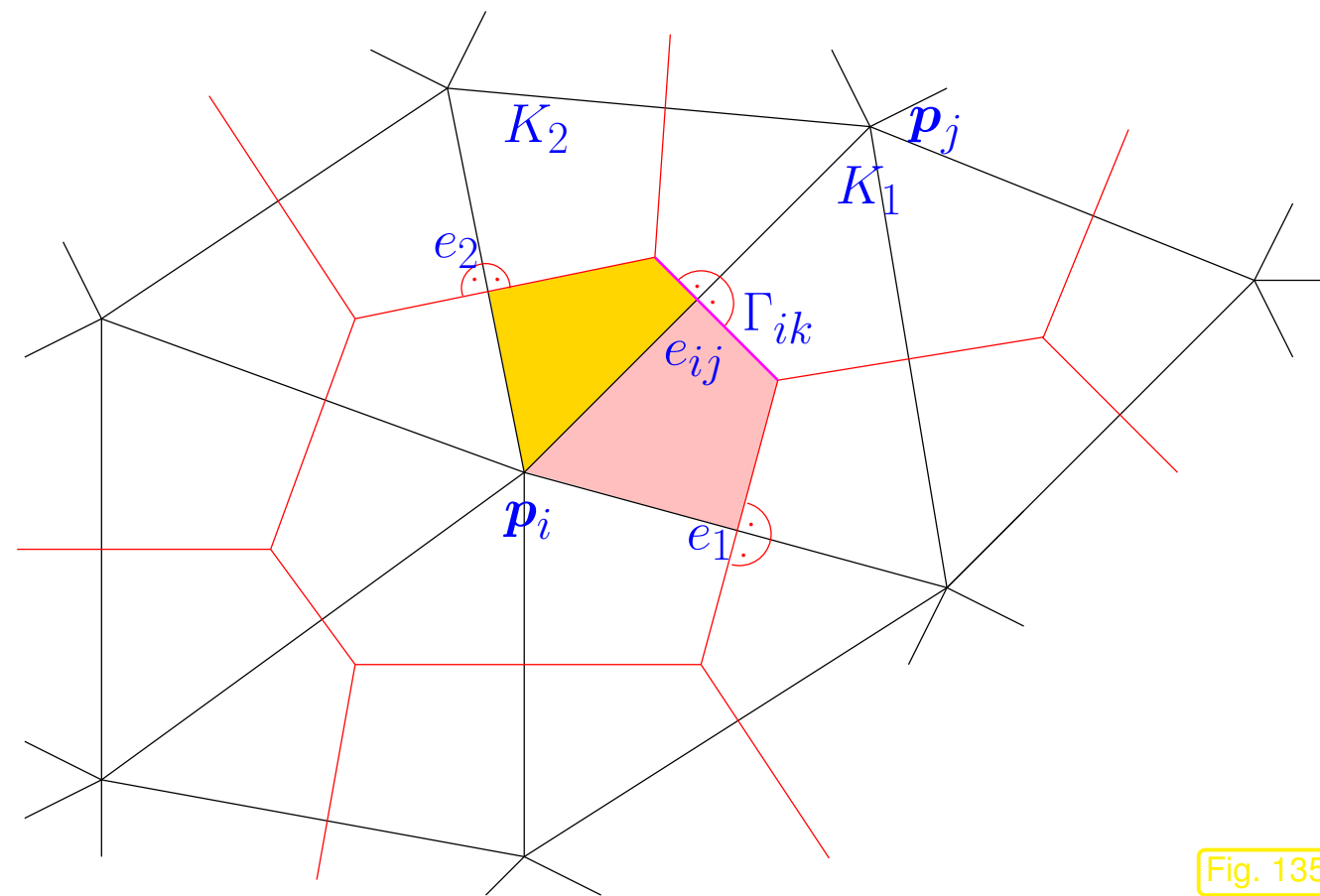


Fig. 135

Now apply Gauss' theorem Thm. 2.4.9 to the domains $C_i \cap K_1$ and $C_i \cap K_2$ (shaded in figure).

Also use again that $\text{grad } b_N^j \equiv \text{const}$ on K_1 and K_2 .

Another important observation; conclusion from $\text{grad } \lambda_i$ -formula from Sect. 3.2.5:

$$\begin{aligned} \text{grad } b_N^j &\perp e_1 \quad \text{in } K_1 , \\ \text{grad } b_N^j &\perp e_2 \quad \text{in } K_2 . \end{aligned}$$

$$\blacktriangleright (\mathbf{A})_{ij} = \frac{1}{2} \int_{e_1} \mathbf{grad} b_{N|K_1}^j \cdot \mathbf{n}_{e_1} dS + \frac{1}{2} \int_{e_{ij}} \mathbf{grad} b_{N|K_1}^j \cdot \mathbf{n}_{e_{ij}}^1 dS \\ + \frac{1}{2} \int_{e_{ij}} \mathbf{grad} b_{N|K_2}^j \cdot \mathbf{n}_{e_{ij}}^2 dS + \frac{1}{2} \int_{e_2} \mathbf{grad} b_{N|K_1}^j \cdot \mathbf{n}_{e_2} dS . \quad (4.2.11)$$


On the other hand, an entry of finite element Galerkin matrix $\tilde{\mathbf{A}}$ based on linear Lagrangian finite element space $\mathcal{S}_1^0(\mathcal{M})$ can be computed as, see Sect. 3.2.5:

$$(\tilde{\mathbf{A}})_{ij} = \int_{K_1} \mathbf{grad} b_N^j \cdot \mathbf{grad} b_N^i d\mathbf{x} + \int_{K_2} \mathbf{grad} b_N^j \cdot \mathbf{grad} b_N^i d\mathbf{x} .$$

Conduct local integration by parts using Green's first formula from Thm. 2.4.11 and taking into account (4.2.10) and the linearity of the local shape functions

$$\blacktriangleright (\tilde{\mathbf{A}})_{ij} = \int_{\partial K_1} (\mathbf{grad} b_{N|K_1}^j \cdot \mathbf{n}_1) b_N^i dS + \int_{\partial K_2} (\mathbf{grad} b_{N|K_2}^j \cdot \mathbf{n}_2) b_N^i dS \\ = \frac{1}{2}|e_1| \mathbf{grad} b_{N|K_1}^j \cdot \mathbf{n}_{e_1} + \frac{1}{2}|e_{ij}| \mathbf{grad} b_{N|K_1}^j \cdot \mathbf{n}_{e_{ij}}^1 + \\ \frac{1}{2}|e_2| \mathbf{grad} b_{N|K_2}^j \cdot \mathbf{n}_{e_2} + \frac{1}{2}|e_{ij}| \mathbf{grad} b_{N|K_2}^j \cdot \mathbf{n}_{e_{ij}}^2 .$$

This is the same value as for $(\mathbf{A})_{ij}$ from (4.2.11)! Similar considerations apply to the diagonal entries $(\mathbf{A})_{ii}$ and $(\tilde{\mathbf{A}})_{ii}$.



The finite volume discretization and the finite element Galerkin discretization spawn the same system matrix for the model problem (4.2.5).

Learning outcomes

The chapter aims to impart

- the gist of the “finite difference approach”: starting from strong form of a partial differential equation replace derivatives by difference quotients anchored on a regular grid (finite lattice).
- awareness that finite difference schemes can usually be recovered as finite element discretization (plus quadrature) on special (regular) meshes.
- the principles of the finite volume discretization of 2nd-order elliptic boundary value problems.

5

Convergence and Accuracy

In this chapter we resume the discussion of Sect. 1.6 of accuracy of a Galerkin solution u_N of a variational boundary value problem.

More precisely, we are going to study *convergence*, see Rem. 1.6.4

Focus: **finite element Galerkin discretization** of *linear* scalar 2nd-order elliptic boundary value problems in 2D, 3D

Prerequisites (what you should know by now):

- Boundary value problems (from equilibrium models, diffusion models): Sects. 2.4, 2.6,
- Variational formulation: Sect. 2.8, see also (2.3.5), (2.8.24), (3.0.1),
- Some Sobolev spaces and their norms: Sect. 2.2
- Abstract Galerkin discretization: Sect. 3.1,
- Lagrangian finite elements: Sects. 3.4, 3.2.

5.1 Galerkin error estimates

Setting: **linear variational problem** (1.4.7) in the form

$$u \in V_0: \quad \mathbf{a}(u, v) = \ell(v) \quad \forall v \in V_0, \quad (3.1.1)$$

- $V_0 \hat{=}$ (real) vector space, a space of functions $\Omega \mapsto \mathbb{R}$ for scalar 2nd-order elliptic variational problems,
- $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R} \hat{=}$ a bilinear form, see Def. 1.3.23,
- $\ell : V_0 \mapsto \mathbb{R} \hat{=}$ a linear form, see Def. 1.3.23,

☞ We want (3.1.1) to be related to a **quadratic minimization problem** (\rightarrow Def. 2.1.26):

Assumption 5.1.1. *The bilinear form $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ in (3.1.1) is symmetric and positive definite (\rightarrow Def. 2.1.32).*

\mathbf{a} supplies an inner product on V_0

\mathbf{a} induces energy norm $\|\cdot\|_{\mathbf{a}}$ on V_0 (\rightarrow Def. 2.1.35)

\Rightarrow We want (3.1.1) to be well posed, see Rem. 2.3.18

Assumption 5.1.2. *The right hand side functional $\ell : V_0 \mapsto \mathbb{R}$ from (3.1.1) is continuous w.r.t. to the energy norm (\rightarrow Def. 2.1.35) induced by \mathbf{a} :*

$$\exists C > 0: |\ell(u)| \leq C \|u\|_{\mathbf{a}} \quad \forall u \in V_0. \quad (2.2.3)$$

\Rightarrow An assumption to appease fastidious mathematicians:

Assumption 5.1.3. V_0 equipped with the energy norm $\|\cdot\|_{\mathbf{a}}$ is a *Hilbert space*, that is, complete.

Theorem 5.1.4 (Existence and uniqueness of solution of linear variational problem).

Under Assumptions 5.1.1–5.1.3 the linear variational problem has a unique solution $u \in V_0$.

This theorem is also known as **Riesz representation theorem** for continuous linear functionals.

Remark 5.1.5 (Well-posed 2nd-order linear elliptic variational problems).

For instance, Assumption 5.1.1 is satisfied for the bilinear form

$$\mathbf{a}(u, v) := \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} v \, d\mathbf{x}, \quad u, v \in H_0^1(\Omega), \quad (5.1.6)$$

and uniformly positive definite (\rightarrow Def. 2.1.12) coefficient tensor $\boldsymbol{\alpha} : \Omega \mapsto \mathbb{R}^{d,d}$, see Sect. 2.1.3.

For the right hand side functional

$$\ell(v) := \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} + \int_{\partial\Omega} h(\mathbf{x})v(\mathbf{x}) \, dS, \quad v \in H^1(\Omega),$$

we found in Sect. 2.2, see (2.2.24), and Rem. 2.9.8 that $f \in L^2(\Omega)$ and $h \in L^2(\partial\Omega)$ ensures Assumption 5.1.2.

Assumption 5.1.3 for \mathbf{a} from (5.1.6) is a deep result in the theory of Sobolev spaces [15, Sect. 5.2.3, Thm. 2].



Now consider Galerkin discretization of (3.1.1) based on Galerkin trial/test space $V_{0,N} \subset V_0$, $N := \dim V_{0,N} < \infty$:

$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N} . \quad (3.1.4)$$

Thm. 3.1.5: existence and uniqueness of Galerkin solution $u_N \in V_{0,N}$

Goal: bound *relevant norm* of **discretization error** $u - u_N$

Here: relevant norm = energy norm $\|\cdot\|_a$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Why is the energy norm a “relevant norm” ?

➤ Bounds of $\|u - u_N\|_a$ provide bounds for the **error in energy**, see Rem. 1.6.10, (1.6.17)

$$\begin{aligned} |J(u) - J(u_N)| &= \frac{1}{2} |\mathbf{a}(u, u) - \mathbf{a}(u_N, u_N)| = \left| \frac{1}{2} \mathbf{a}(u + u_N, u - u_N) \right| \\ &\stackrel{(2.1.37)}{\leq} \frac{1}{2} \|u - u_N\|_a \cdot \|u + u_N\|_a . \end{aligned}$$

(No doubt, energy is a key quantity for the solution of an equilibrium problem, which is defined as the minimizer of a potential energy functional.)

Other “relevant norms” were discussed in Sects. 1.6.1, 2.2:

- the mean square norm or $L^2(\Omega)$ -norm, see Def. 2.2.8,
- the supremum norm or $L^\infty(\Omega)$ -norm, see Def. 1.6.7.

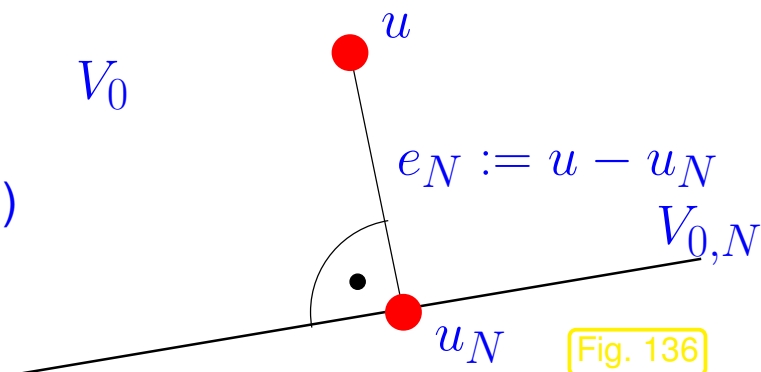
The Galerkin approach allows a remarkably simple bound of the energy norm of the discretization error $u - u_N$:

$$\begin{aligned} \mathbf{a}(u, v) &= \ell(v) & \forall v \in V_0, & & V_{0,N} \subset V_0 & \implies & \mathbf{a}(u - u_N, v_N) = 0 & \forall v_N \in V_{0,N}. \\ \mathbf{a}(u_N, v_N) &= \ell(v_N) & \forall v_N \in V_{0,N} \end{aligned}$$

Galerkin orthogonality

$$\mathbf{a}(u - u_N, v_N) = 0 \quad \forall v_N \in V_{0,N}. \quad (5.1.7)$$

[Geometric meaning for inner product $\mathbf{a}(\cdot, \cdot) \rightarrow$]



Discretization error $e_N := u - u_N$ “ $a(\cdot, \cdot)$ -orthogonal” to discrete trial/test space V_N

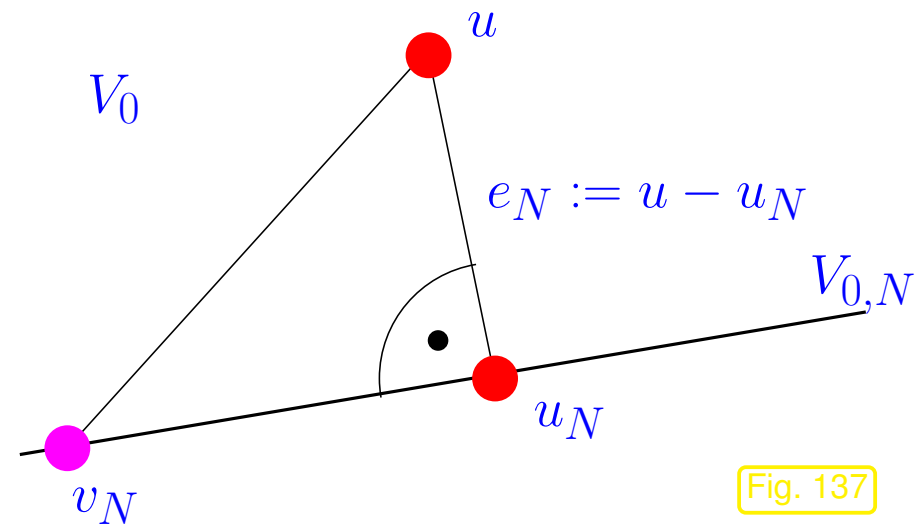


Fig. 137

Remark 5.1.8. If $a(\cdot, \cdot)$ is inner product on V :
“**Phythagoras’ theorem**” \rightarrow Fig. 137

$$\|u - v_N\|_a^2 = \|u - u_N\|_a^2 + \|u_N - v_N\|_a^2 . \tag{5.1.9}$$

(5.1.9) \triangleright ($v_N = 0$) simple formula for computation of energy norm of Galerkin discretization error in numerical experiments with known u .



Theorem 5.1.10 (Cea’s lemma).

Under Assumptions 5.1.1–5.1.3 the energy norm of the Galerkin discretization error satisfies

$$\|u - u_N\|_a = \inf_{v_N \in V_{0,N}} \|u - v_N\|_a .$$

Proof. Use bilinearity of \mathbf{a} and Galerkin orthogonality (5.1.7): for any $v_N \in V_{0,N}$

$$\|u - u_N\|_{\mathbf{a}}^2 = \mathbf{a}(u - u_N, u - u_N) = \mathbf{a}(u - v_N, u - u_N) + \underbrace{\mathbf{a}(v_N - u_N, u - u_N)}_{=0} .$$

Next, use the Cauchy-Schwartz inequality for the inner product \mathbf{a} :

$$\begin{aligned} \mathbf{a}(u, v) &\leq \|u\|_{\mathbf{a}} \|v\|_{\mathbf{a}} \quad \forall u, v \in V_0 . \\ \blacktriangleright \quad \|u - u_N\|_{\mathbf{a}}^2 &\leq \|u - v_N\|_{\mathbf{a}} \cdot \|u - u_N\|_{\mathbf{a}} , \end{aligned}$$

and cancel one factor $\|u - u_N\|_{\mathbf{a}}$. □

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Optimality of Galerkin solutions:

$$\underbrace{\|u - u_N\|_{\mathbf{a}}}_{\substack{\uparrow \\ \text{(norm of) discretization error}}} = \inf_{v_N \in V_{0,N}} \underbrace{\|u - v_N\|_{\mathbf{a}}}_{\substack{\uparrow \\ \text{best approximation error}}} , \quad (5.1.11)$$

☞ To assess accuracy of Galerkin solution: study capability of $V_{0,N}$ to approximate u !

▶ “Monotonicity” of best approximation: consider different trial/test spaces

$$\begin{array}{l} V_{0,N}, V'_{0,N} \subset V_0, \\ V_{0,N} \subset V'_{0,N} \end{array} \Rightarrow \inf_{v_N \in V'_{0,N}} \|u - v_N\|_a \leq \inf_{v_N \in V_{0,N}} \|u - v_N\|_a .$$

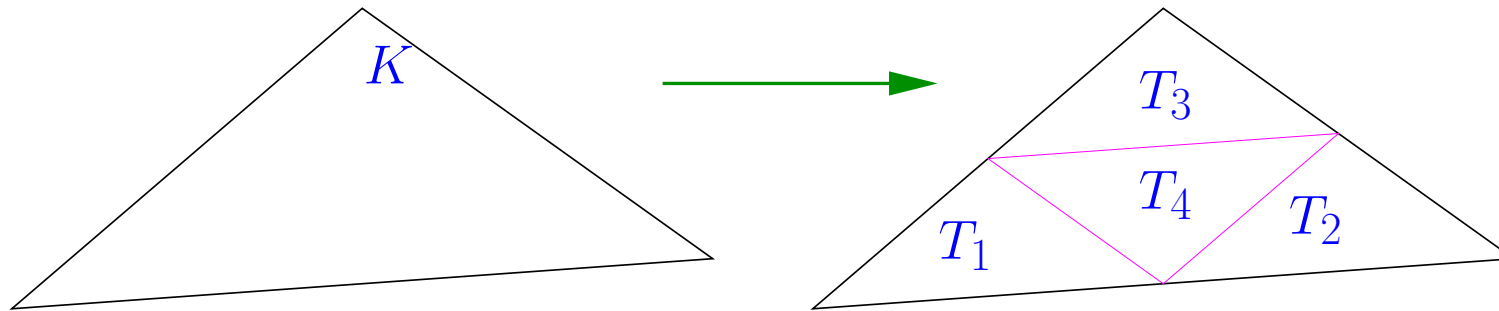
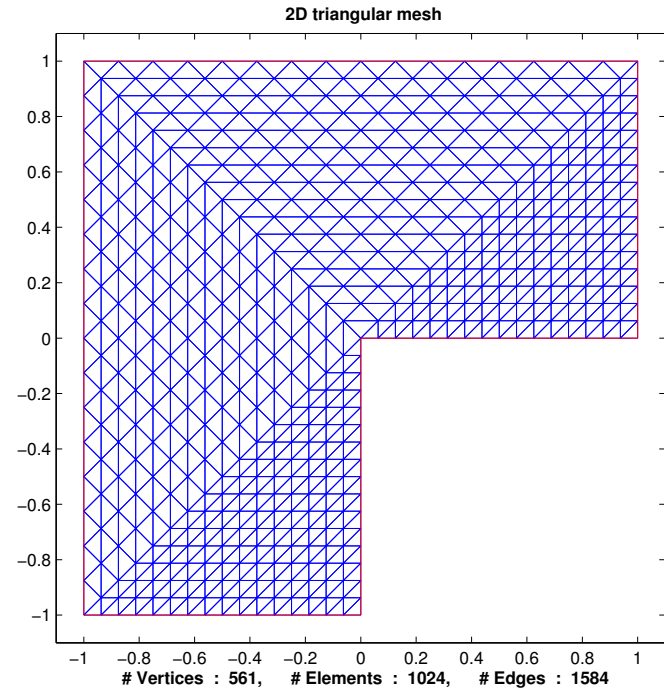
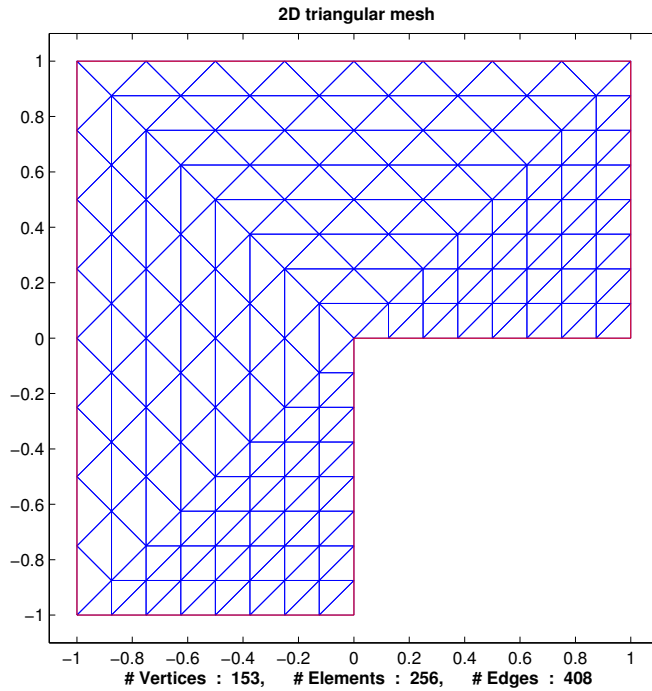
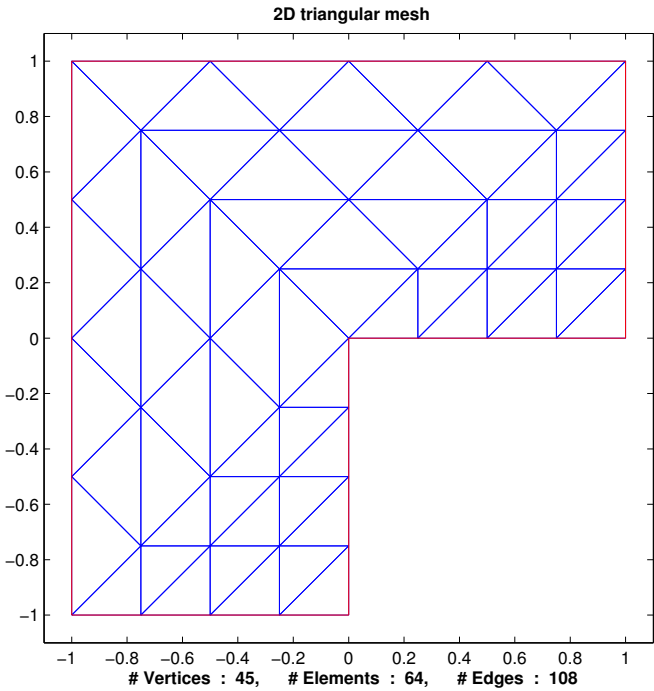
Enhance accuracy by enlarging (“refining”) trial space.

Now return to finite element Galerkin discretization of linear 2nd-order elliptic variational problems.

How to achieve refinement of FE space ?

- **h-refinement:** replace \mathcal{M} (underlying $V_{0,N}$) \rightarrow \mathcal{M}' (underlying larger discrete trial space $V'_{0,N'}$)

Example 5.1.12 (regular refinement of triangular mesh in 2D).



Regular refinement of triangle K into four congruent triangles T_1, T_2, T_3, T_4

- **p-refinement:** replace $V_{0,N} := \mathcal{S}_p^0(\mathcal{M})$, $p \in \mathbb{N}$ with $V'_{0,N} := \mathcal{S}_{p+1}^0(\mathcal{M}) \Rightarrow V_{0,N} \subset V'_{0,N}$

The extreme case of p -refinement amounts to the use of *global* polynomials on Ω as trial and test functions \triangleright (polynomial) **spectral Galerkin method**, see Sect. 1.5.1.1.

Combination of h-refinement and p-refinement ? OF COURSE (**hp-refinement**, [29])

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

5.2 Empirical (asymptotic) convergence of FEM

Recall from Sect. 1.6.2:

Crucial: convergence is an *asymptotic notion* !

sequence of discrete models \Rightarrow sequence of approximate solutions $(u_N^{(i)})_{i \in \mathbb{N}}$
 \Rightarrow study sequence $(\|u_N^{(i)} - u\|)_{i \in \mathbb{N}}$

created by *variation* of a **discretization parameter**:

In this section some numerical experiments will demonstrate

- meaningful notions of “discretization parameters”,
- qualitative behaviors of the sequence $(\|u_N^{(i)} - u\|)_{i \in \mathbb{N}}$ we may expect,

for Lagrangian finite element discretization of linear scalar 2nd-order elliptic variational problems (\rightarrow Sect. 2.8).

Sequences of discrete models will be generated by either h -refinement or p -refinement.

Model problem: Dirichlet problem for Poisson equation:

$$-\Delta u = f \in L^2(\Omega) \quad \text{in } \Omega, \quad u = g \in C^0(\partial\Omega) \quad \text{on } \partial\Omega. \quad (5.2.1)$$

Example 5.2.2 (Convergence of linear and quadratic Lagrangian finite elements in energy norm).

Setting: $\Omega =]0, 1[^2$, $f(x_1, x_2) = 2\pi^2 \sin(\pi x_1) \sin(\pi x_2)$, $\mathbf{x} \in \Omega$, $g = 0$

➤ **Smooth** solution $u(x, y) = \sin(\pi x) \sin(\pi y)$.

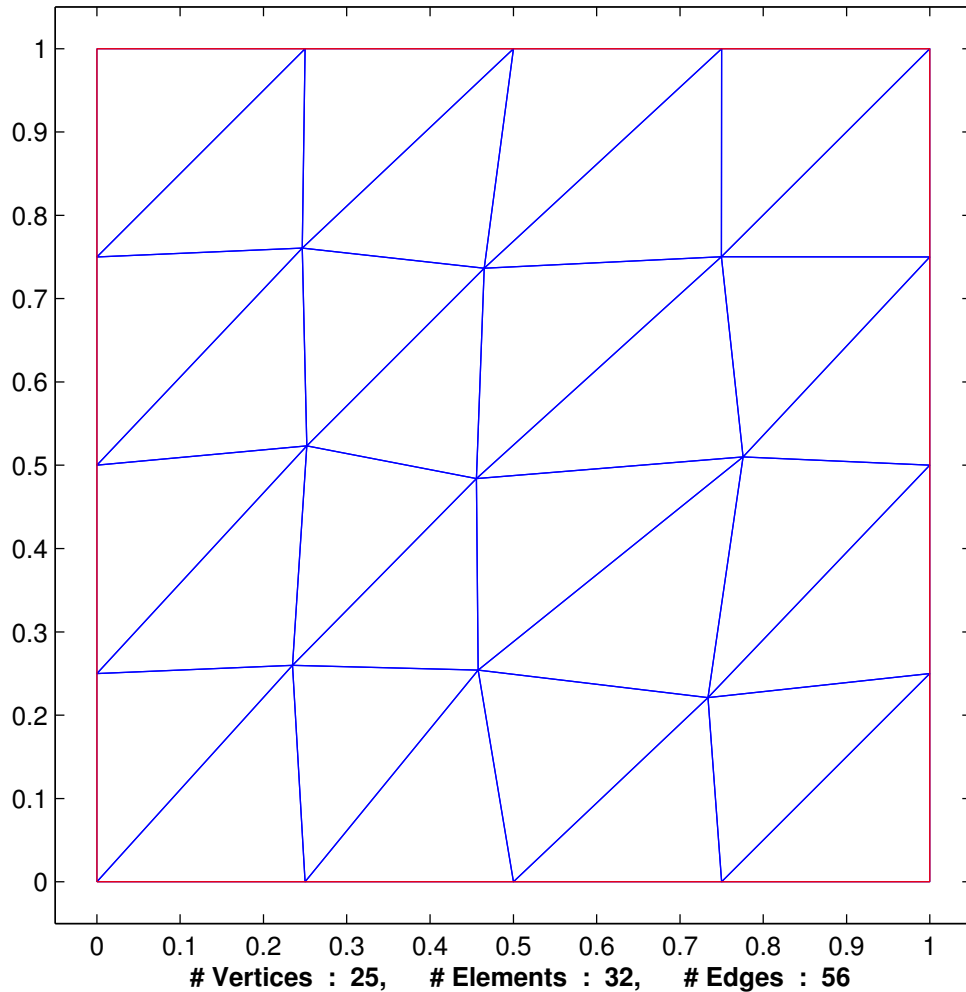
- Galerkin finite element discretization based on triangular meshes and
 - linear Lagrangian finite elements, $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$ (\rightarrow Sect. 3.2),
 - quadratic Lagrangian finite elements, $V_{0,N} = \mathcal{S}_{2,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$ (\rightarrow Ex. 3.4.2),
- quadrature rule (3.5.38) for assembly of local load vectors (\rightarrow Sect. 3.5.4),

Monitored: $H^1(\Omega)$ -semi-norm (\rightarrow Def. 2.2.15) of the Galerkin discretization error $u - u_N$

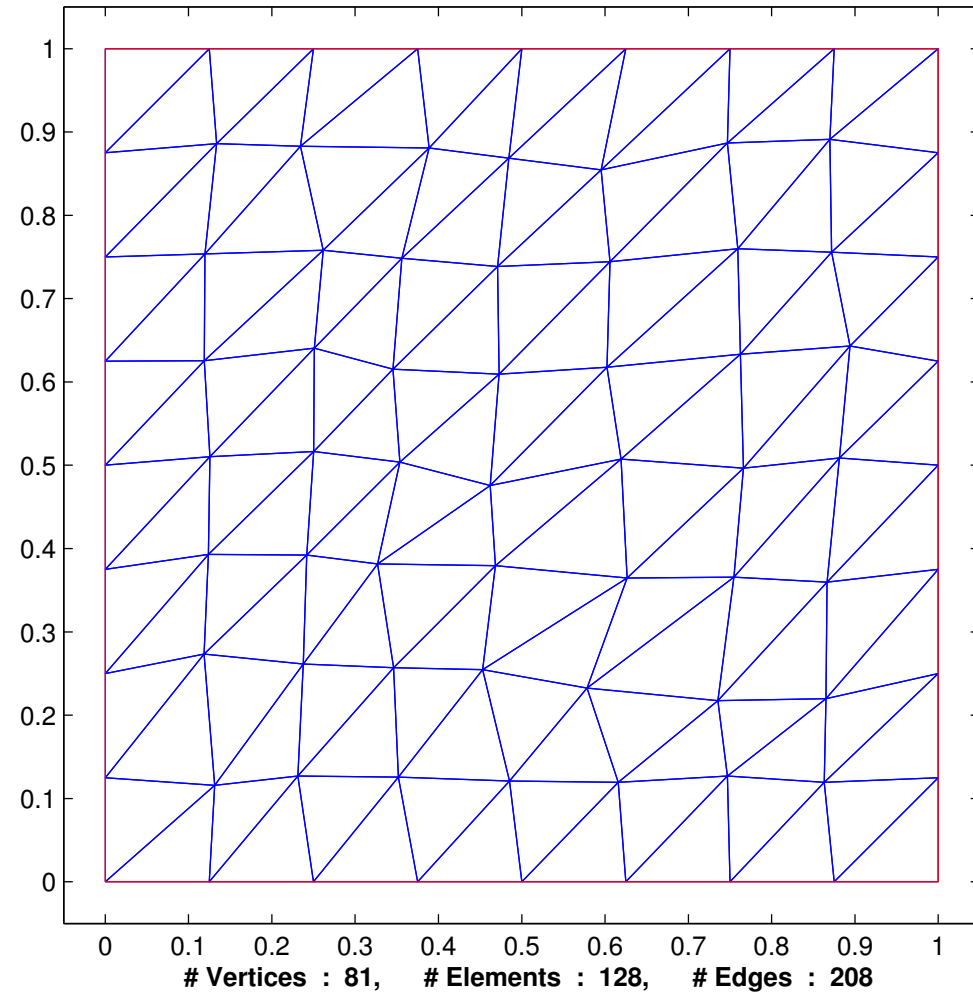
▶ **Approximate** (*) computation of $|u - u_N|_{H^1(\Omega)}$ on a **sequence** of meshes (created by successive regular refinement (\rightarrow Ex. 5.1.12) of coarse initial mesh)

(*): use of local quadrature rule (3.5.38) (on current FE mesh)

2D triangular mesh



2D triangular mesh



Unstructured triangular meshes of $\Omega =]0, 1[^2$ (two coarsest specimens)

Focus on **asymptotics** entails studying a

norm of the discretization error as function of a (real, cardinal) **discretization parameter**.

The discretization parameter must be linked to the **resolution** (“capability to approximate generic solution”) of the Galerkin trial/test space $V_{0,N}$. Possible choices are

- $N := \dim V_{0,N}$ as a measure of the “cost” of a discretization, see Sect. 1.6.2,
- the maximum “size” of mesh cells, expressed by the mesh width $h_{\mathcal{M}}$ (\rightarrow Def. 5.2.3), see below.

Definition 5.2.3 (Mesh width).

*Given a mesh $\mathcal{M} = \{K\}$, its **mesh width** $h_{\mathcal{M}}$ is defined as*

$$h_{\mathcal{M}} := \max\{\text{diam } K : K \in \mathcal{M}\} \quad , \quad \text{diam } K := \max\{|\mathbf{p} - \mathbf{q}| : \mathbf{p}, \mathbf{q} \in K\} .$$

This generalizes the concept of “mesh width” introduced in Sect. 1.5.1.2.

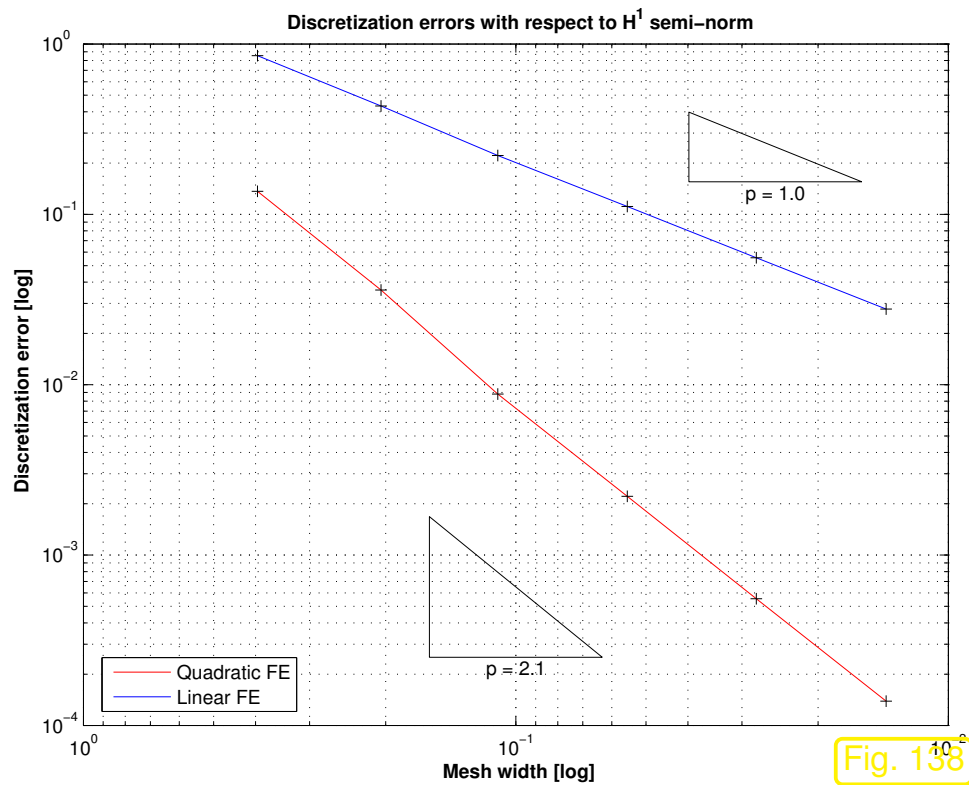


Fig. 138

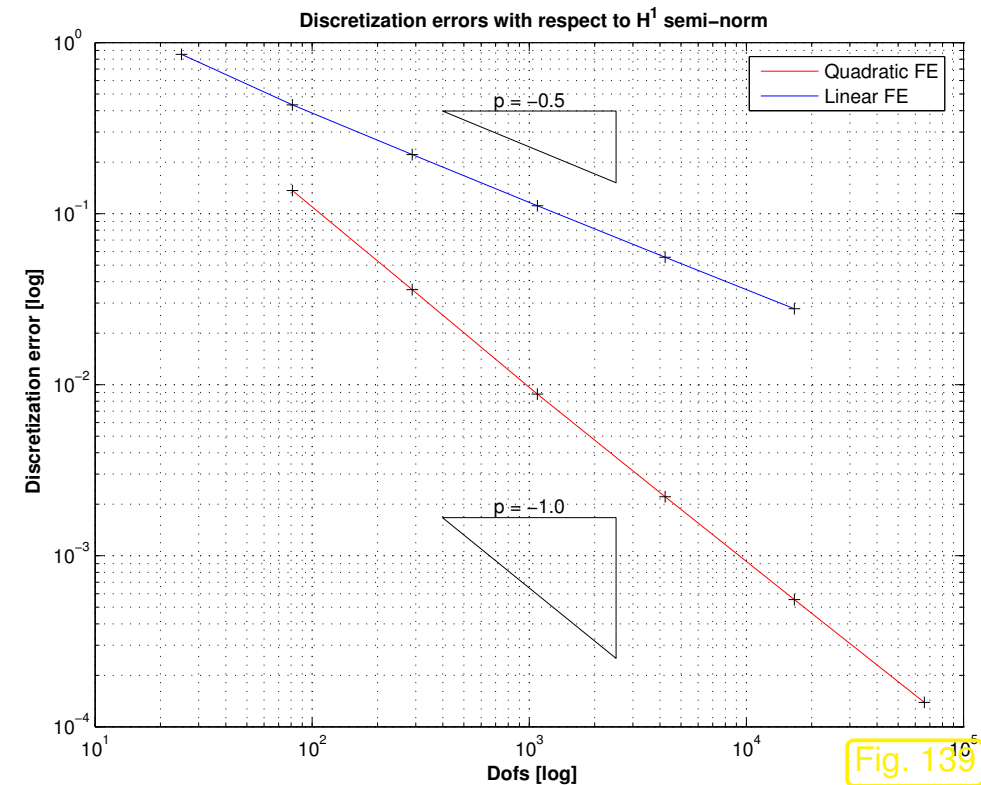


Fig. 139

$H^1(\Omega)$ -semi-norm of discretization error on unit square ($- \leftrightarrow p = 1$, $- \leftrightarrow p = 2$)

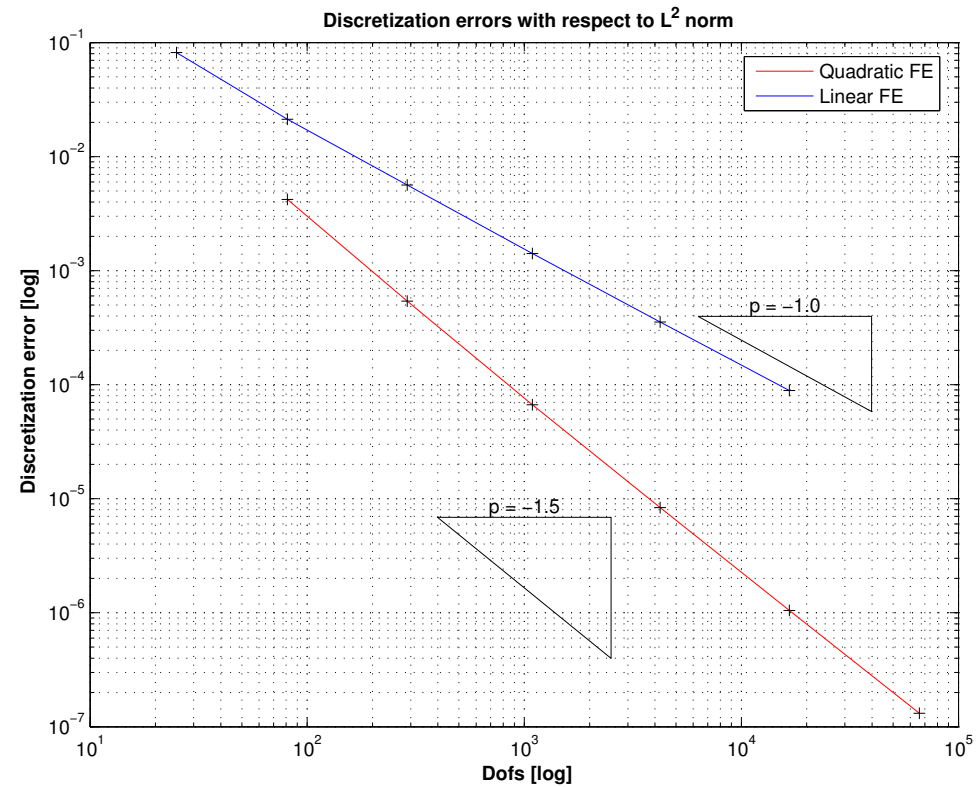
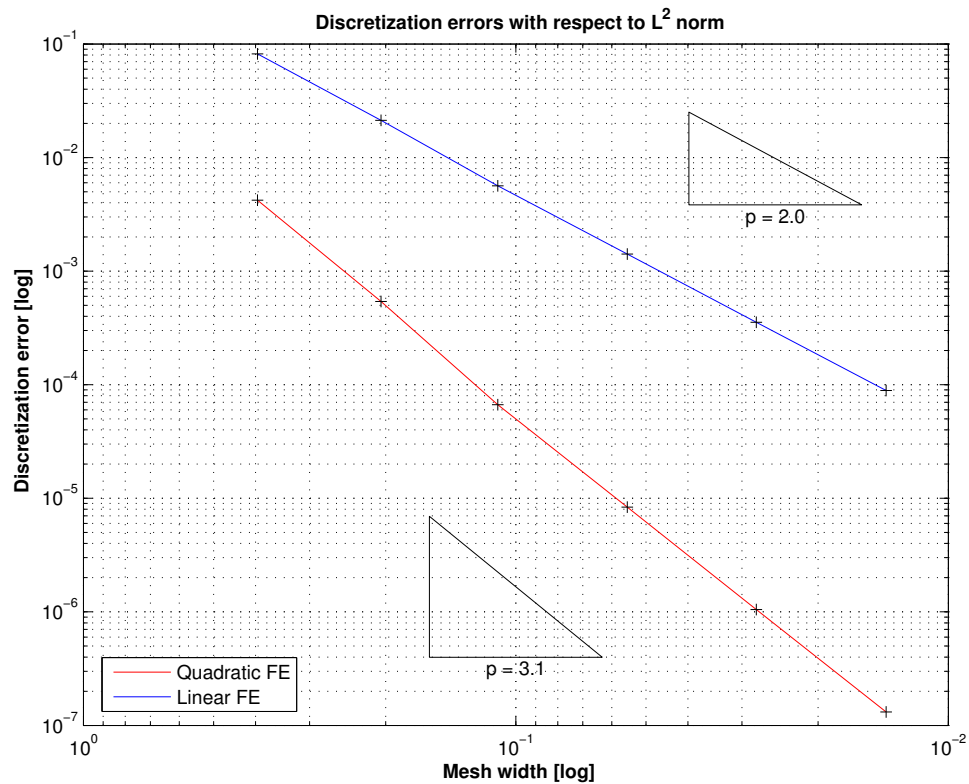
Recall type of convergence (**algebraic convergence** vs. **exponential convergence**) from Def. 1.6.32 and how to detect them in a numerical experiment by inspecting appropriate graphs, see Rem. 1.6.35.

- Observations:
- Algebraic rates of convergence in terms of N and h
 - Quadratic Lagrangian FE converge with double the rate of linear Lagrangian FE

Example 5.2.4 (Convergence of linear and quadratic Lagrangian finite elements in L^2 -norm).

Setting as above in Ex. 5.2.2, $\Omega =]0, 1[$.

Monitored: **asymptotics** of the $L^2(\Omega)$ -semi-norm of the Galerkin discretization error (approximate computation of $\|u - u_N\|_{L^2(\Omega)}$ by means of local quadrature rule (3.5.38) on a sequence of meshes created by successive regular refinement (\rightarrow Ex. 5.1.12) of coarse initial mesh).



$L^2(\Omega)$ -norm of discretization error on unit square (— $\leftrightarrow p = 1$, — $\leftrightarrow p = 2$)

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

- Observations:
- Linear Lagrangian FE ($p = 1$) $\Rightarrow \|u - u_N\|_0 = O(h_{\mathcal{M}}^2) = O(N^{-1})$
 - Quadratic Lagrangian FE ($p = 2$) $\Rightarrow \|u - u_N\|_0 = O(h_{\mathcal{M}}^3) = O(N^{-1.5})$

For the “conversion” of convergence rates with respect to the mesh width $h_{\mathcal{M}}$ and $N := \dim \mathcal{S}_p^0(\mathcal{M})$, note that in 2D for Lagrangian finite element spaces with fixed polynomial degree (\rightarrow Sect. 3.4) and meshes created by global (that is, carried out everywhere) regular refinement

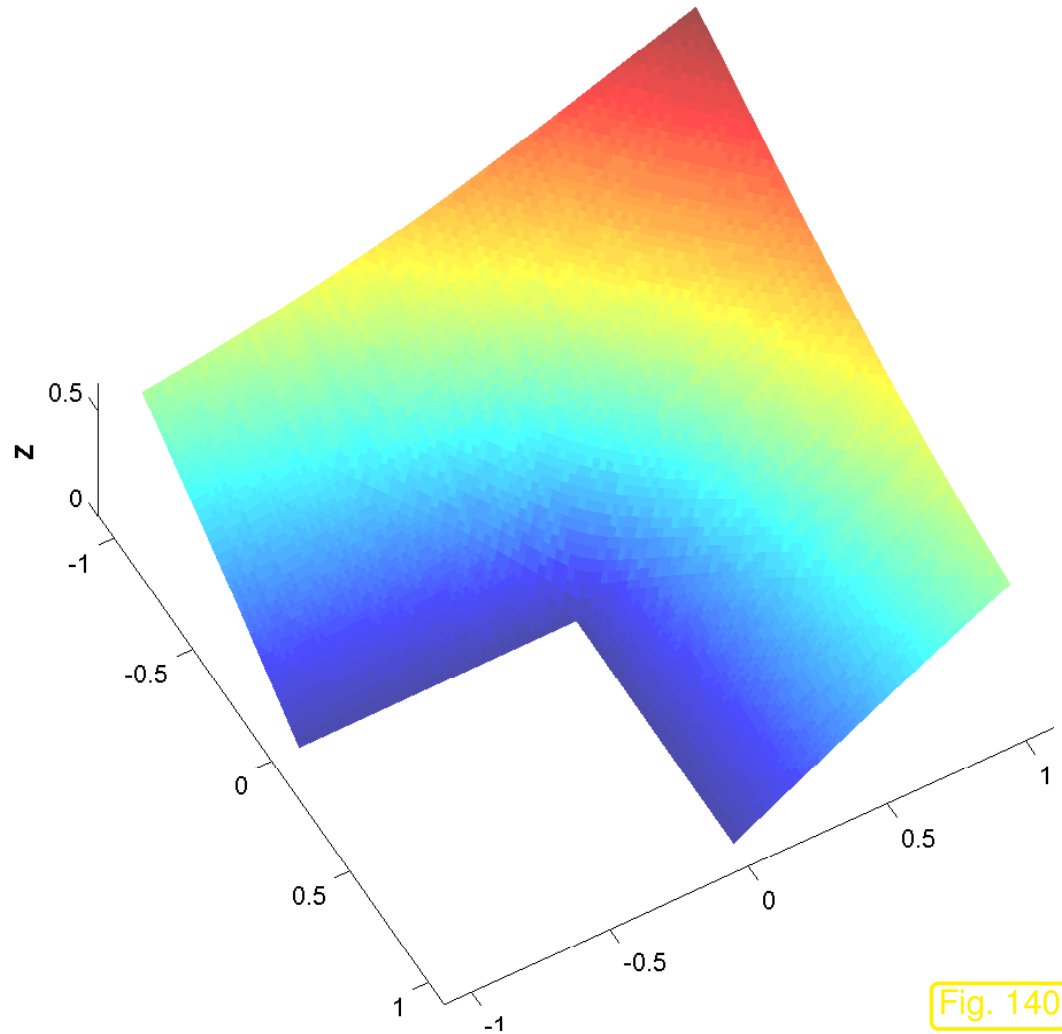
$$N = O(h_{\mathcal{M}}^{-2}) . \tag{5.2.5}$$



Example 5.2.6 (h -convergence of Lagrangian FEM on L-shaped domain).

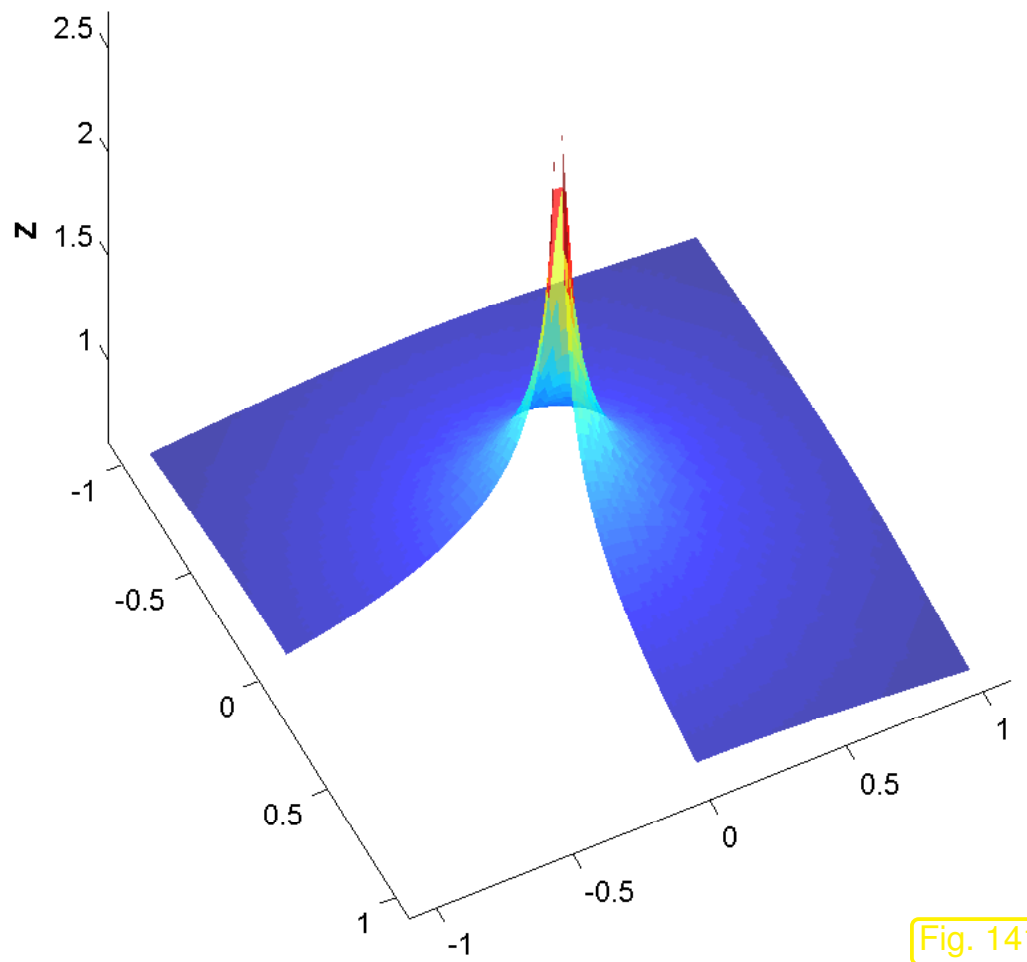
Setting: Model problem (5.2.1) on $\Omega =]-1, 1[^2 \setminus ([0, 1[\times]-1, 0[)$, exact solution (in polar coordinates)

$$u(r, \varphi) = r^{2/3} \sin(2/3\varphi) \quad \blacktriangleright \quad f = 0, g = u|_{\partial\Omega}.$$



Exact solution u

Fig. 140



Norm of gradient $\|\text{grad } u\|$

Fig. 141

Note: $\text{grad } u$ has a singularity at 0 , that is, “ $\|\text{grad } u(0)\| = \infty$ ”.

- Galerkin finite element discretization based on triangular meshes and

- linear Lagrangian finite elements, $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$ (\rightarrow Sect. 3.2),
- quadratic Lagrangian finite elements, $V_{0,N} = \mathcal{S}_{2,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$ (\rightarrow Ex. 3.4.2),
- linear/quadratic interpolation of Dirichlet data to obtain offset function $u_0 \in \mathcal{S}_{p,0}^0(\mathcal{M})$, $p = 1, 2$, see Sect. 3.5.5, Ex. 3.5.43.

Sequence of meshes created by successive regular refinement (\rightarrow Ex. 5.1.12) of coarse initial mesh, see Fig. 142.

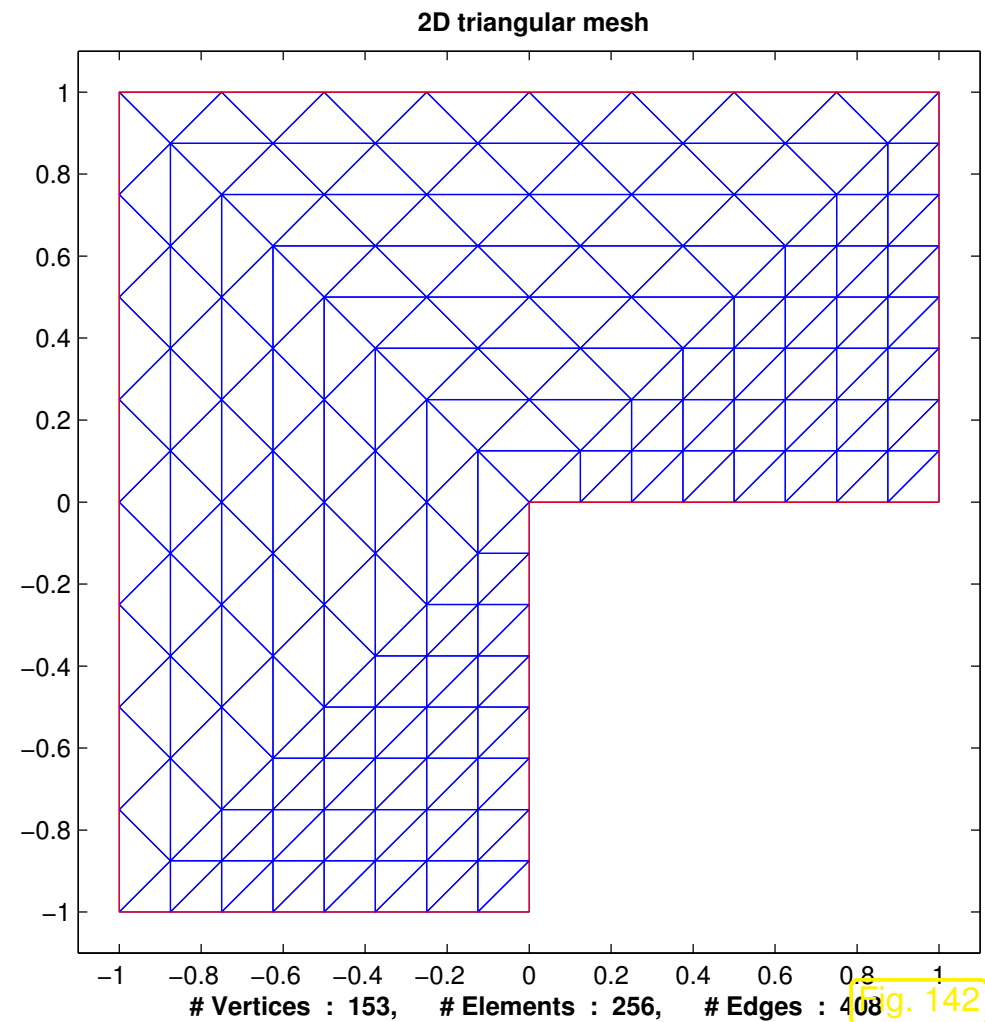
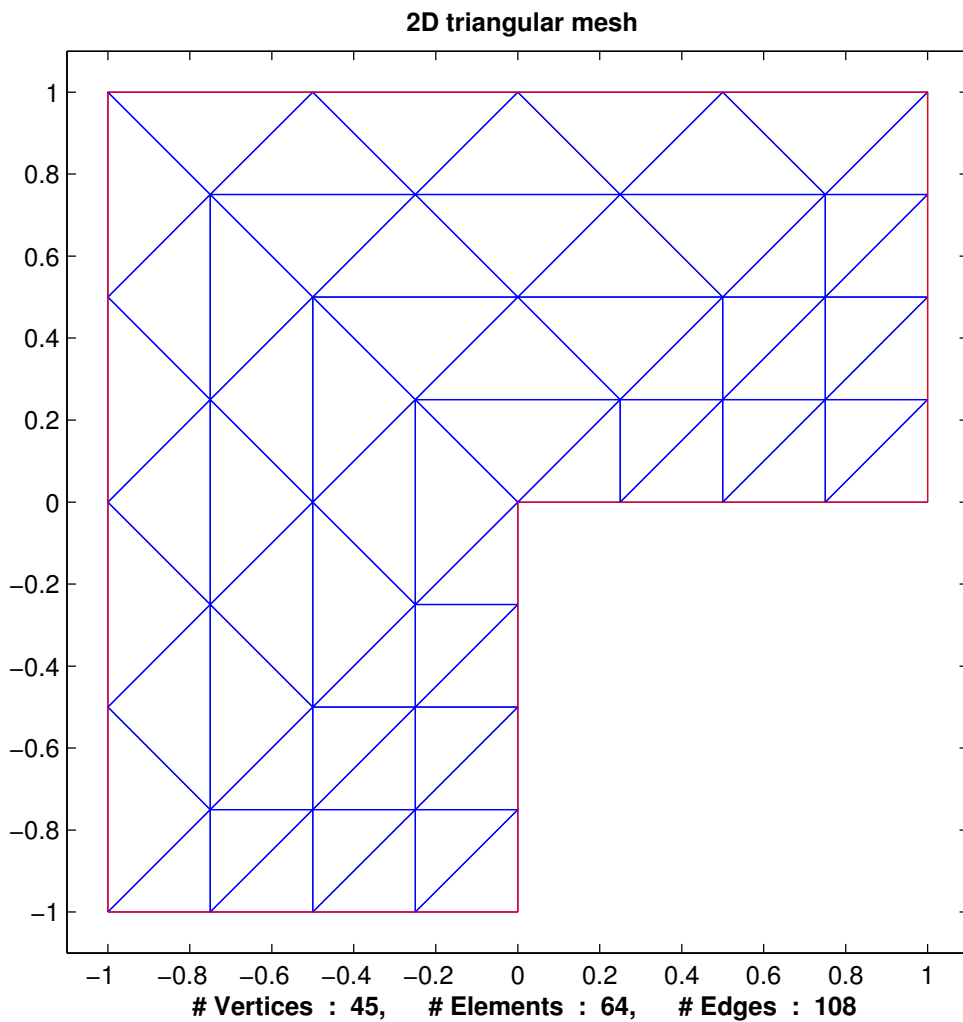
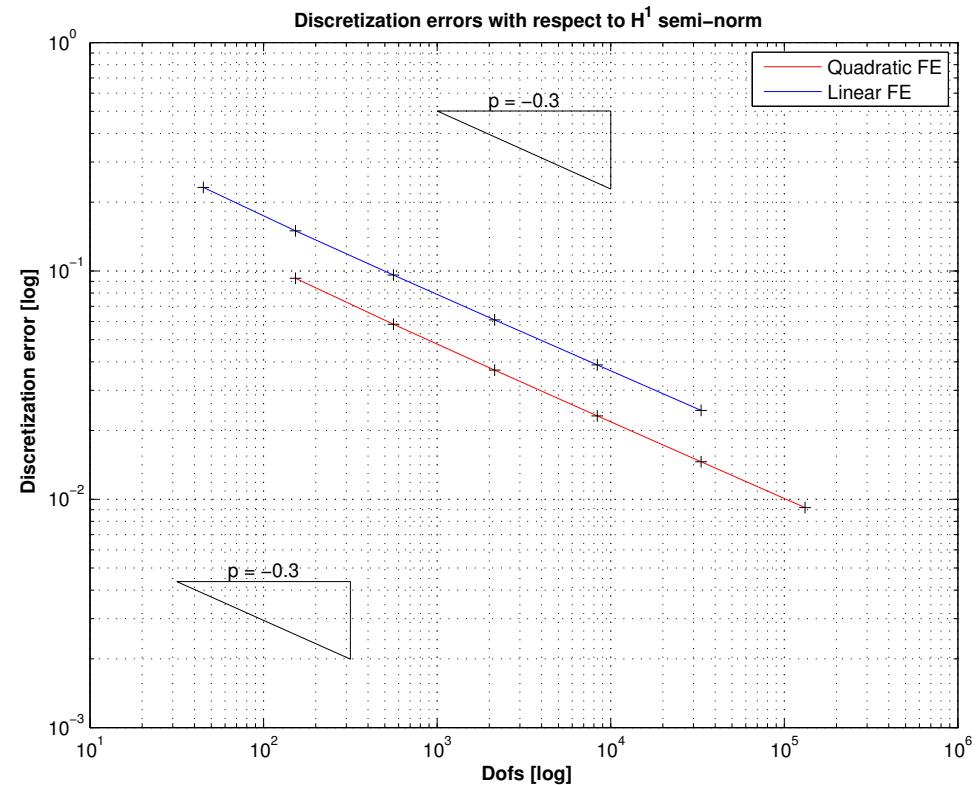
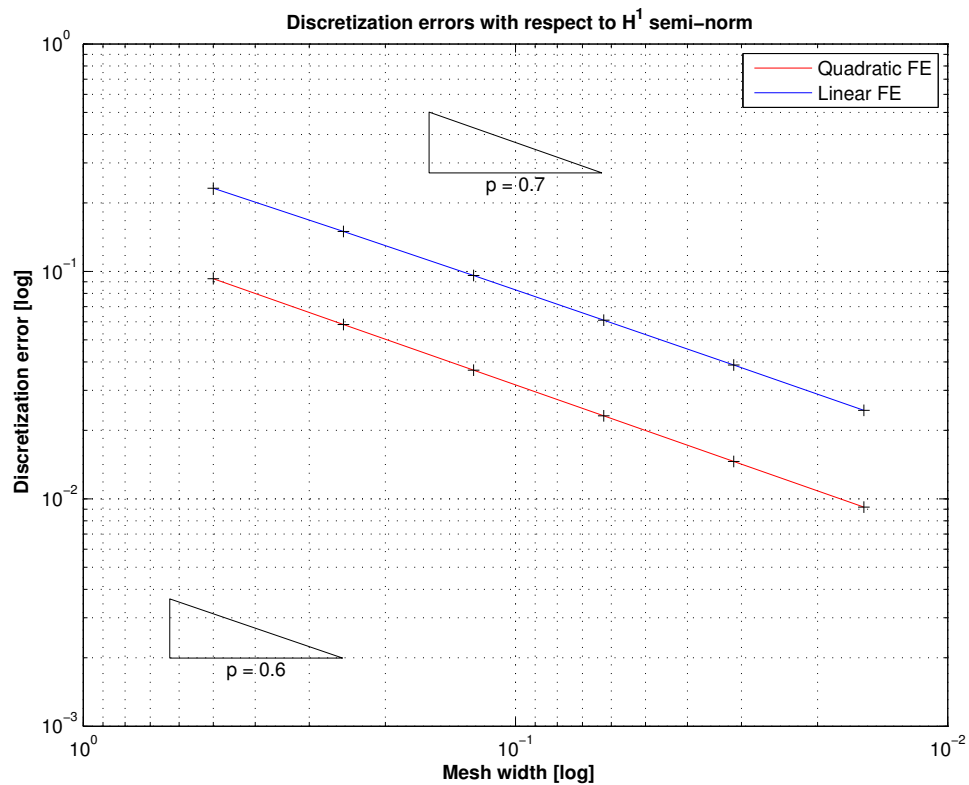


Fig. 142

Unstructured triangular meshes of $\Omega =]-1, 1[^2 \setminus ([0, 1[\times]-1, 0[)$ (two coarsest specimens)

Approximate computation of $|u - u_N|_{H^1(\Omega)}$ by using local quadrature formula (3.5.38) on FE meshes.



$H^1(\Omega)$ -semi-norm of discretization error on “L-shaped” domain (— $\leftrightarrow p = 1$, — $\leftrightarrow p = 2$)

- Observations:
- For **both** $p = 1, 2$: $\|u - u_N\|_1 = O(N^{-1/3})$
 - **No gain** from higher polynomial degree

Conjecture: singularity of $\text{grad } u$ at $\mathbf{x} = 0$ seems to foil faster algebraic convergence of quadratic Lagrangian finite element solutions!

Example 5.2.7 (Convergence of Lagrangian FEM for p -refinement).

- Model BVP as in Ex. 5.2.2 ($\Omega =]0, 1[^2$) and Ex. 5.2.6 (L-shaped domain $\Omega =]-1, 1[^2 \setminus (]0, 1[\times]-1, 0[)$).
- Galerkin finite element discretization based on $\mathcal{S}_p^0(\mathcal{M})$, $p = 1, 2, 3, 5, 6, 7, 8, 9, 10$, built on a *fixed* coarse triangular mesh of Ω .

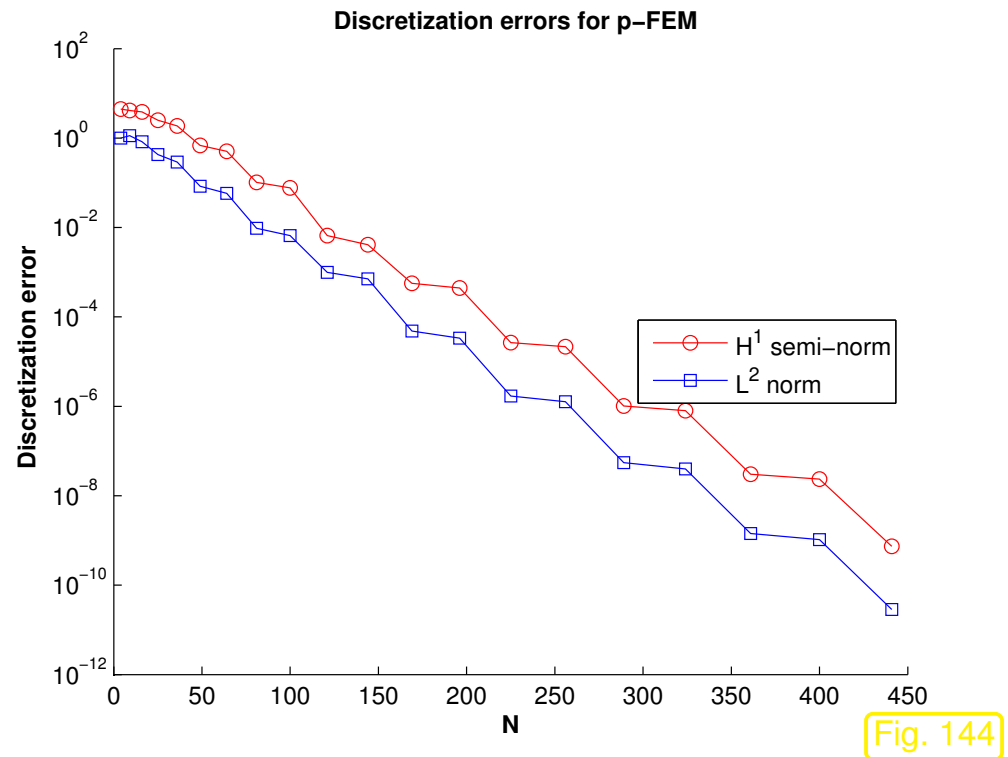
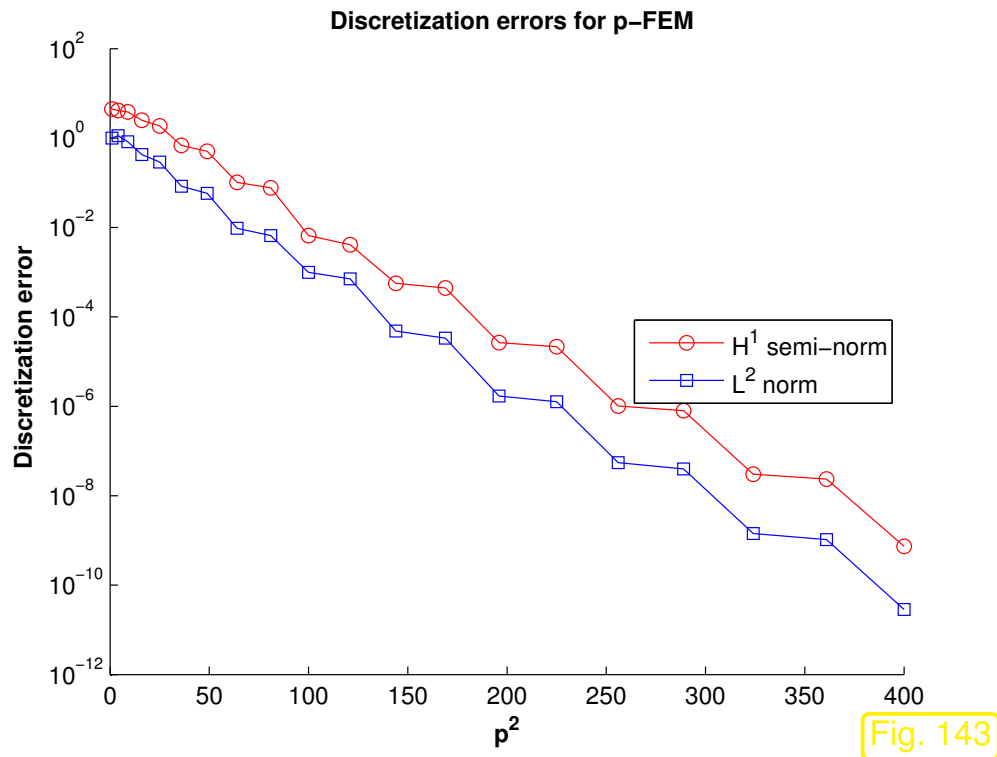
➤ p -refinement

Monitored: $H^1(\Omega)$ -semi-norm (energy norm) and $L^2(\Omega)$ -norm of discretization error as functions of polynomial degree p and $N := \dim \mathcal{S}_p^0(\mathcal{M})$.

(Computation of norms by means of local quadrature rule of order 19!. This renders the error in norm computations introduced by numerical quadrature negligible.)

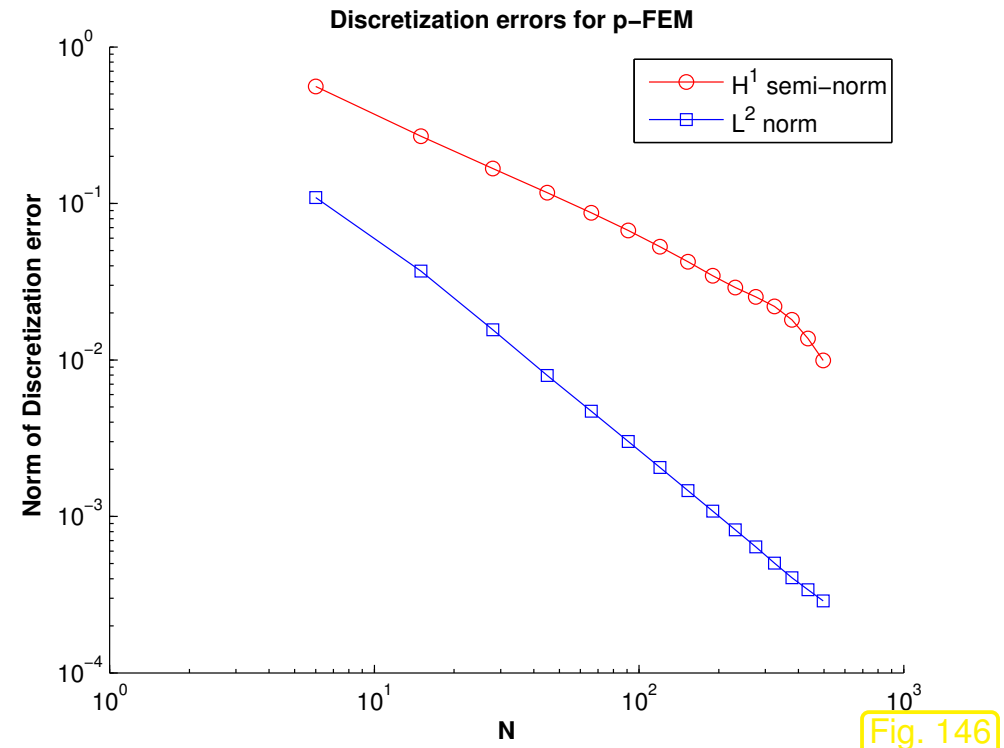
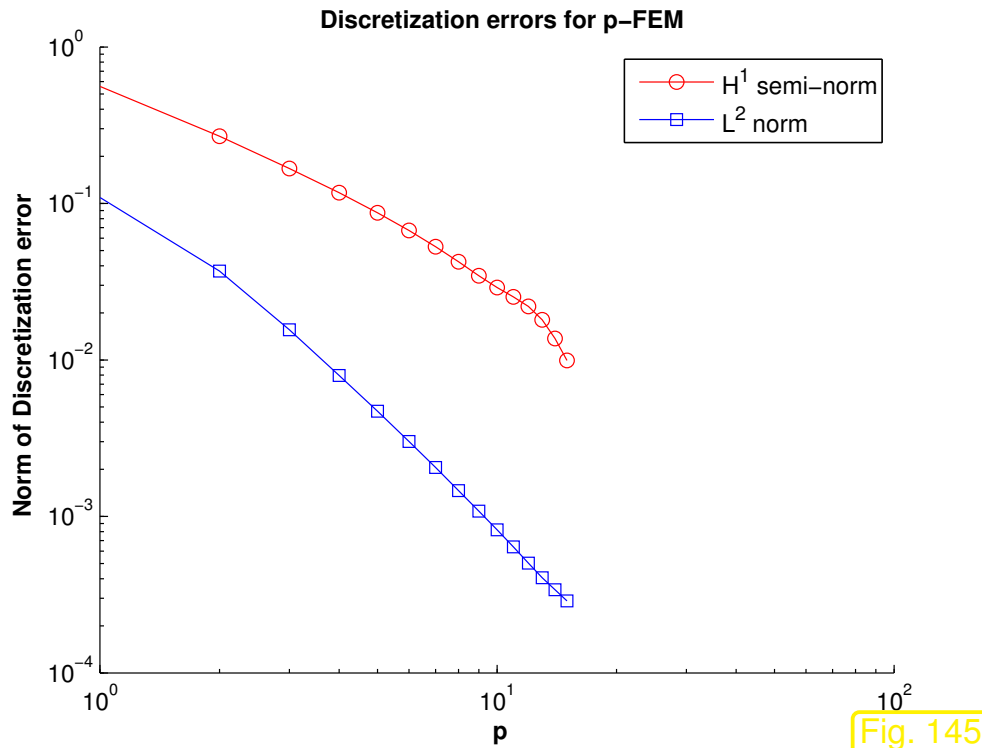
Meaningful discretization parameters for asymptotic study of error norms:

- polynomial degree p for Lagrangian finite element space,
- $N := \dim V_{0,N}$ as a measure of the “cost” of a discretization, see Sect. 1.6.2.



$\Omega =]0, 1[$: behavior of $|u - u_N|_{H^1(\Omega)}$ for different polynomial degrees.
 Lagrangian FEM: p -convergence for smooth (analytic) solution

Observation: exponential convergence of FE discretization error, *cf.* the behavior of the discretization error of spectral collocation and polynomial spectral Galerkin methods in 1D, Ex. 1.6.31.



Lagrangian FEM: p -convergence for solution with singular gradient

R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

Observation: Only algebraic convergence of FE discretization error!

The suspect: “singular behavior” of $\text{grad } u$ at $x = 0$.

5.3 A priori finite element error estimates

We are interested in **a priori estimates** of norms of the discretization error.

A priori estimate: bounds for error norms available **before** computing approximate solutions.



A posteriori estimate: bounds for error norms based on an approximate solution **already computed**.

Results of Sect. 5.1 provide handle on a priori estimate for Galerkin discretization error:

Optimality (5.1.11) of Galerkin solution  a priori error estimates

Thm. 5.1.10 ➤

Estimate energy norm of Galerkin discretization error $u - u_N$
by bounding best approximation error
for exact solution u in finite element space:

$$\underbrace{\|u - u_N\|_a}_{\uparrow} \leq \inf_{v_N \in V_{0,N}} \underbrace{\|u - v_N\|_a}_{\uparrow}, \quad (5.1.11)$$

(norm of) discretization error

best approximation error

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

How to estimate best approximation error

$$\inf_{v_N \in V_{0,N}} \|u - v_N\|_V ?$$

➤ Well, given solution u seek candidate function $w_N \in V_{0,N}$ with

$$\|u - w_N\|_V \approx \inf_{v_N \in V_N} \|u - v_N\|_V .$$

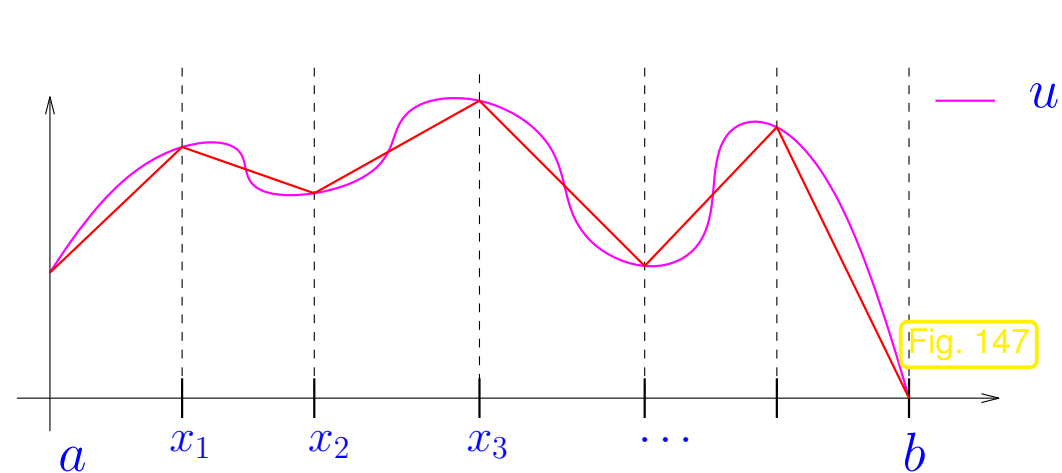
Natural choice:

 w_N by interpolation/averaging of (*unknown, but existing*) u

5.3.1 Estimates for linear interpolation in 1D

Computational domain (\rightarrow Sect. 1.4): interval $\Omega = [a, b]$

Given: mesh of Ω (\rightarrow Sect. 1.5.1.2): $\mathcal{M} := \{]x_{j-1}, x_j[: j = 1, \dots, M\}, M \in \mathbb{N}$



Piecewise linear interpolant of $u \in C^0([a, b])$

$$I_1 u \in \mathcal{S}_1^0(\mathcal{M}), \tag{5.3.1}$$

$$(I_1 u)(x_j) = u(x_j), \quad j = 0, \dots, M. \tag{5.3.2}$$

\triangleright [21, Sect. 3.6.1]

Goal: Bound suitable norm (\rightarrow Sect. 1.6.1) of **interpolation error** $u - I_1 u$ in terms of geometric quantities (*) characterizing \mathcal{M} .

(*): A typical such quantity is the **mesh width** $h_{\mathcal{M}} := \max_j |x_j - x_{j-1}|$

Now we investigate different norms of the interpolation error.

• $\|u - I_1 u\|_{L^\infty([a,b])}$, see [21, Sect. 9.1] and [21, Sect. 9.4.1]: from [21, Thm. 9.1.7] for $n = 1$: for $u \in C^2([a,b])$

$$\max_{x_{j-1} \leq x \leq x_j} u(x) - (I_1 u)(x) = \frac{1}{4} u''(\xi_t) (x_j - x_{j-1})^2, \quad \text{for some } \xi_t \in]x_{j-1}, x_j[, \quad (5.3.3)$$

with **local linear interpolant** $(I_1 u)(x) = \frac{x - x_{j-1}}{x_j - x_{j-1}} u(x_j) - \frac{x_j - x}{x_j - x_{j-1}} u(x_{j-1}). \quad (5.3.4)$

(5.3.3) \blacktriangleright interpolation error estimate in $L^\infty([a,b])$

$$\|u - I_1 u\|_{L^\infty([a,b])} \leq \frac{1}{4} h_{\mathcal{M}}^2 \|u''\|_{L^\infty([a,b])}. \quad (5.3.5)$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

This is obtained by simply taking the maximum over all *local* norms of the interpolation error.

Recall: supremum norm (maximum norm) from Def. 1.6.7

Now, we also want to study other norms of the interpolation error:

• $\|u - I_1 u\|_{L^2([a,b])}$:

Now all mesh cells contribute to this norm:

$$\|u - I_1 u\|_{L^2([a,b])}^2 = \sum_{j=1}^M \|u - I_1 u\|_{L^2([x_{j-1}, x_j])}^2 = \sum_{j=1}^M \int_{x_{j-1}}^{x_j} |(u - I_1 u)(x)|^2 dx, \quad I_1 u \text{ from (5.3.4)}. \quad (5.3.6)$$

➤ Idea:

localization

(Estimate error on individual mesh cells and sum local bounds)

By integrating by parts (1.3.36) twice, for $u \in C^2([x_{j-1}, x_j])$, $x \in [x_{j-1}, x_j]$,

$$\begin{aligned} \int_{x_{j-1}}^x \frac{(x_j - x)(\xi - x_{j-1})}{x_j - x_{j-1}} u''(\xi) d\xi + \int_x^{x_j} \frac{(x - x_{j-1})(x_j - \xi)}{x_j - x_{j-1}} u''(\xi) d\xi \\ = \underbrace{\frac{x_j - x}{x_j - x_{j-1}} u(x_{j-1}) + \frac{x - x_{j-1}}{x_j - x_{j-1}} u(x_j)}_{=I_1 u(x)} - u(x). \quad (5.3.7) \end{aligned}$$

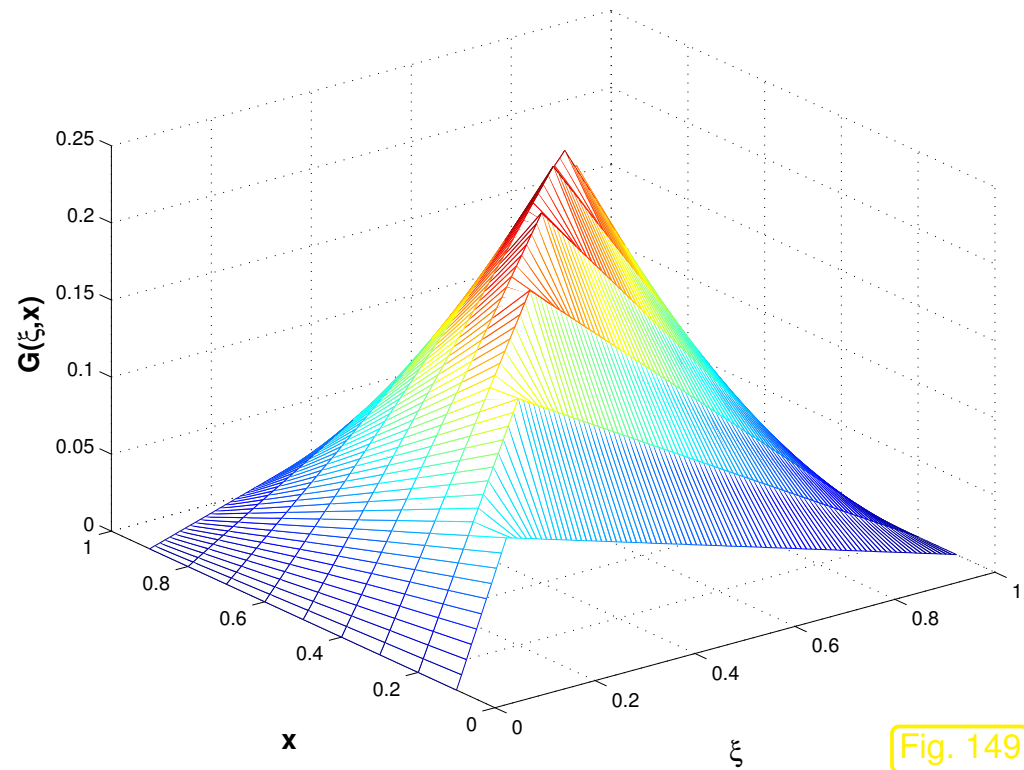
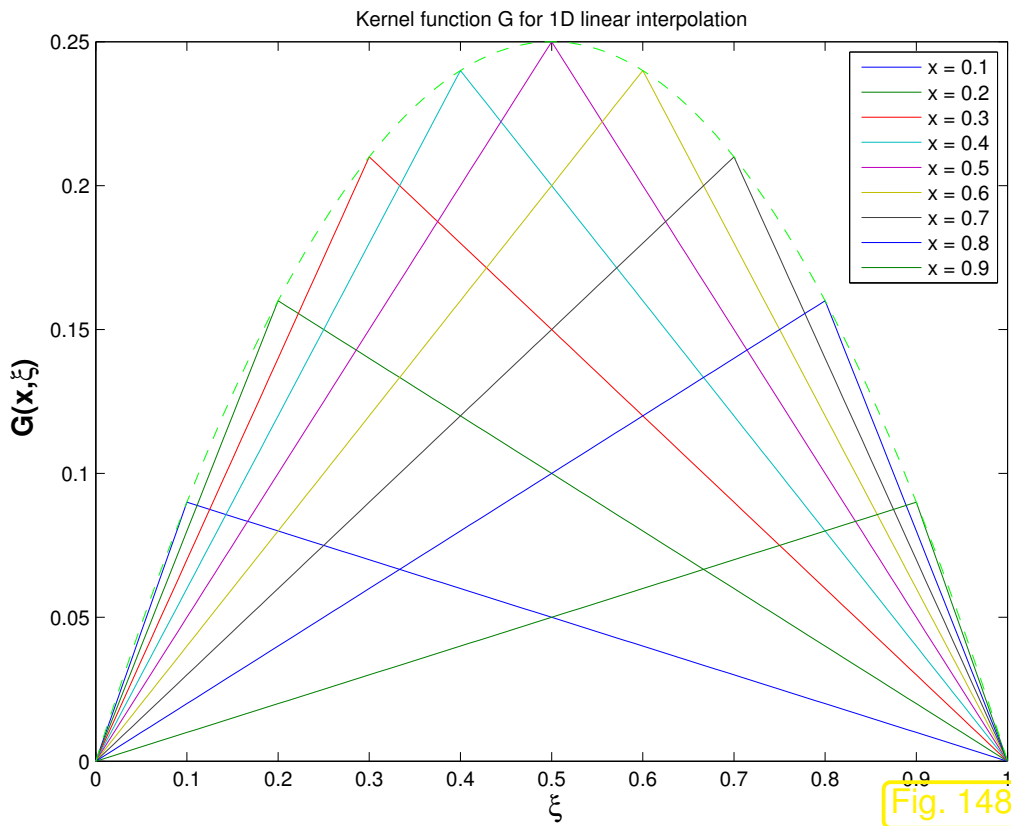
This is a *representation formula* for the local interpolation error $I_1 u - u$ of the form

$$(I_1 u - u)(x) = \int_{x_{j-1}}^{x_j} G(x, \xi) u''(\xi) d\xi.$$

with $G(x, \xi) = \begin{cases} \frac{(x_j - x)(\xi - x_{j-1})}{x_j - x_{j-1}} & \text{for } x_{j-1} \leq \xi < x, \\ \frac{(x - x_{j-1})(x_j - \xi)}{x_j - x_{j-1}} & \text{for } x \leq \xi \leq x_j. \end{cases}$,which satisfies

$$|G(x, \xi)| \leq |x_j - x_{j-1}| \Rightarrow \int_{x_{j-1}}^{x_j} G(x, \xi)^2 d\xi \leq |x_j - x_{j-1}|^3 .$$

Kernel functions G for 1D linear interpolation for $x_{j-1} = 0$, $x_j = 1$.



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

$$\begin{aligned}
 \blacktriangleright \int_{x_{j-1}}^{x_j} |u(x) - \mathbb{I}u(x)|^2 dx &= \int_{x_{j-1}}^{x_j} \left| \int_{x_{j-1}}^{x_j} G(x, \xi) u''(\xi) d\xi \right|^2 dx \\
 &\stackrel{(2.2.24)}{\leq} \int_{x_{j-1}}^{x_j} \left\{ \int_{x_{j-1}}^{x_j} G(x, \xi)^2 d\xi \cdot \int_{x_{j-1}}^{x_j} |u''(\xi)|^2 d\xi \right\} dx,
 \end{aligned} \tag{5.3.8}$$

(5.3.8)
 \Rightarrow

$$\|u - I_1 u\|_{L^2([x_{j-1}, x_j])}^2 = \int_{x_{j-1}}^{x_j} |u(x) - I_1 u(x)|^2 dx \leq |x_j - x_{j-1}|^4 \int_{x_{j-1}}^{x_j} |u''(\xi)|^2 d\xi .$$

(5.3.9)

Apply this estimate on $[x_{j-1}, x_j]$, sum over all cells of the mesh \mathcal{M} and take square root.

$$(5.3.9) \Rightarrow \|u - I_1 u\|_{L^2([a,b])} \leq h_{\mathcal{M}}^2 \|u''\|_{L^2([a,b])} .$$

(5.3.10)

• $|u - I_1 u|_{H^1([a,b])}$:

Differentiate representation formula (5.3.7): for $x_{j-1} < x < x_j$

$$\frac{d}{dx}(I_1 u - u)(x) = \int_{x_{j-1}}^{x_j} -\frac{\xi - x_{j-1}}{x_j - x_{j-1}} u''(\xi) d\xi + \int_{x_{j-1}}^{x_j} \frac{x_j - \xi}{x_j - x_{j-1}} u''(\xi) d\xi .$$

$$\begin{aligned} \blacktriangleright \int_{x_{j-1}}^{x_j} \left| \frac{d}{dx} (\mathbb{I}u - u)(x) \right|^2 dx &= \int_{x_{j-1}}^{x_j} \left| \int_{x_{j-1}}^{x_j} \frac{\partial G}{\partial x}(x, \xi) u''(\xi) d\xi \right|^2 dx \\ &\leq \int_{x_{j-1}}^{x_j} \left\{ \int_{x_{j-1}}^{x_j} \underbrace{\left| \frac{\partial G}{\partial x}(x, \xi) \right|^2}_{\leq 1} d\xi \cdot \int_{x_{j-1}}^{x_j} |u''(\xi)|^2 d\xi \right\} dx . \end{aligned}$$

$$\blacktriangleright |u - \mathbb{I}_1 u|_{H^1([x_{j-1}, x_j])}^2 \leq (x_j - x_{j-1})^2 \int_{x_{j-1}}^{x_j} |u''(\xi)|^2 d\xi . \quad (5.3.11)$$

As above, apply this estimate on $[x_{j-1}, x_j]$, sum over all cells of the mesh \mathcal{M} and take square root.

$$(5.3.11) \Rightarrow |u - \mathbb{I}_1 u|_{H^1([a,b])} \leq h_{\mathcal{M}} \|u''\|_{L^2([a,b])} . \quad (5.3.12)$$

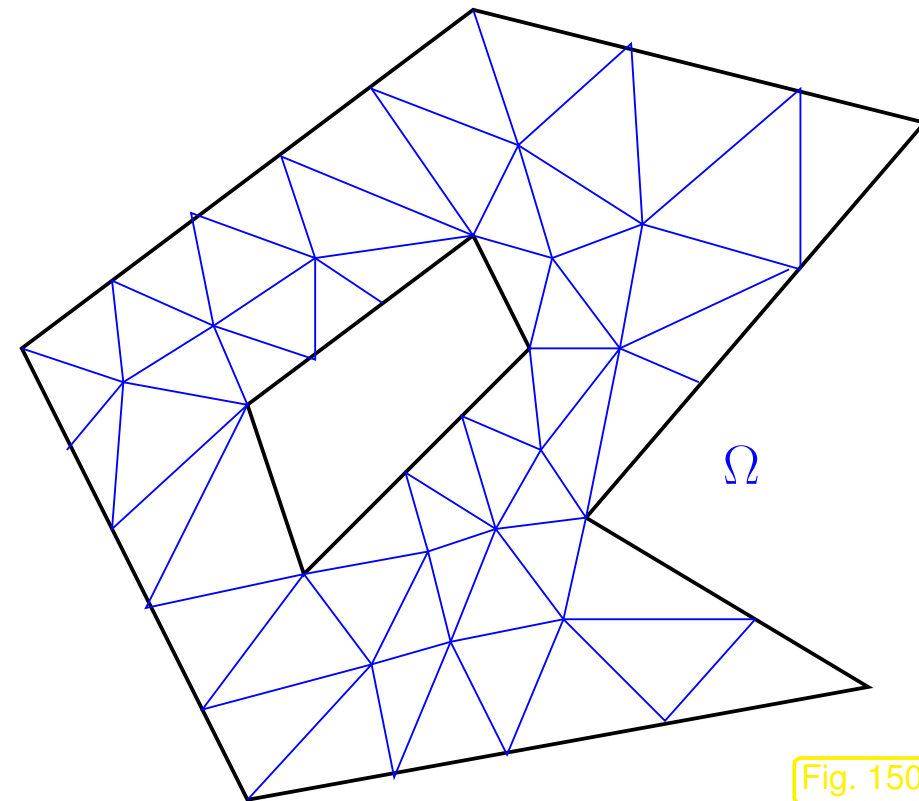
What we learn from this example:

1. We have to rely on **smoothness** of the interpolant u to obtain bounds for norms of the interpolation error.
2. The bounds involve norms of derivatives of the interpolant.
3. For smooth u we find **algebraic convergence** (\rightarrow Def. 1.6.32) of norms of the interpolation error *in terms of mesh width* $h_{\mathcal{M}} \rightarrow 0$.

5.3.2 Error estimates for linear interpolation in 2D

Given:

- polygonal domain $\Omega \subset \mathbb{R}^2$
- triangular mesh \mathcal{M} of Ω (\rightarrow Def. 3.3.1)



Sect 5.3.1 introduced piecewise linear interpolation on a mesh/grid in 1D. The next definition gives the natural 2D counterpart on a triangular mesh, which is closely related to the piecewise linear reconstruction (interpolation) operator from (4.2.6), see Figs. 132, 133.


Definition 5.3.13 (Linear interpolation in 2D).

The linear interpolation operator $I_1 : C^0(\bar{\Omega}) \mapsto \mathcal{S}_1^0(\mathcal{M})$ is defined by

$$I_1 u \in \mathcal{S}_1^0(\mathcal{M}) \quad , \quad I_1 u(\mathbf{p}) = u(\mathbf{p}) \quad \forall \mathbf{p} \in \mathcal{V}(\mathcal{M}) .$$

Recalling the definition of the nodal basis $\mathfrak{B} = \{b_N^{\mathbf{p}} : \mathbf{p} \in \mathcal{V}(\mathcal{M})\}$ of $\mathcal{S}_1^0(\mathcal{M})$ from (3.2.4), where $b_N^{\mathbf{p}}$ is the “tent function” associated with node \mathbf{p} , an equivalent definition is, cf. (3.5.44),

$$I_1 u = \sum_{\mathbf{p} \in \mathcal{V}(\mathcal{M})} u(\mathbf{p}) b_N^{\mathbf{p}} \quad , \quad u \in C^0(\bar{\Omega}) . \quad (5.3.14)$$

Task:  For “sufficiently smooth” $u : \Omega \mapsto \mathbb{R}$ ($\Leftrightarrow u \in C^\infty(\bar{\Omega})$ to begin with) *estimate*

interpolation error norm $\|u - I_1 u\|_{H^1(\Omega)} .$

Idea:

Localization

I_1 local \triangleright first, estimate $\|u - I_1 u\|_{H^1(K)}^2$, $K \in \mathcal{M}$,
then, global estimate via summation as in Sect. 5.3.1.

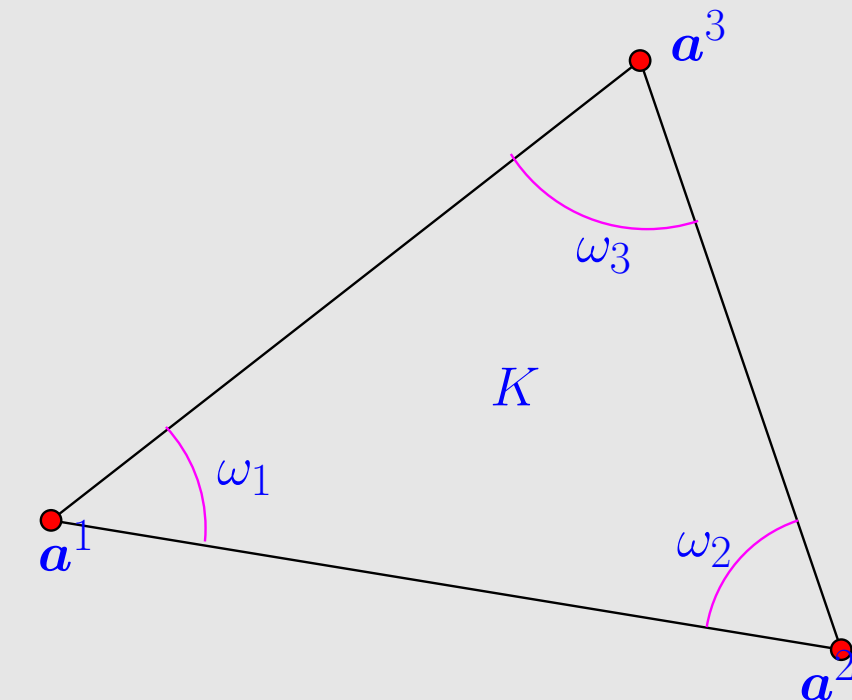
\triangleright Focus on single triangle $K \in \mathcal{M}$

Crucial for localization to work: linear interpolation operator $I_1 : C^0(\bar{\Omega}) \mapsto \mathcal{S}_1^0(\mathcal{M})$ can be defined **purely locally** by

$$I_1 u|_K = u(\mathbf{a}^1)\lambda_1 + u(\mathbf{a}^2)\lambda_2 + u(\mathbf{a}^3)\lambda_3, \quad (5.3.15)$$

for each triangle $K \in \mathcal{M}$ with vertices $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$ ($\lambda_k \hat{=}$ barycentric coordinate functions = local shape functions for $\mathcal{S}_1^0(\mathcal{M})$, see Fig. 72).

Next step, cf. (5.3.7): **representation formula** for local interpolation error.



$u \in C^2(\bar{K})$: by mean value formula $\forall \mathbf{x} \in K$,

$$u(\mathbf{a}^j) = u(\mathbf{x}) + \mathbf{grad} u(\mathbf{x}) \cdot (\mathbf{a}^j - \mathbf{x}) + \int_0^1 (\mathbf{a}^j - \mathbf{x})^\top D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x})(1 - \xi) d\xi, \quad (5.3.16)$$

$$D^2 u(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 u}{\partial x_1^2}(\mathbf{x}) & \frac{\partial^2 u}{\partial x_1 \partial x_2}(\mathbf{x}) \\ \frac{\partial^2 u}{\partial x_1 \partial x_2}(\mathbf{x}) & \frac{\partial^2 u}{\partial x_2^2}(\mathbf{x}) \end{pmatrix} \hat{=} \text{Hessian.}$$

The formula (5.3.16) is easily verified by applying integration by parts

$$f(b) - f(a) = [\xi f'(\xi)]_a^b - \int_a^b \xi f''(\xi) d\xi = f'(a)(b - a) + \int_a^b (b - \xi) f''(\xi) d\xi.$$

to the function $\phi(t) = u(t\mathbf{a}^j + (1-t)\mathbf{x})$ with $a = 0, b = 1$.

Next, use (5.3.16) to replace $u(\mathbf{a}^j)$ in the formula (5.3.15) for local linear interpolation. Also use the identities for the barycentric coordinate functions

$$\sum_{j=1}^3 \lambda_j(\mathbf{x}) = 1 \quad , \quad \mathbf{x} = \sum_{j=1}^3 \mathbf{a}^j \lambda_j(\mathbf{x}) . \quad (5.3.17)$$

$$I_1 u(\mathbf{x}) = \sum_{j=1}^3 u(\mathbf{a}^j) \lambda_j(\mathbf{x}) = u(\mathbf{x}) \cdot \underbrace{\sum_{j=1}^3 \lambda_j(\mathbf{x})}_{=1} + \mathbf{grad} u(\mathbf{x}) \cdot \underbrace{\sum_{j=1}^3 (\mathbf{a}^j - \mathbf{x}) \lambda_j(\mathbf{x})}_{=0} + R(\mathbf{x}) ,$$

with
$$R(\mathbf{x}) := \sum_{j=1}^3 \left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^\top D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x}) (1 - \xi) d\xi \right) \lambda_j(\mathbf{x}) . \quad (5.3.18)$$

Again, as in the case of (5.3.7) for 1D linear interpolation we have arrived at an **integral representation formula** for the local interpolation error:

$$(u - I_1 u)(\mathbf{x}) = \sum_{j=1}^3 \left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^\top D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x}) (1 - \xi) d\xi \right) \lambda_j(\mathbf{x}) . \quad (5.3.19)$$

Together with the triangle inequality, the trivial bound $|\lambda_j| \leq 1$ yields

$$\|u - I_1 u\|_{L^2(K)} \leq \sum_{j=1}^3 \left(\int_K \left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^T D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x})(1 - \xi) d\xi \right)^2 d\mathbf{x} \right)^{\frac{1}{2}}.$$

To estimate an expression of the form

$$\int_K \left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^T D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x})(1 - \xi) d\xi \right)^2 d\mathbf{x}, \quad (5.3.20)$$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

we may assume, without loss of generality, that $\mathbf{a}^j = 0$.

➤ Task: estimate terms (where 0 is a vertex of K !)

$$\int_K \left(\int_0^1 \mathbf{x}^\top D^2 u((1 - \xi)\mathbf{x}) \mathbf{x}(1 - \xi) d\xi \right)^2 d\mathbf{x} = \int_K \left(\int_0^1 \mathbf{x}^\top D^2 u(\xi\mathbf{x}) \mathbf{x} \xi d\xi \right)^2 d\mathbf{x}.$$

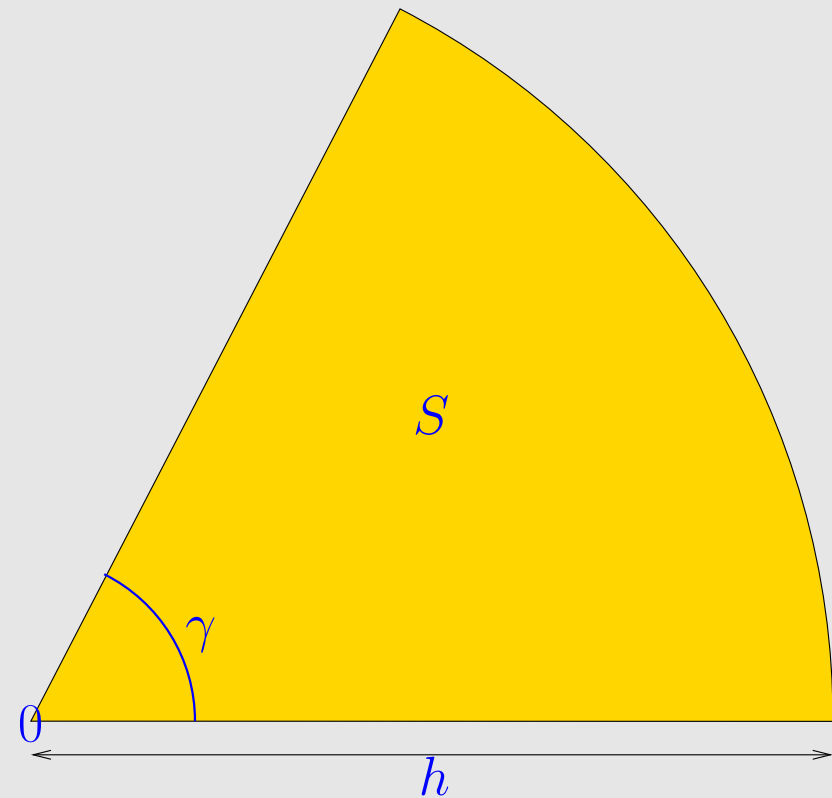
Denote $\gamma \hat{=}$ angle of K at vertex 0 ,
 $h \hat{=}$ length of longest edge of K .

K is contained in the sector

$$S := \left\{ \mathbf{x} = \begin{pmatrix} r \cos \varphi \\ r \sin \varphi \end{pmatrix} : 0 \leq r < h, 0 \leq \varphi \leq \gamma \right\}$$

Lemma 5.3.21. For any $\psi \in L^2(S)$

$$\int_S \left(\int_0^1 |\mathbf{y}|^2 \psi(\tau \mathbf{y}) \tau \, d\tau \right)^2 \, d\mathbf{y} \leq \frac{h^4}{8} \|\psi\|_{L^2(S)}^2 .$$



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Using polar coordinates (r, φ) , $\widehat{\mathbf{s}}_\varphi = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix}$, see [32, Bsp. 8.5.3], and Cauchy-Schwarz inequality (2.2.24):

$$\begin{aligned} \int_S \left(\int_0^1 |\mathbf{y}|^2 \psi(\tau \mathbf{y}) \tau \, d\tau \right)^2 \, d\mathbf{y} &= \int_0^\gamma \int_0^h \left(\int_0^1 r^2 \psi(\tau r \widehat{\mathbf{s}}_\varphi) \tau \, d\tau \right)^2 r \, dr \, d\varphi \\ &= \int_0^\gamma \int_0^h \left(\int_0^r \psi(\sigma \widehat{\mathbf{s}}_\varphi) \sigma \, d\sigma \right)^2 r \, dr \, d\varphi \leq \int_0^\gamma \int_0^h \int_0^r \psi^2(\sigma \widehat{\mathbf{s}}_\varphi) \sigma \, d\sigma \cdot \int_0^r \sigma \, d\sigma \, r \, dr \, d\varphi \end{aligned}$$

$$\leq \frac{1}{2} \int_0^\gamma \int_0^h \psi^2(\sigma \hat{\mathbf{s}}_\varphi) \sigma \, d\sigma d\varphi \cdot \int_0^h r^3 \, dr .$$

Use $|\mathbf{z}^\top \mathbf{A} \mathbf{y}| \leq \|\mathbf{A}\|_F |\mathbf{z}| |\mathbf{y}|$, $\mathbf{A} \in \mathbb{R}^{n,n}$, $\mathbf{z}, \mathbf{y} \in \mathbb{R}^n$, and then apply Lemma 5.3.21 with $\mathbf{y} := \mathbf{x} - \mathbf{a}^j$, $\tau = 1 - \xi$

$$\blacktriangleright \quad \|u - I_1 u\|_{L^2(K)}^2 \leq \frac{3}{8} h_K^4 \left\| \left\| D^2 u \right\|_F \right\|_{L^2(K)}^2, \quad (5.3.22)$$

with **Frobenius matrix norm** $\left\| D^2 u(\mathbf{x}) \right\|_F^2 := \sum_{i,j=1}^2 \left| \frac{\partial^2 u}{\partial x_i \partial x_j}(\mathbf{x}) \right|^2$ (\rightarrow [21, Def. 6.5.35]),

size of triangle $h_K := \text{diam } K := \max\{|\mathbf{p} - \mathbf{q}| : \mathbf{p}, \mathbf{q} \in K\}$

Estimate for gradient: from (5.3.16) we infer the local integral representation formula, which can also be obtained by taking the gradient of (5.3.19).

$$\text{grad } I_1 u(\mathbf{x}) = u(\mathbf{x}) \underbrace{\sum_{j=1}^3 \text{grad } \lambda_j(\mathbf{x})}_{=0} + \underbrace{\sum_{j=1}^3 (\mathbf{a}^j - \mathbf{x})^\top \text{grad } \lambda_j(\mathbf{x}) \cdot \text{grad } u(\mathbf{x})}_{=I} + G(\mathbf{x}),$$

$$\text{with } G(\mathbf{x}) := \sum_{j=1}^3 \underbrace{\left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^\top D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x}) (1 - \xi) \, d\xi \right)}_{(5.3.22)} \text{grad } \lambda_j(\mathbf{x}).$$

Note that $\mathbf{grad} \sum_{j=1}^3 \lambda_j(\mathbf{x}) = \mathbf{grad} 1 = 0$ and

$$\sum_{j=1}^3 \mathbf{grad} \lambda_j(\mathbf{x})(\mathbf{a}^j - \mathbf{x})^\top = \sum_{j=1}^3 \mathbf{grad} \lambda_j(\mathbf{x})(\mathbf{a}^j)^\top = \mathbf{grad} \left(\sum_{j=1}^3 \lambda_j(\mathbf{x}) \mathbf{a}^j \right) = \mathbf{grad} \mathbf{x} = \mathbf{I} .$$

$$(3.5.22) \quad \blacktriangleright \quad \boxed{|\mathbf{grad} \lambda_j(\mathbf{x})| \leq \frac{h_K}{2|K|}, \quad \mathbf{x} \in K} . \quad (5.3.23)$$

$$\blacktriangleright \quad \|\mathbf{grad}(u - l_1 u)\|_{L^2(K)}^2 \leq \frac{h_K^2}{4|K|^2} \|R\|_{L^2(K)}^2 \stackrel{(5.3.22)}{\leq} \frac{3}{8} \frac{h_K^6}{4|K|^2} \left\| \|D^2 u\|_F \right\|_{L^2(K)}^2 . \quad (5.3.24)$$

Summary of *local* interpolation error estimates for linear interpolation according to Def. 5.3.13:

Lemma 5.3.25 (Local interpolation error estimates for 2D linear interpolation).

For any triangle K and $u \in C^2(\overline{K})$ the following holds

$$\|u - I_1 u\|_{L^2(K)}^2 \leq \frac{3}{8} h_K^4 \left\| \left\| D^2 u \right\|_F \right\|_{L^2(K)}^2, \quad (5.3.22)$$

$$\|\mathbf{grad}(u - I_1 u)\|_{L^2(K)}^2 \leq \frac{3}{24} \frac{h_K^6}{|K|^2} \left\| \left\| D^2 u \right\|_F \right\|_{L^2(K)}^2. \quad (5.3.24)$$

New aspect compared to Sect. 5.3.1: *shape* of K enters error bounds of Lemma 5.3.25.

We aim to extract this shape dependence from the bounds.

Definition 5.3.26 (Shape regularity measures).

For a simplex $K \in \mathbb{R}^d$ we define its *shape regularity measure* as the ratio

$$\rho_K := h_K^d : |K| ,$$

and the shape regularity measure of a simplicial mesh $\mathcal{M} = \{K\}$

$$\rho_{\mathcal{M}} := \max_{K \in \mathcal{M}} \rho_K .$$

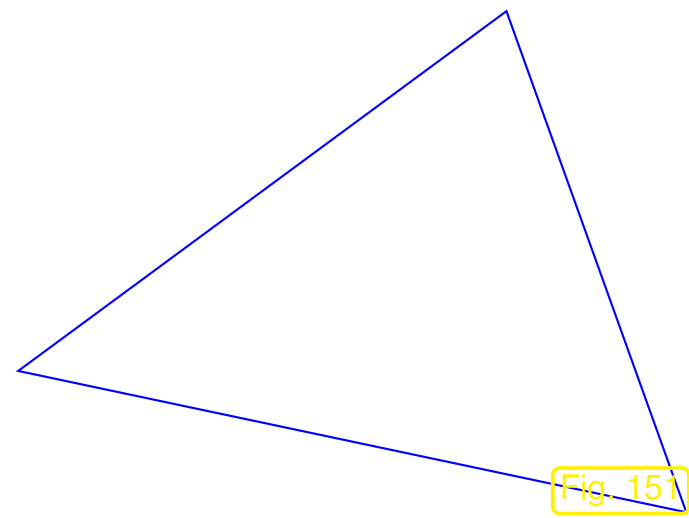
Important:

shape regularity measure ρ_K is an invariant of a similarity class of triangles.

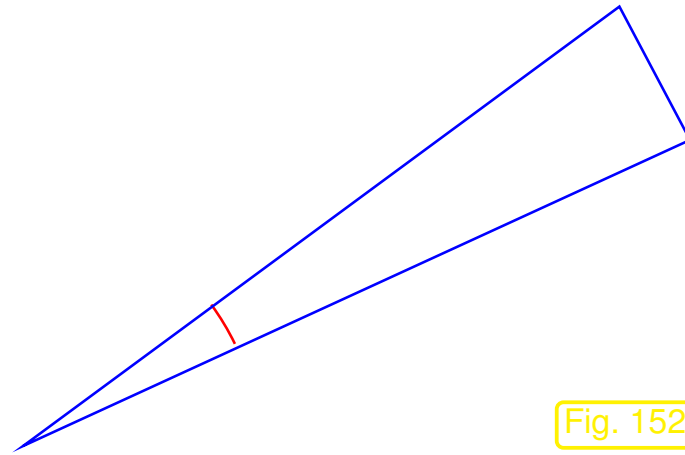
(= if a triangle is transformed by scaling, rotation, and translation, the shape regularity measure does not change)

➤ Sloppily speaking, ρ_K depends only on the shape, not the size of K

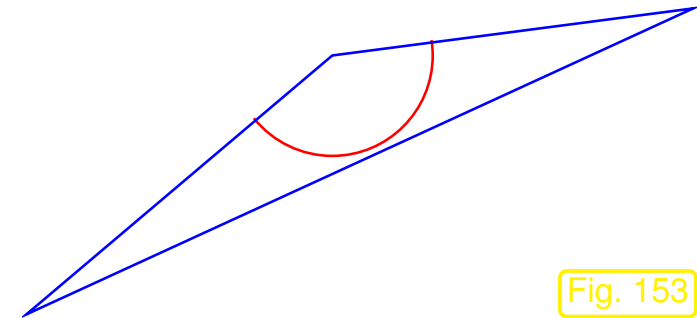
For triangle K : ρ_K large $\Leftrightarrow K$ “distorted” $\Leftrightarrow K$ has small angles



ρ_k small



ρ_k large



ρ_k large

The shape regularity measure $\rho_{\mathcal{M}}$ is often used to gauge the *quality* of meshes produced by mesh generators.

Final step: we add up the local estimates from Lemma 5.3.25 over all triangles of the mesh and take the square root.

Theorem 5.3.27 (Error estimate for piecewise linear interpolation).

For any $u \in C^2(\bar{\Omega})$

$$\begin{aligned} \|u - I_1 u\|_{L^2(\Omega)} &\leq \sqrt{\frac{3}{8}} h_{\mathcal{M}}^2 \left\| \left\| D^2 u \right\|_F \right\|_{L^2(\Omega)}, \\ \|\mathbf{grad}(u - I_1 u)\|_{L^2(\Omega)} &\leq \sqrt{\frac{3}{24}} \rho_{\mathcal{M}} h_{\mathcal{M}} \left\| \left\| D^2 u \right\|_F \right\|_{L^2(\Omega)}. \end{aligned}$$

Remark 5.3.28 (Energy norm and $H^1(\Omega)$ -norm).

Objection! Well, Cea's lemma Thm. 5.1.10 refers to the energy norm, but Thm. 5.3.27 provides estimates in $H^1(\Omega)$ -norm only!

☞ For uniformly positive definite (\rightarrow Def. 2.1.12) and bounded coefficient tensor $\alpha : \Omega \mapsto \mathbb{R}^{d,d}$, cf. (2.1.9),

$$\exists 0 < \alpha^- < \alpha^+ : \alpha^- \|z\|^2 \leq z^T \alpha(x) z \leq \alpha^+ \|z\|^2 \quad \forall z \in \mathbb{R}^d, x \in \Omega,$$

and the energy norm (\rightarrow Def. 2.1.35) induced by

$$\mathbf{a}(u, v) := \int_{\Omega} (\alpha(x) \mathbf{grad} u) \cdot \mathbf{grad} v \, dx, \quad u, v \in H_0^1(\Omega), \quad (5.1.6)$$

we immediately find the **equivalence** (= two-sided uniform estimate)

$$\sqrt{\alpha^-} |v|_{H^1(\Omega)} \leq \|v\|_a \leq \sqrt{\alpha^+} |v|_{H^1(\Omega)} . \quad (5.3.29)$$

Thus, interpolation error estimates in $|\cdot|_{H^1(\Omega)}$ immediately translate into estimates in terms of the energy norm.

5.3.3 The Sobolev scales

Bounds in Thm. 5.3.27 involve $\| \| \| D^2 u \| \|_F \|_{L^2(\Omega)}$ measures smoothness of u



➔ Norms of this type are a tool to measure the **smoothness** of functions (that usually are solutions of elliptic BVP):

Definition 5.3.30 (Higher order Sobolev spaces/norms).

The *m -th order Sobolev norm*, $m \in \mathbb{N}_0$, for $u : \Omega \subset \mathbb{R}^d \mapsto \mathbb{R}$ (sufficiently smooth) is defined by

$$\|u\|_{H^m(\Omega)}^2 := \sum_{k=0}^m \sum_{\alpha \in \mathbb{N}^d, |\alpha|=k} \int_{\Omega} |D^{\alpha}u|^2 d\mathbf{x}, \quad \text{where} \quad D^{\alpha}u := \frac{\partial^{|\alpha|}u}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}.$$

Sobolev space $H^m(\Omega) := \{v : \Omega \mapsto \mathbb{R} : \|v\|_{H^m(\Omega)} < \infty\}.$

Recall: multiindex notation (3.3.4), (3.3.5)

Gripe (\rightarrow Sect. 2.2):

Don't bother me with these Sobolev spaces !

Response: Well, these concepts are pervasive in the numerical analysis literature and you have to be familiar with them.

Reassuring:

Again, it is only the norms $\|u\|_{H^m(\Omega)}$ that matter for us !

Now, we have come across an additional purpose of Sobolev spaces and their norms:

provide framework for
variational formulation of
elliptic BVP
(\rightarrow Sect. 2.2)

Sobolev
spaces

provide norms $\|\cdot\|_{H^m(\Omega)}$ that
measure smoothness of
functions

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Sobolev scale:

$$\dots \subset H^3(\Omega) \subset H^2(\Omega) \subset H^1(\Omega) \subset L^2(\Omega)$$

Observation: bounds in Thm. 5.3.27 = “principal parts” of Sobolev norms, that is, the parts containing the highest partial derivatives.

Definition 5.3.31 (Higher order Sobolev semi-norms).

The *m -th order Sobolev semi-norm*, $m \in \mathbb{N}$, for sufficiently smooth $u : \Omega \mapsto \mathbb{R}$ is defined by

$$|u|_{H^m(\Omega)}^2 := \sum_{\alpha \in \mathbb{N}^d, |\alpha|=m} \int_{\Omega} |D^{\alpha}u|^2 \, dx .$$

Elementary observation: $|p|_{H^m(\Omega)} = 0 \iff p \in \mathcal{P}_{m-1}(\mathbb{R}^d)$

► By density arguments we can rewrite the interpolation error estimates of Thm. 5.3.27 in terms of Sobolev semi-norms:

Corollary 5.3.32 (Error estimate for piecewise linear interpolation in 2D).

Under the assumptions/with notations of Thm. 5.3.27

$$\begin{aligned} \|u - \mathbf{l}_1 u\|_{L^2(\Omega)} &\leq \sqrt{\frac{3}{8}} h_{\mathcal{M}}^2 |u|_{H^2(\Omega)} , \\ |u - \mathbf{l}_1 u|_{H^1(\Omega)} &\leq \sqrt{\frac{3}{24}} \rho_{\mathcal{M}} h_{\mathcal{M}} |u|_{H^2(\Omega)} , \end{aligned} \quad \forall u \in H^2(\Omega) .$$

Remark 5.3.33 (Continuity of interpolation operators).

Apply \triangle -inequality to estimates of Cor. 5.3.32:

$$\|I_1 u\|_{L^2(\Omega)} \leq \|u\|_{L^2(\Omega)} + \sqrt{\frac{3}{8}} h_{\mathcal{M}}^2 |u|_{H^2(\Omega)} \leq 2\|u\|_{H^2(\Omega)}, \quad (5.3.34)$$

if lengths are scaled such that $h_{\mathcal{M}} \leq 1$. Estimate (5.3.34) means that $I_1 : H^2(\Omega) \mapsto L^2(\Omega)$ is a **continuous linear** mapping.

The same conclusion could have been drawn from the following fundamental result:

Theorem 5.3.35 (Sobolev embedding theorem).

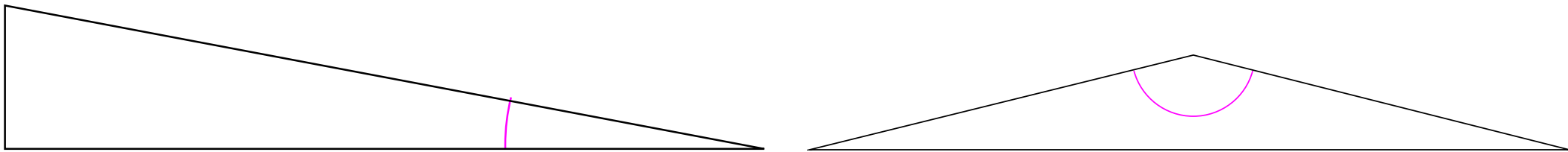
$$m > \frac{d}{2} \Rightarrow H^m(\Omega) \subset C^0(\bar{\Omega}) \quad \wedge \quad \exists C = C(\Omega) > 0: \quad \|u\|_{\infty} \leq C \|u\|_{H^m(\Omega)} \quad \forall u \in H^m(\Omega).$$

On the other hand $I_1 : H^1(\Omega) \mapsto L^2(\Omega)$ is **not** continuous, as we learn from Rem. 2.3.27.



5.3.4 Anisotropic interpolation error estimates

Triangular cells with “bad shape regularity” (ρ_K “large”): very small/large angles:



The estimates of Lemma 5.3.25 might suggest that we face huge local interpolation errors, once ρ_K becomes large.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Issue: are the estimates of Lemma 5.3.25 *sharp* ?

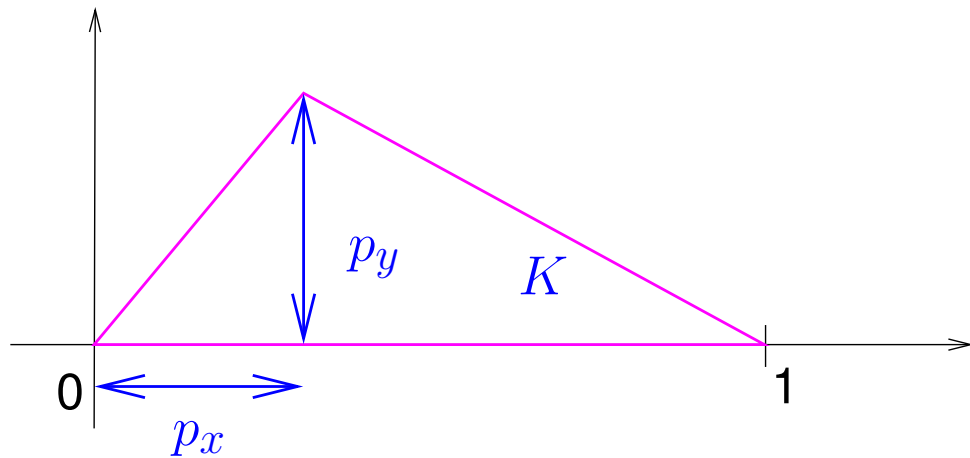
We will try to find this out experimentally by computing the best possible constants in the estimates

$$\|u - I_1 u\|_{L^2(K)} \leq C_{K,2} h_K^2 \|u\|_{H^2(K)}, \quad \|u - I_1 u\|_{H^1(K)} \leq C_K h_K \|u\|_{H^2(K)}.$$

Note: Merely translating, rotating, or scaling K does not affect the constants $C_{K,2}$ and C_K . Therefore, we can restrict ourselves to “canonical triangles”. Every general triangle can be mapped to one of these by translating, rotating, and scaling.

$$C_{K,2} := \sup_{u \in H^2(K) \setminus \{0\}} \frac{\|u - I_1 u\|_{L^2(K)}}{\|u\|_{H^2(K)}}, \quad C_K := \sup_{u \in H^2(K) \setminus \{0\}} \frac{\|u - I_1 u\|_{H^1(K)}}{\|u\|_{H^2(K)}},$$

on triangle $K := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} p_x \\ p_y \end{pmatrix} \right\}$.



Sampling the space of “canonical” triangles
(modulo similarity)

$$0 \leq p_x, p_y \leq 1.$$

+ Numerical computation of $C_K, C_{K,2}$

implementation by A. Inci (spectral polynomial
Galerkin method)

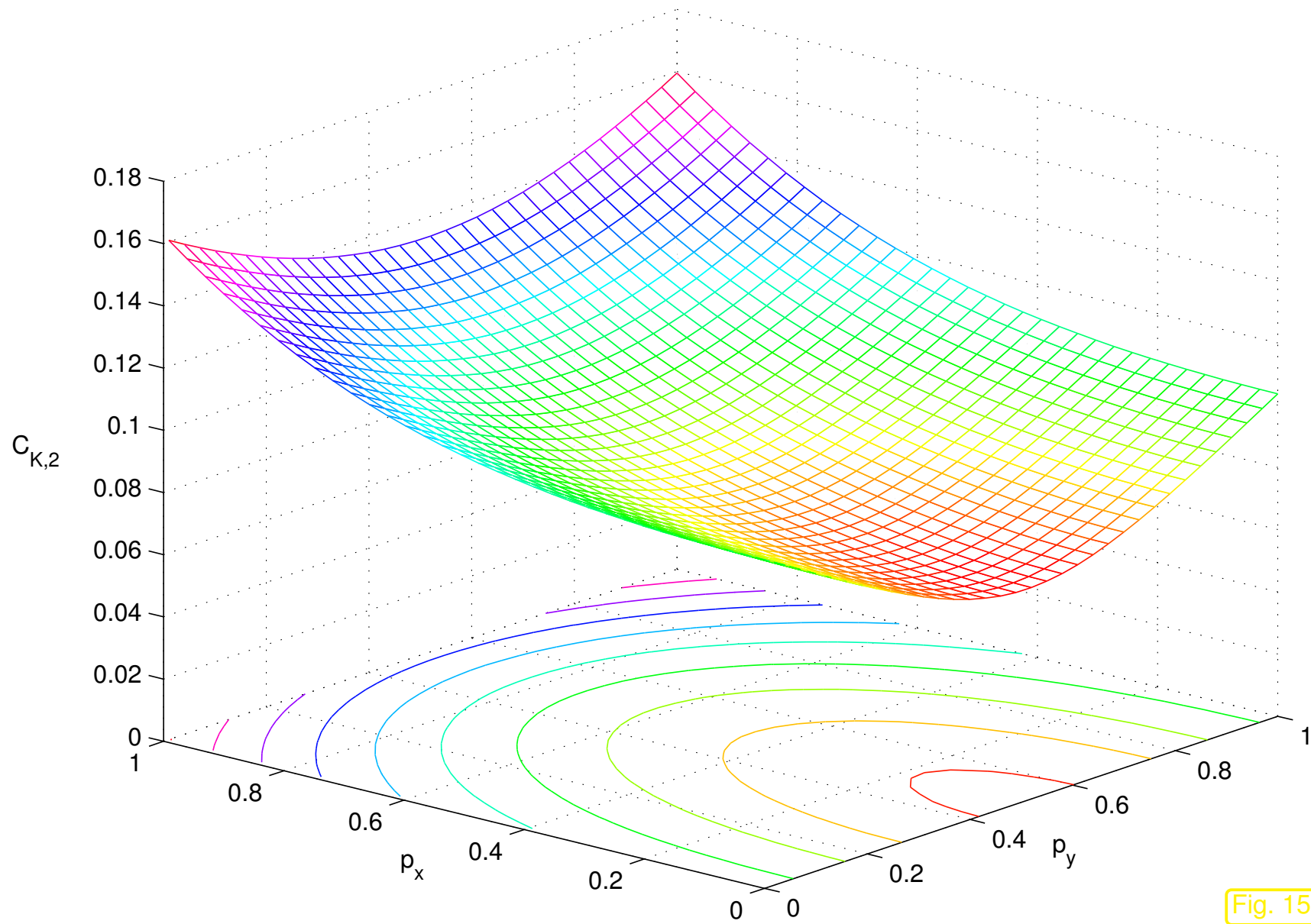
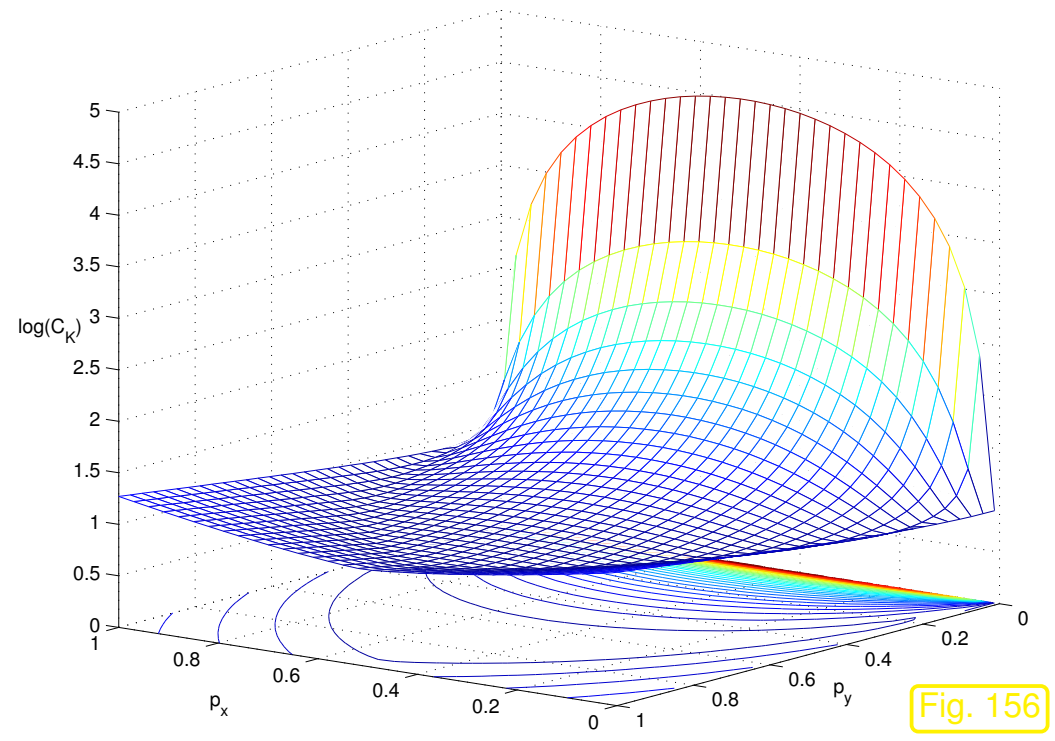
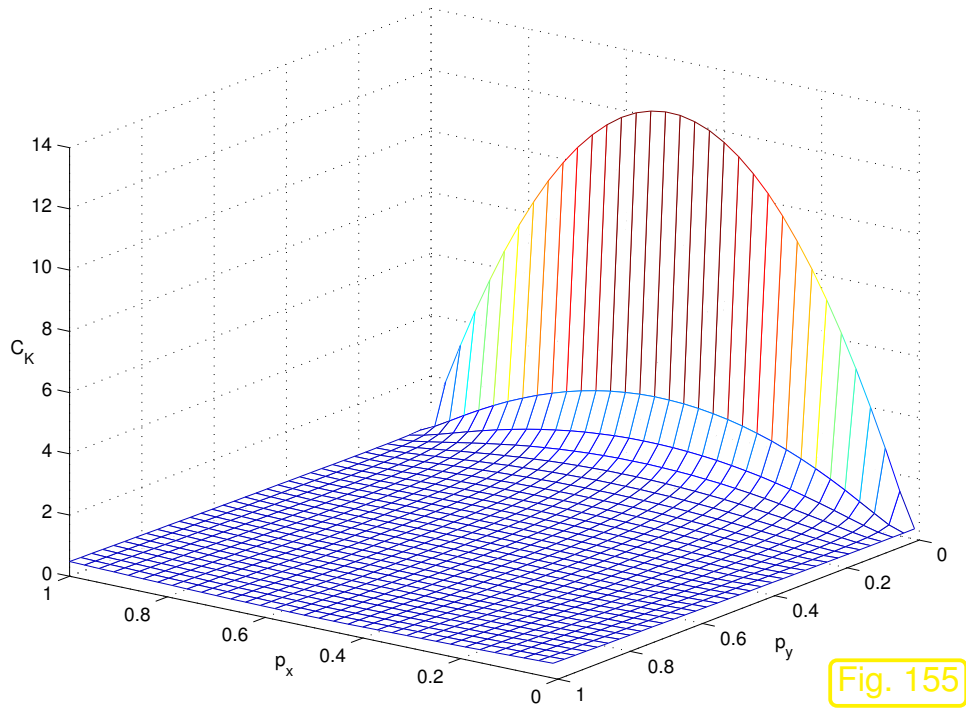


Fig. 154



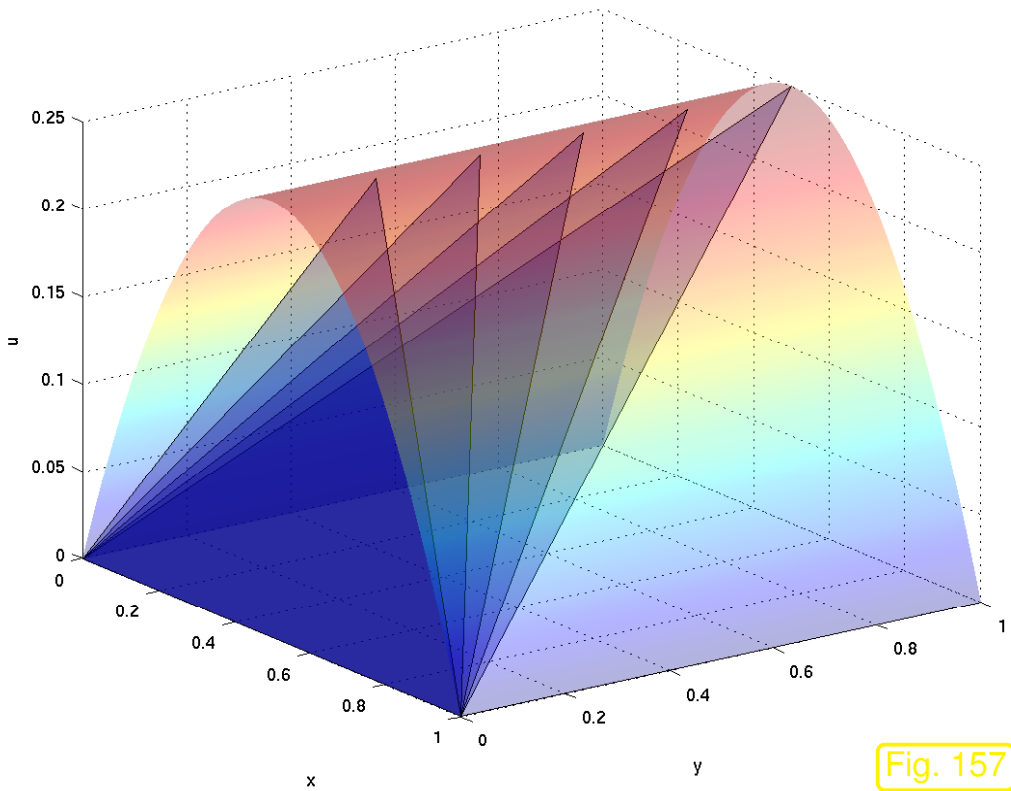
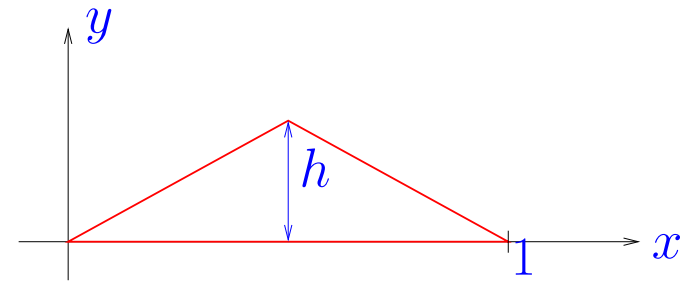


Fig. 157



triangle $K := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1/2 \\ h \end{pmatrix} \right\}$, $h > 0$,
 $u(x, y) = x(1 - x)$, $0 < x < 1$.

◁ linear interpolant of u on K as $h \rightarrow 0$

R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

The interpolant becomes steeper and steeper as $h \rightarrow 0$:

▶ $\|u\|_{H^2(K)}^2 = \frac{3031}{1440}h$, $\|u - I_1u\|_{H^1(K)}^2 = \frac{29}{2880}h + \frac{1}{12}h + \frac{1}{32}h^{-1}$, $\|u - I_1u\|_{L^2(K)}^2 = \frac{29}{2889}h$

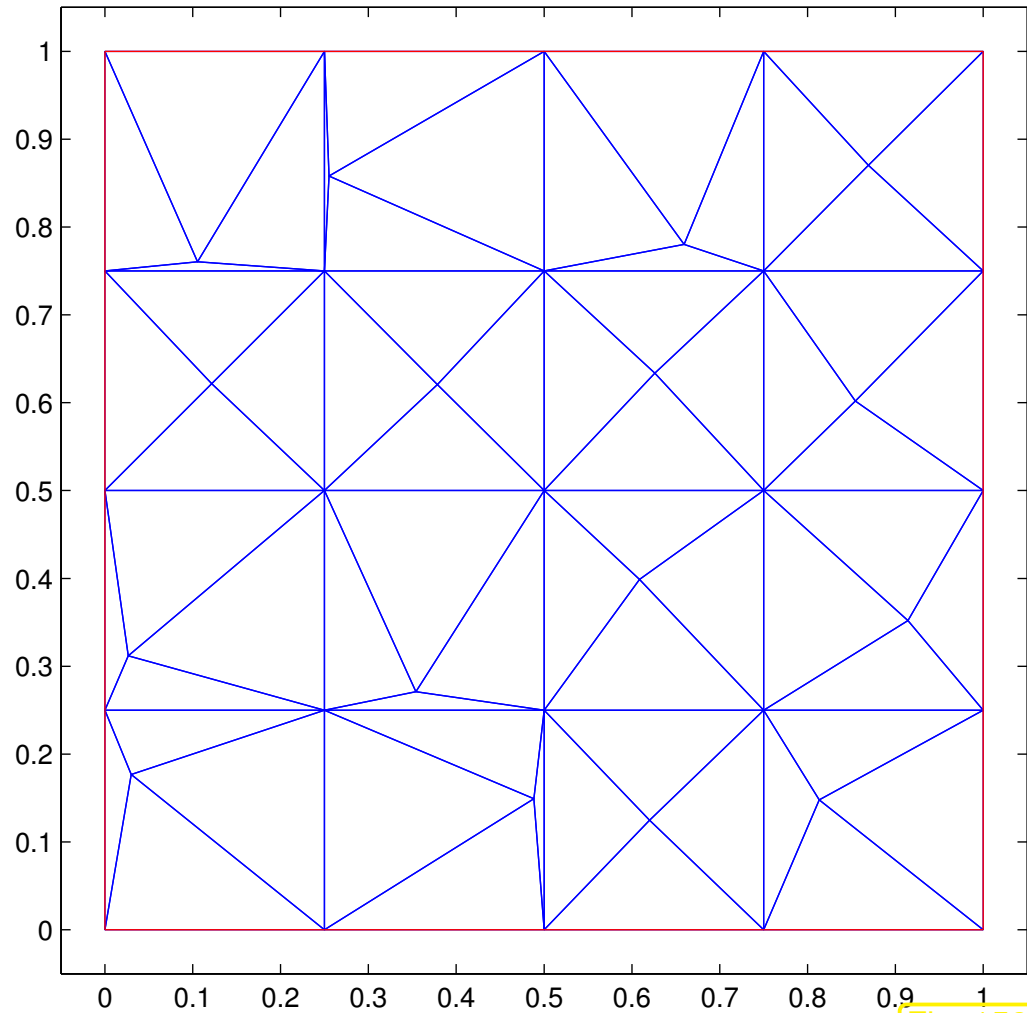
$$\frac{\|u - \mathbb{I}_1 u\|_{H^1(K)}^2}{\|u\|_{H^2(K)}^2} \geq \frac{269}{6062} + \frac{45}{3031} h^{-2}, \quad \frac{\|u - \mathbb{I}_1 u\|_{L^2(K)}^2}{\|u\|_{H^2(K)}^2} = \frac{29}{6062}.$$

Example 5.3.36 (Good accuracy on “bad” meshes).

$\Omega =]0, 1[^2$, $u(x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2)$, BVP $-\Delta u = f$, $u|_{\partial\Omega} = 0$, finite element Galerkin discretization on triangular meshes, $V_N = \mathcal{S}_{1,0}^0(\mathcal{M})$.

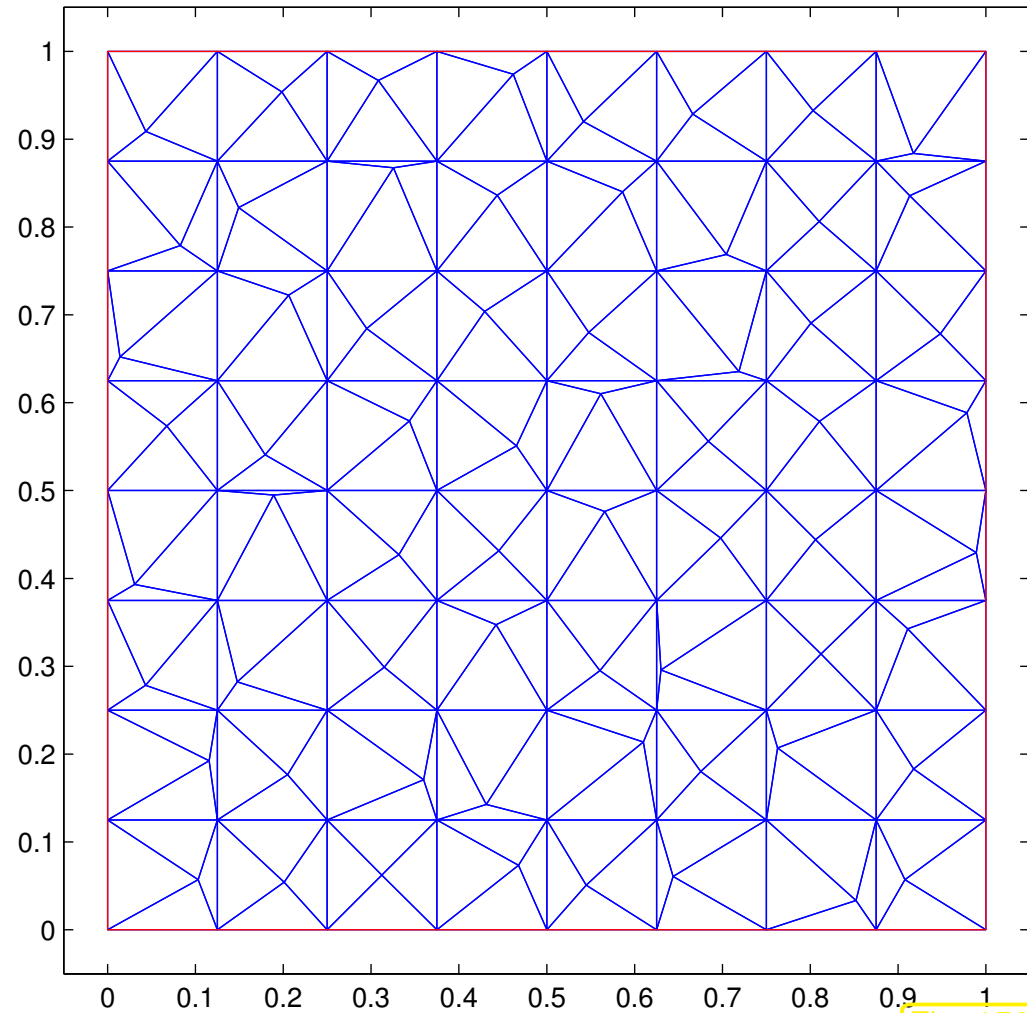
👉 meshes created by random distortion of tensor product grids

2D triangular mesh



Vertices : 41, # Elements : 64, # Edges : 56 **Fig. 158**

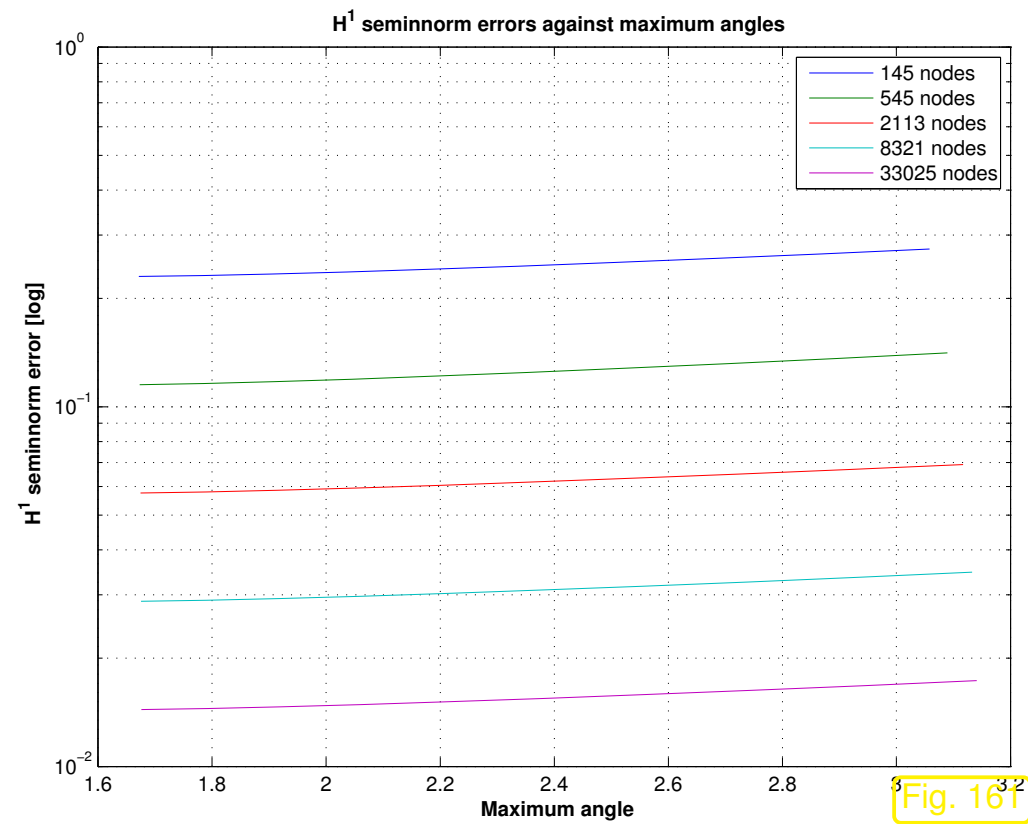
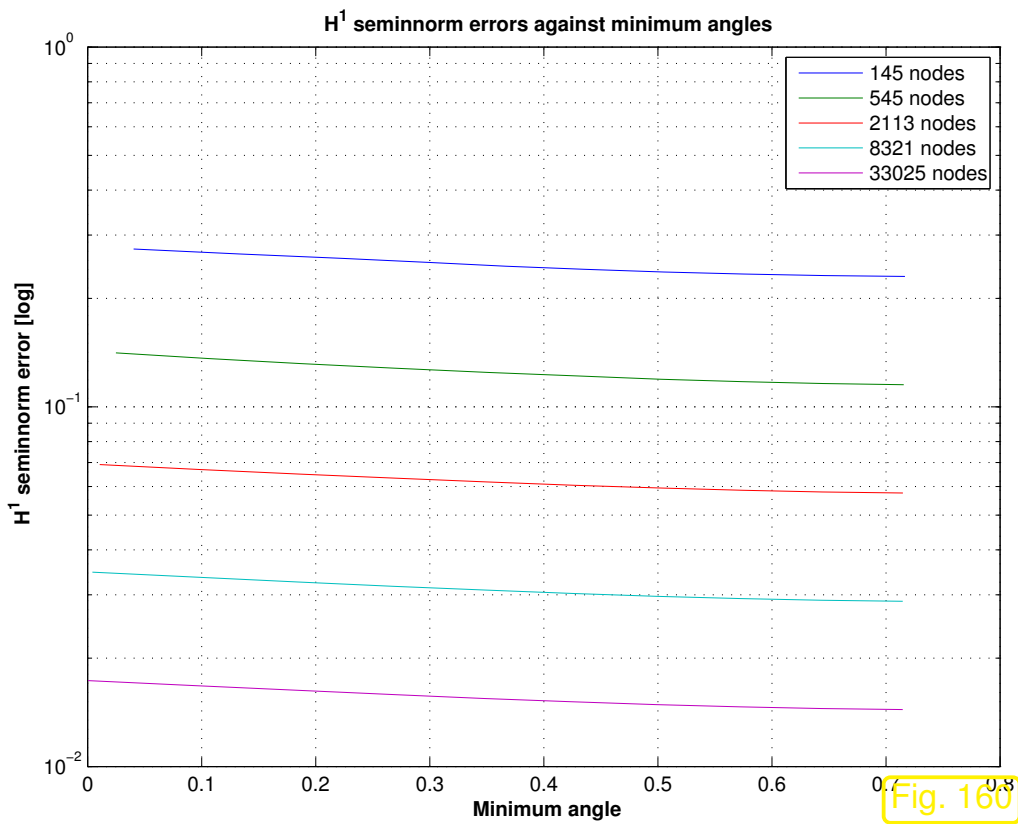
2D triangular mesh



Vertices : 145, # Elements : 256, # Edges : 208 **Fig. 159**

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Monitored: for different mesh resolutions, $H^1(\Omega)$ -seminorm of discretization error as function of smallest/largest angle in the mesh.

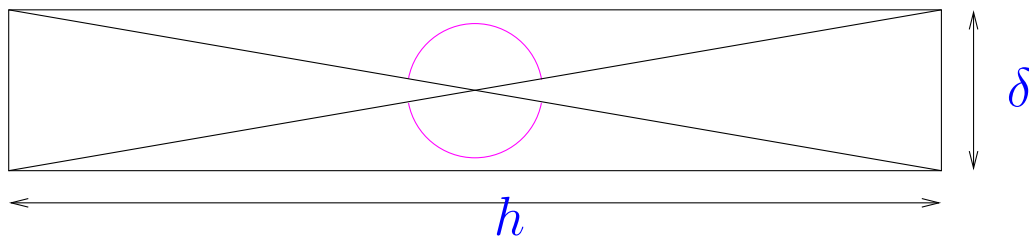
Observation: Accuracy does *not* suffer much from distorted elements !

Example 5.3.37 (Gap between interpolation error and best approximation error).

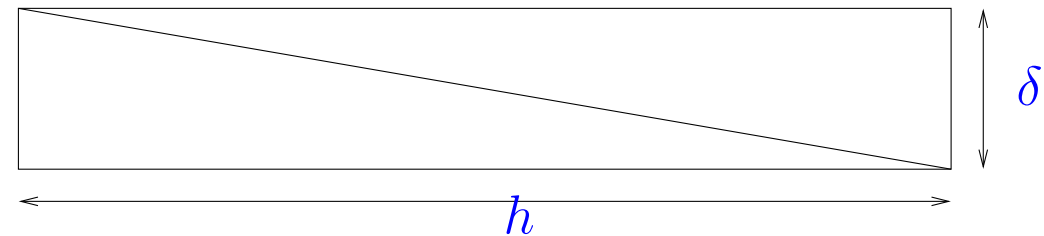
Ex. 5.3.36 raises doubts whether the interpolation error can be trusted to provide good, that is, reasonably sharp bounds for the best approximation error.

In this example we will see that

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_1 \ll \|u - I_p u\|_{H^1(\Omega)} \quad \text{is possible !}$$



Elementary cell of “bad mesh” \mathcal{M}_{bad}



Elementary cell of “good mesh” $\mathcal{M}_{\text{good}}$

On “bad” mesh : $\sup_{u \in H^2(\Omega)} \frac{\|u - I_1 u\|_{H^1(\Omega)}}{\|u\|_{H^2(\Omega)}} \rightarrow \infty$ as $h/\delta \rightarrow \infty$,

On “good” mesh : $\sup_{u \in H^2(\Omega)} \frac{\|u - I_1 u\|_{H^1(\Omega)}}{\|u\|_{H^2(\Omega)}} \quad \text{uniformly bounded in } h/\delta.$

Yet, $\inf_{v_N \in \mathcal{S}_1^0(\mathcal{M}_{\text{bad}})} \|u - v_N\|_{H^1(\Omega)} \leq \inf_{v_N \in \mathcal{S}_1^0(\mathcal{M}_{\text{good}})} \|u - v_N\|_{H^1(\Omega)} \quad \forall u \in H^2(\Omega).$



5.3.5 General approximation error estimates

In Sect. 5.3.2 we only examined the behavior of norms of the interpolation error for piecewise linear interpolation into $\mathcal{S}_1^0(\mathcal{M})$, that is, the case of Lagrangian finite elements of degree $p = 1$.

However, Ex. 5.2.2 sent the clear message that quadratic Lagrangian finite elements achieve faster convergence of the energy norm of the Galerkin discretization error, see Fig. 138, 139.



On the other hand quadratic finite elements could not deliver faster convergence in Ex. 5.2.6.

In this section we learn about theoretical results that shed light on these observations and extend the results of Sect. 5.3.2.

Remark 5.3.38 (L^∞ interpolation error estimate in 1D).

The faster convergence of quadratic Lagrangian FE in Ex. 5.2.2 does not come as a surprise: recall the estimate from [21, Eq. 9.4.6]:

$$\|u - I_p u\|_{L^\infty([a,b])} \leq \frac{h_{\mathcal{M}}^{p+1}}{(p+1)!} \|u^{(p+1)}\|_{L^\infty([a,b])} \quad \forall u \in C^{p+1}([a,b]),$$

where $I_p u$ is the \mathcal{M} -piecewise polynomial interpolant of u of local degree p . It generalizes (5.3.5).

$$\|u - I_p u\|_{L^\infty([a,b])} = O(h_{\mathcal{M}}^{p+1}) !$$

Remark 5.3.39 (Local interpolation onto higher degree Lagrangian finite element spaces).

\mathcal{M} : triangular/tetrahedral/quadrilateral/hybrid mesh of domain Ω (\rightarrow Sect. 3.3.1)

Recall (\rightarrow Sect. 3.4): nodal basis functions of p -th degree Lagrangian finite element space $\mathcal{S}_p^0(\mathcal{M})$ defined via **interpolation nodes**, cf. (3.4.3).

Set of interpolation nodes: $\mathcal{N} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\} \subset \bar{\Omega}$, $N = \dim \mathcal{S}_p^0(\mathcal{M})$.

\triangleright General **nodal Lagrangian interpolation operator**

$$I_p : \begin{cases} C^0(\bar{\Omega}) \mapsto \mathcal{S}_p^0(\mathcal{M}) \\ u \mapsto I_p(u) := \sum_{l=1}^N u(\mathbf{p}_l) b_N^l \end{cases},$$

where b_N^l are the nodal basis functions.

$$(3.4.3) \Rightarrow I_p(u)(\mathbf{p}_l) = u(\mathbf{p}_l), \quad l = 1, \dots, N \quad (\text{Interpolation!}).$$

By virtue of the location of the interpolation nodes, see Ex. 3.4.2, Ex. 3.4.5, and Fig. 104, the nodal interpolation operators are purely local:

$$\forall K \in \mathcal{M}: \quad I_p u|_K = \sum_{i=1}^Q u(\mathbf{q}_i^K) b_i^K, \quad (5.3.40)$$

$\mathbf{q}_i^K, i = 1, \dots, Q$ = local interpolation nodes in cell $K \in \mathcal{M}$, see Ex. 3.4.2, Ex. 3.4.5, and Fig. 104,
 $b_i^K, i = 1, \dots, Q$ = local shape functions: $b_i^K(\mathbf{q}_j^K) = \delta_{ij}$.

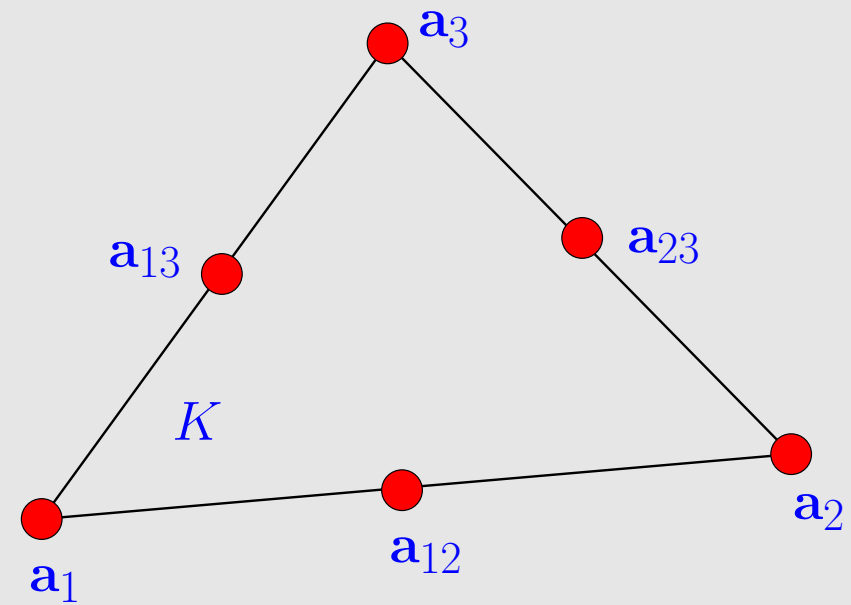
Example 5.3.41 (Piecewise quadratic interpolation). \rightarrow Ex. 3.4.2

triangle $K = \text{convex}\{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3\}$, $p = 2$

\Rightarrow local quadratic interpolation:

$$I_2 u|_K = - \sum_{i=1}^3 \lambda_i (1 - 2\lambda_i) u(\mathbf{a}^i) + \sum_{1 \leq i < j \leq 3} 4\lambda_i \lambda_j u\left(\frac{1}{2}(\mathbf{a}^i + \mathbf{a}^j)\right).$$

local shape functions, see (3.4.4)



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

The following theorem summarized best approximation results for **affine equivalent** Lagrangian FE spaces $\mathcal{S}_p^0(\mathcal{M})$ (\rightarrow Sect. 3.4) on mesh \mathcal{M} of a bounded polygonal/polyhedral domain $\Omega \subset \mathbb{R}^d$. It is the result of many years of research in approximation theory, see [29, Sect. 3.3], [1].

Theorem 5.3.42 (Best approximation error estimates for Lagrangian finite elements).

Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, be a bounded polygonal/polyhedral domain equipped with a mesh \mathcal{M} consisting of simplices or parallelepipeds. Then, for each $k \in \mathbb{N}$, there is a constant $C > 0$ depending only on k and the shape regularity measure $\rho_{\mathcal{M}}$ such that

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_{H^1(\Omega)} \leq C \left(\frac{h_{\mathcal{M}}}{p} \right)^{\min\{p+1, k\} - 1} \|u\|_{H^k(\Omega)} \quad \forall u \in H^k(\Omega). \quad (5.3.43)$$

This theorem is a typical example of finite element analysis results that you can find in the literature. It is important to know what kind of information can be gleaned from statements like that of Thm. 5.3.42.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 5.3.44 (“Generic constants”).

A statement like (5.3.43) is typical of a priori error estimates in the numerical analysis literature, which often come in the form

$$\|u - u_N\|_X \leq C \cdot \text{“discretization parameter”} \cdot \|u\|_Y,$$

where

- $C > 0$ is not specified precisely or only claimed to exist (though, in principle, they could be computed),

- C must neither depend on the exact solution u nor the discrete solution u_N ,

- the possible dependence of C on problem parameters or discretization parameters

Such constants $C > 0$ are known as **generic constants**. Customarily, different generic constants are even denoted by the same symbol (“ C ” is most common).

Remark 5.3.45 (Nature of a priori estimates). → Sect. 1.6.2

Cea’s lemma, Thm. 5.1.10 ➤ Thm. 5.3.42 implies a priori estimates of the energy norm of the finite element Galerkin discretization error (see also Rem. 5.3.28) of the form

$$\|u - u_N\|_a \leq C \left(\frac{h_{\mathcal{M}}}{p} \right)^{\min\{p+1, k\} - 1} \|u\|_{H^k(\Omega)}, \quad (5.3.46)$$

where u is the exact solution of the discretized 2nd-order elliptic boundary value problem.



(5.3.46) does not give concrete information about $\|u - u_N\|_a$, because

- we do not know the value of the “generic constant” $C > 0$, see Rem. 5.3.44,
- as u is unknown, a bound for $\|u\|_{H^k(\Omega)}$ may not be available.

A priori error estimates like (5.3.46) exhibit only the *trend* of the (norm of) the discretization error as discretization parameters $h_{\mathcal{M}}$ (mesh width), p (polynomial degree) are varied.



Remark 5.3.48. The estimate of Thm. 5.3.42 is *sharp*: the powers of $h_{\mathcal{M}}$ and p cannot be increased.



What questions can Thms. 5.3.42 and (5.3.46) answer? What do they tell us about the accuracy and *efficiency* of a Lagrangian finite element Galerkin discretization of a 2nd-order elliptic BVP? Closely related discussions have been developed for numerical quadrature, see [21, Sect. 10.3], and

higher order single step methods for initial value problems from ODEs, see [21, Rem. 12.4.1]. You are advised to review these passages in order to understand the parallels.

Question 5.3.49. *What computational effort buys us what error (measured in energy norm)?*

Bad luck (\rightarrow Rem. 5.3.45): actual error norm remains elusive! Therefore, rephrase the question so that it fits the available information about the effect of changing discretization parameters on the error:

Question 5.3.50. *What **increase** in computational effort buys us a prescribed **decrease** of the (energy norm of the) error?*

The answer to this question offers an a priori gauge of the *asymptotic efficiency* of a discretization method.

Convention: computational effort \approx number of unknowns $N = \dim \mathcal{S}_p^0(\mathcal{M})$ (**problem size**)

Framework: family \mathbb{M} of simplicial meshes of domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, created by *global* regular refinement of a single initial mesh

Global regular refinement of a simplicial mesh (\rightarrow Ex. 5.1.12)

- avoids greater distortion of “child cells” w.r.t. their parents,
- spawns meshes with fairly uniform size h_K of cells.

$$\exists C > 0: \quad \rho_{\mathcal{M}} \leq C, \quad \forall \mathcal{M} \in \mathbb{M}.$$

$$\exists C > 0: \quad \max\{h_K/h_{K'}, K, K' \in \mathcal{M}\} \leq C,$$

Now, for meshes $\in \mathbb{M}$, we investigate “ N -dependence”, $N = \dim \mathcal{S}_p^0(\mathcal{M})$, of energy norm of finite element discretization error:

Counting argument $N = \dim \mathcal{S}_p^0(\mathcal{M}) \approx p^d h_{\mathcal{M}}^{-d} \Rightarrow \boxed{\frac{h_{\mathcal{M}}}{p} \approx N^{-1/d}}. \quad (5.3.51)$

dimensions of local spaces, Lemma 3.3.6 $\sim \#\mathcal{M} \sim \#\mathcal{V}(\mathcal{M}), \mathcal{E}(\mathcal{M})$ etc.

Notation: $\approx \hat{=}$ uniform equivalence on the set \mathbb{M} , that is, each side can be bounded by a constant times the other, and the constants can be chosen independently of the mesh $\mathcal{M} \in \mathbb{M}$

Example 5.3.52 (Dimensions of Lagrangian finite element spaces on triangular meshes).

$d = 2$: for triangular meshes \mathcal{M} , by Lemma 3.3.6

$$\dim \mathcal{S}_p^0(\mathcal{M}) = \#\{\text{nodes}(\mathcal{M})\} + \#\{\text{edges}(\mathcal{M})\} (p - 1) + \#\mathcal{M} \frac{1}{2}(p - 1)(p - 2) .$$

1 basis function per vertex

$p - 1$ basis functions per edge

$\frac{1}{2}(p - 1)(p - 2)$ “interior” basis functions

Geometric considerations: the number of triangles sharing a vertex can be bounded in terms of $\rho_{\mathcal{M}}$, because $\rho_{\mathcal{M}}$ implies a lower bound for the smallest angles of the triangular cells.

$$\exists C = C(\rho_{\mathcal{M}}): \#\{K_j \in \mathcal{M}: \overline{K}_i \cap \overline{K}_j \neq \emptyset\} \leq C \quad (i = 1, 2, \dots, \#\mathcal{M}) .$$

If every vertex belongs only to a small number of triangles, the number $\#\{\text{nodes}(\mathcal{M})\}$ can be bounded by $C \cdot \#\mathcal{M}$, where $C > 0$ will depend on $\rho_{\mathcal{M}}$ only. The same applies to the edges.

$$\blacktriangleright \quad \#\{\text{nodes}(\mathcal{M})\}, \#\{\text{edges}(\mathcal{M})\} \approx \#\mathcal{M} .$$



$$\dim \mathcal{S}_p^0(\mathcal{M}) \approx (\#\mathcal{M})p^2 , \quad (5.3.53)$$

with constants hidden in \approx depending on $\rho_{\mathcal{M}}$ only.

Now, we merge (5.3.46) and (5.3.51):

$$\boxed{u \in H^k(\Omega)} \quad \text{Thm. 5.3.42} \quad \Rightarrow \quad \inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_{H^1(\Omega)} \leq CN^{-\frac{\min\{p, k-1\}}{d}} \|u\|_{H^k(\Omega)} , \quad (5.3.54)$$

with $C > 0$ depending *only* on d, p, k , and $\rho_{\mathcal{M}}$.

(5.3.54) ➤ **algebraic convergence** (→ Def. 1.6.32) in problem size

$$\left(\text{rate } \frac{\min\{p, k - 1\}}{d}\right)$$

We observe that

- the rate of convergence is limited by the polynomial degree p of the Lagrangian FEM,
- the rate of convergence is limited by the smoothness of the exact solution u , measured by means of the Sobolev index k , see Sect. 5.3.3,
- the rate of convergence will be worse for $d = 3$ than for $d = 2$, the effect being more pronounced for small k or p .

Answer to Question 5.3.50:

Assumption: a priori error estimate (5.3.54) is *sharp*

$$\exists C = C(u, \dots) > 0: \quad \text{error norm}(N) \approx CN^{-\frac{\min\{p, k-1\}}{d}} \quad \forall \mathcal{M} \in \mathbb{M}.$$

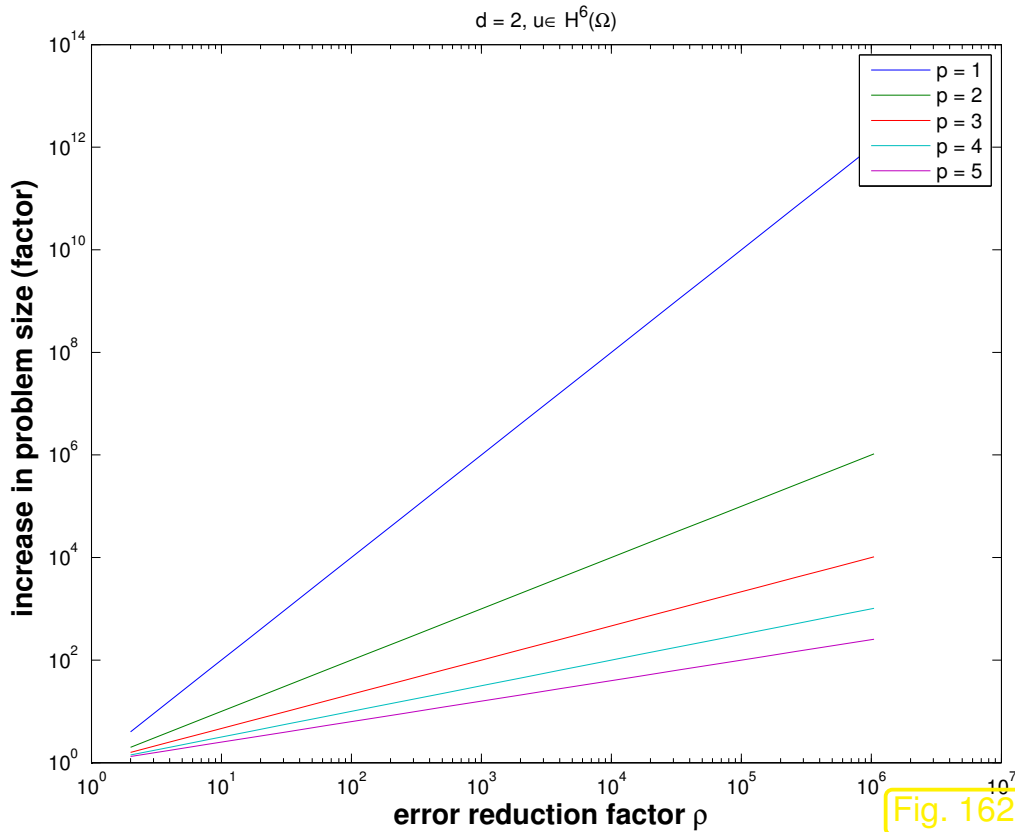
$$\blacktriangleright \frac{\text{error norm}(N_1)}{\text{error norm}(N_2)} \approx \left(\frac{N_1}{N_2}\right)^{-\frac{\min\{p, k-1\}}{d}}.$$

reduction of (the energy norm of) the error by a factor $\rho > 1$

requires

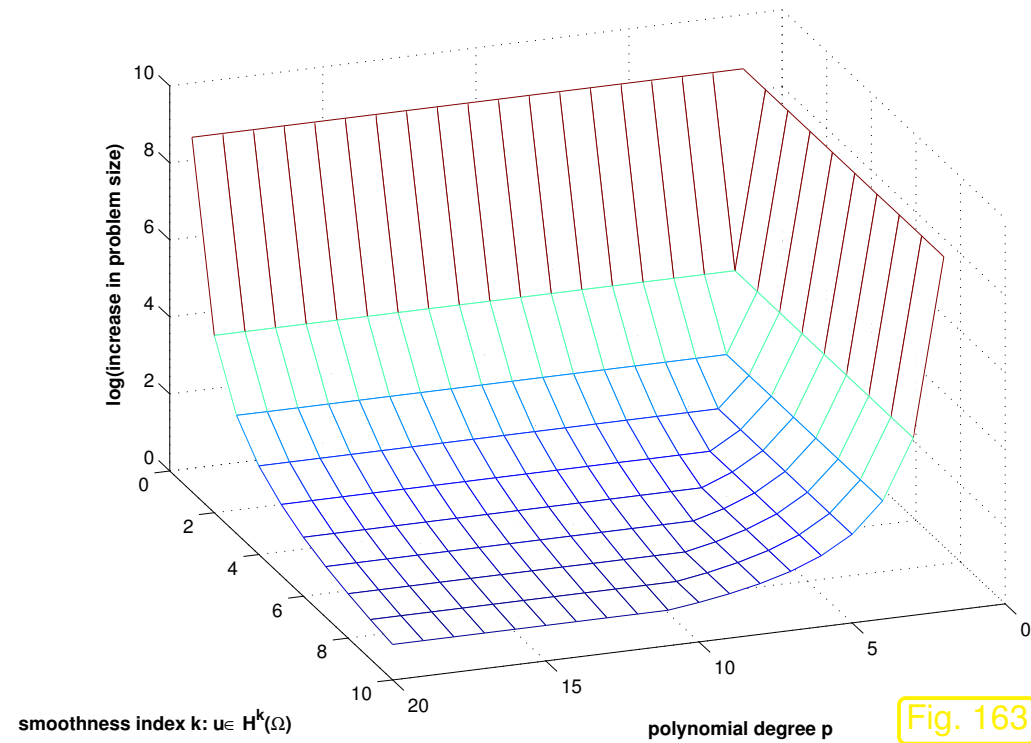
increase of the problem size

by factor $\rho^{\frac{d}{\min\{p, k-1\}}}$



exact solution $u \in H^6(\Omega)$

Fig. 162



error reduction by factor $\rho = 100$

Fig. 163

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Discussion:

Solution $u \in H^k(\Omega)$ \triangleright optimal asymptotic efficiency for $p = k - 1$

Remark 5.3.55 (General asymptotic estimates).

Recall (\rightarrow Sect. 1.6.2):

convergence is an asymptotic notion

Now we deduce **asymptotic** estimates for the best approximation errors from Thm. 5.3.42, and (5.3.54), in particular, for the case $N \rightarrow \infty$:

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

• h-refinement: p fixed, $h_{\mathcal{M}} \rightarrow 0$ for $\mathcal{M} \in \mathbb{M}$:

(5.3.54) \Rightarrow algebraic convergence w.r.t. N

\blacktriangleright $p \leq k - 1$ \blacktriangleright

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_1 = O(N^{-p/d}) \quad (5.3.56)$$

☞ $k \leq p + 1$ ►

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_1 = O(N^{-(k-1)/d}) \quad (5.3.57)$$

Note: for very smooth solution u , i.e. $k \gg 1$, polynomial degree p limits speed of convergence

- p-refinement: $\mathcal{M} \in \mathbb{M}$ fixed, $p \rightarrow \infty$:

☞ p large ►

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_1 = O(N^{-(k-1)/d}) \quad (5.3.58)$$

Note: arbitrarily fast (**super-**)algebraic convergence for very smooth solutions $u \in C^\infty(\bar{\Omega})$



5.4 Elliptic regularity theory

Crudely speaking, in Sect. 5.3.5 we saw that the asymptotic behavior of the Lagrangian finite element Galerkin discretization error (for 2nd-order elliptic BVPs) can be predicted provided that

- we use families of meshes, whose cells have rather uniform size and whose shape regularity measure is uniformly bounded,
- we have an *idea about the smoothness of the exact solution u* , that is, we know $u \in H^k(\Omega)$ for a (maximal) k , see Thm. 5.3.42.

Knowledge about the mesh can be taken for granted, but

how can we guess the smoothness of the (unknown !) exact solution u ?

A (partial) answer is given in this section.

Focus: Scalar 2nd-order elliptic BVP with homogeneous Dirichlet boundary conditions

$$-\operatorname{div}(\sigma(\mathbf{x}) \mathbf{grad} u) = f \quad \text{in } \Omega \quad , \quad u = g \quad \text{on } \partial\Omega .$$

To begin with, we summarize the available information:

➤ Known:

 u solves BVP

+

Information about coefficient σ ,
domain Ω , source function f ,
boundary data g


u will belong to a certain **class of functions** (e.g. subspace $S \subset V$)

Example 5.4.1 (Elliptic lifting result in 1D).

$d = 1$, $\Omega =]0, 1[$, coefficient $\sigma \equiv 1$, homogeneous Dirichlet boundary conditions:

$$u'' = f \quad , \quad u(0) = u(1) = 0 \quad .$$

Obvious:

$$f \in H^k(\Omega) \quad \Rightarrow \quad u \in H^{k+2}(\Omega) \quad (\text{a lifting theorem})$$



Can this be generalized to higher dimensions $d > 1$?

Partly so:

Theorem 5.4.2 (Smooth elliptic lifting theorem).

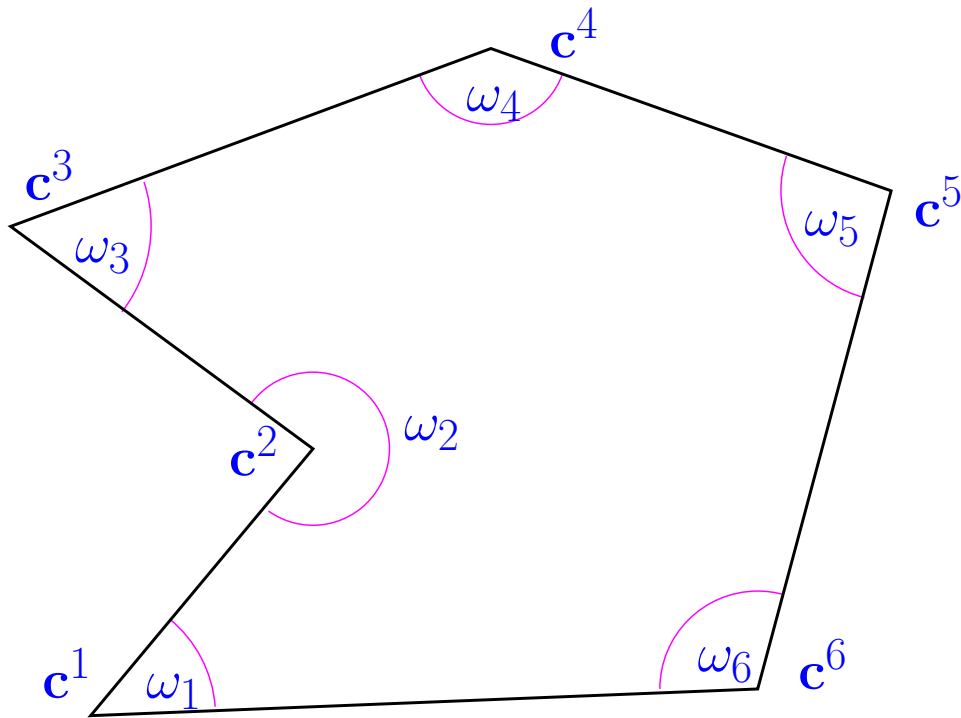
If $\partial\Omega$ is C^∞ -smooth, ie. possesses a local parameterization by C^∞ -functions, and $\sigma \in C^\infty(\bar{\Omega})$, then, for any $k \in \mathbb{N}$,

$$u \in H_0^1(\Omega) \quad \text{and} \quad -\operatorname{div}(\sigma \mathbf{grad} u) \in H^k(\Omega) \quad \Rightarrow \quad u \in H^{k+2}(\Omega) .$$
$$u \in H^1(\Omega) , \quad -\operatorname{div}(\sigma \mathbf{grad} u) \in H^k(\Omega) \quad \text{and} \quad \mathbf{grad} u \cdot \mathbf{n} = 0 \quad \text{on} \quad \partial\Omega$$

In addition, for such u there is $C = C(k, \Omega, \sigma)$ such that

$$\|u\|_{H^{k+2}(\Omega)} \leq C \|\operatorname{div}(\sigma \mathbf{grad} u)\|_{H^k(\Omega)} .$$

What about non-smooth $\partial\Omega$?



These are very common in engineering applications (“CAD-geometries”).

◁ polygonal domain with corners c^i

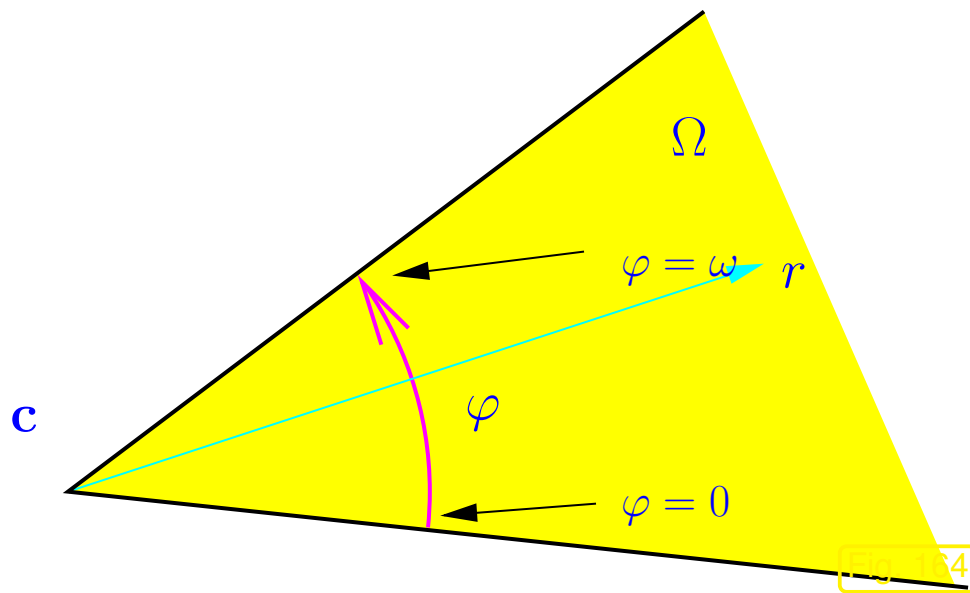
How will the corners affect the smoothness of solutions of

$$u \in H_0^1(\Omega): \quad \Delta u = f \in C^\infty(\bar{\Omega})?$$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Example 5.4.3 (Corner singular functions).



corner singular function

$$u_s(r, \varphi) = r^{\frac{\pi}{\omega}} \sin\left(\frac{\pi}{\omega}\varphi\right), \quad (5.4.4)$$

$$r \geq 0, \quad 0 \leq \varphi \leq \omega.$$

(in local polar coordinates)

► $u_s = 0$ on $\partial\Omega$ locally at \mathbf{c} !

Straightforward computation:

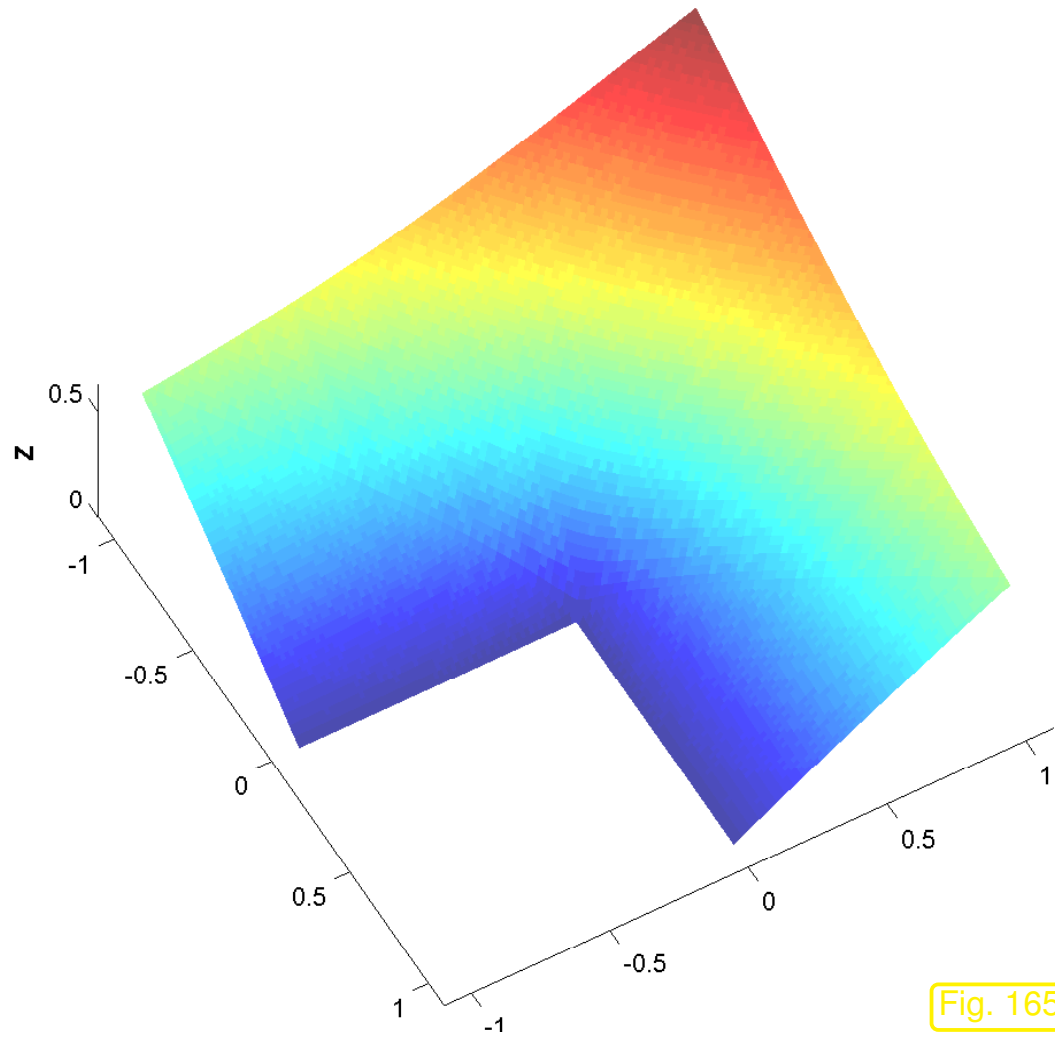
$$\Delta u_s = 0 \quad \text{in } \Omega!$$

To see this recall: Δ in polar coordinates:

$$\Delta u = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \varphi^2}. \quad (5.4.5)$$

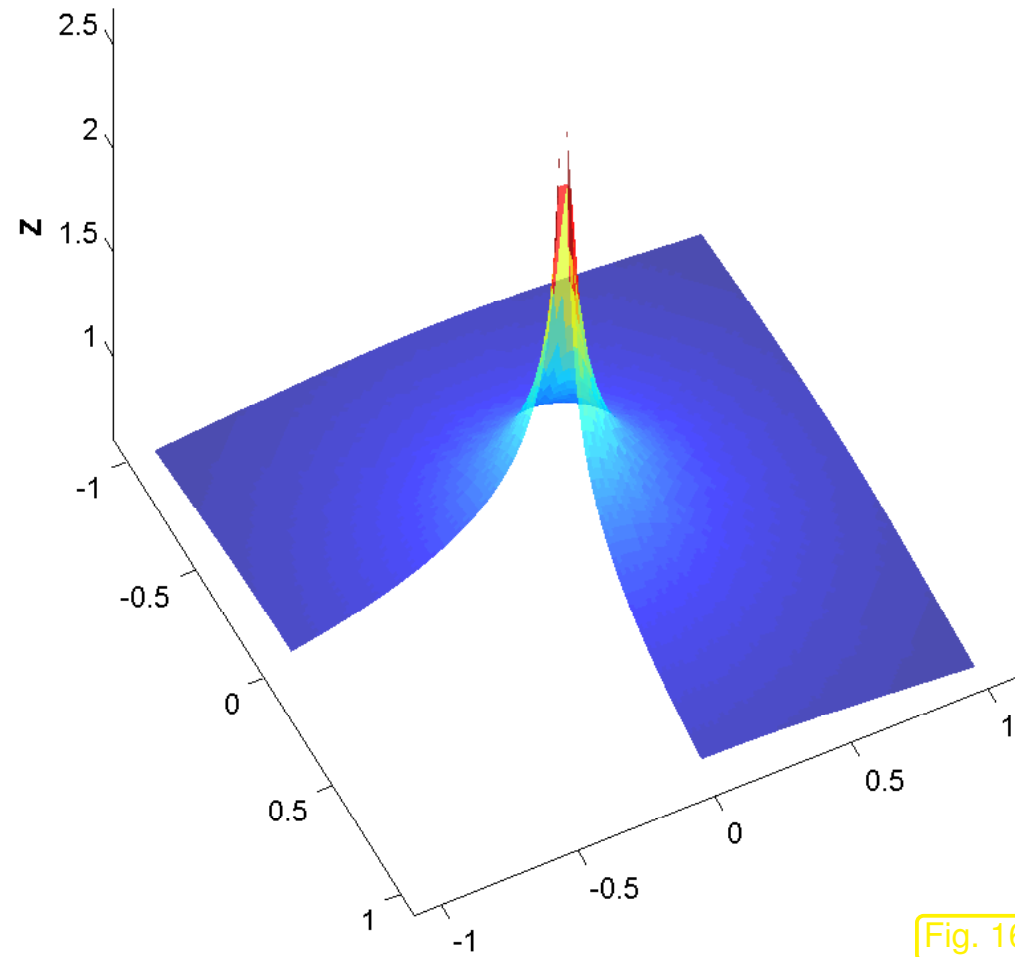
$$\begin{aligned} \stackrel{(5.4.4)}{\implies} \Delta u_s(r, \varphi) &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\pi}{\omega} r^{\frac{\pi}{\omega}-1} \sin\left(\frac{\pi}{\omega}\varphi\right) \right) + \frac{1}{r^2} r^{\frac{\pi}{\omega}} \frac{\partial}{\partial \varphi} \cos\left(\frac{\pi}{\omega}\varphi\right) \frac{\pi}{\omega} \\ &= \left(\frac{\pi}{\omega}\right)^2 r^{\frac{\pi}{\omega}-2} \sin\left(\frac{\pi}{\omega}\varphi\right) - \left(\frac{\pi}{\omega}\right)^2 r^{\frac{\pi}{\omega}-2} \sin\left(\frac{\pi}{\omega}\varphi\right) = 0. \end{aligned}$$

What is “singular” about these functions? Plot them for $\omega = \frac{3\pi}{2}$, cf. Ex. 5.2.6



u_s for $\omega = \frac{3\pi}{2}$

Fig. 165



$\|\text{grad } u_s\|$ for $\omega = \frac{3\pi}{2}$

Fig. 166

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Recall gradient (2.3.33) in polar coordinates

$$\mathbf{grad} u = \frac{\partial u}{\partial r} \mathbf{e}_r + \frac{1}{r} \frac{\partial u}{\partial \varphi} \mathbf{e}_\varphi . \quad (2.3.33)$$

$$(5.4.4) \quad \implies \mathbf{grad} u_s(r, \varphi) = \frac{\pi}{\omega} r^{\frac{\pi}{\omega}-1} \left(\sin\left(\frac{\pi}{\omega}\varphi\right) \mathbf{e}_r + \cos\left(\frac{\pi}{\omega}\varphi\right) \mathbf{e}_\varphi \right) .$$

$$\omega > \pi \text{ (“re-entrant corner”) } \implies \text{“grad } u_s(0) = \infty\text{”}$$

How does this “blow-up” of the gradient affect the **Sobolev regularity** (that is, the smoothness as expressed through “ $u_s \in H^k(\Omega)$ ”) of the corner singular function u_s ?

 R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

We try to compute $|u|_{H^2(D)}$, with (in polar coordinates, see Fig. 164)

$$D := \{(r, \varphi) : 0 < r < 1, 0 < \varphi < \omega\} .$$

By tedious computations we find

$$\omega > \pi \implies \int_D \left\| D^2 u_s(r, \varphi) \right\|_F^2 r d(r, \varphi) = \infty .$$

$$\stackrel{\text{Def. 5.3.30}}{\implies} \left\{ \omega > \pi \implies u_s \notin H^2(D) \right\} .$$



Bad news: With the exception of concocted examples,
corner singular functions like (5.4.4) will be present in the solution of linear scalar
2nd-order elliptic BVP on polygonal domains!

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

The meaning of “being present” is elucidated in the following theorem:

Theorem 5.4.6 (Corner singular function decomposition).

Let $\Omega \subset \mathbb{R}^2$ be a polygon with J corners \mathbf{c}^i . Denote the polar coordinates in the corner \mathbf{c}^i by (r_i, φ_i) and the inner angle at the corner \mathbf{c}^i by ω_i . Additionally, let $f \in H^l(\Omega)$ with $l \in \mathbb{N}_0$ and $l \neq \lambda_{ik} - 1$, where the λ_{ik} are given by the **singular exponents**

$$\lambda_{ik} = \frac{k\pi}{\omega_i} \quad \text{for } k \in \mathbb{N}. \quad (5.4.7)$$

Then $u \in H_0^1(\Omega)$ with $-\Delta u = f$ in Ω can be decomposed

$$u = u^0 + \sum_{i=1}^J \psi(r_i) \sum_{\lambda_{ik} < l+1} \kappa_{ik} s_{ik}(r_i, \varphi_i), \quad \kappa_{ik} \in \mathbb{R}, \quad (5.4.8)$$

with **regular part** $u^0 \in H^{l+2}(\Omega)$, cut-off functions $\psi \in C^\infty(\mathbb{R}^+)$ ($\psi \equiv 1$ in a neighborhood of 0), and corner singular functions

$$\begin{aligned} \lambda_{ik} \notin \mathbb{N}: s_{ik}(r, \varphi) &= r^{\lambda_{ik}} \sin(\lambda_{ik}\varphi), \\ \lambda_{ik} \in \mathbb{N}: s_{ik}(r, \varphi) &= r^{\lambda_{ik}} (\ln r) \sin(\lambda_{ik}\varphi). \end{aligned} \quad (5.4.9)$$

$\Omega \subset \mathbb{R}^2$ has **re-entrant corners**

\Rightarrow

if u solves $\Delta u = f$ in Ω , $u = 0$ on $\partial\Omega$,
then $u \notin H^2(\Omega)$ in general.

Theorem 5.4.10 (Elliptic lifting theorem on convex domains).

If $\Omega \subset \mathbb{R}^d$ convex, $u \in H_0^1(\Omega)$, $\Delta u \in L^2(\Omega) \Rightarrow u \in H^2(\Omega)$.

Terminology: if conclusion of Thm. 5.4.10 true \rightarrow Dirichlet problem **2-regular**.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

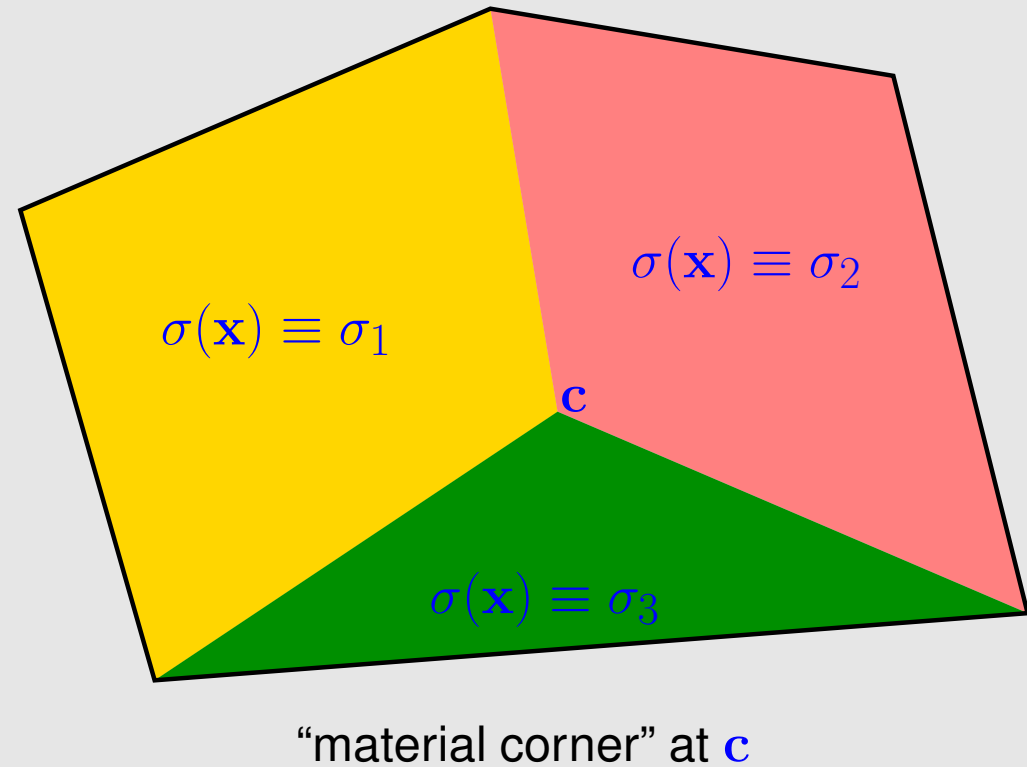
SAM, ETHZ

Similar lifting theorems also hold for Neumann BVPs, BVPs with *smooth* coefficients.

Remark 5.4.11 (Causes for non-smoothness of solutions of elliptic BVPs).

Causes for poor Sobolev regularity of solution u of BVPs for $-\operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u) = f$:

- Corner of $\partial\Omega$, see above
- Discontinuities of σ
→ singular functions at “material corners”
- Mixed boundary conditions
- Non-smooth source function f



5.5 Variational crimes

Variational crime = replacing (exact) discrete (linear) variational problem

$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = f(v_N) \quad \forall v_N \in V_{0,N}, \quad (3.1.4)$$

with **perturbed variational problem**

$$\tilde{u}_N \in V_{0,N}: \quad \mathbf{a}_N(\tilde{u}_N, v_N) = f_N(v_N) \quad \forall v_N \in V_{0,N}. \quad (5.5.1)$$

▶ perturbation of Galerkin solution u_N \mapsto perturbed solution $\tilde{u}_N \in V_{0,N}$

Approximations $\mathbf{a}_N(\cdot, \cdot) \approx \mathbf{a}(\cdot, \cdot)$, $f_N(\cdot) \approx f(\cdot)$ due to

- use of numerical quadrature \rightarrow Sect. 3.5.4,
- approximation of boundary $\partial\Omega$ \rightarrow Sect. 3.6.4.

We are all sinners! Variational crimes are *inevitable* in practical FEM, recall Rem. 1.5.6!

Which “variational petty crimes” can be tolerated?

5.5.1 Impact of numerical quadrature

Model problem: on polygonal/polyhedral $\Omega \subset \mathbb{R}^d$:

$$u \in H_0^1(\Omega): \quad \mathbf{a}(u, v) := \int_{\Omega} \sigma(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = f(v) := \int_{\Omega} f v \, d\mathbf{x} . \quad (5.5.3)$$

Assumptions: σ satisfies (2.5.4), $\sigma \in C^0(\bar{\Omega})$, $f \in C^0(\bar{\Omega})$

- Galerkin finite element discretization, $V_N := \mathcal{S}_p^0(\mathcal{M})$ on simplicial mesh \mathcal{M}
- Approximate evaluation of $\mathbf{a}(u_N, v_N)$, $f(v_N)$ by a fixed stable local numerical quadrature rule (\rightarrow Sect. 3.5.4)bigskip
 - perturbed bilinear form \mathbf{a}_N , right hand side f_N (see (5.5.1))

Focus: h -refinement (key discretization parameter is the mesh width $h_{\mathcal{M}}$)

Example 5.5.4 (Impact of numerical quadrature on finite element discretization error).

$$\Omega =]0, 1[^2, \sigma \equiv 1, f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y), (x, y)^T \in \Omega$$

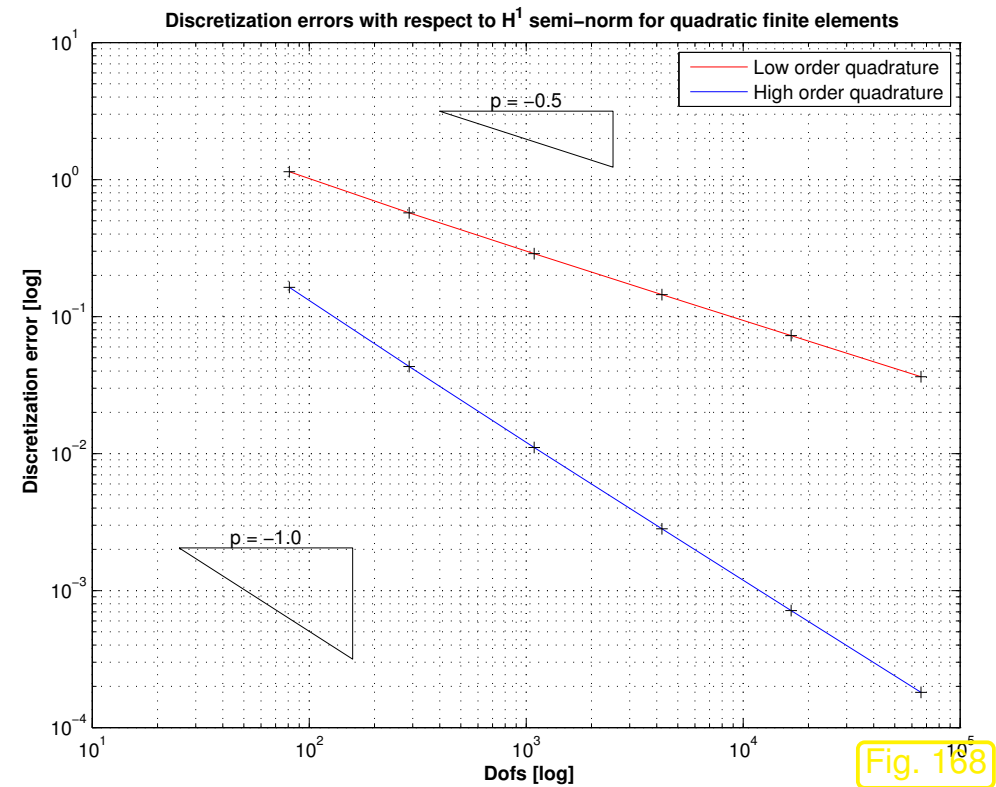
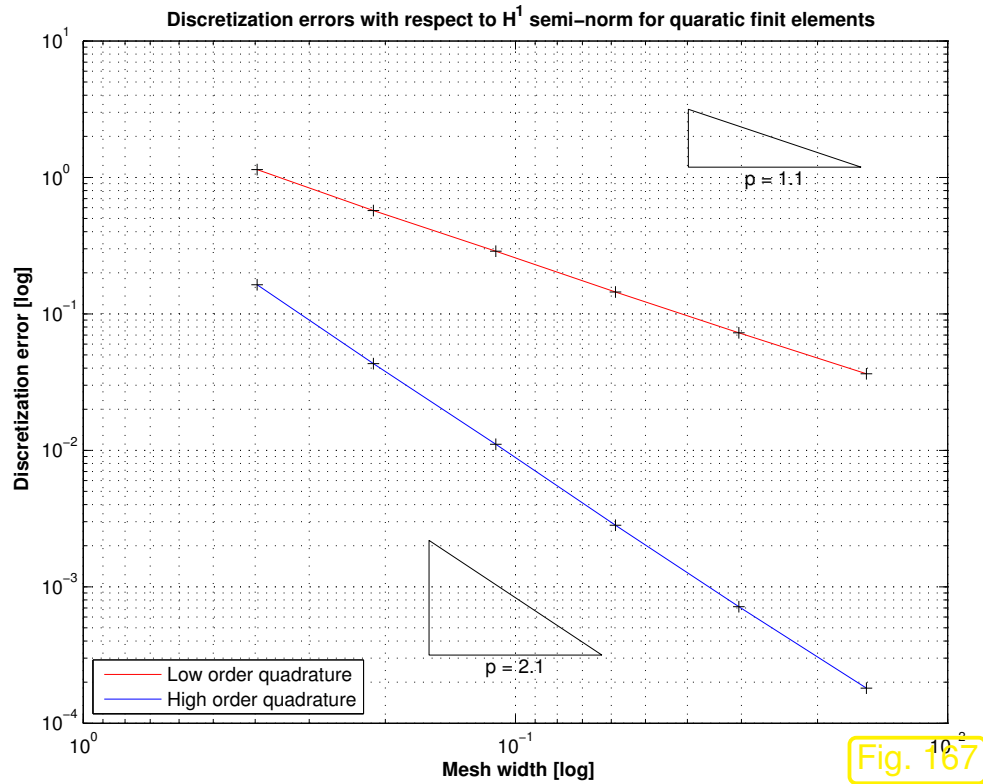
➤ solution $u(x, y) = \sin(\pi x) \sin(\pi y), g = 0.$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

Details of numerical experiment:

SAM, ETHZ

- *Quadratic* Lagrangian FE ($V_N = \mathcal{S}_2^0(\mathcal{M})$) on triangular meshes \mathcal{M} , obtained by regular refinement
- “Exact” evaluation of bilinear form by very high order quadrature
- f_N from one point quadrature rule (3.5.37) *of order 2*



$H^1(\Omega)$ -norm of discretization error on unit square (— \leftrightarrow rule (3.5.37), — \leftrightarrow rule (3.5.38))

Observation: Use of quadrature rule of order 2 \Rightarrow Algebraic rate of convergence (w.r.t. N) drops from $\alpha = 1$ to $\alpha = 1/2$!

Finite element theory [10, Ch. 4,§4.1] tells us that the Guideline 5.5.2 can be met, if the local numerical quadrature rule has sufficiently high order. The quantitative results can be condensed into the following rules of thumb:

$\|u - u_N\|_1 = O(h_{\mathcal{M}}^p)$ at best \blacktriangleright Quadrature rule of order $2p - 1$ sufficient for f_N .

$\|u - u_N\|_1 = O(h_{\mathcal{M}}^p)$ at best \blacktriangleright Quadrature rule of order $2p - 1$ sufficient for \mathbf{a}_N .

5.5.2 Approximation of boundary

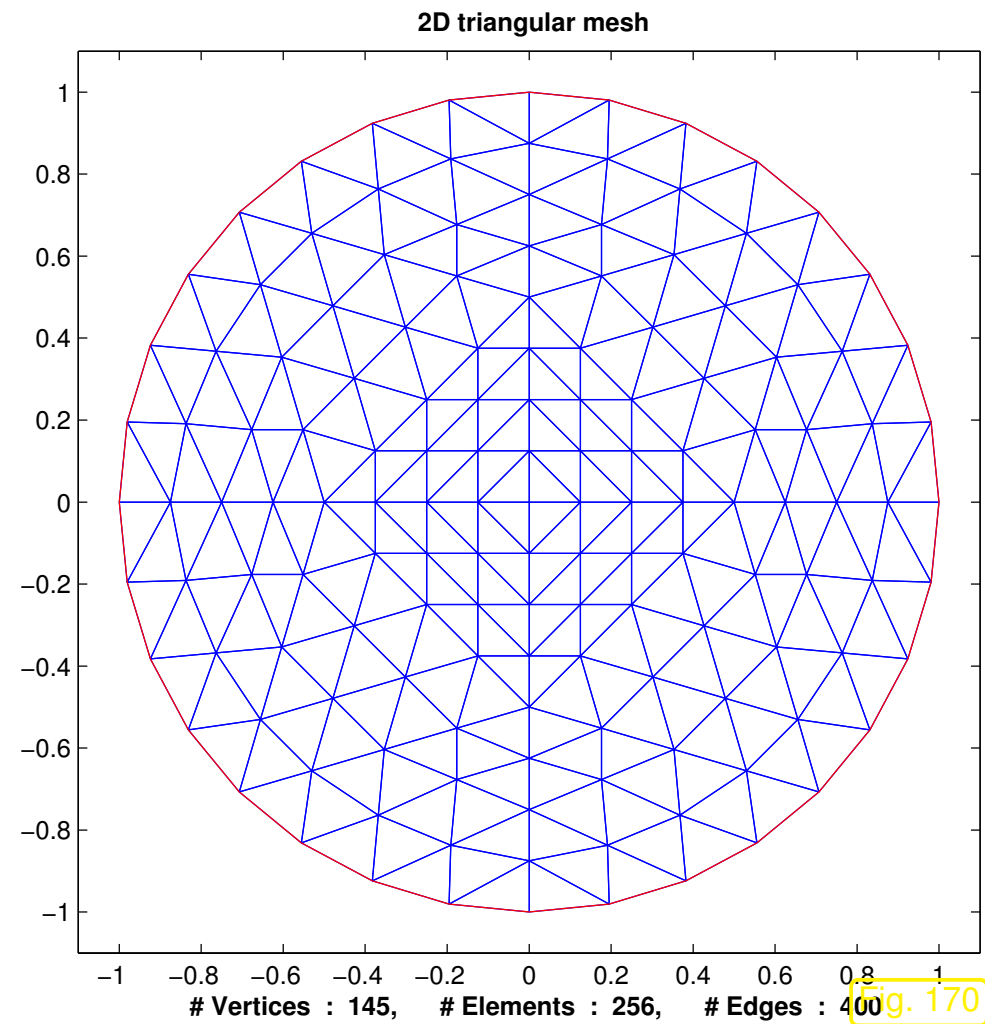
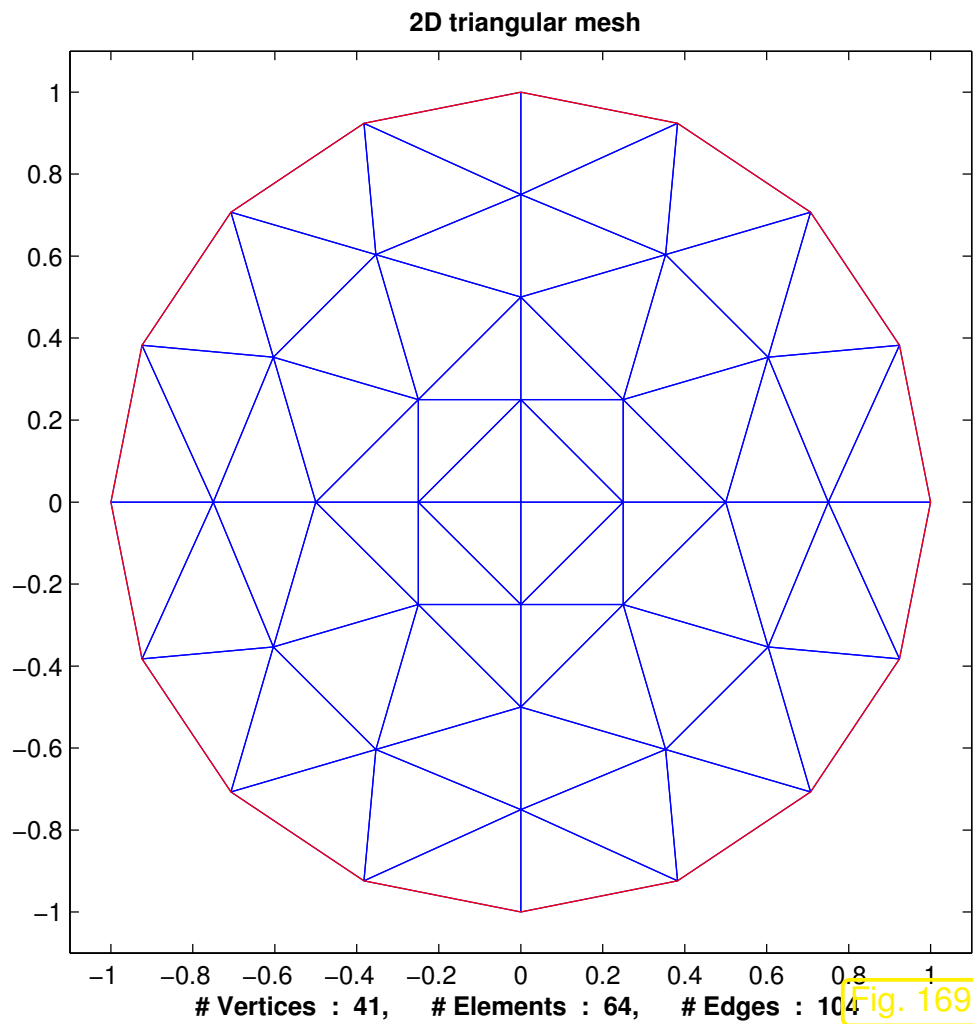
We focus on 2nd-order scalar linear variational problems as in the previous section.

Example 5.5.5 (Impact of linear boundary fitting on FE convergence).

Setting: $\Omega := B_1(0) := \{\mathbf{x} \in \mathbb{R}^2: |\mathbf{x}| < 1\}$, $u(r, \varphi) = \cos(r\pi/2)$ (polar coordinates)
➤ $f = \frac{\pi}{2r} \sin(r\pi/2) + \frac{\pi}{2} \cos(r\pi/2)$

- Sequences of unstructured triangular meshes \mathcal{M} obtained by regular refinement (of coarse mesh with 4 triangles) + linear boundary fitting.
- Galerkin FE discretization based on $V_N := \mathcal{S}_{1,0}^0(\mathcal{M})$ or $V_N := \mathcal{S}_{2,0}^0(\mathcal{M})$.
- Recorded: approximate norm $|u - u_N|_{1,\Omega_h}$, evaluated using numerical quadrature rule (3.5.38).

(FE solution extended beyond the domain covered by \mathcal{M} (“mesh interior”) to Ω (“full domain”) by means of polynomial extrapolation.)



Linearly boundary fitted unstructured triangular meshes of $\Omega = B_1(0)$.

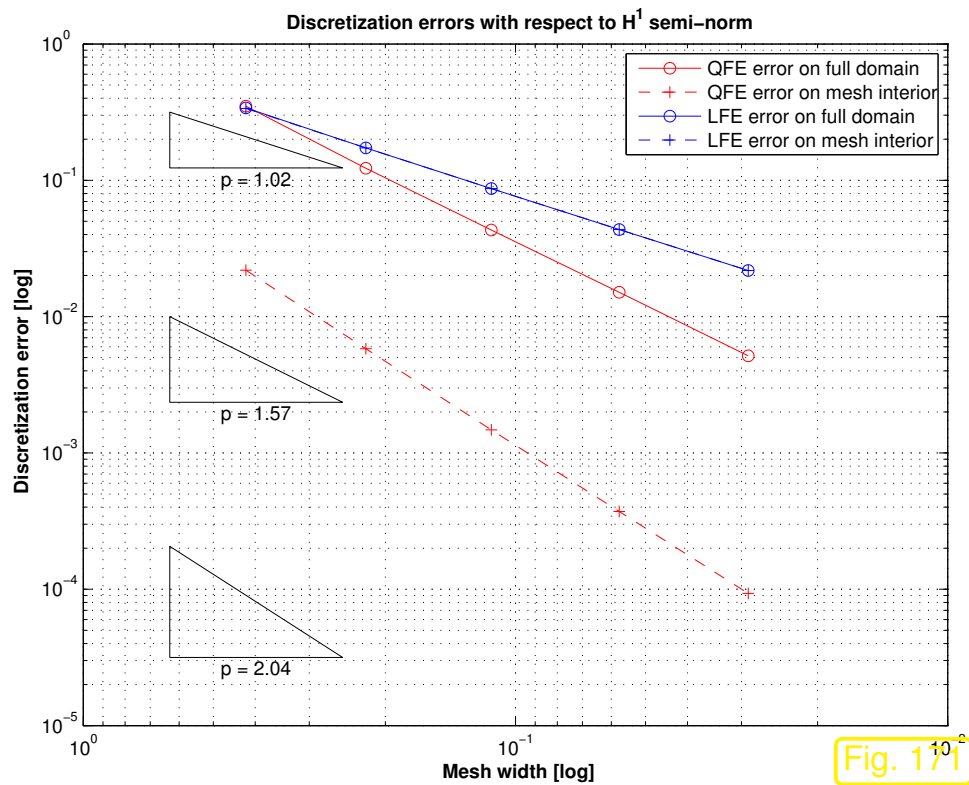


Fig. 171

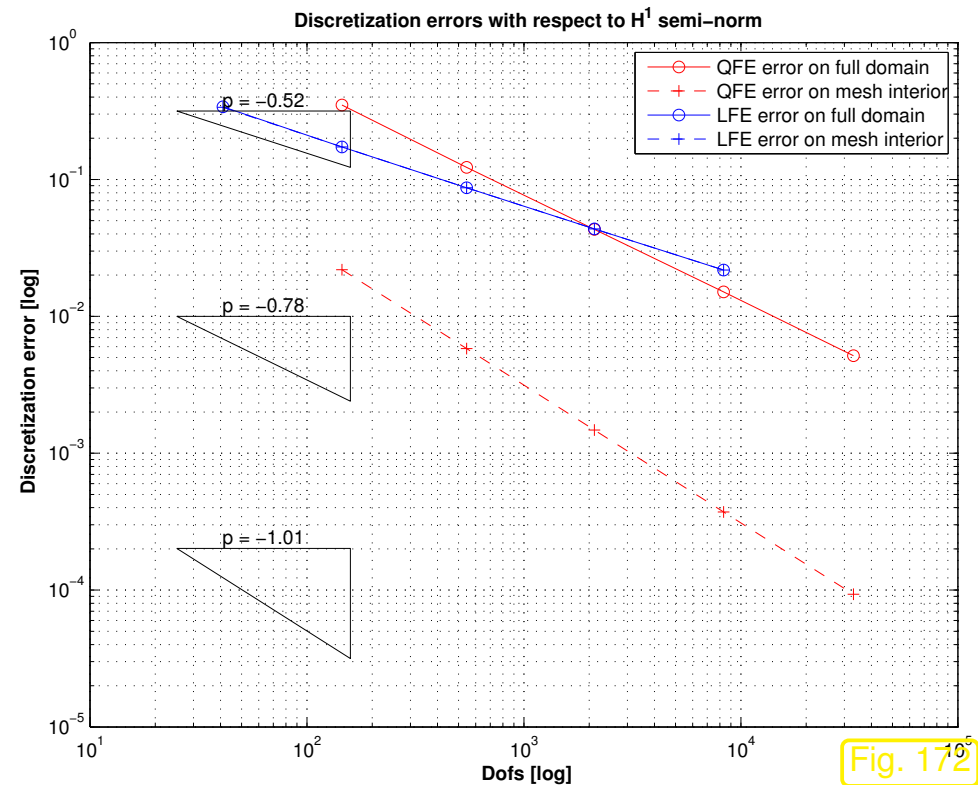


Fig. 172

$H^1(\Omega)$ -norm of discretization error on unit ball ($- \leftrightarrow p = 1$, $- \leftrightarrow p = 2$)

Dashed lines in Figs. 171, 172: error norms computed on polygonal domain covered by the mesh $\neq \Omega$; this spurious “error norm” suggests no deterioration of the convergence!



If $V_{0,N} = \mathcal{S}_p^0(\mathcal{M})$, use boundary fitting with polynomials of degree p .

5.6 Duality techniques

5.6.1 Linear output functionals

Adopt abstract setting of Sect. 5.1:

linear variational problem (1.4.7) in the form

$$u \in V_0: \quad \mathbf{a}(u, v) = \ell(v) \quad \forall v \in V_0, \quad (3.1.1)$$

- $V_0 \hat{=}$ (real) vector space, a space of functions $\Omega \mapsto \mathbb{R}$ for scalar 2nd-order elliptic variational problems, usually “energy space” $H^1(\Omega)/H_0^1(\Omega)$, see Sect. 2.2
- $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R} \hat{=}$ a bilinear form, see Def. 1.3.23,
- $\ell : V_0 \mapsto \mathbb{R} \hat{=}$ a linear form, see Def. 1.3.23,
- Assumptions 5.1.1, 5.1.2, 5.1.3 are supposed to hold \triangleright existence, uniqueness, and stability of solution u by Thm. 5.1.4.

(Examples of 2nd-order linear BVPs discussed in Rem. 5.1.5, Sect. 2.8)

Galerkin discretization using $V_{0,N} \subset V_0 \triangleright$ discrete variational problem

$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = f(v_N) \quad \forall v_N \in V_{0,N} . \quad (3.1.4)$$

New twist: we are interested mainly/only in the *number* $F(u)$, where

$F : V_0 \mapsto \mathbb{R}$ is an **output functional**.

Mathematical terminology: **functional** $\hat{=}$ mapping from a function space into \mathbb{R}

Example 5.6.1 (Output functionals).

Some output functionals for solutions of PDEs commonly encountered in applications:

- mean values, see Ex. 5.6.4 below
- total heat flux through a surface (for heat conduction model \rightarrow Sect. 2.5), see Ex. 5.6.13 below
- total surface charge of a conducting body (for electrostatics \rightarrow Sect. 2.1.2)
- total heat production (Ohmic losses) by stationary currents
- total force on a charged conductor (for electrostatics \rightarrow Sect. 2.1.2)
- lift and drag in computational fluid dynamics (aircraft simulation)
- and many more . . .



We consider output functionals with special properties, which are rather common in practice:

Assumption 5.6.2 (Linearity of output functional).

The output functional F is a **linear** form (\rightarrow Def. 1.3.23) on V_0 .

To put the next assumption into context, please recall Ass. 5.1.2 and Rem. 2.3.18.

Assumption 5.6.3 (Continuity of output functional).

The output functional is **continuous** w.r.t. the energy norm in the sense that

$$\exists C_f > 0: |F(v)| \leq C_f \|v\|_a \quad \forall v \in V_0 .$$

Now consider Galerkin discretization of (3.1.1) based on Galerkin trial/test space $V_{0,N} \subset V_0$, $N := \dim V_{0,N} < \infty$ \triangleright discrete variational problem

$$u_N \in V_{0,N}: \mathbf{a}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N} . \quad (3.1.4)$$

What would you dare to sell as an approximation of $F(u)$? Of course, ...

Galerkin solution $u_N \in V_{0,N}$ \mapsto approximate output value $F(u_N)$

How accurate is $F(u_N)$, that is, how big is the **output error** $|F(u) - F(u_N)|$?

Linearity (\rightarrow Ass. 5.6.2) and continuity Ass. 5.6.3 conspire to furnish a very simple estimate

$$|F(u) - F(u_N)| \leq C_f \|u - u_N\|_a .$$

 A priori estimates for $\|u - u_N\|_a$ \Rightarrow estimates for $|F(u) - F(u_N)|$

Hence, Thm. 5.3.42 immediately tells us the asymptotic convergence of linear and continuous output functionals defined for solutions of 2nd-order scalar elliptic BVPs and Lagrangian finite element discretization.

Example 5.6.4 (Approximation of mean temperature).

Heat conduction model (\rightarrow Sect. 2.5), scaled heat conductivity $\kappa \equiv 1$, on domain $\Omega =]0, 1[^2$, fixed temperature $u = 0$ on $\partial\Omega$:

$$-\Delta u = f \quad \text{in } \Omega \quad , \quad u = 0 \quad \text{on } \partial\Omega .$$

Heat source function $f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$, $(x, y)^T \in \Omega$

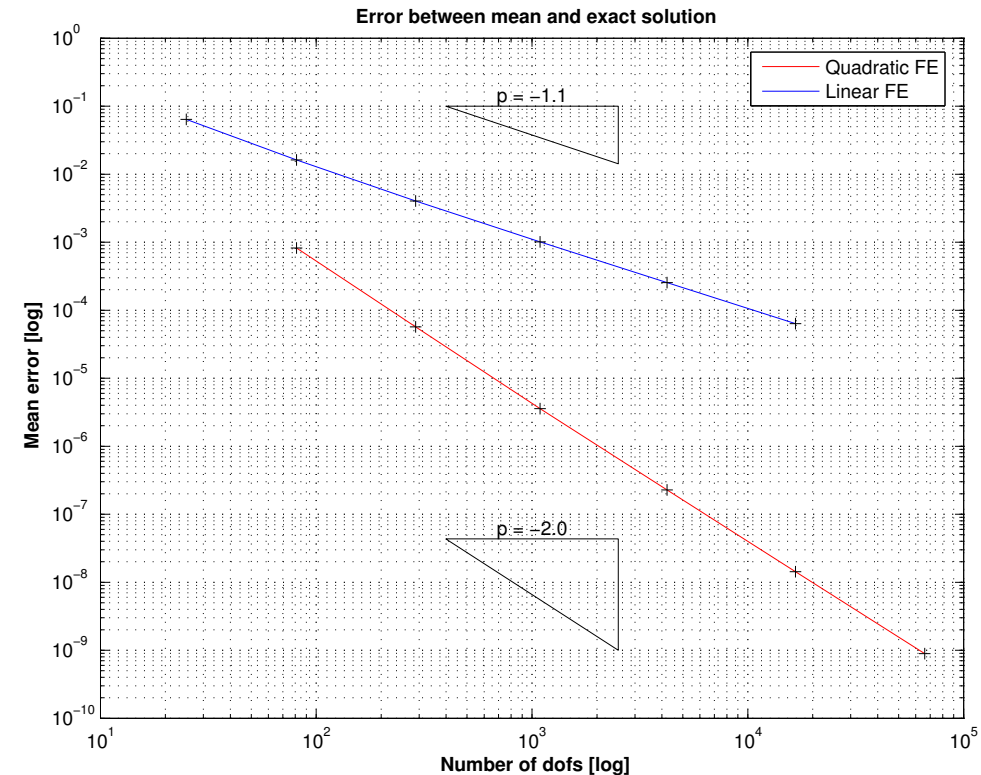
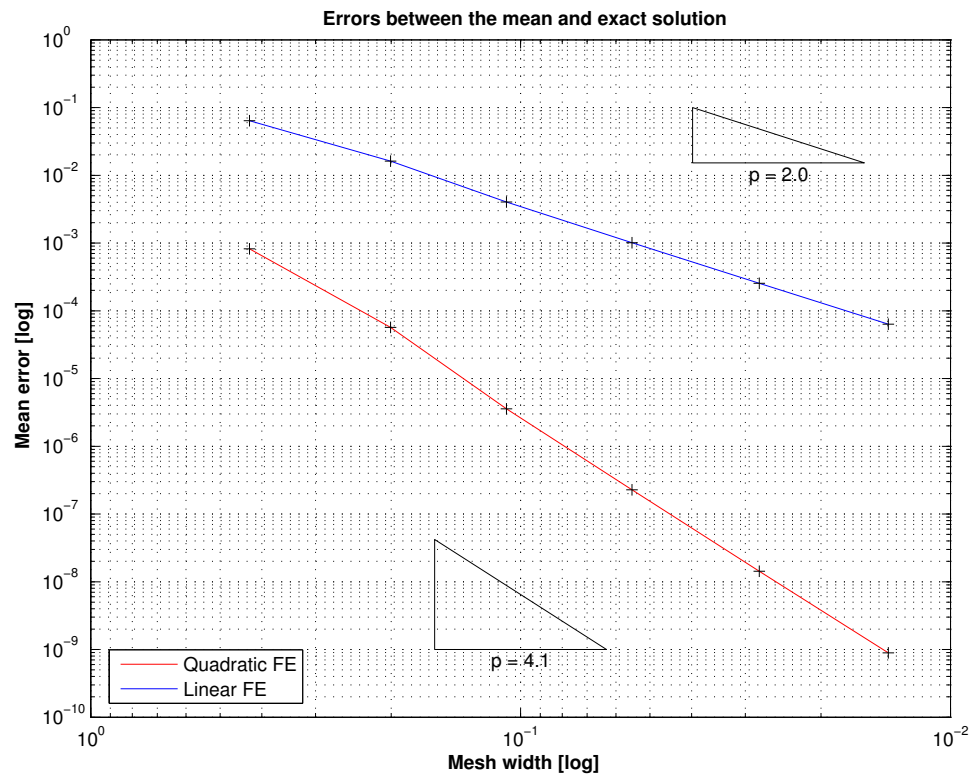
\triangleright solution $u(x, y) = \sin(\pi x) \sin(\pi y)$.

mean temperature $F(u) = \frac{1}{|\Omega|} \int_{\Omega} u \, d\mathbf{x} .$

Details of finite element Galerkin discretization:

- Sequence of triangular meshes \mathcal{M} created by regular refinement.
- Galerkin discretization: $V_{0,N} := \mathcal{S}_{1,0}^0(\mathcal{M})$ (linear Lagrangian finite elements \rightarrow Sect. 3.2).
- Quadrature rule (3.5.38) of order 6 for assembly of right hand side vector (more than sufficiently accurate \rightarrow guidelines from Sect. 5.5.1)

Expected: algebraic convergence in $h_{\mathcal{M}}$ with rate 1 of approximate mean temperature



Error in mean value on unit square ($- \leftrightarrow p = 1, - \leftrightarrow p = 2$)

Observation: Mean value converges twice as fast as expected: algebraic convergence $O(h_{\mathcal{M}}^2)$!



Theorem 5.6.5 (Duality estimate for linear functional output).

Define the **dual solution** $g_F \in V_0$ to F as solution of

$$g_F \in V_0: \quad a(v, g_F) = F(v) \quad \forall v \in V_0 .$$

Then

$$|F(u) - F(u_N)| \leq \|u - u_N\|_a \inf_{v_N \in V_{0,N}} \|g_F - v_N\|_a . \quad (5.6.6)$$

Proof. For **any** $v_N \in V_{0,N}$:

$$F(u) - F(u_N) = a(u - u_N, g_F) \stackrel{(*)}{=} a(u - u_N, g_F - v_N) \leq \|u - u_N\|_a \|g_F - v_N\|_a .$$

(*) ← by **Galerkin orthogonality** (5.1.7). □

▼

If g_F can be approximated well in $V_{0,N}$, then the **output error** can converge $\rightarrow 0$ (much) faster than $\|u - u_N\|_a$.

Example 5.6.7 (Approximation of mean temperature cnt'd). \rightarrow Ex. 5.6.4

- The mean temperature functional (5.6.6) is obviously linear \rightarrow Ass. 5.6.2
- By the Cauchy-Schwarz inequality (2.2.24) it clearly satisfies Ass. 5.6.3 even with $\|\cdot\|_a = \|\cdot\|_{L^2(\Omega)}$, let alone for $\|\cdot\|_a = \|\cdot\|_{H^1(\Omega)}$ on $H_0^1(\Omega)$.

What is $g_F \in H_0^1(\Omega)$ in this case? By Thm. 5.6.5 it is the solution of the variational problem

$$\int_{\Omega} \mathbf{grad} g_F \cdot \mathbf{grad} v \, d\mathbf{x} = F(v) = \frac{1}{|\Omega|} \int_{\Omega} v \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega).$$

The associated 2nd-order BVP reads

$$-\Delta g_F = \frac{1}{|\Omega|} \quad \text{in } \Omega, \quad g_F = 0 \quad \text{on } \partial\Omega.$$

Now recall the elliptic lifting theory Thm. 5.4.10 for convex domains: since $\Omega =]0, 1[^2$ is convex, we conclude $g_F \in H^2(\Omega)$.

► By interpolation estimate of Thm. 5.3.27 ($\mathbb{I}_1 \hat{=}$ linear interpolation onto $\mathcal{S}_1^0(\mathcal{M})$)

$$\inf_{v_N \in \mathcal{S}_1^0(\mathcal{M})} |g_F - v_N|_{H^1(\Omega)} \leq |g_F - \mathbb{I}_1 g_F|_{H^1(\Omega)} \leq Ch_{\mathcal{M}} |g_F|_{H^2(\Omega)},$$

where $C > 0$ may depend on Ω and the shape regularity measure (\rightarrow Def. 5.3.26) of \mathcal{M} .

Plug this into the **duality estimate** (5.6.6) of Thm. 5.6.5 and note that $u \in H^2(\Omega)$ by virtue of Thm. 5.4.10 and $f \in L^2(\Omega)$:

$$\blacktriangleright \quad |F(u) - F(u_N)| \leq Ch_{\mathcal{M}} \cdot \underbrace{|u - u_N|_{H^1(\Omega)}}_{\leq Ch_{\mathcal{M}} \text{ if } u \in H^2(\Omega)} \leq Ch_{\mathcal{M}}^2,$$

where the “generic constant” $C > 0$ depends only on $\Omega, u, \rho_{\mathcal{M}}$.

Again, by the elliptic lifting theory Thm. 5.4.10 we infer that $u \in H^2(\Omega)$ holds for this example since $f \in L^2(\Omega)$.



5.6.2 Case study: Boundary flux computation

Model problem (process engineering):

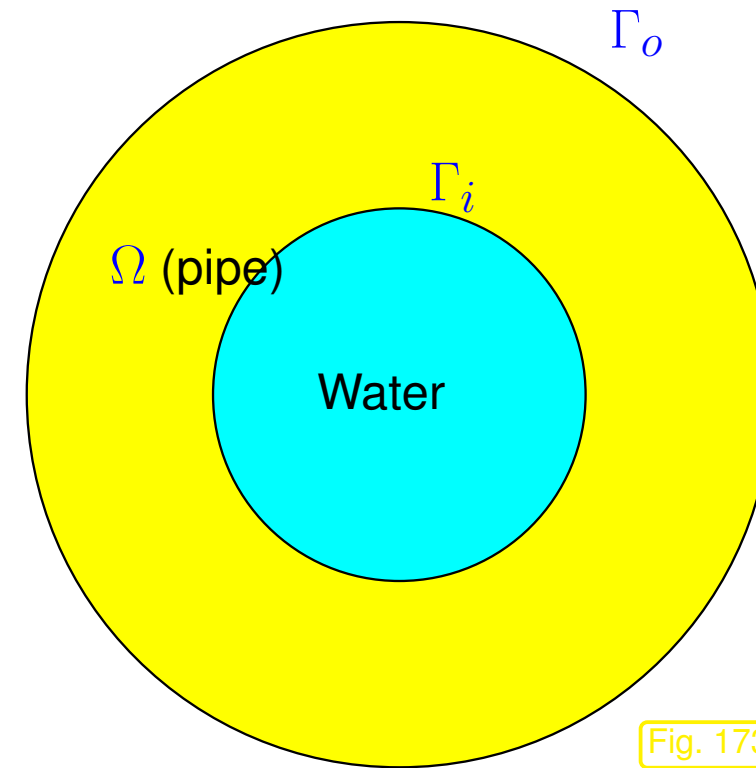
Long pipe carrying turbulent flow of coolant (water)

$\Omega \subset \mathbb{R}^2$: cross-section of pipe

κ : (scaled) heat conductivity of pipe material (assumed homogeneous, $\kappa = \text{const}$)

Assumption: Constant temperatures u_o, u_i at outer/inner wall Γ_o, Γ_i of pipe

Task: Compute heat flow pipe \rightarrow water



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Mathematical model: elliptic boundary value for stationary heat conduction (\rightarrow Sect. 2.5)

$$-\operatorname{div}(\kappa \mathbf{grad} u) = 0 \quad \text{in } \Omega, \quad u = u_x \quad \text{on } \Gamma_x, \quad x \in \{i, o\}. \quad (5.6.8)$$

$$\text{Heat flux through } \Gamma_i: \quad J(u) := \int_{\Gamma_i} \kappa \mathbf{grad} u \cdot \mathbf{n} \, dS. \quad (5.6.9)$$

Relate to abstract framework: $(5.6.8) \cong (3.1.1), \quad V_0 \cong H_0^1(\Omega) \quad (\rightarrow \text{Sect. 2.8})$

(Actually, $u \in H^1(\Omega)$, but by means of offset functions we can switch to the variational space $H_0^1(\Omega)$, see Sects. 2.1.3, 3.5.5.)

Numerical method: finite element computation of heat conduction in pipe
(e.g. linear Lagrangian finite element Galerkin discretization, Sect. 3.2)

Expectation: Algebraic convergence $|J(u) - J(u_N)| = O(h_{\mathcal{M}}^2)$ for regular h -refinement

This expectation is based on the analogy to Ex. 5.6.4 (Approximation of mean temperature), where duality estimates yielded $O(h_{\mathcal{M}}^2)$ convergence of the mean temperature error in the case of Galerkin discretization by means of linear Lagrangian finite elements on a sequence of meshes obtained by regular refinement. Now, it seems, we can follow the same reasoning.

Example 5.6.10 (Computation of heat flux).

- Setting: model problem “heat flux pipe *to* water”, see (5.6.8) and Fig. 173.
- Linear output functional from (5.6.9)

- Domain $\Omega = B_{R_o}(0) \setminus B_{R_i}(0) := \{\mathbf{x} \in \mathbb{R}^2: R_i < |\mathbf{x}| < R_o\}$ with $R_o = 1$ and $R_i = 1/2$
- Dirichlet boundary data $u_i = 60^\circ\text{C}$ on Γ_i , $u_o = 10^\circ\text{C}$ on Γ_o , heat source $f \equiv 0$, heat conductivity $\kappa \equiv 1$.
- Exact solution: $u(r, \varphi) = C_1 \ln(r) + C_2$, with $C_1 := (u_o - u_i)/(\ln R_i - \ln R_o)$,
- Exact heat flux: $J = 2\pi\kappa C_1$, $C_2 := (\ln R_o u_i - \ln R_i u_o)/(\ln R_i - \ln R_o)$.

Details of linear Lagrangian finite element Galerkin discretization:

- Sequences of unstructured triangular meshes \mathcal{M} obtained by regular refinement of coarse mesh (from grid generator).
- Galerkin FE discretization based on $V_{0,N} := \mathcal{S}_{1,0}^0(\mathcal{M})$.
- Approximate evaluation of $\mathbf{a}(u_N, v_N)$, $f(v_N)$ by six point quadrature rule (3.5.38) (“overkill quadrature”, see Sect. 5.5.1)
- Approximate evaluation of $J(u_N)$ by 4 point Gauss-Legendre quadrature rule on boundary edges of \mathcal{M} .
- Linear boundary approximation (circle replaced by polygon).
- Recorded: errors $|J - J(u_N)|$ on sequence of meshes.

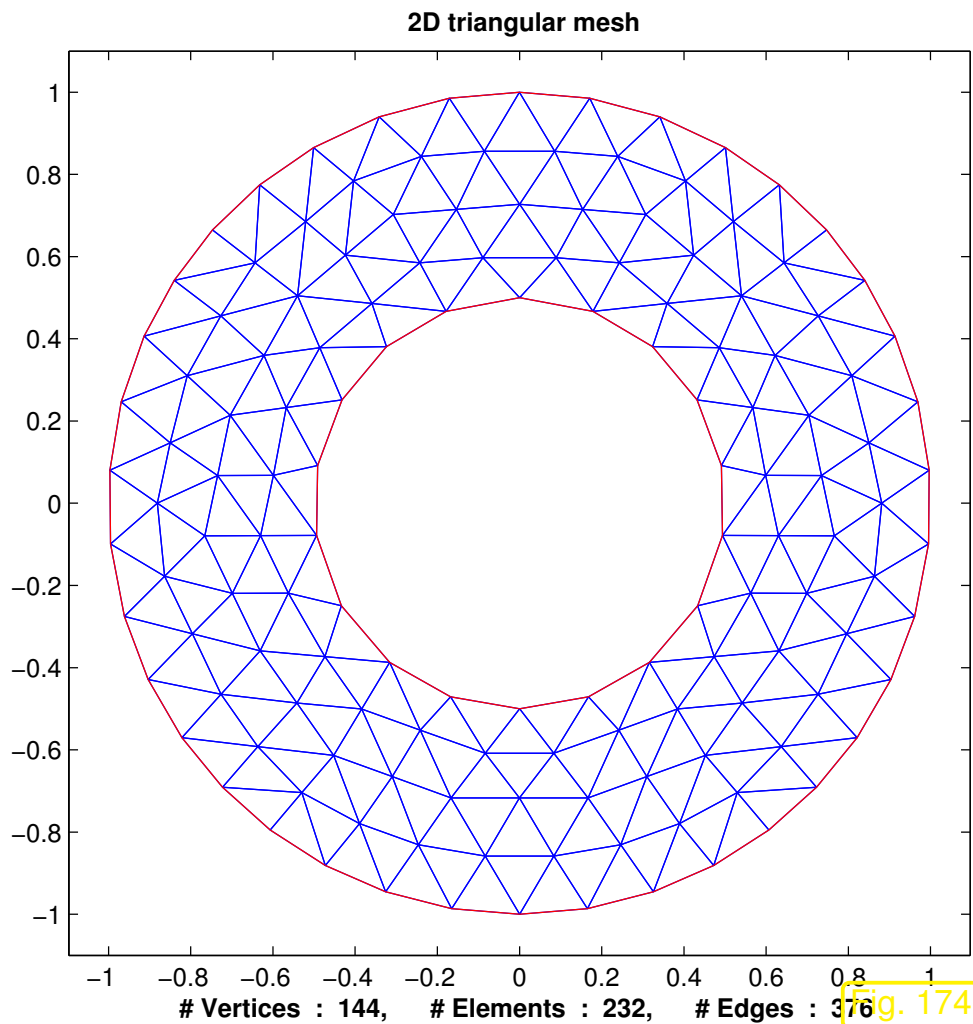


Fig. 174

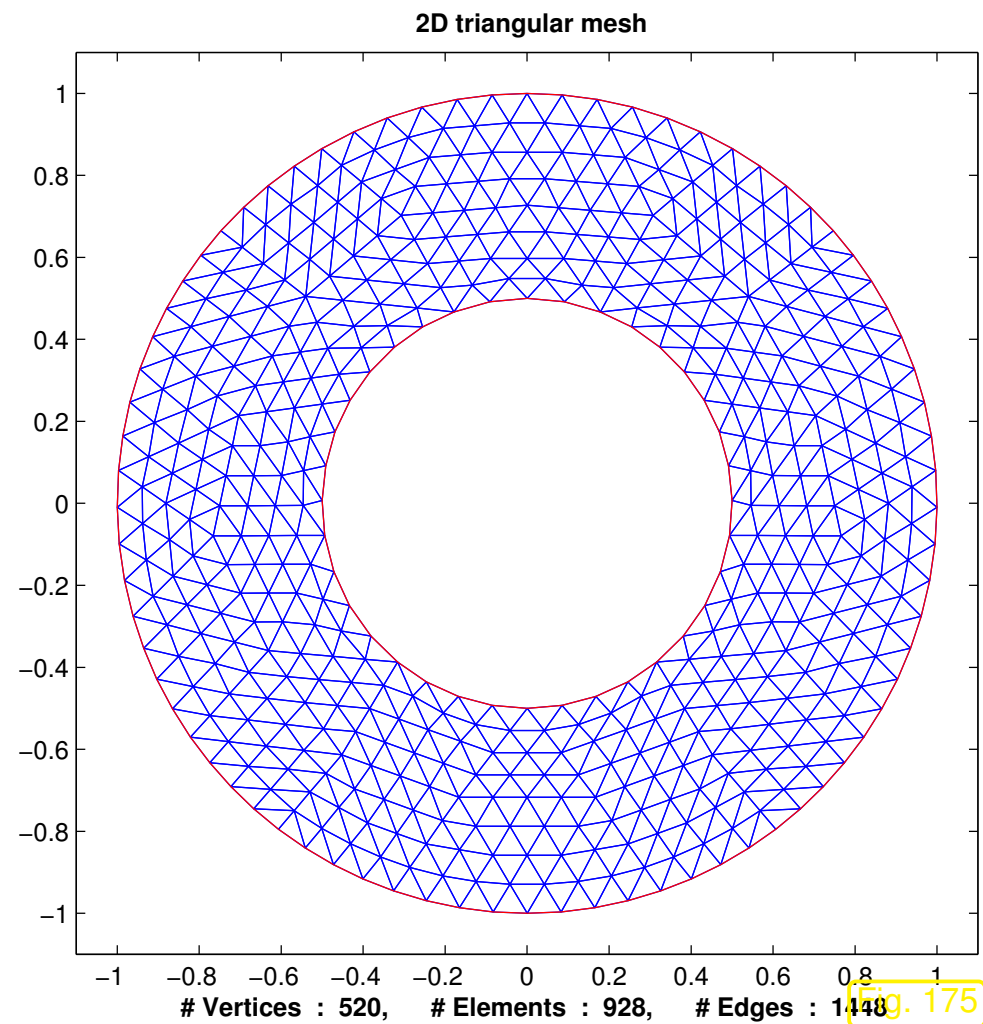
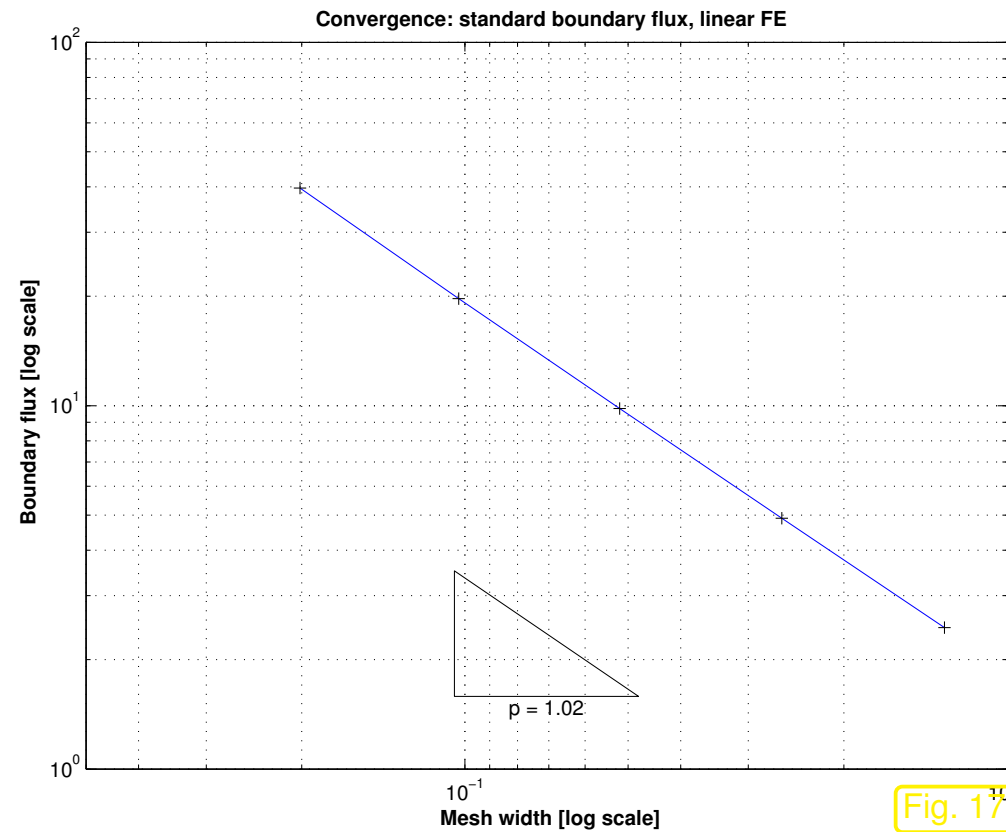


Fig. 175

Unstructured triangular meshes for $\Omega = B_1(0) \setminus B_{1/2}(0)$ (two coarsest specimens).



Observation:

Algebraic convergence of output error for J from (5.6.9) *only with rate 1* (in mesh width h_M)!

(This is not the fault of the piecewise linear boundary approximation, which is sufficient when using piecewise linear Lagrangian finite elements, see Sect. 5.5.2.)

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs



SAM, ETHZ

Why was our expectation mistaken ?

Suspicion: the output functional J fails to meet requirements of duality estimates of Thm. 5.6.5:

boundary flux functional J from (5.6.9) is **not** continuous on $H^1(\Omega)$!



Example 5.6.11 (Non-continuity of boundary flux functional).

Idea: find $u \in H^1(\Omega)$, for which “ $J(u) = \infty$ ”,
cf. investigation of non-continuity of point evaluation functional on $H^1(\Omega) \rightarrow$ Rem. 2.3.27.

On $\Omega = \{\mathbf{x} \in \mathbb{R}^2: \|\mathbf{x}\| < 1\}$ (unit disk) consider

$$u(\mathbf{x}) = (1 - \|\mathbf{x}\|)^\alpha =: g(\|\mathbf{x}\|), \quad \frac{1}{2} < \alpha < 1,$$

and the boundary flux functional (5.6.9) on $\partial\Omega$.

☞ On the one hand, using the expression (2.3.33) for the gradient in polar coordinates,

$$J_0(v) = \int_{\partial\Omega} \frac{\partial u}{\partial r}(\mathbf{x}) \, dS(\mathbf{x}) = 2\pi \alpha (1 - r)^{\alpha-1} \Big|_{r=1} \text{ “} = \infty \text{”}.$$

☞ On the other hand, straightforward computation of improper integral using (2.3.36):

$$\begin{aligned}
 |u|_{H^1(\Omega)}^2 &= \int_{\Omega} \|\mathbf{grad} u(\mathbf{x})\|^2 d\mathbf{x} = 2\pi \int_0^1 |g'(r)|^2 r dr = 2\pi\alpha^2 \int_0^1 (1-r)^{2\alpha-2} r dr \\
 &= 2\pi\alpha^2 \int_0^1 s^{2\alpha-2} (1-s) ds = 2\pi\alpha \left[\frac{s^{2\alpha-1}}{2\alpha-1} - \frac{s^{2\alpha}}{2\alpha} \right]_{s=0}^{s=1} = 2\pi \frac{1}{2\alpha-1} < \infty .
 \end{aligned}$$

Def. 2.2.18
 \implies

$u \in H^1(\Omega)$ ($u \in C^0(\bar{\Omega})$ and $u \in C^\infty(\Omega \setminus \{0\})$!).

Ex. 5.6.11 \triangleright Thm. 5.6.5 cannot be applied



(Potentially) poor convergence of flux obtained from straightforward evaluation of

$J(u_N)$ for FE solution $u_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$!




Apparently there is no remedy, because the boundary flux functional (5.6.9) seems to be enforced on

us by the problem: we are not allowed to tinker with it, are we?

Trick:

use fixed **cut-off function** $\psi \in C^0(\bar{\Omega}) \cap H^1(\Omega)$, $\psi \equiv 1$ on Γ_i , $\psi|_{\Gamma_o} = 0$

$$\int_{\Gamma_i} \kappa \mathbf{grad} u \cdot \mathbf{n} \, dS = \int_{\Gamma_i} (\kappa \mathbf{grad} u \cdot \mathbf{n}) \psi \, dS = \int_{\Omega} \underbrace{\operatorname{div}(\kappa \mathbf{grad} u)}_{=0} \psi + \kappa \mathbf{grad} u \cdot \mathbf{grad} \psi \, d\mathbf{x}$$

 use $J^*(u) := \int_{\Omega} \kappa \mathbf{grad} u \cdot \mathbf{grad} \psi \, d\mathbf{x} . \tag{5.6.12}$

Obviously (*): $J^* : H^1(\Omega) \mapsto \mathbb{R}$ continuous & $J^*(u) = J(u)$ for solution of (5.6.8)

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

(*): By the Cauchy-Schwarz inequality (2.2.24), since $\kappa = \text{const}$,

$$|J^*(u)| \leq \kappa \|\mathbf{grad} u\|_{L^2(\Omega)} \|\mathbf{grad} \psi\|_{L^2(\Omega)} \leq C |u|_{H^1(\Omega)},$$

with $C := \kappa \|\mathbf{grad} \psi\|_{L^2(\Omega)}$, which is a constant independent of u , as ψ is a fixed function.

Objection: You cannot just tamper with the output functional of a problem just because you do not like it!

Retort: Of course, one can replace the output function J with another one J^* as long as

$$J(u) = J^*(u) \quad \text{for the exact solution } u \text{ of the BVP,}$$

because the objective is not to “evaluate J ”, but to obtain an approximation for $J(u)$!

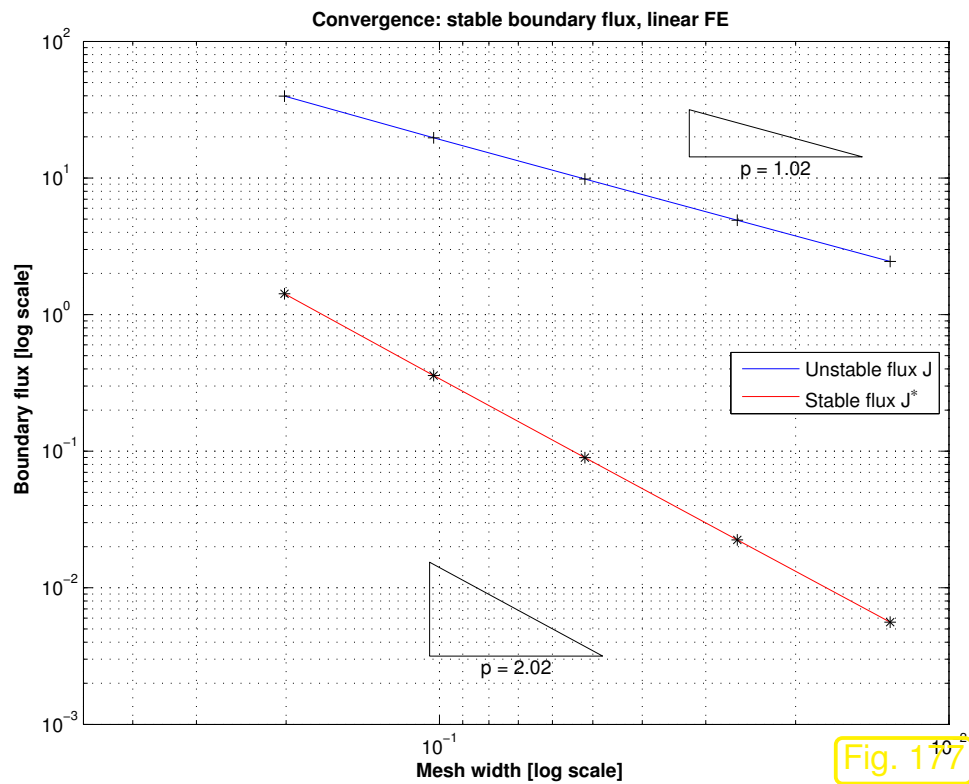
Example 5.6.13 (Computation of heat flux cnt'd). \rightarrow Ex. 5.6.13

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Further details on flux evaluation:

- Galerkin FE discretization based on $V_{0,N} := \mathcal{S}_{1,0}^0(\mathcal{M})$ or $V_{0,N} := \mathcal{S}_{2,0}^0(\mathcal{M})$.
- Approximate evaluation of $J^*(u_N)$ by six point quadrature rule (3.5.38) (“overkill quadrature”, see Sect. 5.5.1)
- Cut-off function with linear decay in radial direction
- Recorded: errors $|J - J(u_N)|$ and $|J - J^*(u_N)|$.



◁ Convergence of $|J(u) - J(u_N)|$ and $|J(u) - J^*(u_N)|$ for linear Lagrangian finite element discretization.

Additional observations:

- Algebraic convergence $|J(u) - J^*(u_N)| = O(h_{\mathcal{M}}^2)$ (rate 2 !) for alternative output functional J^* from (5.6.12).
- Dramatically reduced output error!

Remark 5.6.14 (Finding continuous replacement functionals).

Now you will ask: How can we find good (continuous) replacement functionals, if we are confronted with an unbounded output functional on the energy space?

Unfortunately, there is *no recipe*, and sometimes it does not seem to be possible to find a suitable J^* at all, for instance in the case of point evaluation, *cf.* Rem. 2.3.27.

Good news: another opportunity to show off how smart you are!

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs



SAM, ETHZ

5.6.3 L^2 -estimates

So far we have only studied the energy norm ($\leftrightarrow H^1(\Omega)$ -norm, see Rem. 5.3.28) of the finite element discretization error for 2nd-order elliptic BVP.

The reason was the handy tool of Cea's lemma Thm. 5.1.10.

What about error estimates in other “relevant norms”, e.g.,

- in the mean square norm or $L^2(\Omega)$ -norm, see Def. 2.2.8,
- in the supremum norm or $L^\infty(\Omega)$ -norm, see Def. 1.6.7?

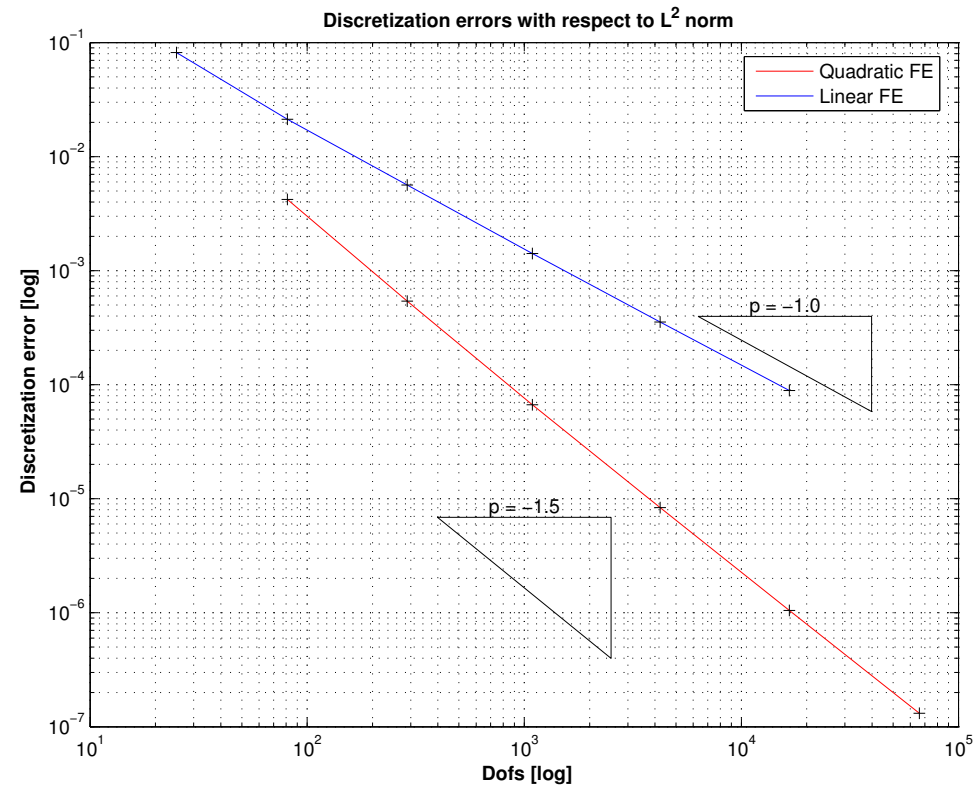
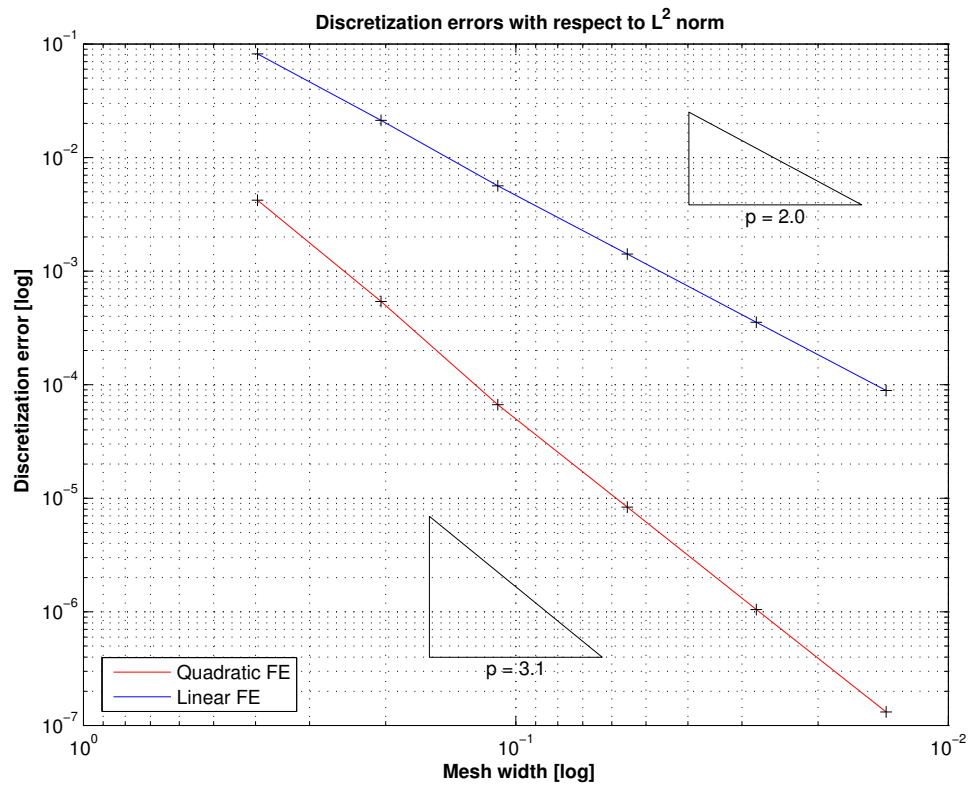
In this section we tackle $\|u - u_N\|_{L^2(\Omega)}$. We largely reuse the abstract framework of Sect. 5.6.1: linear variational problem (3.1.1) with exact solution $u \in V_0$, Galerkin finite element solution $u_N \in V_{0,N}$, see p. 568, and the special framework of linear 2nd-order elliptic BVPs, see Rem. 5.1.5: concretely,

$$a(u, v) := \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x}, \quad u, v \in H_0^1(\Omega).$$

Example 5.6.15 (L^2 -convergence of FE solutions). \rightarrow Ex. 5.2.4

Setting: $\Omega =]0, 1[^2$, $D \equiv 1$, $f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$, $(x, y)^\top \in \Omega$
➤ $u(x, y) = \sin(\pi x) \sin(\pi y)$.

- Sequence of triangular meshes \mathcal{M} , created by regular refinement.
- FE Galerkin discretization based on $\mathcal{S}_{1,0}^0(\mathcal{M})$ or $\mathcal{S}_2^0(\mathcal{M})$.
- Quadrature rule (3.5.38) for assembly of local load vectors (\rightarrow Sect. 3.5.4).
- Approximate $L^2(\Omega)$ -norm by means of quadrature rule (3.5.38).



$L^2(\Omega)$ -norm of discretization error on unit square ($- \leftrightarrow p = 1$, $- \leftrightarrow p = 2$)

- Observations:
- Linear Lagrangian FE ($p = 1$) $\Rightarrow \|u - u_N\|_0 = O(N^{-1})$
 - Quadratic Lagrangian FE ($p = 2$) $\Rightarrow \|u - u_N\|_0 = O(N^{-1.5})$

Remark 5.6.16 (L^2 interpolation error).

Recall the interpolation error estimate of Thm. 5.3.27

$$\|u - I_1 u\|_{L^2(\Omega)} = O(h_{\mathcal{M}}^2) \quad \text{vs.} \quad |u - I_1 u|_{H^1(\Omega)} = O(h_{\mathcal{M}}),$$

on a family of meshes with uniformly bounded shape regularity measure.

☞ Higher rate of algebraic convergence of the interpolation error when measured in the **weaker** $L^2(\Omega)$ -norm compared to the **stronger** $H^1(\Omega)$ -norm.

Therefore a similar observation in the case of the finite element approximation error is not so surprising.



Now we supply a rigorous underpinning and explanation of the behavior of $\|u - u_N\|_{L^2(\Omega)}$ that we have observed and expect.

Idea: Consider special **continuous linear output functional**

$$F(v) := \int_{\Omega} v \cdot (u - u_N) \, d\mathbf{x} \quad !$$

This functional is highly relevant for L^2 -estimates, because

$$F(u) - F(u_N) = \|u - u_N\|_{L^2(\Omega)}^2 \quad !$$

➤ estimates for the output error will provide bounds for $\|u - u_N\|_{L^2(\Omega)}$!

Note: Both u and u_N are *fixed* functions $\in H^1(\Omega)$!

➤ Linearity of F (\rightarrow Ass. 5.6.2) is obvious.

➤ Continuity $F : H_0^1(\Omega) \mapsto \mathbb{R}$ (\rightarrow Ass. 5.6.3) is clear, use Cauchy-Schwarz inequality (2.2.24).

Duality estimate of Thm. 5.6.5 can be applied:

Thm. 5.6.5



$$F(u) - F(u_N) = \|u - u_N\|_{L^2(\Omega)}^2 \leq C |u - u_N|_{H^1(\Omega)} \inf_{v_N \in V_{0,N}} |g_F - v_N|_{H^1(\Omega)}, \quad (5.6.17)$$

where $C > 0$ may depend only on κ , and the **dual solution** $g_F \in H_0^1(\Omega)$ satisfies

$$\begin{aligned} a(g_F, v) = F(v) \quad \forall v \in V_0 &\Leftrightarrow \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} g_F \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} v(u - u_N) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) \\ &\Downarrow \\ -\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} g_F) &= u - u_N \quad \text{in } \Omega, \quad g_F = 0 \quad \text{on } \partial\Omega. \end{aligned} \quad (5.6.18)$$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Assumption 5.6.19 (2-regularity of homogeneous Dirichlet problem).

We assume that the homogeneous Dirichlet problem with coefficient κ is **2-regular** on Ω : There is $C > 0$, which depends on Ω only such that

$$\begin{aligned} u \in H_0^1(\Omega) \\ \operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) \in L^2(\Omega) \end{aligned} \Rightarrow u \in H^2(\Omega) \quad \text{and} \quad |u|_{H^2(\Omega)} \leq C \|\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u)\|_{L^2(\Omega)}.$$

By the elliptic lifting theorem for convex domains Thm. 5.4.10 we know

$$\kappa C^1\text{-smooth} \quad \& \quad \Omega \text{ convex} \quad \implies \quad \text{Ass. 5.6.19 is satisfied.}$$

Ass. 5.6.19 in conjunction with (5.6.18) yields

$$|g_F|_{H^2(\Omega)} \leq C \|u - u_N\|_{L^2(\Omega)}, \quad (5.6.22)$$

where $C > 0$ depends only on Ω .

Now we can appeal to the general best approximation theorem for Lagrangian finite element spaces Thm. 5.3.42:

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} |g_F - v_N|_{H^1(\Omega)} \leq C \frac{h_{\mathcal{M}}}{p} |g_F|_{H^2(\Omega)} \stackrel{(5.6.22)}{\leq} C \frac{h_{\mathcal{M}}}{p} \|u - u_N\|_{L^2(\Omega)}, \quad (5.6.23)$$

where the “generic constants” $C > 0$ depend only on Ω and the shape regularity measure $\rho_{\mathcal{M}}$ (\rightarrow Def. 5.3.26).

Combine (5.6.17) and (5.6.23) and cancel one power of $\|u - u_N\|_{L^2(\Omega)}$:

With $C > 0$ depending only on Ω , κ , and the shape regularity measure $\rho_{\mathcal{M}}$ we conclude

$$\text{Ass. 5.6.19} \Rightarrow \|u - u_N\|_{L^2(\Omega)} \leq C \frac{h_{\mathcal{M}}}{p} \|u - u_N\|_{H^1(\Omega)}.$$

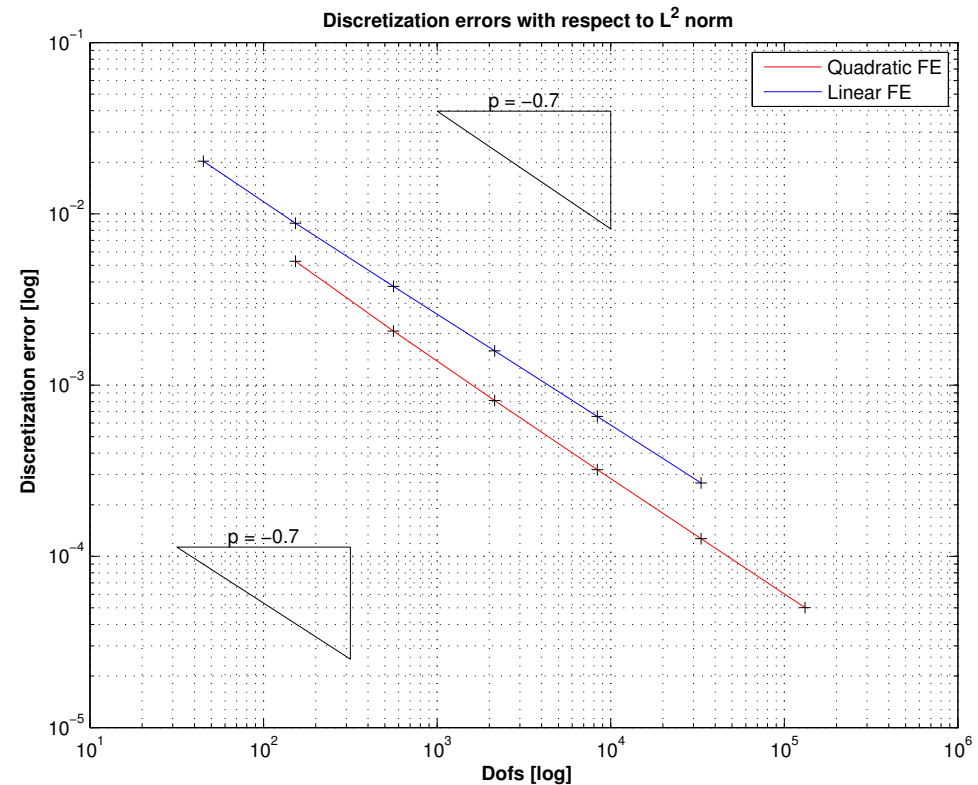
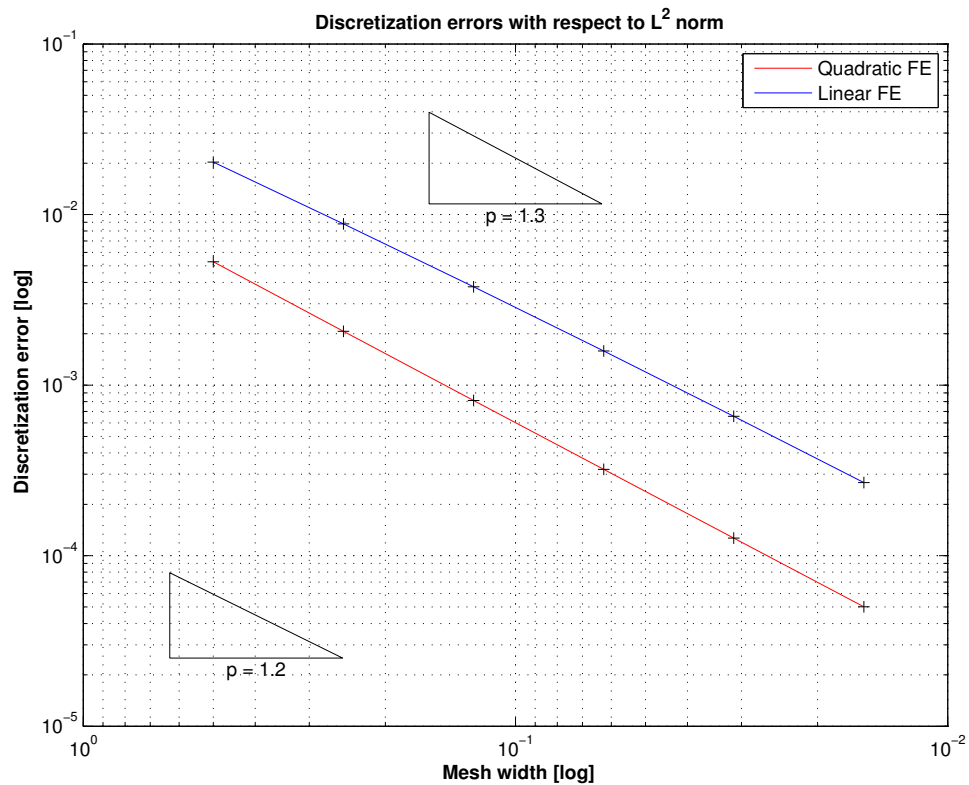
for h -refinement: gain of one factor $O(h_{\mathcal{M}})$ (vs. $H^1(\Omega)$ -estimates)

Is it important to assume 2-regularity, Ass. 5.6.19 or merely a technical requirement of the theoretical approach?

Example 5.6.24 (L^2 -estimates on non-convex domain). cf. Ex. 5.2.6

Setting: $\Omega =]-1, 1[^2 \setminus (]0, 1[\times]-1, 0[)$, $D \equiv 1$, $u(r, \varphi) = r^{2/3} \sin(2/3\varphi)$ (polar coordinates)
 ➤ $f = 0$, Dirichlet data $g = u|_{\partial\Omega}$.

Finite element Galerkin discretization and evaluations as in Ex. 5.6.15.



$L^2(\Omega)$ -norm of discretization error on “L-shaped” domain (— $\leftrightarrow p = 1$, — $\leftrightarrow p = 2$)

Observation: For both ($p = 1, 2$) \Rightarrow algebraic convergence $\|u - u_N\|_0 = O(N^{-2/3})$

Comparison with Ex. 5.2.6: for both linear and quadratic Lagrangian FEM

$$\|u - u_N\|_{L^2(\Omega)} = O(N^{-2/3}) \iff \|u - u_N\|_{H^1(\Omega)} = O(N^{-1/3}),$$

that is, we again observe a doubling of the rate of convergence for the weaker norm.

No gain through the use of quadratic FEM, because of limited smoothness of both u and dual solution g_F . For both the solution and the dual solution the gradient will have a singularity at 0 .



5.7 Discrete maximum principle

So far we have investigated the **accuracy** of finite element Galerkin solutions: we studied relevant norms $\|u - u_N\|$ of the discretization error.

Now new perspective:

structure preservation by FEM

To what extent does the finite element solution u_N inherit key structural properties of the solution u of a 2nd-order scalar elliptic BVP?

This issue will be discussed for a special structural property of the solution of the linear 2nd-order elliptic BVP (inhomogeneous Dirichlet problem) in variational form (\rightarrow Sect. 2.8)

$$u \in \tilde{g} + H_0^1(\Omega): \quad \mathbf{a}(u, v) := \int_{\Omega} \kappa \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega). \quad (5.7.1)$$

where $\tilde{g} \hat{=}$ offset function, extension of Dirichlet data $g \in C^0(\partial\Omega)$, see Sect. 2.3.1, (2.3.7),
 $\kappa \hat{=}$ bounded and uniformly positive definite diffusion coefficient, see (2.5.4).

(5.7.1) \longleftrightarrow BVP (PDE-form)

$$-\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) = f \quad \text{in } \Omega \quad , \quad u = g \quad \text{on } \partial\Omega .$$

Recall (\rightarrow Sect. 2.5): (5.7.1) models *stationary* temperature distribution in body, when temperature on its surface is prescribed by g .

- Intuition:
- In the absence of heat sources maximal and minimal temperature attained on surface.
 - In the presence of a heat source ($f \geq 0$) the temperature minimum will be attained on surface $\partial\Omega$.
 - If $f \leq 0$ (heat sink), then the maximal temperature will be attained on the surface.

In fact this is a theorem, cf. Sect. 2.7.

Theorem 5.7.2 (Maximum principle for 2nd-order elliptic BVP).

For $u \in C^0(\bar{\Omega}) \cap H^1(\Omega)$ holds the *maximum principle*

$$\begin{aligned} -\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) \geq 0 &\implies \min_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) = \min_{\mathbf{x} \in \Omega} u(\mathbf{x}) , \\ -\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) \leq 0 &\implies \max_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) = \max_{\mathbf{x} \in \Omega} u(\mathbf{x}) . \end{aligned}$$

$$\Delta u = 0$$



Maximum/minimum on $\partial\Omega$

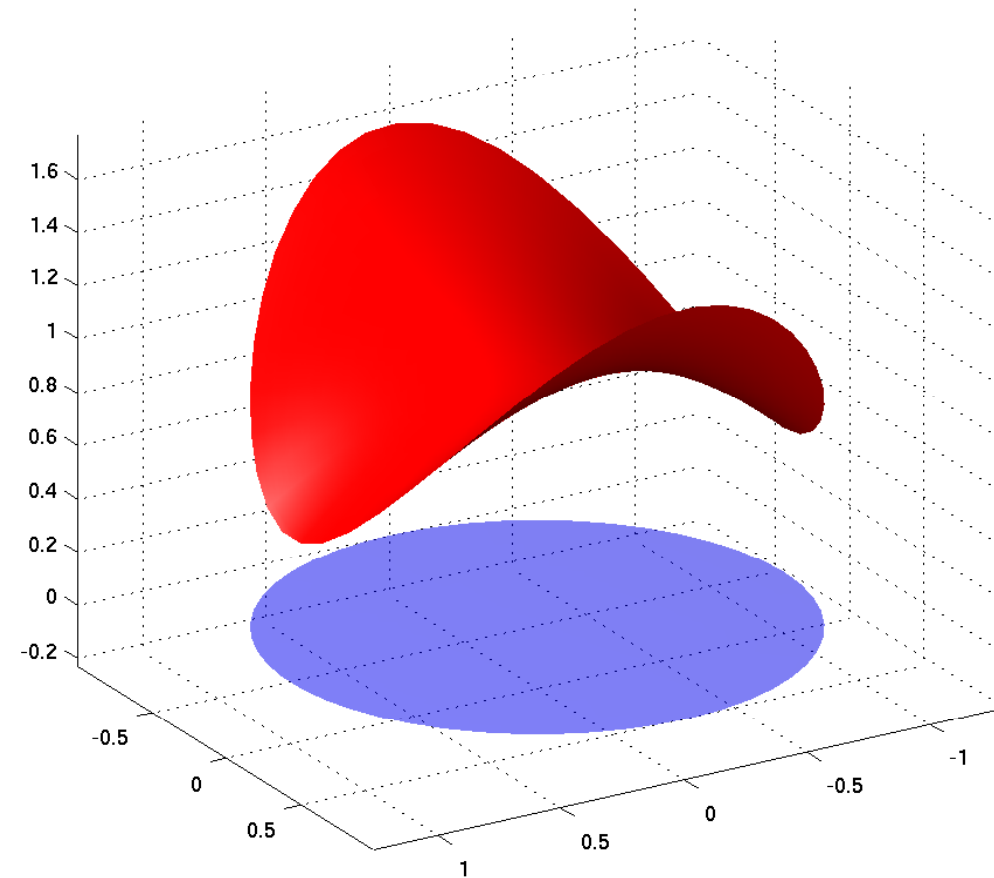


Fig. 178

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Proof. (for the case $-\operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) = 0$)

Sect. 2.1.3 \triangleright u solves quadratic minimization problem

$$u = \operatorname{argmin}_{\substack{v \in H^1(\Omega) \\ v = g \text{ on } \partial\Omega}} \int_{\Omega} \kappa(\mathbf{x}) \|\operatorname{grad} v(\mathbf{x})\|^2 \, d\mathbf{x} .$$

If u had a global maximum at \mathbf{x}^* in the interior of Ω , that is

$$\exists \delta > 0: \quad u(\mathbf{x}^*) \geq \max_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) + \delta .$$

Now “chop off” the maximum and define

$$w(\mathbf{x}) := \min\{u(\mathbf{x}), u(\mathbf{x}^*) - \delta\} , \quad \mathbf{x} \in \Omega .$$

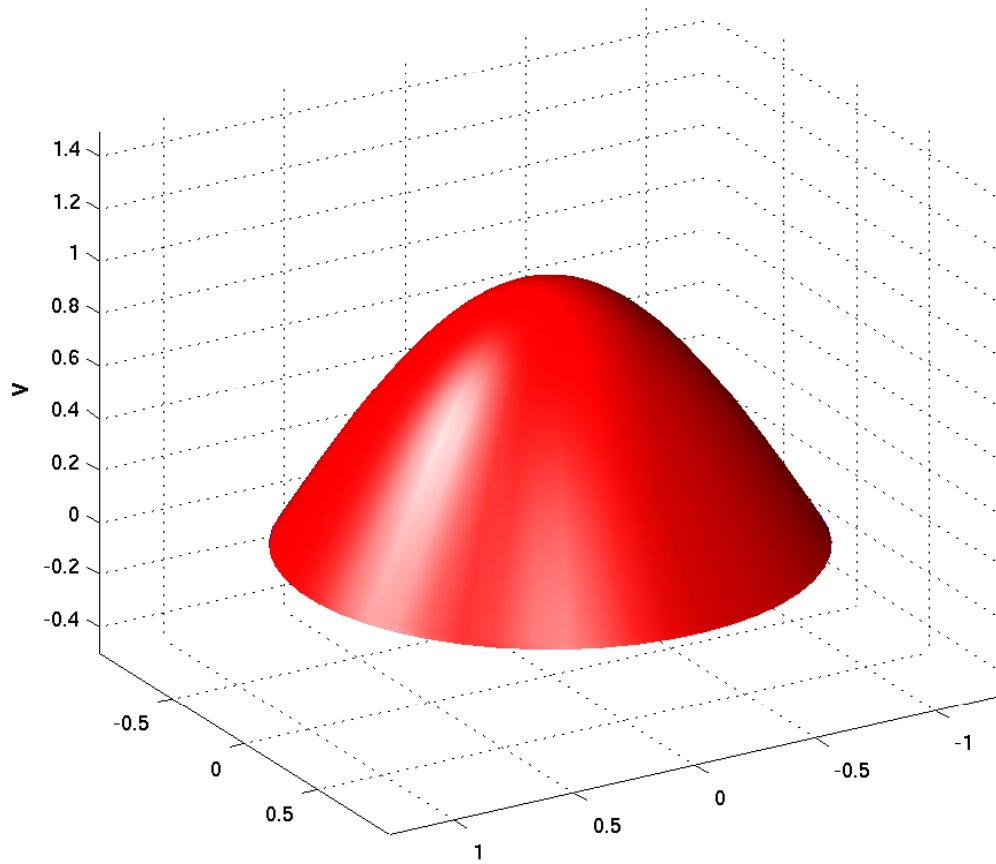


Fig. 179

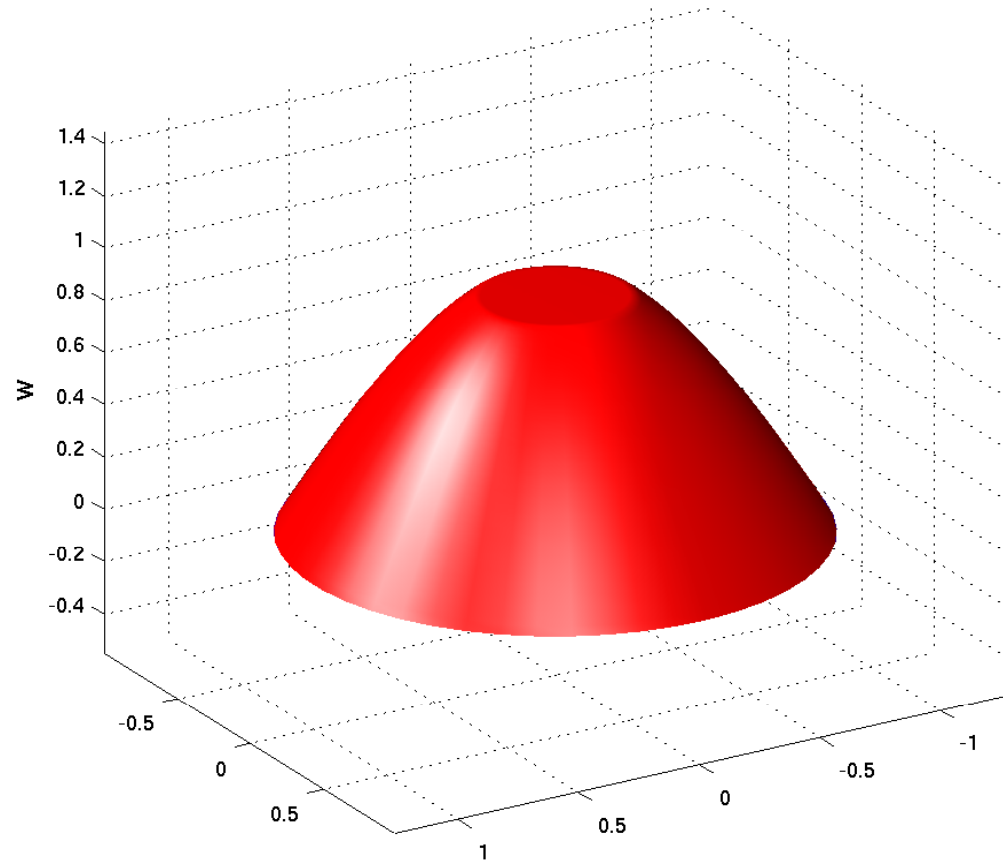


Fig. 180

$$\int_{\Omega} \kappa(\mathbf{x}) \|\mathbf{grad} w(\mathbf{x})\|^2 \, d\mathbf{x} \geq \int_{\Omega} \kappa(\mathbf{x}) \|\mathbf{grad} v(\mathbf{x})\|^2 \, d\mathbf{x} .$$

also belong to $H^1(\Omega)$. However

$$\int_{\Omega} \kappa(\mathbf{x}) \|\mathbf{grad} w(\mathbf{x})\|^2 \, d\mathbf{x} < \int_{\Omega} \kappa(\mathbf{x}) \|\mathbf{grad} u(\mathbf{x})\|^2 \, d\mathbf{x} ,$$

which contradicts the definition of u as the global minimizer of the quadratic energy functional. \square

Now we consider a finite element Galerkin discretization of (5.7.1) by means of linear Lagrangian finite elements (\rightarrow Sect. 3.4), using offset functions supported near $\partial\Omega$ as explained in Sect. 3.5.5.

\triangleright finite element Galerkin solution $u_N \in \mathcal{S}_1^0(\mathcal{M}) \subset C^0(\bar{\Omega})$

Issue: does u_N satisfy a **maximum principle**, that is, can we conclude

$$\begin{aligned} f \geq 0 &\implies \min_{\mathbf{x} \in \partial\Omega} u_N(\mathbf{x}) = \min_{\mathbf{x} \in \Omega} u_N(\mathbf{x}) , \\ f \leq 0 &\implies \max_{\mathbf{x} \in \partial\Omega} u_N(\mathbf{x}) = \max_{\mathbf{x} \in \Omega} u_N(\mathbf{x}) ? \end{aligned} \tag{5.7.3}$$

Recall from Sect. 4.1: finite difference discretization of

$$-\Delta u = 0 \quad \text{in } \Omega :=]0, 1[^2, \quad u = g \quad \text{on } \partial\Omega,$$

on an $M \times M$ tensor product mesh

$$\mathcal{M} := \{[(i-1)h, ih] \times [(j-1)h, jh], 1 \leq i, j \leq M\}, \quad M \in \mathbb{N}.$$

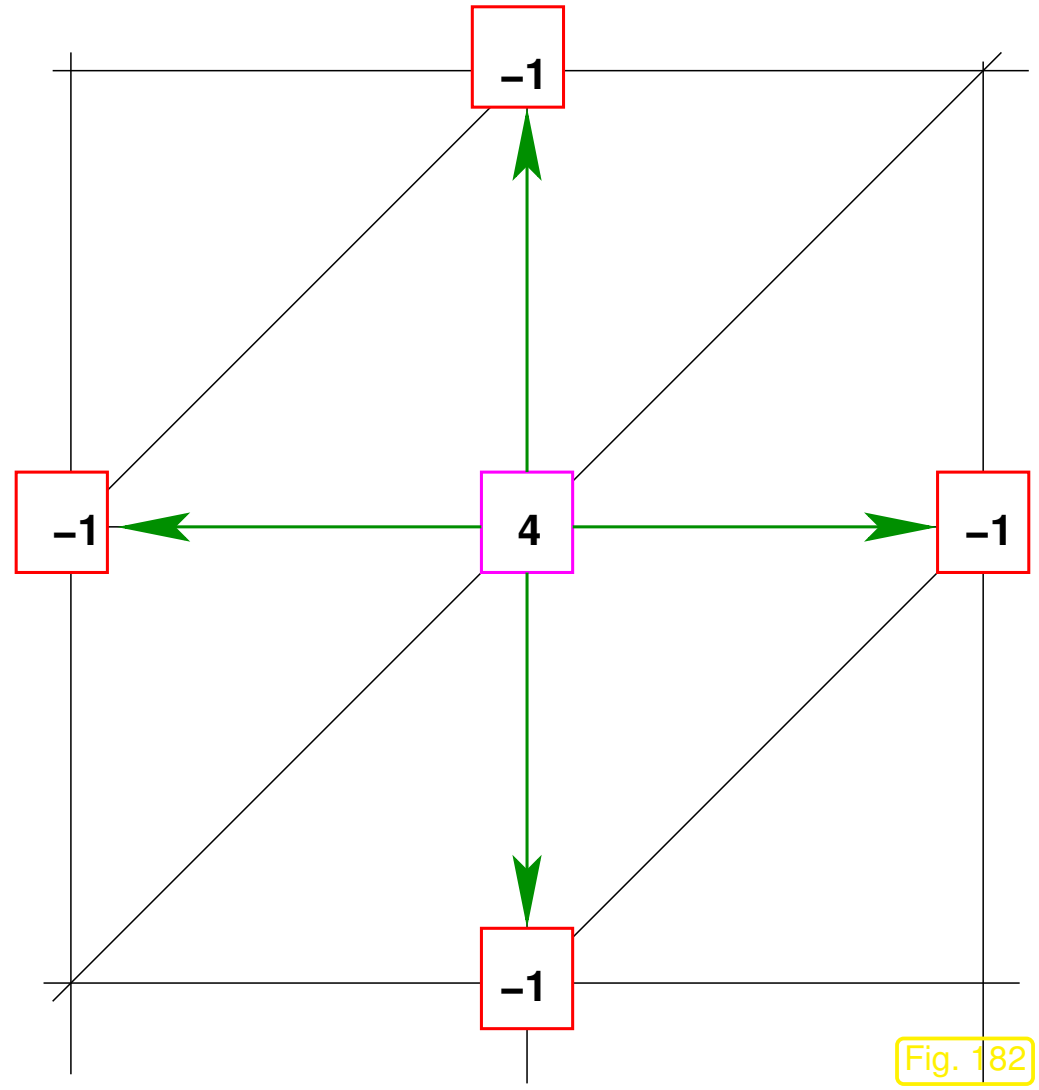
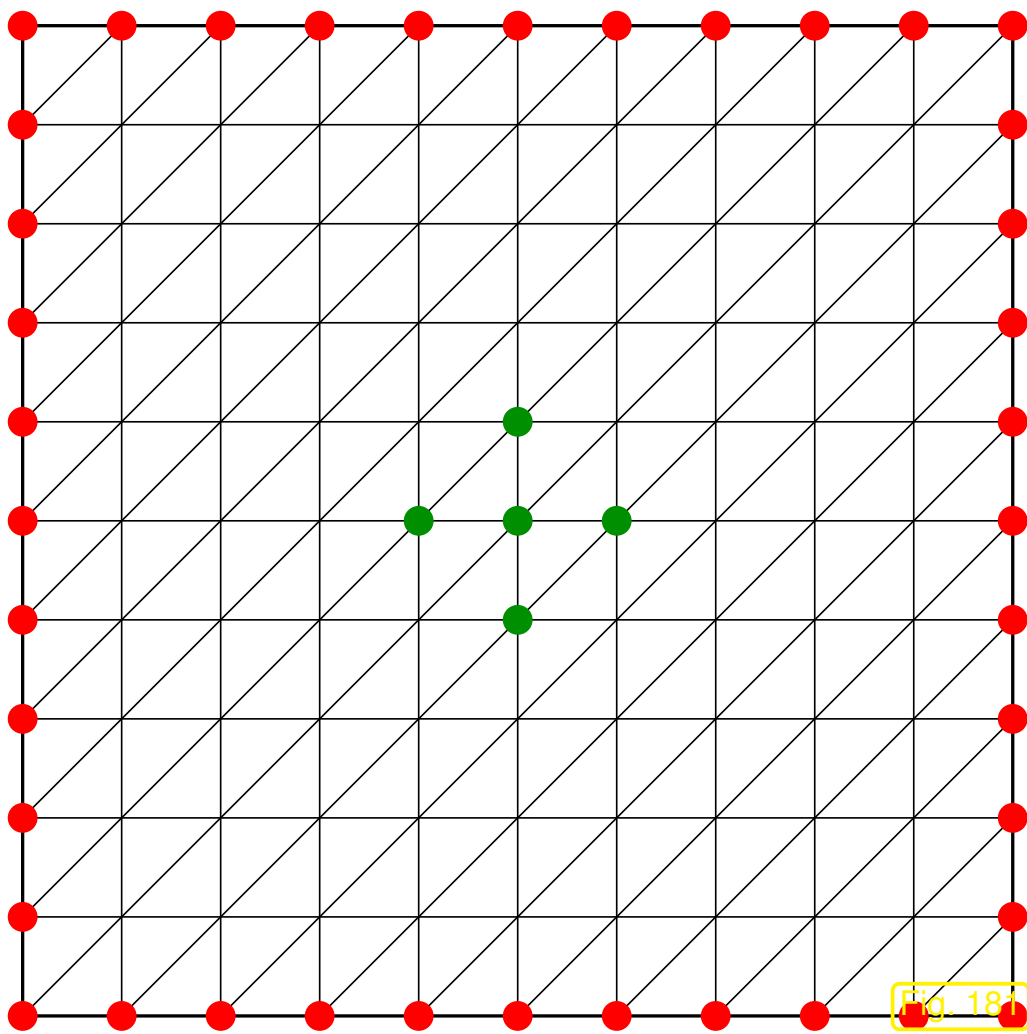
Unknowns in the finite difference method: $\mu_{ij} \approx u((ih, jh)^T), 1 \leq i, j \leq M-1.$

Unknowns are solutions of a linear system of equations, see (4.1.2)

$$\frac{1}{h^2} (4\mu_{i,j} - \mu_{i-1,j} - \mu_{i+1,j} - \mu_{i,j-1} - \mu_{i,j+1}) = 0, \quad 1 \leq i, j \leq M-1, \quad (5.7.5)$$

where values corresponding to points on the boundary are gleaned from g :

$$\mu_{0,j} := g(0, hj), \quad \mu_{M,j} := g(1, hj), \quad \mu_{i,0} := g(hi, 0), \quad \mu_{i,M} := g(hi, 1), \quad 1 \leq i, j < M.$$



The finite difference solution $(\mu_{i,j})_{1 \leq i,j < M}$ will attain its maximal value somewhere:

$$\exists n, m \in \{1, \dots, M - 1\}: \mu_{n,m} = \mu_{\max} := \max_{0 \leq i,j \leq M} \mu_{i,j} .$$

Assume: $(nh, mh)^T$ in the interior $\Leftrightarrow 1 \leq n, m < M$

Be aware of the following two facts:

$$\mu_{n-1,m}, \mu_{n+1,m}, \mu_{n,m-1}, \mu_{n,m+1} \leq \mu_{n,m}, \quad (5.7.6)$$

$$\mu_{n,m} = \frac{1}{4}(\mu_{n-1,m} + \mu_{n+1,m} + \mu_{n,m-1} + \mu_{n,m+1}) \quad (\text{average!}).$$

$$\Downarrow \leftarrow \text{“averaging argument”} \quad (5.7.7)$$

$$\mu_{n-1,m} = \mu_{n+1,m} = \mu_{n,m-1} = \mu_{n,m+1} = \mu_{n,m}! \quad (5.7.8)$$

The same argument can now target the neighboring grid points $((n-1)h, mh)^T$, $((n+1)h, mh)^T$, $(nh, (m-1)h)^T$, $(nh, (m+1)h)^T$. By induction we find:

$$\blacktriangleright \quad \mu_{i,j} = \mu_{\max} \quad \forall 0 \leq i, j \leq M,$$

that is, the finite difference solution has to be *constant*!

\blacktriangleright The finite difference solution can attain its maximum in the interior only in the case of constant boundary data g !



Maximum principle satisfied for $f = 0$!

Now we try to generalize the considerations of the previous example to the discretization by means of *linear Lagrangian finite elements* on a triangular mesh (of a polygonal domain $\Omega \subset \mathbb{R}^2$) see Sect. 3.2.

$$\tilde{\mathbf{A}} \in \mathbb{R}^{M,M} \hat{=} \mathcal{S}_1^0(\mathcal{M})\text{-Galerkin matrix for } \mathbf{a} \text{ from (5.7.1)} \quad (M := \#\mathcal{V}(\mathcal{M}))$$

Row of this matrix connects all values $\mu_j = u_N(\mathbf{x}^j)$ of Galerkin solution $u_N \in \mathcal{S}_1^0(\mathcal{M})$ according to

$$(\tilde{\mathbf{A}})_{ii}\mu_i + \sum_{j \neq i} (\tilde{\mathbf{A}})_{ij}\mu_j = (\vec{\varphi})_i, \quad \mathbf{x}^i \text{ interior node},$$

where $\mu_j := g(\mathbf{x}^j)$ for $\mathbf{x}^j \in \partial\Omega$.

The above averaging argument from Ex. 5.7.4 carries over, if the entries of $\tilde{\mathbf{A}}$ satisfy the following conditions:

$$\bullet \quad (\tilde{\mathbf{A}})_{ii} > 0 \quad (\text{positive diagonal}) , \quad (5.7.9)$$

$$\bullet \quad (\tilde{\mathbf{A}})_{ij} \leq 0 \quad \text{for } j \neq i \quad (\text{non-positive off-diagonal entries}) , (5.7.10)$$

$$\bullet \quad \sum_j (\tilde{\mathbf{A}})_{ij} = 0 , \text{ if } \mathbf{x}^i \text{ is interior node .} \quad (5.7.11)$$

(Recall [21, Def. 2.7.7]: matrix $\tilde{\mathbf{A}}$ satisfying (5.7.9)–(5.7.11) is **diagonally dominant**.)

averaging argument \blacktriangleright $u_N(\mathbf{x}^i) = \max_{\mathbf{y} \in \mathcal{V}(\mathcal{M})} u_N(\mathbf{y})$ can only hold for an interior node \mathbf{x}^i ,
if $\mu_N = \text{const.}$

\blacktriangleright Since $u_N \in \mathcal{S}_1^0(\mathcal{M})$ attains its extremal values at nodes of the mesh, the maximum principles holds for it in the case $f = 0$ provided that (5.7.9)–(5.7.11) are satisfied.

More general case $f \leq 0$:

$$\blacktriangleright \quad (\vec{\varphi})_i = \int_{\Omega} f(\mathbf{x}) b_N^i(\mathbf{x}) \, d\mathbf{x} \leq 0 , \quad \text{since } b_N^i \geq 0 .$$

Then the averaging argument again rules out the existence of an interior maximum for a non-constant solution. The case $f \geq 0$ follows similarly.

When will (5.7.9)–(5.7.11) hold for $\mathcal{S}_1^0(\mathcal{M})$ -Galerkin matrix?

First consider

$$\kappa \equiv 1, \quad \leftrightarrow \quad -\Delta u = f$$

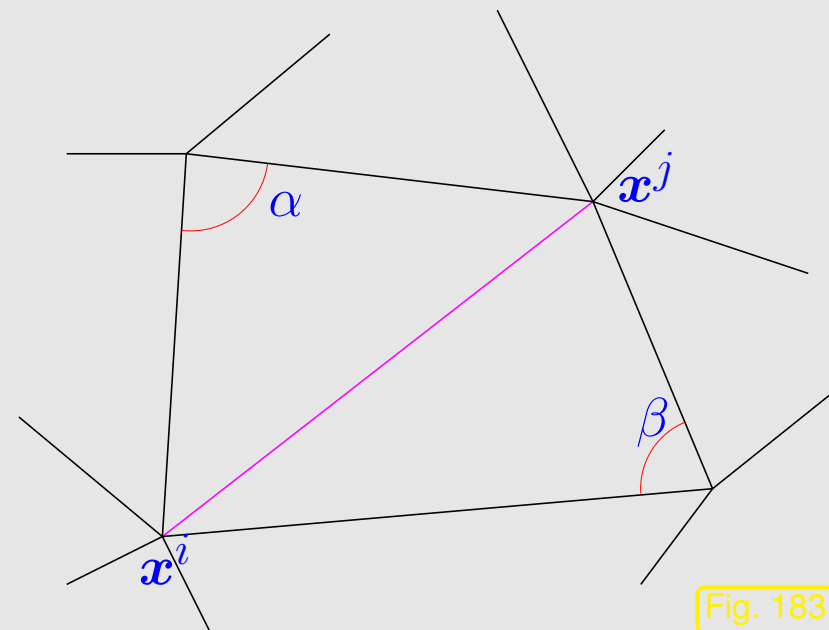
(The linear finite element discretization of this BVP was scrutinized in Sect. 3.2)

From formula (3.2.11) for element matrix & assembly, see Fig. 74:

$$(\tilde{\mathbf{A}})_{ij} = -\cot \alpha - \cot \beta = -\frac{\sin(\alpha + \beta)}{\sin \alpha \sin \beta}.$$

↓

$$(\tilde{\mathbf{A}})_{ij} \leq 0 \quad \Leftrightarrow \quad \alpha + \beta < \pi.$$



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Moreover

$$\sum_{\mathbf{x} \in \mathcal{V}(\mathcal{M})} b_N^{\mathbf{x}} \equiv 1 \quad \Rightarrow \quad \sum_j (\tilde{\mathbf{A}})_{ij} = 0 \quad (\Leftrightarrow (5.7.11)).$$

The condition (5.7.9) $\Leftrightarrow (\tilde{\mathbf{A}})_{ii} > 0$ is straightforward.

Theorem 5.7.12 (Maximum principle for linear FE solution of Poisson equation).

The linear finite element solution of

$$-\Delta u = 0 \quad \text{in } \Omega \subset \mathbb{R}^2, \quad u = g \quad \text{on } \partial\Omega,$$

*on a triangular mesh \mathcal{M} satisfies the **maximum principle** (5.7.3), if \mathcal{M} is a Delaunay triangulation.*

Remark 5.7.13 (Maximum principle for linear FE for 2nd-order elliptic BVPs).

For $\mathcal{S}_1^0(\mathcal{M})$ -Galerkin discretization of (5.7.1) on triangular mesh, the conditions (5.7.9)–(5.7.11) are fulfilled,

if all angles of triangles of $\mathcal{M} \leq \frac{\pi}{2}$.



Remark 5.7.14 (Maximum principle for higher order Lagrangian FEM).

Even when using p -degree Lagrangian finite elements with nodal basis functions associated with interpolation nodes, see Sect. 3.4.1, the discrete maximum principle will fail to hold on *any mesh* for $p > 1$.



Learning Outcomes

Essential knowledge and skills acquired in this chapter:

- State, prove and understand Cea's Lemma and its relevance for the finite element Galerkin discretization of elliptic BVP.
- Known the meaning of h -refinement and p -refinement.

- Ability to predict the asymptotic algebraic convergence of the energy norm and L^2 -norm the finite element discretization error for scalar 2nd-order elliptic BVP.
- Familiarity with features of an elliptic BVP (corners, discontinuous coefficients) that can thwart the fastest possible convergence of a Lagrangian finite element discretization for h -refinement.
- Knowledge of how to choose the appropriate order of quadrature and boundary approximation so as to preserve the optimal rate of convergence (for h -refinement).
- Use duality techniques to obtain improved error estimates for the evaluation of linear and continuous output functionals

6

2nd-Order Linear Evolution Problems

Now we study scalar linear partial differential equations for which *one* coordinate direction is special and identified with **time** and denoted by the independent variable t . The other coordinates are regarded as **spatial coordinates** and designated by $\boldsymbol{x} = (x_1, \dots, x_d)^T$.

➤ solution will be a “function of time and space”: $u = u(\boldsymbol{x}, t)$

The domain for such PDEs will have **tensor product structure** (tensor product of spatial domain and a bounded time interval):

Computational domain:

$$\tilde{\Omega} := \Omega \times]0, T[\subset \mathbb{R}^{d+1}.$$

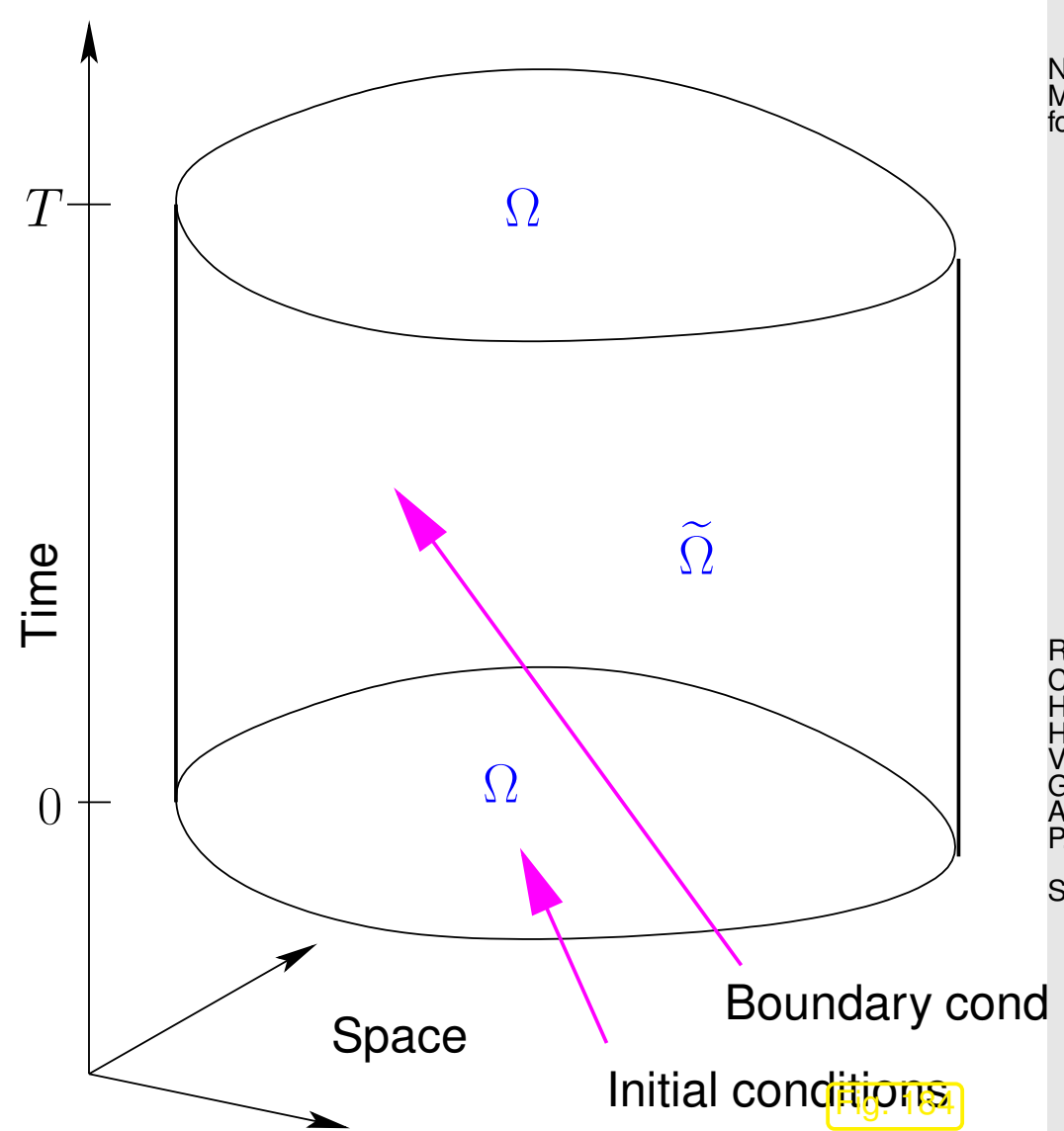
▶ **space-time cylinder**

$\Omega \subset \mathbb{R}^d \hat{=} \text{spatial domain}$ (satisfying assumptions of Sect. 2.1.1)

$T > 0 \hat{=} \text{final time}$

On $\Omega \times \{0\} \rightarrow \text{initial conditions,}$

on $\partial\Omega \times]0, T[\rightarrow \text{(spatial) boundary conditions.}$



PDE for $u(\mathbf{x}, t)$

+

initial conditions

+

boundary conditions

= evolution problem

Note: No boundary conditions on $\Omega \times \{T\}$ (“final conditions”) are prescribed: time is supposed to have a “direction” that governs the flow of information in the evolution problem.

▶ evolution problems (on bounded spatial domains) are also known as
initial-boundary value problems (IBVP).

Remark 6.0.1 (Initial time).

Why do we always pick initial time $t = 0$?

The modelled physical systems will usually be time-invariant, so that we are free to shift time. Remember the analogous situation with **autonomous** ODE, see [21, Sect. 12.1].



6.1 Parabolic initial-boundary value problems

6.1.1 Heat equation

Sect. 2.5 treated *stationary* heat conduction: no change of temperature with time (temporal equilibrium)

Now we consider the evolution of a temperature distribution $u = u(\mathbf{x}, t)$.

- $\Omega \subset \mathbb{R}^d$: space occupied by solid body (bounded spatial computational domain),
- $\kappa = \kappa(\mathbf{x})$: (spatially varying) heat conductivity ($[\kappa] = \frac{\text{W}}{\text{Km}}$),
- $T > 0$: final time for “observation period” $[0, T]$,
- $u_0 : \Omega \mapsto \mathbb{R}$: **initial** temperature distribution in Ω ,
- $g : \partial\Omega \times [0, T] \mapsto \mathbb{R}$: **surface temperature**, varying in space and time: $g = g(\mathbf{x}, t)$,
- $f : \Omega \times [0, T] \mapsto \mathbb{R}$: time-dependent heat source/sink ($[f] = \frac{\text{W}}{\text{m}^3}$): $f = f(\mathbf{x}, t)$.

Goal: derive PDE governing *transient* heat conduction.

Conservation of energy:

$$\frac{d}{dt} \int_V \rho u \, d\mathbf{x} + \int_{\partial V} \mathbf{j} \cdot \mathbf{n} \, dS = \int_V f \, d\mathbf{x} \quad \text{for all "control volumes" } V \quad (6.1.1)$$

energy stored in V

power flux through ∂V

heat generation in V

$\rho = \rho(\mathbf{x})$: (spatially varying) **heat capacity** ($[\rho] = \text{JK}^{-1}$), uniformly positive, cf. (2.5.4).

As in Sect. 2.5, now apply Gauss' Theorem Thm. 2.4.9 to the power flux integral in (6.1.1). This converts the surface integral to a volume integral over $\text{div } \mathbf{j}$ and we get

$$\frac{d}{dt} \int_V \rho u \, d\mathbf{x} + \int_V \text{div } \mathbf{j} \, d\mathbf{x} = \int_V f \, d\mathbf{x} \quad \text{for all "control volumes" } V$$

Now appeal to another version of the fundamental lemma of the calculus of variations, see Lemma 2.4.15, this time involving piecewise constant test functions.

► Local form of energy balance law (**Heat equation**)

$$\frac{\partial}{\partial t}(\rho u)(\mathbf{x}, t) + (\text{div}_{\mathbf{x}} \mathbf{j})(\mathbf{x}, t) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega}. \quad (6.1.2)$$

The heat flux is linked to temperature variations by Fourier's law:

$$\mathbf{j}(\mathbf{x}) = -\kappa(\mathbf{x}) \mathbf{grad} u(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (2.5.3)$$

From here we let all differential operators like **grad** and **div** act on the spatial independent variable \mathbf{x} . As earlier, the independent variables \mathbf{x} and t will be omitted frequently. Watch out!

Now, plug (2.5.3) into (6.1.2).



$$\frac{\partial}{\partial t}(\rho u) - \operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) = f \quad \text{in} \quad \tilde{\Omega} := \Omega \times]0, T[. \quad (6.1.3)$$

+ **Dirichlet boundary conditions** (fixed surface temperature) on $\partial\Omega \times]0, T[$:

$$u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \text{for} \quad (\mathbf{x}, t) \in \partial\Omega \times]0, T[. \quad (6.1.4)$$

+ initial conditions for $t = 0$:

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{for all} \quad \mathbf{x} \in \Omega. \quad (6.1.5)$$

Terminology: (6.1.2) & (6.1.4) & (6.1.5) is a specimen of a

2nd-order **parabolic** initial-boundary value problem

Remark 6.1.6 (Compatible boundary and initial data).

Natural regularity requirements for Dirichlet data g :

g continuous in time and space

Natural compatibility requirement at initial time and $u_0 \in C^0(\bar{\Omega})$

$$g(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \forall \mathbf{x} \in \partial\Omega .$$



Remark 6.1.7 (Boundary conditions for 2nd-order parabolic IBVPs).

Physical intuition for transient heat conduction:

On $\partial\Omega]0, T[$ we can impose any of the boundary conditions discussed in Sect. 2.6:

- Dirichlet boundary conditions $u(\mathbf{x}, t) = g(\mathbf{x}, t)$, see (6.1.4) (fixed surface temperature),
- Neumann boundary conditions $\mathbf{j}(\mathbf{x}, t) \cdot \mathbf{n} = -h(\mathbf{x}, t)$ (fixed heat flux through surface),
- radiation boundary conditions $\mathbf{j}(\mathbf{x}, t) \cdot \mathbf{n} = \Psi(u(\mathbf{x}, t))$,

and any combination of these as discussed in Ex. 2.6.7, yet, *only one* of them at any part of $\partial\Omega \times]0, T[$, see Rem. 2.6.6.

6.1.2 Spatial variational formulation

Now we study the linear 2nd-order parabolic initial-boundary value problem with pure Dirichlet boundary conditions, introduced in the preceding section:

$$\frac{d}{dt}(\rho u) - \operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[, \quad (6.1.3)$$

$$u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \text{for } (\mathbf{x}, t) \in \partial\Omega \times]0, T[, \quad (6.1.4)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \Omega . \quad (6.1.5)$$

Assume: Homogeneous Dirichlet boundary conditions $g = 0$

The general case can be reduced to this by using the offset function trick, see Sect. 3.5.5, and solve the parabolic initial-boundary value problem for $w(\mathbf{x}, t) := u(\mathbf{x}, t) - \tilde{g}(\mathbf{x}, t)$, where $\tilde{g}(\cdot, t)$ is an extension of the Dirichlet data g to $\tilde{\Omega}$. Then w will satisfy homogeneous Dirichlet boundary conditions and solve an evolution equation with a modified source function $\tilde{f}(\mathbf{x}, t)$.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Now we pursue the formal derivation of the *spatial* variational formulation of (6.1.3)–(6.1.4).

The steps completely mirror those discussed in Sect. 2.8

STEP 1: *test PDE with functions* $v \in H_0^1(\Omega)$

(do not test, where the solution is known, that is, on the boundary $\partial\Omega$)

Note: test function does *not depend on time*: $v = v(\boldsymbol{x})!$

STEP 2: *integrate* over domain Ω

STEP 3: *perform integration by parts* in space

(by using Green's first formula, Thm. 2.4.11)

STEP 4: [optional] *incorporate boundary conditions* into boundary terms

For the concrete PDE (6.1.3) and boundary conditions (6.1.4) refer to Ex. 2.8.1, for more general boundary conditions to Ex. 2.8.8.


Spatial variational form of (6.1.3)–(6.1.4): seek $t \in]0, T[\mapsto u(t) \in H_0^1(\Omega)$

$$\int_{\Omega} \rho(\mathbf{x}) \dot{u}(t) v \, d\mathbf{x} + \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u(t) \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x}, t) v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega), \quad (6.1.8)$$

$$u(0) = u_0 \in H_0^1(\Omega). \quad (6.1.9)$$

Be aware: $u(t) \hat{=}$ function space $(H_0^1(\Omega))$ -valued function on $]0, T[$.

Also note that **grad** acts on the spatial independent variables that are suppressed in the notation $u(t)$.

 Notation: $\dot{u}(t) = \frac{\partial u}{\partial t}(t) \hat{=}$ (partial) derivative w.r.t. time.

Shorthand notation (with obvious correspondences):

$$t \in]0, T[\mapsto u(t) \in V_0 \quad : \quad \begin{cases} \mathbf{m}(\dot{u}(t), v) + \mathbf{a}(u(t), v) = \ell(t)(v) & \forall v \in V_0, \\ u(0) = u_0 \in V_0. \end{cases} \quad (6.1.10)$$

Again, here $\ell(t) \hat{=}$ linear form valued function on $]0, T[$.

Concretely:

$$\mathbf{m}(u, v) := \int_{\Omega} \rho(\mathbf{x}) \dot{u}(t) v \, d\mathbf{x} , \quad u, v \in H_0^1(\Omega) ,$$

$$\mathbf{a}(u, v) := \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u(t) \cdot \mathbf{grad} v \, d\mathbf{x} , \quad u, v \in H_0^1(\Omega) ,$$

$$\ell(t)(v) := \int_{\Omega} f(\mathbf{x}, t) v(\mathbf{x}) \, d\mathbf{x} , \quad v \in H_0^1(\Omega) .$$

Note that both \mathbf{m} and \mathbf{a} are *symmetric, positive definite* bilinear forms (\rightarrow Def. 2.1.32).

Equivalent formulation, since the bilinear form \mathbf{m} does not depend on time:

$$t \in]0, T[\mapsto u(t) \in V_0 \quad : \quad \begin{cases} \frac{d}{dt} \mathbf{m}(u(t), v) + \mathbf{a}(u(t), v) = \ell(t)(v) & \forall v \in V_0, \\ u(0) = u_0 \in V_0. \end{cases} \quad (6.1.11)$$

Now we are concerned with the **stability** of parabolic evolution problems: We investigate whether $\|u\|_{H^1(\Omega)}$ stays bounded for all times in the case $f \equiv 0$.

For the sake of simplicity:

consider $\rho \equiv 1$ and $\kappa \equiv 1$

(General case is not more difficult, because both ρ and κ are bounded and uniformly positive, see (2.5.4).)

By the first Poincaré-Friedrichs inequality Thm. 2.2.25

$$\exists \gamma > 0: \quad |v|_{H^1(\Omega)}^2 \geq \gamma \|v\|_{L^2(\Omega)}^2 \quad \forall v \in H_0^1(\Omega). \quad (6.1.12)$$

In fact, Thm. 2.2.25 reveals $\gamma = \text{diam}(\Omega)^{-2}$.

Remark 6.1.13 (Differentiating bilinear forms with time-dependent arguments).

Consider (temporally) smooth $u : [0, T] \mapsto V_0$, $v : [0, T] \mapsto V_0$ and a *symmetric* bilinear form $\mathbf{b} : V_0 \times V_0 \mapsto \mathbb{R}$.

What is $\frac{d}{dt}\mathbf{b}(u(t), v(t))$?

Formal Taylor expansion:

$$\begin{aligned}\mathbf{b}(u(t + \tau), v(t + \tau)) &= \mathbf{b}(u(t) + \dot{u}(t)\tau + O(\tau^2), v(t) + \dot{v}(t)\tau + O(\tau^2)) \\ &= \mathbf{b}(u(t), v(t)) + \tau(\mathbf{b}(\dot{u}(t), v(t)) + \mathbf{b}(u(t), \dot{v}(t))) + O(\tau^2) .\end{aligned}$$

$$\blacktriangleright \lim_{\tau \rightarrow 0} \frac{\mathbf{b}(u(t + \tau), v(t + \tau)) - \mathbf{b}(u(t), v(t))}{\tau} = \mathbf{b}(\dot{u}(t), v(t)) + \mathbf{b}(u(t), \dot{v}(t)) .$$

This is a general **product rule**, see [21, Eq. 4.4.5].



Lemma 6.1.18 (Decay of solutions of parabolic evolutions).

For $\rho \equiv 1$, $\kappa \equiv 1$, and $f \equiv 0$ the solution $u(t)$ of (6.1.8) satisfies

$$\|u(t)\|_{L^2(\Omega)} \leq e^{-\gamma t} \|u_0\|_{L^2(\Omega)} \quad , \quad |u(t)|_{H^1(\Omega)} \leq e^{-\gamma t} |u_0|_{H^1(\Omega)} \quad \forall t \in]0, T[.$$

Proof. Multiply the solution of the parabolic IBVP with an exponential weight function:

$$w(t) := \exp(\gamma t)u(t) \in H_0^1(\Omega) \quad \Rightarrow \quad \dot{w} := \frac{dw}{dt}(t) = \gamma w(t) + \exp(\gamma t)\frac{du}{dt}(t) \quad , \quad (6.1.19)$$

solves the parabolic IBVP

$$\begin{aligned} \mathbf{m}(\dot{w}, v) + \tilde{\mathbf{a}}(w, v) &= 0 \quad \forall v \in V \quad , \\ w(0) &= u_0 \quad , \end{aligned} \quad (6.1.20)$$

with $\tilde{\mathbf{a}}(w, v) = \mathbf{a}(w, v) - \gamma \mathbf{m}(w, v)$, γ from (6.1.12). To see this, use that $u(t)$ solves (6.1.11) with $f \equiv 0$ (elementary calculation).

Note: $(6.1.12) \Rightarrow \tilde{\mathbf{a}}(v, v) \geq 0 \quad \forall v \in V$

Exponential decay of $\|\cdot\|_{L^2(\Omega)}$ -norm of solution:

$$\frac{d}{dt} \frac{1}{2} \|w\|_{L^2(\Omega)}^2 = \frac{d}{dt} \frac{1}{2} \mathbf{m}(w, w) \stackrel{\text{Rem. 6.1.13}}{=} \mathbf{m}(\dot{w}, w) = -\tilde{\mathbf{a}}(w, w) \leq 0 \quad (6.1.21)$$

This confirms that $t \mapsto \|w\|_{L^2(\Omega)}(t)$ is a decreasing function, which involves

$$(6.1.21) \quad \Rightarrow \quad \|w(t)\|_{L^2(\Omega)} \leq \|w(0)\|_{L^2(\Omega)} ,$$

and the first assertion of the Lemma is evident. Next, we verify the exponential decay of $|\cdot|_{H^1(\Omega)}$ -norm of solution by a similar trick:

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} \|w\|_{\tilde{a}}^2 &\stackrel{\text{Rem. 6.1.13}}{=} \tilde{a}\left(\frac{d}{dt}w, w\right) = -\mathbf{m}\left(\frac{d}{dt}w, \frac{d}{dt}w\right) \leq 0 \quad \Rightarrow \quad \|w(t)\|_{\tilde{a}} \leq \|w(0)\|_{\tilde{a}} , \\ \blacktriangleright \quad |w(t)|_{H^1(\Omega)}^2 &\leq |w(0)|_{H^1(\Omega)}^2 - \underbrace{\gamma(\|w(0)\|_{L^2(\Omega)}^2 - \|w(t)\|_{L^2(\Omega)}^2)}_{\geq 0 \text{ by (6.1.21)}} . \end{aligned}$$

\blacktriangleright Exponential decrease of energy during parabolic evolution without excitation
("Parabolic evolutions dissipate energy")

6.1.3 Method of lines

Idea: Apply **Galerkin discretization** (\rightarrow Sect. 3.1) to abstract linear parabolic variational problem (6.1.11).

$$t \in]0, T[\mapsto u(t) \in V_0 \quad : \quad \begin{cases} \mathbf{m}(\dot{u}(t), v) + \mathbf{a}(u(t), v) = \ell(t)(v) & \forall v \in V_0, \\ u(0) = u_0 \in V_0. \end{cases} \quad (6.1.11)$$

1st step: replace V_0 with a finite dimensional subspace $V_{0,N}$, $N := \dim V_{0,N} < \infty$

► Discrete parabolic evolution problem

$$t \in]0, T[\mapsto u(t) \in V_{0,N} \quad : \quad \begin{cases} \mathbf{m}(\dot{u}_N(t), v_N) + \mathbf{a}(u_N(t), v_N) = \ell(t)(v_N) & \forall v_N \in V_{0,N}, \\ u_N(0) = \text{projection/interpolant of } u_0 \text{ in } V_{0,N}. \end{cases} \quad (6.1.22)$$

2nd step: introduce (ordered) basis $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ of $V_{0,N}$

$$(6.1.22) \quad \Rightarrow \quad \begin{cases} \mathbf{M} \left\{ \frac{d}{dt} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = \vec{\varphi}(t) & \text{for } 0 < t < T, \\ \vec{\mu}(0) = \vec{\mu}_0. \end{cases} \quad (6.1.23)$$

- ▷ s.p.d. stiffness matrix $\mathbf{A} \in \mathbb{R}^{N,N}$, $(\mathbf{A})_{ij} := \mathbf{a}(b_N^j, b_N^i)$ (independent of time),
- ▷ s.p.d. **mass matrix** $\mathbf{M} \in \mathbb{R}^{N,N}$, $(\mathbf{M})_{ij} := \mathbf{m}(b_N^j, b_N^i)$ (independent of time),
- ▷ source (load) vector $\vec{\varphi}(t) \in \mathbb{R}^N$, $(\vec{\varphi}(t))_i := \ell(t)(b_N^i)$ (time-dependent),
- ▷ $\vec{\mu}_0 \hat{=} \text{coefficient vector of a projection of } u_0 \text{ onto } V_{0,N}$.

For the concrete linear parabolic evolution problem (6.1.8)–(6.1.9) and spatial finite element discretization based on a finite element trial/test space $V_{0,N} \subset H^1(\Omega)$ we can compute

- the mass matrix \mathbf{M} as the Galerkin matrix for the bilinear form $(u, v) \mapsto \int_{\Omega} \rho(\mathbf{x}) uv \, d\mathbf{x}$, $u, v \in L^2(\Omega)$,
- the stiffness matrix \mathbf{A} as Galerkin matrix arising from the bilinear form $(u, v) \mapsto \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x}$, $u, v \in H^1(\Omega)$.

The calculations are explained in Sects. 3.5.3 and 3.5.4 and may involve numerical quadrature.

Note:

(6.1.23) is an ordinary differential equation (ODE) for $t \mapsto \vec{\mu}(t) \in \mathbb{R}^N$

Conversion (6.1.11) \rightarrow (6.1.23) through Galerkin discretization *in space only* is known as **method of lines**.

(6.1.23) $\hat{=}$ A **semi-discrete** evolution problem

Discretized in space \longleftrightarrow but still continuous in time

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 6.1.24 (Spatial discretization options).

Beside the Galerkin approach any other method for spatial discretization of 2nd-order elliptic BVPs can be used in the context of the method of lines: the matrices **A**, **M** may also be generated by finite differences (\rightarrow Sect. 4.1), finite volume methods (\rightarrow Sect. 4.2), or collocation methods (\rightarrow Sect. 1.5.2).



6.1.4 Timestepping

For implementation we need a **fully discrete** evolution problem. This requires additional discretization in time:

semi-discrete evolution problem (6.1.23) + timestepping \blacktriangleright **fully discrete** evolution problem

Benefit of method of lines: we can apply already known integrators for initial value problems for ODEs to (6.1.23).

First, refresh central concepts from numerical integration of initial value problems for ODEs, see [21, Ch. 12], [21, Ch. 13]:

- single step methods of order p , see [21, Def. 12.2.12] and [21, Thm. 12.3],
- explicit and implicit Runge-Kutta single step methods, see [21, Sect. 12.4], [21, Sect. ??], encoded by Butcher scheme [21, Eq. 12.4.9], [21, Eq 13.3.6].
- the notion of a **stiff problem** (\rightarrow [21, Notion 13.2.17]),
- the definition of the **stability function** of a single step method, see [21, Thm. 13.3.7],
- the concept of **L-stability** [21, Def 13.3.9] and how to verify it for Runge-Kutta methods.

6.1.4.1 Single step methods

Recall: single step methods (\rightarrow [21, Def. 12.2.12])

- are based on a **temporal mesh** $\{0 = t_0 < t_1 < \dots < t_{M-1} < t_M := T\}$ (with local timestep size $\tau_j = t_j - t_{j-1}$),
- compute sequence $(\vec{\mu}^{(j)})_{j=0}^M$ of approximations $\vec{\mu}^{(j)} \approx \mu(t_j)$ to the solution of (6.1.23) at the nodes of the temporal mesh according to

$$\vec{\mu}^{(j)} := \Psi^{t_{j-1}, t_j} \vec{\mu}^{(j-1)} := \Psi(t_{j-1}, t_j, \vec{\mu}^{(j-1)}), \quad j = 1, \dots, M,$$

where Ψ is the **discrete evolution** defining the single step method, see [21, Def. 12.2.12].

Example 6.1.28 (Euler timestepping). \rightarrow [21, Sect. 12.2]

We target the initial value problem

$$\begin{aligned} \mathbf{M} \left\{ \frac{d}{dt} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) &= \vec{\varphi}(t) \quad \text{for } 0 < t < T, \\ \vec{\mu}(0) &= \vec{\mu}_0. \end{aligned} \quad (6.1.23)$$

Explicit Euler method [21, Eq. 12.2.4]: replace $\frac{d}{dt}$ in (6.1.23) with forward difference quotient, see [21, Rem. 12.2.5]:

$$(6.1.23) \quad \blacktriangleright \quad \mathbf{M} \vec{\mu}^{(j)} = \mathbf{M} \vec{\mu}^{(j-1)} - \tau_j (\mathbf{A} \vec{\mu}^{(j-1)} - \vec{\varphi}(t_{j-1})), \quad j = 1, \dots, M-1. \quad (6.1.29)$$

Implicit Euler method [21, Eq. 12.2.8]: replace $\frac{d}{dt}$ in (6.1.23) with backward difference quotient

$$(6.1.23) \quad \blacktriangleright \quad \mathbf{M} \vec{\mu}^{(j)} = \mathbf{M} \vec{\mu}^{(j-1)} - \tau_j (\mathbf{A} \vec{\mu}^{(j)} - \vec{\varphi}(t_j)), \quad j = 1, \dots, M-1. \quad (6.1.30)$$

Note that both (6.1.29) and (6.1.30) require the solution of a linear system of equations in each step

$$(6.1.29): \quad \vec{\mu}^{(j)} = \vec{\mu}^{(j-1)} + \tau_j \mathbf{M}^{-1} (\vec{\varphi}(t_{j-1}) - \mathbf{A} \vec{\mu}^{(j-1)}),$$

$$(6.1.30): \quad \vec{\mu}^{(j)} = (\tau_j \mathbf{A} + \mathbf{M})^{-1} \left(\mathbf{M} \vec{\mu}^{(j-1)} + \tau_j \vec{\varphi}(t_j) \right).$$

Recall [21, Sect. 12.3]: both Euler method are of first order.



Example 6.1.31 (Crank-Nicolson timestepping).

Crank-Nicolson method = implicit midpoint rule: replace $\frac{d}{dt}$ in (6.1.23) with symmetric difference quotient and average right hand side:

$$\begin{aligned} \mathbf{M} \left\{ \frac{d}{dt} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) &= \vec{\varphi}(t) \\ \Downarrow \\ \mathbf{M} \frac{\vec{\mu}^{(j)} - \vec{\mu}^{(j-1)}}{\tau} &= -\frac{1}{2} \mathbf{A} \left(\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)} \right) + \frac{1}{2} (\vec{\varphi}(t_j) + \vec{\varphi}(t_{j-1})) . \end{aligned} \quad (6.1.32)$$

This yields a method that is 2nd-order consistent.



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Generalization of Euler methods:

Runge-Kutta single step methods → [21, Sect. 12.4], [21, Sect. 13.3]

Definition 6.1.33 (General Runge-Kutta method). \rightarrow [21, Def. 13.3.5]

For coefficients $b_i, a_{ij} \in \mathbb{R}$, $c_i := \sum_{j=1}^s a_{ij}$, $i, j = 1, \dots, s$, $s \in \mathbb{N}$, the discrete evolution $\Psi^{s,t}$ of an s -stage Runge-Kutta single step method (RK-SSM) for the ODE $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$, is defined by

$$\mathbf{k}_i := \mathbf{f}\left(t + c_i\tau, \mathbf{y} + \tau \sum_{j=1}^s a_{ij}\mathbf{k}_j\right), \quad i = 1, \dots, s, \quad \Psi^{t, t+\tau}\mathbf{y} := \mathbf{y} + \tau \sum_{i=1}^s b_i\mathbf{k}_i.$$

The $\mathbf{k}_i \in \mathbb{R}^d$ are called *increments*.

 R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Shorthand notation for s -stage Runge-Kutta methods: **Butcher scheme** \rightarrow [21, Eq. 13.3.6]

$$\frac{\mathbf{c} \mid \mathfrak{A}}{\mathbf{b}^T} \hat{=} \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & \dots & a_{1s} \\ c_2 & a_{21} & \ddots & & & a_{2s} \\ \vdots & \vdots & & \ddots & & \vdots \\ c_s & a_{s1} & \vdots & & & a_{ss} \\ \hline & b_1 & b_2 & \dots & \dots & b_s \end{array}, \quad \mathbf{c}, \mathbf{b} \in \mathbb{R}^s, \quad \mathfrak{A} \in \mathbb{R}^{s,s}. \quad (6.1.34)$$

Concretely for linear parabolic evolution: application of s -stage Runge-Kutta method to

$$\mathbf{M} \left\{ \frac{d}{dt} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = \vec{\varphi}(t) \Leftrightarrow \dot{\vec{\mu}} = \underbrace{\mathbf{M}^{-1} (\vec{\varphi}(t) - \mathbf{A} \vec{\mu}(t))}_{=\mathbf{f}(t, \vec{\mu})}. \quad (6.1.23)$$

Then simply plug this into the formulas of Def. 6.1.33.

► Timestepping scheme for (6.1.23): compute $\vec{\mu}^{(j+1)}$ from $\vec{\mu}^{(j)}$ through

$$\vec{\kappa}_i \in \mathbb{R}^N: \quad \mathbf{M}\vec{\kappa}_i + \sum_{m=1}^s \tau a_{im} \mathbf{A}\vec{\kappa}_m = \vec{\varphi}(t_j + c_i\tau) - \mathbf{A}\vec{\mu}^{(j)}, \quad i = 1, \dots, s, \quad (6.1.35)$$

$$\vec{\mu}^{(j+1)} = \vec{\mu}^{(j)} + \tau \sum_{m=1}^s \vec{\kappa}_m b_m. \quad (6.1.36)$$

Note: For an implicit RK-method (6.1.35) is a linear system of equations of size Ns .

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

6.1.4.2 Stability

Example 6.1.37 (Convergence of Euler timestepping).

Parabolic evolution problem in one spatial dimension (IBVP):

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad \text{in } [0, 1] \times]0, 1[, \quad (6.1.38)$$

6.1
p. 638

$$u(t, 0) = u(t, 1) = 0 \quad \text{for } 0 \leq t \leq 1, \quad u(0, x) = \sin(\pi x) \quad \text{for } 0 < x < 1. \quad (6.1.39)$$

► exact solution $u(t, x) = \exp(-\pi^2 t) \sin(\pi x).$ (6.1.40)

- Spatial finite element Galerkin discretization by means of linear finite elements ($V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$) on equidistant mesh \mathcal{M} with meshwidth $h := \frac{1}{N} \rightarrow$ Sect. 1.5.1.2.
- $u_{N,0} := I_1 u_0$ by linear interpolation on \mathcal{M} , see Sect. 5.3.1.
- Timestepping by explicit and implicit Euler method (6.1.29), (6.1.30) with uniform timestep $\tau := \frac{1}{M}$.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Galerkin matrices, see (1.5.86):

$$\mathbf{A} = \frac{1}{h} \begin{pmatrix} 2 & -1 & 0 & & & & & 0 \\ -1 & 2 & -1 & & & & & \\ 0 & \cdots & \cdots & \cdots & & & & \\ & & & & \cdots & \cdots & \cdots & 0 \\ & & & & & -1 & 2 & -1 \\ 0 & & & & & 0 & -1 & 2 \end{pmatrix}, \quad \mathbf{M} = \frac{h}{6} \begin{pmatrix} 4 & 1 & 0 & & & & & 0 \\ 1 & 4 & 1 & & & & & \\ 0 & \cdots & \cdots & \cdots & & & & \\ & & & & \cdots & \cdots & \cdots & 0 \\ & & & & & 1 & 4 & 1 \\ 0 & & & & & 0 & 1 & 4 \end{pmatrix}.$$

Code 6.1.41: Euler timestepping for (6.1.38)

```
1 function [errex,errimp] = sinevl(N,M,u)
2 % Solve fully discrete two-point parabolic evolution problem (6.1.38)
3 % in  $[0,1] \times ]0,1[$ . Use both explicit and implicit Euler method for timestepping
4 % N: number of spatial grid cells
5 % M: number of timesteps
6 % u: handle of type @(t,x) to exact solution
7
8 if (nargin < 3), u = @(t,x) (exp(-(pi^2)*t) .* sin(pi*x)); end %
   Exact solution
9
10 h = 1/N; tau = 1/M; % Spatial and temporal meshwidth
11 x = h:h:1-h; % Spatial grid, interior points
12
13 % Finite element stiffness and mass matrix
14 Amat = gallery('tridiag',N-1,-1,2,-1)/h;
15 Mmat = h/6*gallery('tridiag',N-1,1,4,1);
16 Xmat = Mmat+tau*Amat;
17
18 mu0 = u(0,x)'; % Discrete initial value
19 mui = mu0; mue = mu0;
20
21 %Timestepping
```

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ


```
22 erre = 0; erri = 0;
23 for k=1:M
24     mue = mue - tau*(Mmat\ (Amat*mue)); % explicit Euler step
25     mui = Xmat\ (Mmat*mui);           % implicit Euler step
26     utk = u(k*tau,x)';
27     erre = erre + norm(mue-utk)^2;    % Computation of error norm
28     erri = erri + norm(mui-utk)^2;
29 end
30
31 errex = sqrt(erre*h*tau);
32 errimp = sqrt(erri*h*tau);
```

Evaluation of approximate space-time L^2 -norm of the discretization error:

$$\text{err}^2 := h\tau \cdot \sum_{j=1}^M \sum_{i=1}^{N-1} |u(t_j, x_i) - \mu_i^{(j)}|^2. \quad (6.1.42)$$

Space-time (discrete) L^2 -norm of error for **explicit Euler** timestepping:

$N \backslash M$	50	100	200	400	800	1600	3200
5	Inf	0.009479	0.006523	0.005080	0.004366	0.004011	0.003834
10	Inf	Inf	Inf	Inf	0.001623	0.001272	0.001097
20	Inf	Inf	Inf	Inf	Inf	Inf	0.000405
40	Inf	Inf	Inf	Inf	Inf	Inf	Inf
80	Inf	Inf	Inf	Inf	Inf	Inf	Inf
160	Inf	Inf	Inf	Inf	Inf	Inf	Inf
320	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Space-time (discrete) L^2 -norm of error for **implicit Euler** timestepping:

$N \backslash M$	50	100	200	400	800	1600	3200
5	0.007025	0.001828	0.000876	0.002257	0.002955	0.003306	0.003482
10	0.009641	0.004500	0.001826	0.000461	0.000228	0.000575	0.000749
20	0.010303	0.005175	0.002509	0.001149	0.000461	0.000116	0.000058
40	0.010469	0.005345	0.002681	0.001321	0.000634	0.000289	0.000116
80	0.010511	0.005387	0.002724	0.001364	0.000677	0.000332	0.000159
160	0.010521	0.005398	0.002734	0.001375	0.000688	0.000343	0.000170
320	0.010524	0.005400	0.002737	0.001378	0.000691	0.000346	0.000172

Explicit Euler timestepping: we observe a glaring **instability** (exponential blow-up) in case of *large timestep combined with fine mesh*.

Implicit Euler timestepping: no blow-up at any combination of spatial and temporal mesh width.



Example 6.1.43 (`ode45` for discrete parabolic evolution).

Same IBVP and spatial discretization as in Ex. 6.1.37.

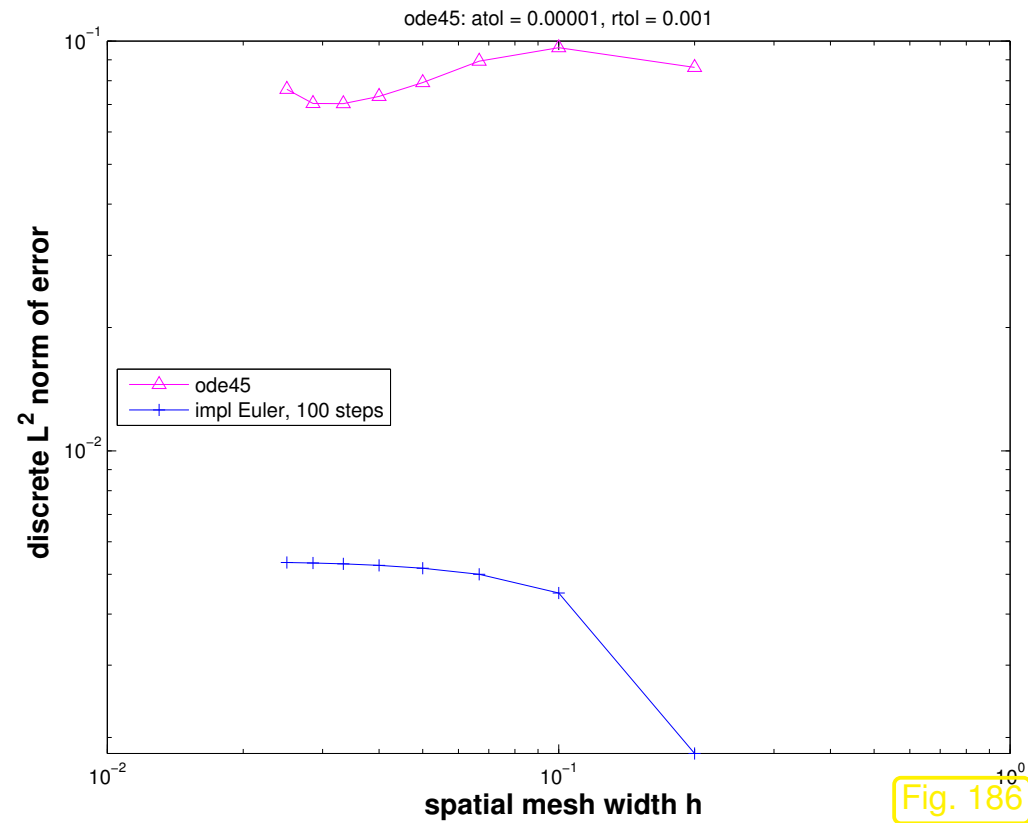
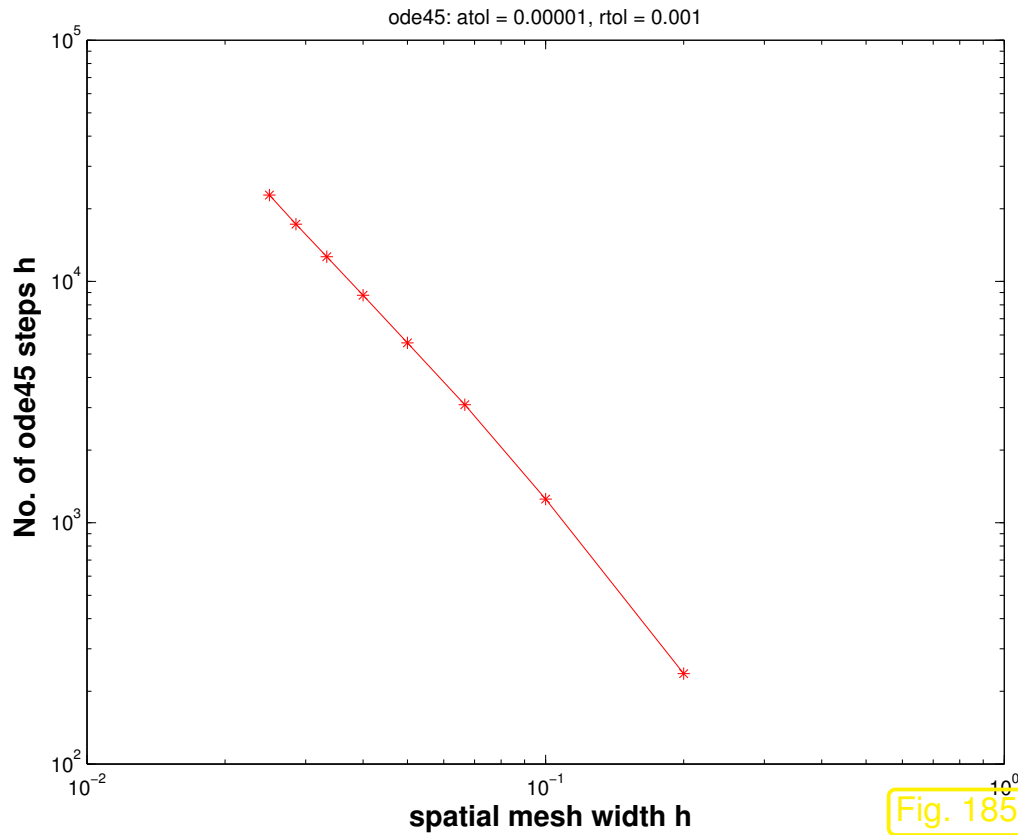
Adaptive Runge-Kutta timestepping by MATLAB standard integrator `ode45`.

Monitored:

- Number of timesteps as a function on spatial meshwidth h ,
- discrete L^2 -error (6.1.42).

Code 6.1.44: ode45 applied semi-discrete (6.1.38)

```
1 function [Nsteps,err] = peode45(N,tol,u)
2 % Solving fully discrete two-point parabolic evolution problem (6.1.38)
3 % in  $[0,1] \times ]0,1[$  by means of adaptiv MATLAB standard Runge-Kutta integrator.
4 if (nargin < 3), u = @(t,x) (exp(-(pi^2)*t) .* sin(pi*x)); end %
   Exact solution
5
6 % Finite element stiffness and mass matrix, see Sect. 1.5.1.2
7 h = 1/N; % spatial meshwidth
8 Amat = gallery('tridiag',N-1,-1,2,-1)/h;
9 Mmat = h/6*gallery('tridiag',N-1,1,4,1);
10 x = h:h:1-h; % Spatial grid, interior points
11
12 mu0 = u(0,x)'; % Discrete initial value
13 fun = @(t,muv) -(Mmat \ (Amat*muv)); % right hand side of ODE
14
15 opts = odeset('reltol',tol,'abstol',0.01*tol);
16 [t,mu] = ode45(fun,[0,1],mu0,opts);
17
18 Nsteps = length(t);
19 [T,X] = meshgrid(t,x); err = norm(mu'-u(T,X),'fro');
```



Observations:

- `ode45`: dramatic increase of no. of timesteps for $h_M \rightarrow 0$ without gain in accuracy.
- Implicit Euler achieves better accuracy with only 100 equidistant timesteps!



This reminds us of the **stiff initial value problems** studied in [21, Thm. 13.2]:

Notion 6.1.45 (Stiff IVP). \rightarrow [21, Notion 13.2.17]

*An initial value problem for an ODE is called **stiff**, if stability imposes much tighter timestep constraints on explicit single step methods than the accuracy requirements.*

Admittedly, this is a fuzzy notion. Yet, it cannot be fleshed out on the abstract level, but has to be discussed for concrete evolution problem, which is done next.

Let us try to understand, why semi-discrete parabolic evolutions (6.1.23) arising from the method of lines lead to stiff initial value problems.

Technique: **Diagonalization**, cf. [21, Eq. 13.2.5]

(Recall the concept of a “square root” $\mathbf{M}^{1/2}$ of an s.p.d. matrix \mathbf{M} , see [21, Sect. 5.3])

\mathbf{A}, \mathbf{M} symmetric positive definite $\Rightarrow \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$ symmetric positive definite .

[21, Cor. 6.1.9] $\Rightarrow \exists$ orthogonal $\mathbf{T} \in \mathbb{R}^{N,N}$: $\mathbf{T}^\top \mathbf{M}^{-1/2} \mathbf{A} \mathbf{M}^{-1/2} \mathbf{T} = \mathbf{D} := \text{diag}(\lambda_1, \dots, \lambda_N)$,

where the $\lambda_i > 0$ are *generalized eigenvalues* for $\mathbf{A}\vec{\xi} = \lambda\mathbf{M}\vec{\xi} \blacktriangleright \lambda_i \geq \gamma$ for all i (γ is the constant introduced in (6.1.12)).

\blacktriangleright Transformation (“diagonalization”) of (6.1.23) based on substitution $\vec{\eta} := \mathbf{T}^\top \mathbf{M}^{1/2} \vec{\mu}$:

$$(6.1.23) \quad \vec{\eta} := \mathbf{T}^\top \mathbf{M}^{1/2} \vec{\mu} \implies \frac{d}{dt} \vec{\eta}(t) + \mathbf{D} \vec{\eta} = \mathbf{T}^\top \mathbf{M}^{-1/2} \vec{\varphi}(t) . \quad (6.1.46)$$

\blacktriangleright Since \mathbf{D} is *diagonal*, (6.1.46) amounts to N decoupled scalar ODEs (for eigencomponents η_i of $\vec{\mu}$).

Note: for $\vec{\varphi} \equiv 0, \lambda > 0$: $\eta_i(t) = \exp(-\lambda_i t) \eta_i(0) \rightarrow 0$ for $t \rightarrow \infty$

As in [21, Thm. 13.2.7] this transformation can be applied to the explicit Euler timestepping (6.1.29) (for $\vec{\varphi} \equiv 0$, uniform timestep $\tau > 0$)

$$\vec{\mu}^{(j)} = \vec{\mu}^{(j-1)} - \tau \mathbf{M}^{-1} \mathbf{A} \vec{\mu}^{(j-1)} \quad \vec{\eta} := \mathbf{T}^\top \mathbf{M}^{1/2} \vec{\mu} \quad \blacktriangleright \quad \vec{\eta}^{(j)} = \vec{\eta}^{(j-1)} - \tau \mathbf{D} \vec{\eta}^{(j-1)} ,$$

that is, the decoupling of eigencomponents carries over to the explicit Euler method: for $i = 1, \dots, N$

$$\eta_i^{(j)} = \eta_i^{(j-1)} - \tau \lambda_i \eta_i^{(j-1)} \Rightarrow \boxed{\eta_i^{(j)} = (1 - \tau \lambda_i)^j \eta_i^{(0)}}. \quad (6.1.47)$$

$$|1 - \tau \lambda_i| < 1 \Leftrightarrow \lim_{j \rightarrow \infty} \eta_i^{(j)} = 0. \quad (6.1.48)$$

The condition $|1 - \tau \lambda_i| < 1$ enforces a

$$\text{timestep size constraint: } \tau < \frac{2}{\lambda_i} \quad (6.1.49)$$

in order to achieve the qualitatively correct behavior $\lim_{j \rightarrow \infty} \eta_i^{(j)} = 0$ and to avoid blow-up $\lim_{j \rightarrow \infty} |\eta_i^{(j)}| = \infty$:

the timestep size constraint (6.1.49) is necessary *only* for the sake of stability (not in order to guarantee a prescribed accuracy).

This accounts to the observed blow-ups in Ex. 6.1.37. On the other hand, adaptive stepsize control [21, Sect. 12.5] manages to ensure the timestep constraint, but the expense of prohibitively small timesteps that render the method *grossly inefficient*, if some of the λ_i are large.

The next numerical demonstrations and Lemma show that $\lambda_{\max} := \max_i \lambda_i$ will inevitably become huge for finite element discretization on fine meshes.

Example 6.1.50 (Behavior of generalized eigenvalues of $\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu}$).

Bilinear forms associated with parabolic IBVP and homogeneous Dirichlet boundary conditions

$$\mathbf{a}(u, v) = \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, dx \quad , \quad \mathbf{m}(u, v) = \int_{\Omega} u(x)v(x) \, dx \quad , \quad u, v \in H_0^1(\Omega) .$$

Linear finite element Galerkin discretization, see Sect. 1.5.1.2 for 1D, and Sect. 3.2 for 2D.

Numerical experiments in 1D & 2D:

- $\Omega =]0, 1[$, equidistant meshes \rightarrow Ex. 6.1.37
- “disk domain” $\Omega = \{\mathbf{x} \in \mathbb{R}^2: \|\mathbf{x}\| < 1\}$, sequence of regularly refined meshes.

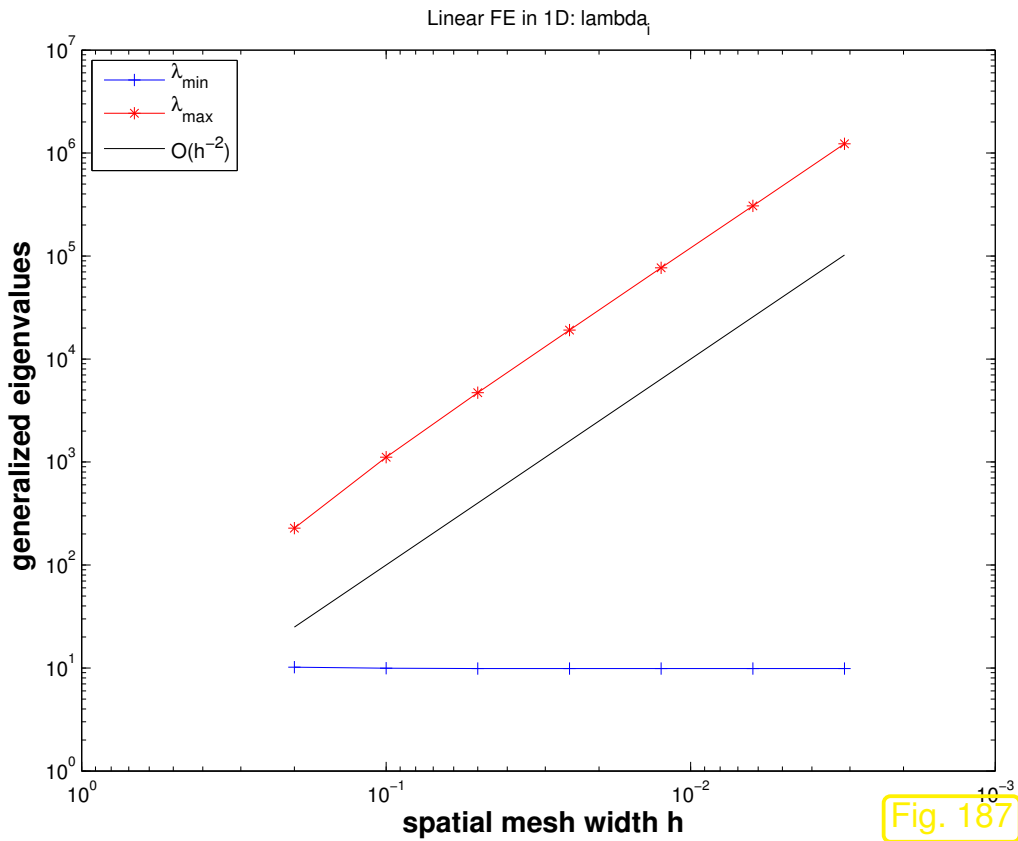
Monitored: largest and smallest generalized eigenvalue

Code 6.1.51: Computation of extremal generalized eigenvalues

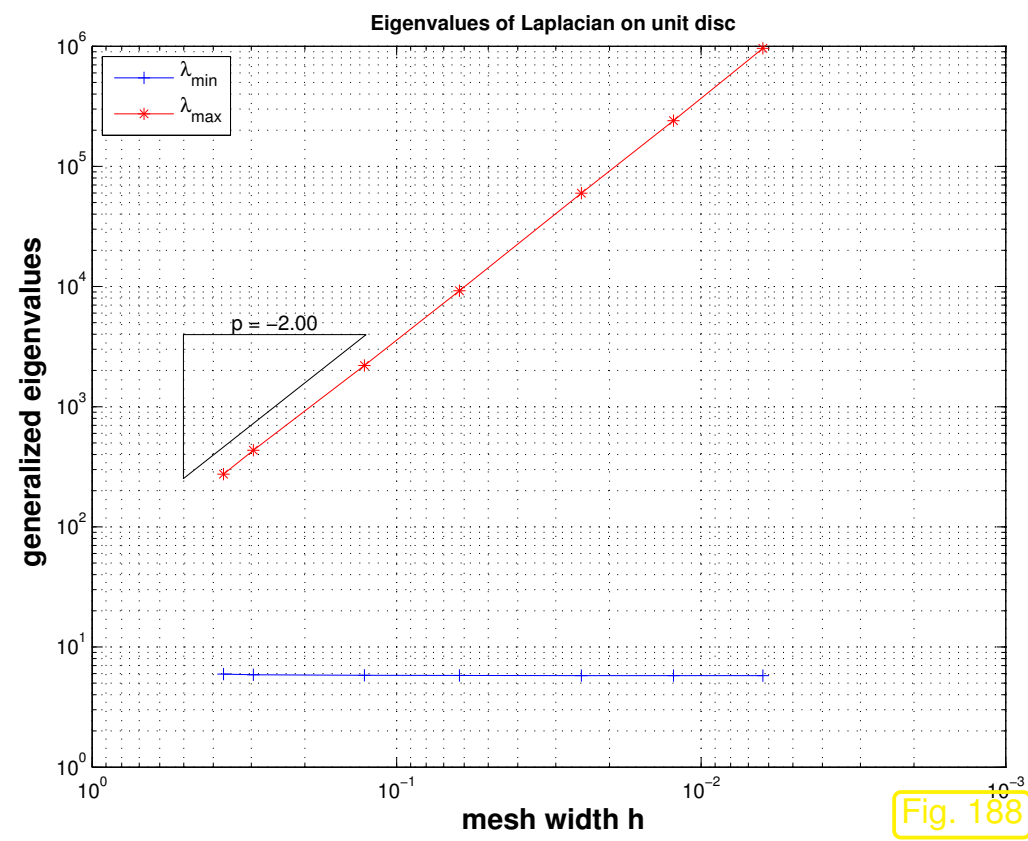
```
1 % LehrFEM MATLAB script for computing Dirichlet eigenvalues of Laplacian
2 % on a unit disc domain.
3
4 GD_HANDLE = @(x,varargin) zeros ( size (x,1),1); % Zero Dirichlet data
5 H0 = [ .25 .2 .1 .05 .02 .01 0.005]'; % target mesh widths
6 NRef = length (H0); % Number of refinement steps
7
8 % Variables for mesh widths and eigenvalues
9 M_W = zeros (NRef,1); lmax = M_W; lmin = M_W;
10
11 % Main refinement loop
12 for iter = 1:NRef
13
14 % Set parameters for mesh
15     C = [0 0]; % Center of circle
16     R = 1; % Radius of circle
17     BBOX = [-1 -1; 1 1]; % Bounding box
18     DHANDLE = @dist_circ; % Signed distance function
19     HHANDLE = @h_uniform; % Element size function
20     FIXEDPOS = []; % Fixed boundary vertices of the mesh
21     DISP = 0; % Display flag
22
```

```
23 % Mesh generation
24 Mesh =
    init_Mesh(BBOX,H0(iter),DHANDLE,HHANDLE,FIXEDPOS,DISP,C,R);
25 Mesh = add_Edges(Mesh); % Provide edge information
26 Loc = get_BdEdges(Mesh); % Obtain indices of edges on  $\partial\Omega$ 
27 Mesh.BdFlags = zeros(size(Mesh.Edges,1),1);
28 Mesh.BdFlags(Loc) = -1; % Flag boundary edges
29 Mesh.ElemFlag = zeros(size(Mesh.Elements,1),1);
30 M_W(iter) = get_MeshWidth(Mesh); % Get mesh width
31
32 fprintf('Mesh on level %i: %i elements, h =
    %f\n',iter,size(Mesh,1),M_W(iter));
33 % Assemble stiffness matrix and mass matrix
34 A = assemMat_LFE(Mesh,@STIMA_Lapl_LFE,P706());
35 M = assemMat_LFE(Mesh,@MASS_LFE,P706());
36 % Incorporate Dirichlet boundary data (nothing to do here)
37 [U,FreeNodes] = assemDir_LFE(Mesh,-1,GD_HANDLE);
38 A = A(FreeNodes,FreeNodes);
39 M = M(FreeNodes,FreeNodes);
40
41 % Use MATLAB's built-in eigs-function to compute the
42 % extremal eigenvalues, see [21, Sect. 6.4].
43 NEigen = 6;
```

```
44 d = eigs(A,M,NEigen,'sm'); lmin(iter) = min(d);
45 d = eigs(A,M,NEigen,'lm'); lmax(iter) = max(d);
46 end
47
48 figure; plot(M_W,lmin,'b-+',M_W,lmax,'r-*'); grid on;
49 set(gca,'XScale','log','YScale','log','XDir','reverse');
50 title('\bf Eigenvalues of Laplacian on unit disc');
51 xlabel('\bf mesh width h','fontsize',14);
52 ylabel('\bf generalized eigenvalues','fontsize',14);
53 legend('\lambda_{min}','\lambda_{max}','Location','NorthWest')
54 p = polyfit(log(M_W),log(lmax),1);
55 add_Slope(gca,'east',p(1));
56
57 print -depsc2 '../.../Slides/NPDEpics/geneigdisklfe.eps';
```



$$\Omega =]0, 1[$$



$$\Omega = \{ \mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| < 1 \}$$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Observation:

- $\lambda_{\min} := \min_i \lambda_i$ does hardly depend on the mesh width.
- $\lambda_{\max} := \max_i \lambda_i$ displays a $O(h_{\mathcal{M}}^{-2})$ growth as $h_{\mathcal{M}} \rightarrow 0$

Remark 6.1.52 (Spectrum of elliptic operators).

The observation made in Ex. 6.1.50 is not surprising!

To understand why, let us translate the generalized eigenproblem “back to the ODE/PDE level”:

$$\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu} \tag{6.1.53}$$



$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = \lambda \mathbf{m}(u_N, v_N) \quad \forall v_N \in V_{0,N} .$$

← “undo Galerkin discretization”

$$u \in H_0^1(\Omega): \quad \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = \lambda \int_{\Omega} u \cdot v \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) .$$



$$-\Delta u = \lambda u \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega, \tag{6.1.54}$$

which is a so-called **elliptic eigenvalue problem**.

It is easily solved in 1D on $\Omega =]0, 1[$:

$$(6.1.54) \hat{=} \quad \frac{d^2 u}{dx^2}(x) = \lambda u(x), \quad 0 < x < 1, \quad u(0) = u(1) = 0.$$
$$\Rightarrow \quad u_k(x) = \sin(k\pi x) \quad \leftrightarrow \quad \lambda_k = (\pi k)^2, \quad k \in \mathbb{N}.$$

Note that we find an infinite number of eigenfunctions and eigenvalues, parameterized by $k \in \mathbb{N}$. The eigenvalues tend to ∞ for $k \rightarrow \infty$:

$$\lambda_k = O(k^2) \quad \text{for } k \rightarrow \infty.$$

Of course, (6.1.53) can have a finite number of eigenvectors only. Crudely speaking, they correspond to those eigenfunctions $u_k(x) = \sin(k\pi x)$ that can be resolved by the mesh (if u_k “oscillates too much”, then it cannot be represented on a grid). These are the first N so that we find in 1D for an equidistant mesh

$$\lambda_{\max} = O(N^2) = O(h_{\mathcal{M}}^{-2}).$$

This is heuristics, but the following Lemma will a precise statement.

Lemma 6.1.55 (Behavior of of generalized eigenvalues).

Let \mathcal{M} be a simplicial mesh and \mathbf{A} , \mathbf{M} denote the Galerkin matrices for the bilinear forms $\mathbf{a}(u, v) = \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x}$ and $\mathbf{m}(u, v) = \int_{\Omega} u(x)v(x) \, d\mathbf{x}$, respectively, and $V_{0,N} := \mathcal{S}_{p,0}^0(\mathcal{M})$. Then the smallest and largest generalized eigenvalues of $\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu}$, denoted by λ_{\min} and λ_{\max} , satisfy

$$\frac{1}{\text{diam}(\Omega)^2} \leq \lambda_{\min} \leq C \quad , \quad \lambda_{\max} \geq Ch_{\mathcal{M}}^{-2} \quad ,$$

where the “generic constants” (\rightarrow Rem. 5.3.44) depend only on the polynomial degree p and the shape regularity measure $\rho_{\mathcal{M}}$.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Proof. (partial) We rely on the **Courant-Fischer min-max theorem** [21, Thm. 6.3.39] that, among other consequences, expresses the boundaries of the spectrum of a symmetric matrix through the extrema of its Rayleigh quotient

$$\mathbf{T} = \mathbf{T}^T \in \mathbb{R}^{N,N} \quad \Rightarrow \quad \lambda_{\min}(\mathbf{T}) = \min_{\vec{\xi} \in \mathbb{R}^N \setminus \{0\}} \frac{\vec{\xi}^T \mathbf{T} \vec{\xi}}{\vec{\xi}^T \vec{\xi}} \quad , \quad \lambda_{\max}(\mathbf{T}) = \max_{\vec{\xi} \in \mathbb{R}^N \setminus \{0\}} \frac{\vec{\xi}^T \mathbf{T} \vec{\xi}}{\vec{\xi}^T \vec{\xi}} \quad .$$

Apply this to the generalized eigenvalue problem

$$\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu} \quad \vec{\zeta} := \mathbf{M}^{1/2}\vec{\mu} \quad \Leftrightarrow \quad \underbrace{\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}}_{=: \mathbf{T}} \vec{\zeta} = \lambda\vec{\zeta} .$$

$$\blacktriangleright \quad \lambda_{\min} = \min_{\vec{\mu} \neq 0} \frac{\vec{\mu}^T \mathbf{A} \vec{\mu}}{\vec{\mu}^T \mathbf{M} \vec{\mu}}, \quad \lambda_{\max} = \max_{\vec{\mu} \neq 0} \frac{\vec{\mu}^T \mathbf{A} \vec{\mu}}{\vec{\mu}^T \mathbf{M} \vec{\mu}} . \quad (6.1.56)$$

As a consequence we only have to find bounds for the extrema of a **generalized Rayleigh quotient**, cf. [21, Eq. 6.3.35]. This generalized Rayleigh quotient can be expressed as

$$\frac{\vec{\mu}^T \mathbf{A} \vec{\mu}}{\vec{\mu}^T \mathbf{M} \vec{\mu}} = \frac{\mathbf{a}(u_N, u_N)}{\mathbf{m}(u_N, u_N)}, \quad \vec{\mu} \hat{=} \text{coefficient vector for } u_N . \quad (6.1.57)$$

Now we discuss a lower bound for λ_{\max} , which can be obtained by inserting a suitable *candidate function* into (6.1.57).

Discussion for special setting: $V_{0,N} = \mathcal{S}_1^0(\mathcal{M})$ on triangular mesh \mathcal{M}

Candidate function: “tent function” $u_N = b_N^i$ (\rightarrow Sect. 3.2.3) for some node $\mathbf{x}^i \in \mathcal{V}(\mathcal{M})$ of the mesh!

By elementary computations as in Sect. 3.2.5 we find

$$\mathbf{a}(b_N^i, b_N^i) \approx C \quad , \quad \mathbf{m}(b_N^i, b_N^i) \leq C \max_{K \in \mathcal{U}(\mathbf{x}^i)} h_K^2 , \quad (6.1.58)$$

where the generic constants $C > 0$ depend on the shape regularity measure $\rho_{\mathcal{M}}$ only.

$$(6.1.56) \ \& \ (6.1.58) \quad \Rightarrow \quad \lambda_{\max} \geq Ch_{\mathcal{M}}^{-2} . \quad \square$$

Lemma 6.1.55 \blacktriangleright timestep constraint (6.1.49) unacceptable to semi-discrete parabolic evolutions!

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

From [21, Sect. 13.3] we already know that some *implicit* single step methods are not affected by stability induced timestep constraints.

Recall [21, Ex. 13.3.1]: apply diagonalization technique, see (6.1.46), to implicit Euler timestepping with uniform timestep $\tau > 0$

$$\vec{\mu}^{(j)} = \vec{\mu}^{(j-1)} - \tau \mathbf{M}^{-1} \mathbf{A} \vec{\mu}^{(j)} \quad \vec{\eta} := \mathbf{T}^{\top} \mathbf{M}^{1/2} \vec{\mu} \quad \blacktriangleright \quad \vec{\eta}^{(j)} = \vec{\eta}^{(j-1)} - \tau \mathbf{D} \vec{\eta}^{(j)} ,$$

that is, the decoupling of eigencomponents carries over to the implicit Euler method: for $i = 1, \dots, N$

$$\eta_i^{(j)} = \eta_i^{(j-1)} - \tau \lambda_i \eta_i^{(j)} \Rightarrow \boxed{\eta_i^{(j)} = \left(\frac{1}{1 + \tau \lambda_i} \right)^j \eta_i^{(0)}} . \tag{6.1.59}$$

$$\left[\left| \frac{1}{1 + \tau \lambda_i} \right| < 1 \quad \text{and} \quad \lambda_i > 0 \Rightarrow \right] \lim_{j \rightarrow \infty} \eta_i^{(j)} = 0 \quad \forall \tau > 0 . \tag{6.1.60}$$

This diagonalization trick can be applied to general Runge-Kutta single step methods (RKSSM, \rightarrow Def. 6.1.33). Loosely speaking, the following diagram commutes

$$\begin{array}{ccc} \mathbf{M} \frac{d}{dt} \vec{\mu} + \mathbf{A} \mu = 0 & \xrightarrow{\text{transformation } \vec{\eta} = \mathbf{T}^T \mathbf{M}^{1/2} \vec{\mu}} & \frac{d}{dt} \eta_i = -\lambda_i \eta_i, \quad i = 1, \dots, N \\ \text{RK-SSM} \downarrow & & \downarrow \text{RK-SSM} \\ \vec{\mu}^{(j)} = \Psi^\tau \vec{\mu}^{(j-1)} & \xrightarrow{\text{transformation } \vec{\eta} = \mathbf{T}^T \mathbf{M}^{1/2} \vec{\mu}} & \vec{\eta}_i^{(j)} = \tilde{\Psi}^\tau \vec{\eta}_i^{(j-1)}, \quad i = 1, \dots, N . \end{array} \tag{6.1.61}$$

The bottom line is

that we have to study the behavior of the RK-SSM *only* for linear scalar ODEs $\dot{y} = -\lambda y, \lambda > 0$.

This is the gist of the **model problem analysis** discussed in [21, Sect. 13.3].

There we saw that everything boils down to inspecting the modulus of a rational **stability function** on \mathbb{C} , see [21, Thm. 13.3.7]. This gave rise to the concept of **L-stability**, see [21, Def. 13.3.9]. Here, we will not delve into a study of stability functions.

Necessary condition for suitability of a single step method for semi-discrete parabolic evolution problem (6.1.23) (“method of lines”):

The discrete evolution $\Psi_\lambda^\tau : \mathbb{R} \mapsto \mathbb{R}$ of the single step method applied to the scalar ODE $\dot{y} = -\lambda y$ satisfies

$$\lambda > 0 \quad \Rightarrow \quad \lim_{j \rightarrow \infty} (\Psi_\lambda^\tau)^j y_0 = 0 \quad \forall y_0 \in \mathbb{R}, \quad \forall \tau > 0. \quad (6.1.62)$$

Definition 6.1.63 ($L(\pi)$ -stability).

*A single step method satisfying (6.1.62) is called **$L(\pi)$ -stable**.*

Example 6.1.64 ($L(\pi)$ -stable Runge-Kutta single step methods).

Simplest example: implicit Euler timestepping (6.1.30).

Some commonly used higher order methods, specified through their Butcher schemes, see (6.1.34):

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

RADAU-3 scheme (order 3)

(6.1.65)

$$\begin{array}{c|cc} \lambda & \lambda & 0 \\ 1 & 1 - \lambda & \lambda \\ \hline & 1 - \lambda & \lambda \end{array}, \quad \lambda := 1 - \frac{1}{2}\sqrt{2}, \quad (6.1.66)$$

SDIRK-2 scheme (order 2)

More examples → [21, Ex. 13.3.15]



6.1.5 Convergence

Why should one prefer complicated implicit $L(\pi)$ -stable Runge-Kutta single step methods (\rightarrow Ex. 6.1.64) to the simple implicit Euler method?

Silly question! Because these methods deliver “better accuracy”!

However, we need some clearer idea of what is meant by this. To this end, we now study the dependence of (a norm of) the discretization error for a parabolic IBVP on the parameters of the spatial and temporal discretization.

Example 6.1.67 (Convergence of fully discrete timestepping in one spatial dimension).

- $\frac{d}{dt}u - u'' = f(t, x)$ on $]0, 1[\times]0, 1[$
- exact solution $u(x, t) = (1 + t^2)e^{-\pi^2 t} \sin(\pi x)$, source term accordingly
- Linear finite element Galerkin discretization equidistant mesh, see Sect. 1.5.1.2, $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$,
- piecewise linear spatial approximation of source term $f(x, t)$
- implicit Euler timestepping (\rightarrow Ex. 6.1.28) with uniform timestep $\tau > 0$

Monitored: error norm $\left(\tau \sum_{j=1}^M |u - u_N(\tau j)|_{H^1(\Omega)}^2 \right)^{\frac{1}{2}}$.

The norms $|u - u_N(\tau j)|_{H^1(\Omega)}$ were approximated by high order local quadrature rules, whose impact can be neglected.

◁ $h_{\mathcal{M}}$ - and τ -dependence of error norm

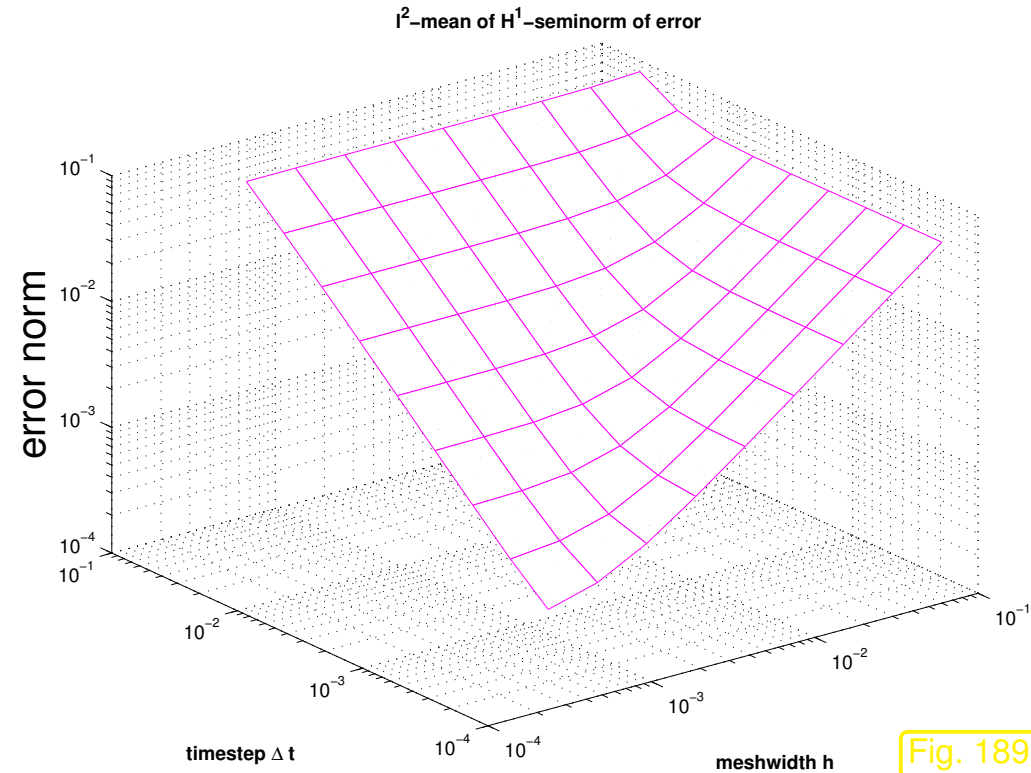


Fig. 189

Obervation:

τ small: error norm $\approx h_{\mathcal{M}}$

$h_{\mathcal{M}}$ small: error norm $\approx \tau$

The error seems to behave like

$$\text{error norm} \approx C_1 h_{\mathcal{M}} + C_2 \tau . \quad (6.1.68)$$

Recall from Sect. 5.3.5, Thm. 5.1.10, Thm. 5.3.42:

energy norm of spatial finite element discretization error $O(h_{\mathcal{M}})$ for $h_{\mathcal{M}} \rightarrow 0$

Since the implicit Euler method is *first order consistent* we expect

temporal timestepping error $O(\tau)$

(6.1.68) ➤ conjecture: total error is **sum** of spatial and temporal discretization error.

From Fig. 189 we draw the compelling conclusion:

- for big mesh width $h_{\mathcal{M}}$ (spatial error dominates) further reduction of timestep size τ is useless,
- if timestep τ is large (temporal error dominates), refinement of the finite element space does not yield a reduction of the total error.



Example 6.1.69 (Higher order timestepping for 1D heat equation).

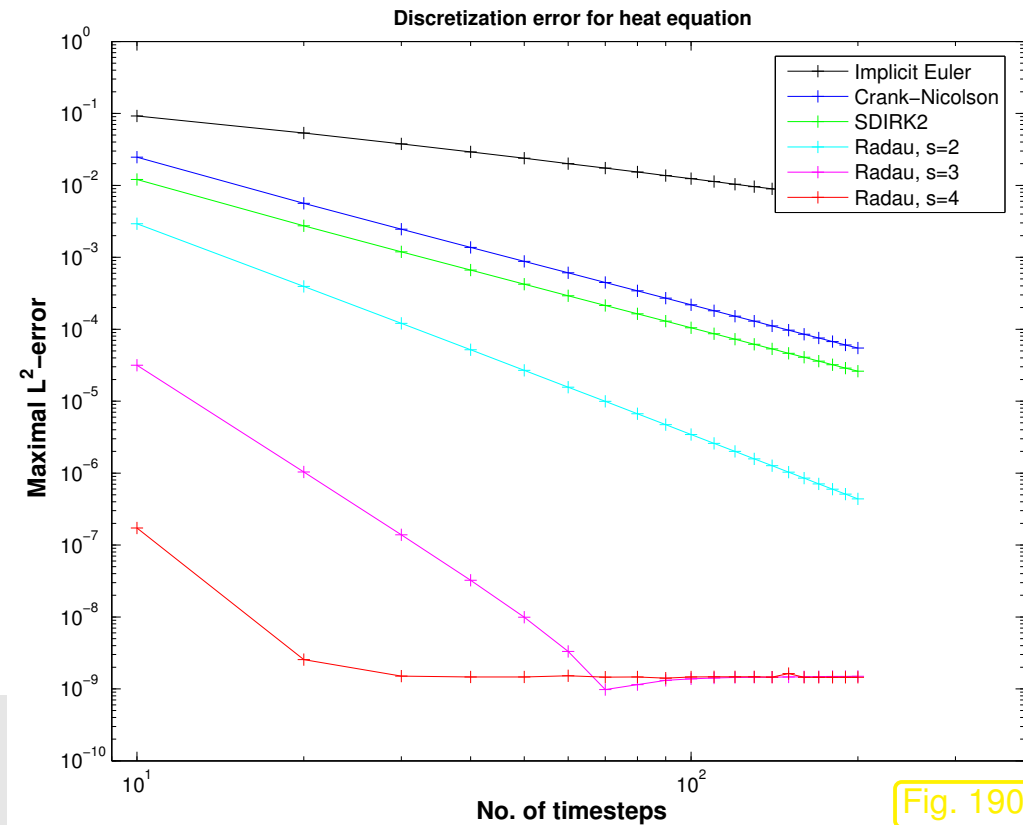
- same IBVP as in Ex. 6.1.67
- spatial discretization on equidistant grid, *very small meshwidth* $h = 0.5 \cdot 10^{-4}$, $V_N = \mathcal{S}_{1,0}^0(\mathcal{M})$

Various timestepping methods

(➤ different **orders of consistency**)

- implicit Euler timestepping (6.1.30), first order
- Crank-Nicolson-method (6.1.32), order 2
- SDIRK-2 timestepping (→ Ex. 6.1.64), order 2
- Gauss-Radau-Runge-Kutta collocation methods with s stages, order $2s - 1$

Note: all methods $L(\pi)$ -stable (→ Def. 6.1.63), except for Crank-Nicolson-method.



Monitored: $\max_j \left\| u(t_j) - u_N^{(j)} \right\|_{L^2([0,1])}$ (evaluated by high order quadrature)



“Meta-theorem” 6.1.70 (Convergence of solutions of fully discrete parabolic evolution problems).

Assume that

- the solution of the parabolic IBVP (6.1.3)–(6.1.5) is “sufficiently smooth” (both in space and time),
- its spatial Galerkin finite element discretization relies on degree p Lagrangian finite elements (\rightarrow Sect. 3.4) on uniformly shape-regular families of meshes,
- timestepping is based on an $L(\pi)$ -stable single step method of order q with uniform timestep $\tau > 0$.

Then we can expect an asymptotic behavior of the total discretization error according to

$$\left(\tau \sum_{j=1}^M |u - u_N(\tau j)|_{H^1(\Omega)}^2 \right)^{\frac{1}{2}} \leq C (h_{\mathcal{M}}^p + \tau^q), \quad (6.1.71)$$

where $C > 0$ must not depend on $h_{\mathcal{M}}, \tau$.

rigorous statement of facts. More details in [22].

A message contained in (6.1.71):

$$\text{total discretization error} = \text{spatial error} + \text{temporal error}$$

Rem. 5.3.45 still applies: (6.1.71) does not give information about actual error, but only about the **trend** of the error, when discretization parameters $h_{\mathcal{M}}$ and τ are varied.

► Nevertheless, as in the case of the a priori error estimates of Sect. 5.3.5, we can draw conclusions about optimal refinement strategies in order to achieve prescribed *error reduction*.

As in Sect. 5.3.5 we make the **assumption** that the estimates (6.1.71) are sharp for all contributions to the total error and that the constants are the same (!)

$$\begin{aligned} \text{contribution of spatial error} &\approx Ch_{\mathcal{M}}^p, \quad h_{\mathcal{M}} \hat{=} \text{mesh width } (\rightarrow \text{Def. 5.2.3}), \\ \text{contribution of temporal error} &\approx C\tau^q, \quad \tau \hat{=} \text{timestep size}. \end{aligned} \tag{6.1.74}$$

This suggests the following change of $h_{\mathcal{M}}, \tau$ in order to achieve *error reduction* by a factor of $\rho > 1$:

$$\begin{aligned} \text{reduce mesh width by factor } \rho^{1/p} \\ \text{reduce timestep by factor } \rho^{1/q} \end{aligned} \xrightarrow{(6.1.74)} \text{error reduction by } \rho > 1. \tag{6.1.75}$$

Guideline: spatial and temporal resolution have to be adjusted in tandem

Remark 6.1.76 (Potential inefficiency of conditionally stable single step methods).

Terminology: A timestepping scheme is labelled **conditionally stable**, if blow-up can be avoided by using sufficient small timesteps (timestep constraint).

Now we can answer the question, why a stability induced timestep constraint like

$$\tau \leq O(h_{\mathcal{M}}^{-2}) \quad (6.1.77)$$

can render a single step method grossly inefficient for integrating semi-discrete parabolic IBVPs.

(6.1.75) ➤ in order to reduce the error by a fixed factor ρ one has to reduce both timestep and meshwidth by some other fixed factors (asymptotically). More concretely, for the timestep τ :

(6.1.75) ➤ **accuracy** requires reduction of τ by a factor $\rho^{1/q}$

(6.1.77) ➤ **stability** entails reduction of τ by a factor $(\rho^{1/p})^2 = \rho^{2/p}$.

$$\frac{1}{q} < \frac{2}{p} \Rightarrow \text{stability enforces smaller timestep than required by accuracy}$$

$$\Rightarrow \text{timestepping is } \textit{inefficient!}$$

► Faced with conditional stability (6.1.77), then for the sake of efficiency
use *high-order spatial discretization* combined with *low order timestepping*.

However, this may not be easy to achieve

- because high-order timestepping is much simpler than high-order spatial discretization,
- because limited spatial smoothness of exact solution (\rightarrow results of Sect. 5.4 apply!) may impose a limit on q in (6.1.71).

Concretely: 5th-order `ode45` timestepping ($q = 5$) $\xrightarrow{\frac{1}{q} = \frac{2}{p}}$ use degree-10 Lagrangian FEM!

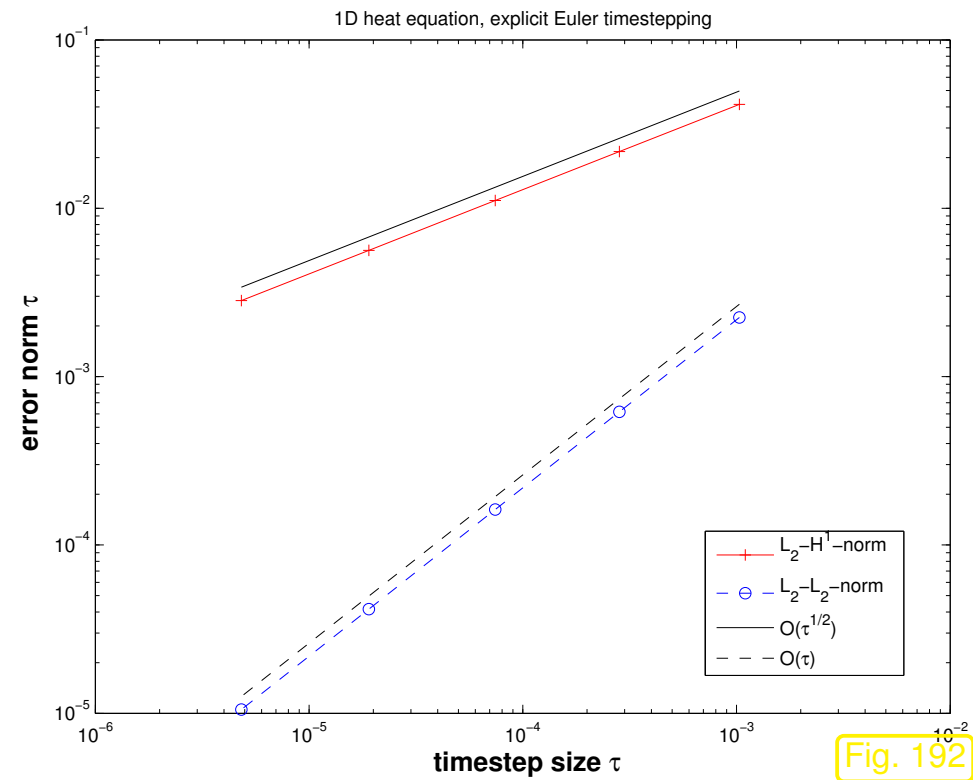
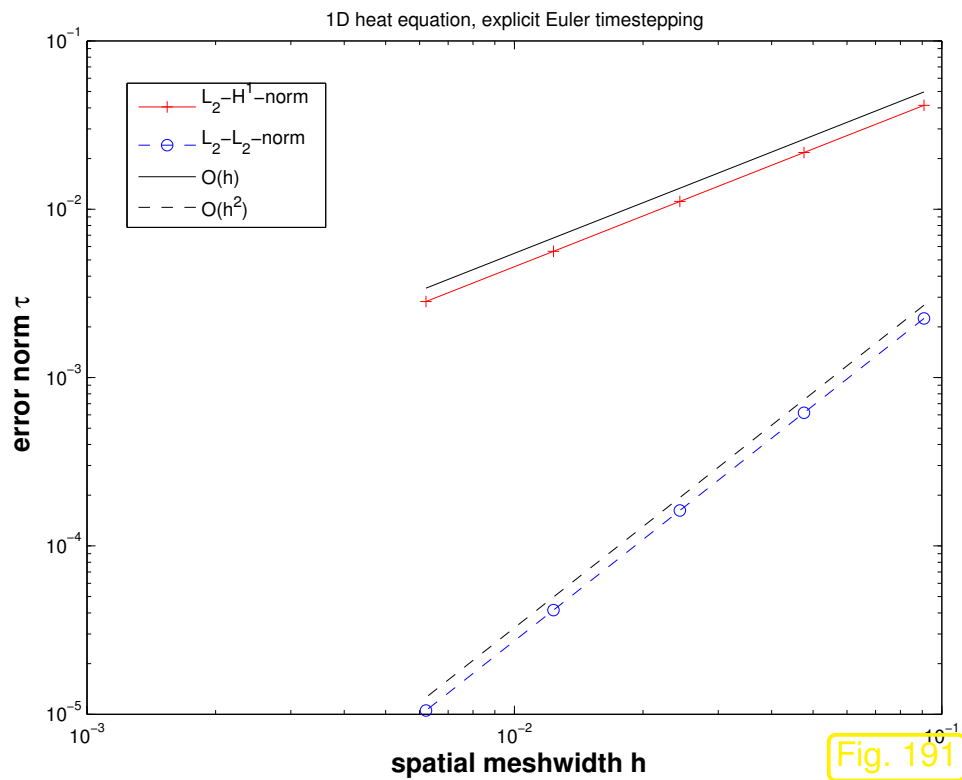
Parabolic IBVP of Ex. 6.1.67:

- $\frac{d}{dt}u - u'' = f(t, x)$ on $]0, 1[\times]0, 1[$
- exact solution $u(x, t) = (1 + t^2)e^{-\pi^2 t} \sin(\pi x)$, source term accordingly
- Linear finite element Galerkin discretization equidistant mesh, see Sect. 1.5.1.2, $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$
- piecewise linear spatial approximation of source term $f(x, t)$
- *explicit* Euler timestepping (6.1.29) with uniform timestep $\tau \sim h^2$ **close to the stability limit.**

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Monitored: error norms $\left(\tau \sum_{j=1}^M |u - u_N(\tau j)|_{H^1(]0,1])}^2 \right)^{\frac{1}{2}}$, $\left(\tau \sum_{j=1}^M \|u - u_N(\tau j)\|_{L^2(]0,1])}^2 \right)^{\frac{1}{2}}$.



In comparison with Ex. 6.1.67: degraded rate of convergence $O(\sqrt{\tau})$ for L^2-H^1 space-time norm.



Lemma 6.1.18 teaches that in the absence of time-dependent sources the rate of change of temperature will decay exponentially in the case of heat conduction.

Now we will encounter a class of evolution problems where temporal and spatial fluctuations will not be damped and will persist for good:

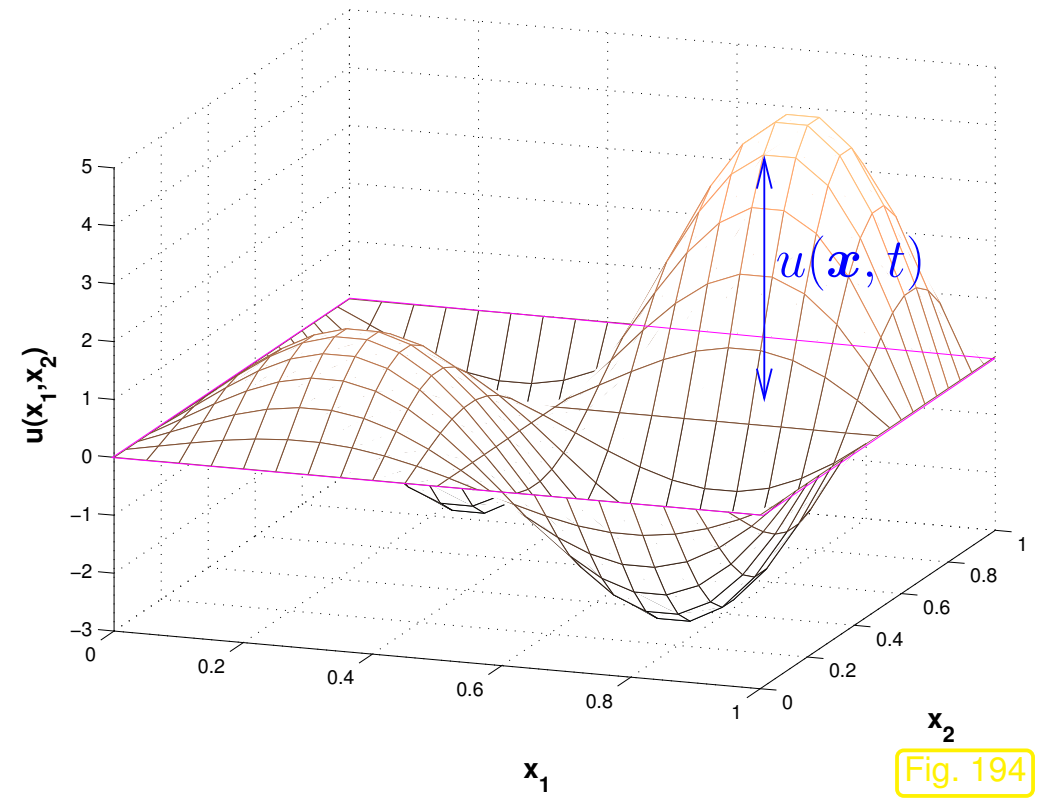
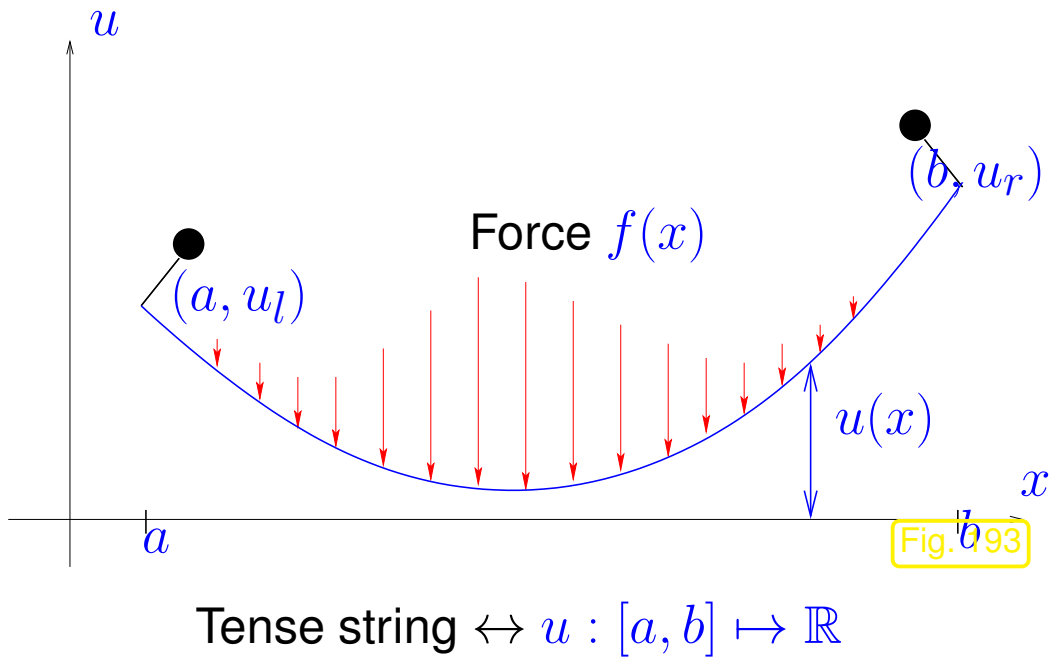
This will be the class of linear conservative wave propagation problems

As before these initial-boundary value problems (IBVP) will be posed on a space time cylinder $\tilde{\Omega} := \Omega \times]0, T[\subset \mathbb{R}^{d+1}$ (\rightarrow Fig. 184), where $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, is a bounded spatial domain as introduced in the context of elliptic boundary value problems, see Sect. 2.1.1.

The unknown will be a function $u = (\boldsymbol{x}, t) : \tilde{\Omega} \mapsto \mathbb{R}$.

Recall:

- Tense string model (\rightarrow Sect. 1.4), shape of string described by continuous displacement function $u : [a, b] \mapsto \mathbb{R}$, $u \in H^1([a, b])$.
- Taut membrane model (\rightarrow Sect. 2.1.1), shape of membrane given by displacement function $u : \Omega \mapsto \mathbb{R}$, $u \in H^1(\Omega)$, over base domain $\Omega \subset \mathbb{R}^2$.



R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

In Sect. 2.1.3 we introduced the general variational formulation: with Dirichlet data (elevation of frame) given by $g \in C^0(\partial\Omega)$,

$$V := \{v \in H^1(\Omega) : v|_{\partial\Omega} = g\}$$

we seek

$$u \in V: \int_{\Omega} \sigma(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x}, \quad \forall v \in H_0^1(\Omega), \quad (6.2.1)$$

where $f : \Omega \mapsto \mathbb{R} \hat{=} \text{density of vertical force}$,

$\sigma : \Omega \mapsto \mathbb{R} \hat{=} \text{uniformly positive stiffness coefficient (characteristic of material of the membrane)}$.

Now we switch to a *dynamic setting*: we allow variation of displacement with time, $u = u(\mathbf{x}, t)$, the membrane is allowed to vibrate.

Recall (secondary school): **Newton's second law of motion** (law of inertia)

$$F = m a \quad (6.2.2)$$
$$\text{force} = \text{mass} \cdot \text{acceleration} \quad (6.2.3)$$

Apply this in a local version (stated for densities) to membrane

$$\text{force density} \quad f(\mathbf{x}, t) = \rho(\mathbf{x}) \cdot \frac{\partial^2 u}{\partial t^2}(\mathbf{x}, t), \quad (6.2.4)$$

- where
- $\rho : \Omega \mapsto \mathbb{R}^+ \hat{=}$ uniformly positive **mass density** of membrane, $[\rho] = \text{kg m}^{-2}$,
 - $\ddot{u} := \frac{\partial^2 u}{\partial t^2} \hat{=}$ vertical acceleration (second temporal derivative of position).

Now, we assume that the force f in (2.3.4) is due to inertia forces only and express these using (6.2.4):

$$(2.3.4) \quad \blacktriangleright \quad (6.2.4) \quad \int_{\Omega} \sigma(\mathbf{x}) \mathbf{grad} u(\mathbf{x}, t) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} = - \int_{\Omega} \rho(\mathbf{x}) \cdot \frac{\partial^2 u}{\partial t^2}(\mathbf{x}, t) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) .$$

Why the “—”-sign? Because, here the inertia force enters as a **reaction** force.

\blacktriangleright **Linear wave equation** in variational form (Dirichlet boundary conditions):

$$u \in V(t): \quad \int_{\Omega} \rho(\mathbf{x}) \cdot \frac{\partial^2 u}{\partial t^2}(\mathbf{x}, t) v(\mathbf{x}) \, d\mathbf{x} + \int_{\Omega} \sigma(\mathbf{x}) \mathbf{grad} u(\mathbf{x}, t) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x} = 0 \quad \forall v \in H_0^1(\Omega) \quad (6.2.5)$$

mass density
stiffness

$$u \in V(t): \quad \mathbf{m}(\ddot{u}, v) + \mathbf{a}(u, v) = 0 \quad \forall v \in V_0 \quad (6.2.6)$$

where

$$V(t) := \{v :]0, T[\mapsto H^1(\Omega) : v(\mathbf{x}, t) = g(\mathbf{x}, t) \text{ for } \mathbf{x} \in \partial\Omega, 0 < t < T\}$$

(with continuous time-dependent Dirichlet data $g : \partial\Omega \times]0, T[\mapsto \mathbb{R}$.)

Undo integration by parts by reverse application of Green's first formula Thm. 2.4.11:

$$(6.2.5) \Rightarrow \int_{\Omega} \left\{ \frac{\partial^2 u}{\partial t^2}(\mathbf{x}, t) - \operatorname{div}_{\mathbf{x}}(\sigma(\mathbf{x}) \mathbf{grad}_{\mathbf{x}} u)(\mathbf{x}, t) \right\} v(\mathbf{x}) \, d\mathbf{x} = 0 \quad \forall v \in H_0^1(\Omega) . \quad (6.2.7)$$

Here it is indicated that the differential operators **grad** and **div** act on the spatial independent variable \mathbf{x} only. This will tacitly be assumed below.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

Now appeal to the fundamental lemma of calculus of variations in higher dimensions Lemma 2.4.15.

$$(6.2.7) \xrightarrow{\text{Lemma 2.4.15}} \frac{\partial^2 u}{\partial t^2} - \operatorname{div}(\sigma(\mathbf{x}) \mathbf{grad} u) = 0 \quad \text{in } \tilde{\Omega} . \quad (6.2.8)$$

(6.2.8) is called a (homogeneous) **wave equation**. A general wave equation is obtained, when an addition exciting vertical force density $f = f(\mathbf{x}, t)$ comes into play:

$$\frac{\partial^2 u}{\partial t^2} - \operatorname{div}(\sigma(\mathbf{x}) \mathbf{grad} u) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega} . \quad (6.2.9)$$

SAM, ETHZ

The wave equations (6.2.8), (6.2.9) have to be supplemented by

- **spatial boundary conditions:** $v(\mathbf{x}, t) = g(\mathbf{x}, t)$ for $\mathbf{x} \in \partial\Omega$, $0 < t < T$,
- **two initial conditions**

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad , \quad \frac{\partial u}{\partial t}(\mathbf{x}, 0) = v_0 \quad \text{for } \mathbf{x} \in \Omega \quad ,$$

with initial data $u_0, v_0 \in H^1(\Omega)$, satisfying the compatibility conditions $u_0(\mathbf{x}) = g(\mathbf{x}, 0)$ for $\mathbf{x} \in \partial\Omega$.

(6.2.8) & boundary conditions & initial conditions = **hyperbolic evolution problem**

Hey, why do we need **two** initial conditions in contrast to the heat equation?

Remember that

- (6.2.8) is a **second-order equation** also in time (whereas the heat equation is merely first-order),

- for second order ODEs $\ddot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$ we need **two** initial conditions

$$\mathbf{y}(0) = \mathbf{y}_0 \quad \text{and} \quad \dot{\mathbf{y}}(0) = \mathbf{v}_0, \quad (6.2.10)$$

in order to get a well-posed initial value problem, see [21, Rem. 12.1.15].

The physical meaning of the initial conditions (6.2.10) in the case of the membrane model is

- $u_0 \hat{=}$ initial displacement of membrane, $u_0 \in H^1(\Omega)$ “continuous”,
- $v_0 \hat{=}$ initial vertical velocity of membrane.

Remark 6.2.11 (Boundary conditions for wave equation).

Rem. 6.1.7 also applies to the wave equation (6.2.8):

On $\partial\Omega \times]0, T[$ we can impose any of the boundary conditions discussed in Sect. 2.6:

- Dirichlet boundary conditions $u(\mathbf{x}, t) = g(\mathbf{x}, t)$ (membrane attached to frame),
- Neumann boundary conditions $\mathbf{j}(\mathbf{x}, t) \cdot \mathbf{n} = 0$ (free boundary, Rem. 2.4.24)
- radiation boundary conditions $\mathbf{j}(\mathbf{x}, t) \cdot \mathbf{n} = \Psi(u(\mathbf{x}, t))$,

and any combination of these as discussed in Ex. 2.6.7, yet, *only one* of them at any part of $\partial\Omega \times]0, T[$, see Rem. 2.6.6.



Remark 6.2.12 (Wave equation as first order system in time).

Usual procedure [21, Rem. 12.1.15]: higher-order ODE can be converted into first-order ODEs by introducing derivatives as additional solution components. This approach also works for the second-order (in time) wave equation (6.2.8):

Additional unknown:

$$\text{velocity} \quad v(\mathbf{x}, t) = \frac{\partial u}{\partial t}(\mathbf{x}, t)$$

$$\frac{\partial^2 u}{\partial t^2} - \operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u) = 0 \quad \blacktriangleright \quad \begin{cases} \dot{u} = v, \\ \dot{v} = \operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u) \end{cases} \quad \text{in } \tilde{\Omega} \quad (6.2.13)$$

with initial conditions

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad , \quad v(\mathbf{x}, 0) = v_0(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega . \quad (6.2.14)$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

6.2.2 Wave propagation

Constant coefficient wave equation for $d = 1$, $\Omega = \mathbb{R}$ (“Cauchy problem”)

$$c > 0: \quad \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad , \quad u(x, 0) = u_0(x) \quad , \quad \frac{\partial u}{\partial t}(x, 0) = v_0(x) \quad , \quad x \in \mathbb{R} . \quad (6.2.15)$$

Change of variables: $\xi = x + ct$, $\tau = x - ct$: $\tilde{u}(\xi, \tau) := u\left(\frac{\xi+\tau}{2}, \frac{\xi-\tau}{2c}\right)$. Applying the chain rule we immediately see

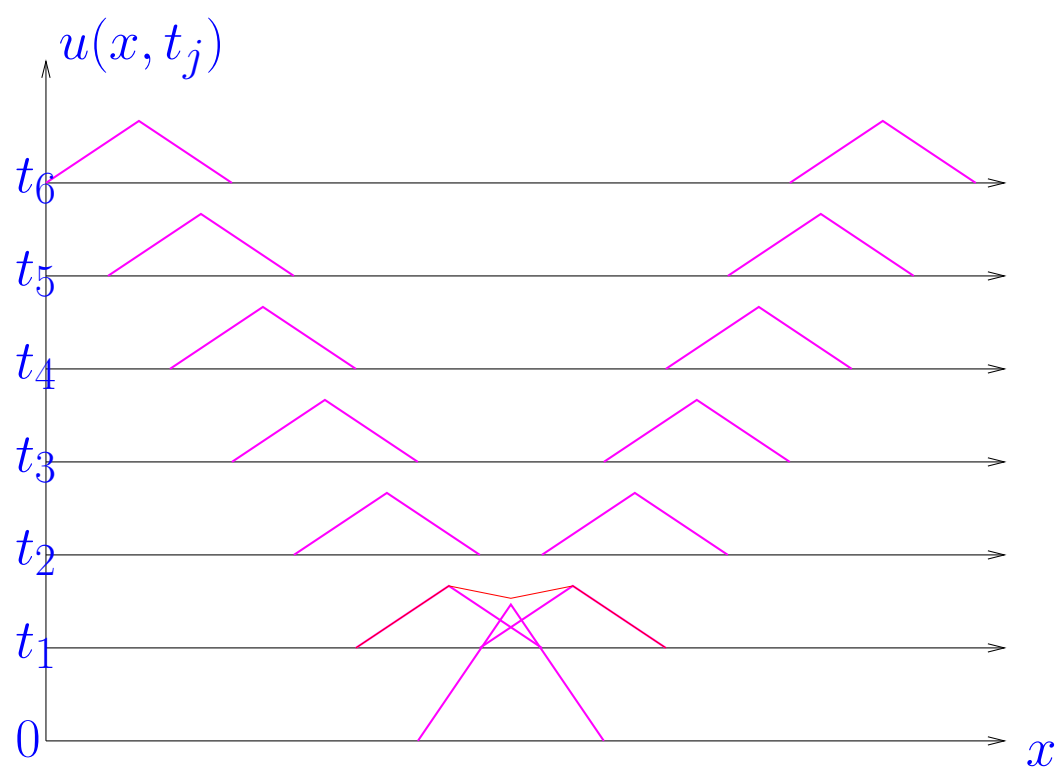
$$u \text{ satisfies (6.2.15)} \quad \blacktriangleright \quad \frac{\partial^2 \tilde{u}}{\partial \xi \partial \tau} = 0 \quad \Rightarrow \quad \tilde{u}(\xi, \tau) = F(\xi) + G(\tau),$$

for **any** $F, G \in C^2(\mathbb{R})$!

 ← matching initial data

$$u(x, t) = \frac{1}{2}(u_0(x + ct) + u_0(x - ct)) + \frac{1}{2} \int_{x-ct}^{x+ct} v_0(s) ds. \quad (6.2.16)$$

(6.2.16) = d'Alembert solution of Cauchy problem (6.2.15).



$v_0 = 0$ ➤ initial data u_0 travel with speed c in opposite directions

finite speed of propagation is typical feature of solutions of wave equations

Note: (6.2.16) meaningful even for discontinuous u_0, v_0 !
 ➤ “generalized solutions” !

R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

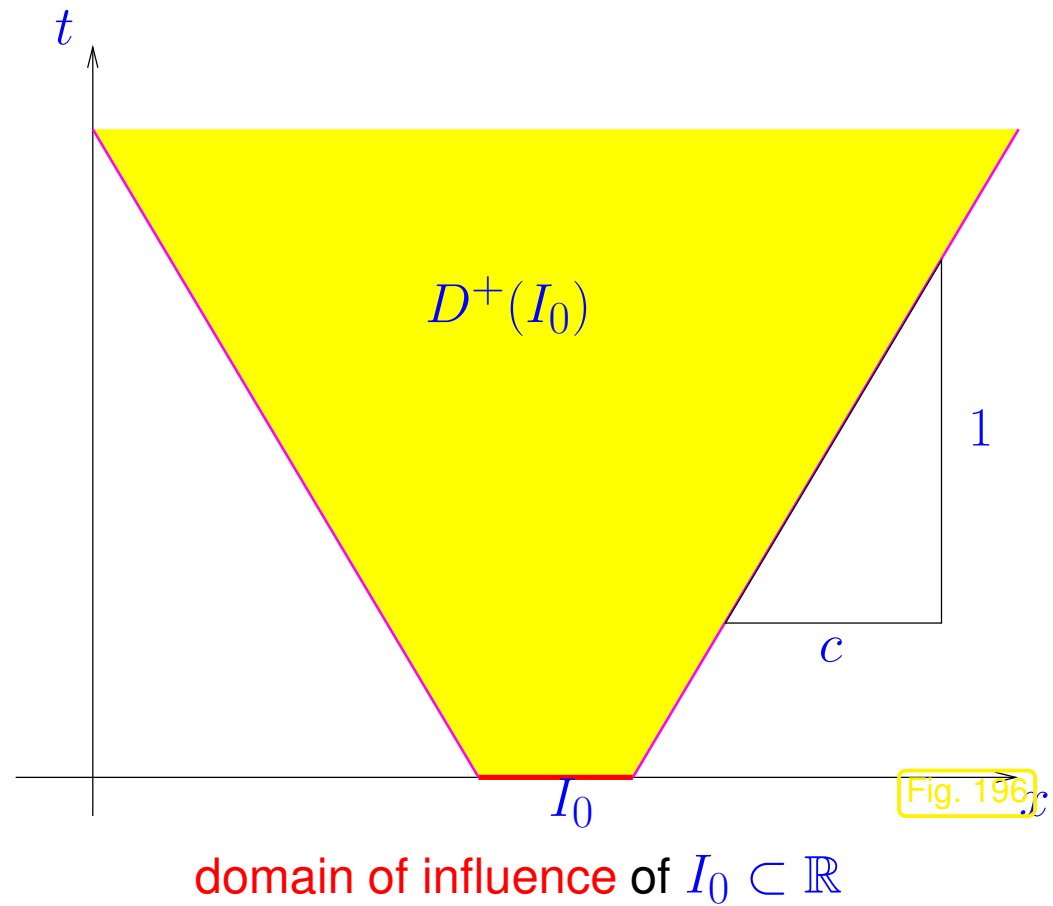
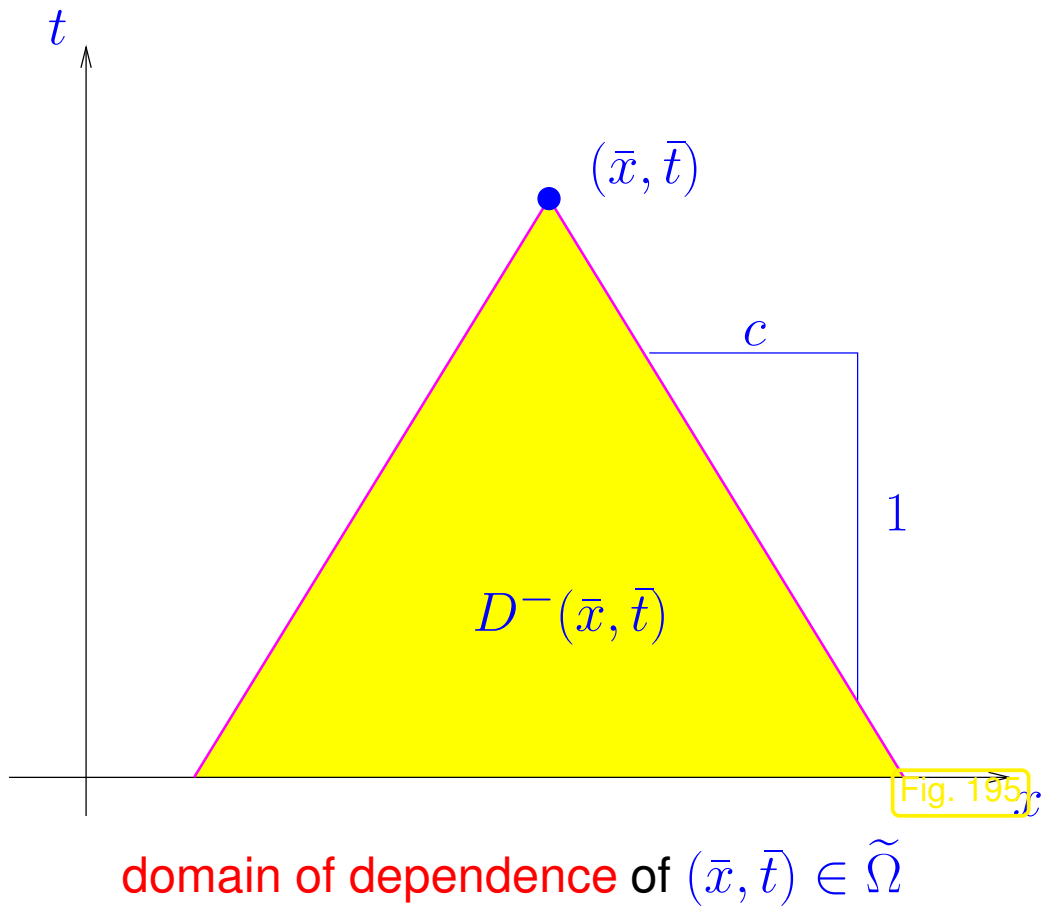
finite speed of propagation ➤ “point value” $u(\bar{x}, \bar{t}), (\bar{x}, \bar{t}) \in \tilde{\Omega}$, may not depend on initial values outside proper subdomain of Ω !

Example 6.2.17 (Domain of dependence/influence for 1D wave equation, constant coefficient case).

Consider $d = 1$, initial-boundary value problem (6.2.15) for wave equation:

$$c > 0: \quad \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad , \quad u(x, 0) = u_0(x) \quad , \quad \frac{\partial u}{\partial t}(x, 0) = v_0(x) \quad , \quad x \in \mathbb{R} . \quad (6.2.15)$$

Intuitive: from D'Alembert formula (6.2.16)



Domain of dependence: the value of the solution in (\bar{x}, \bar{t}) (●) will depend only on data in the yellow triangle in Fig. 195.

Domain of influence: initial data in I_0 will be relevant for the solution only in the yellow triangle in Fig. 196.



Theorem 6.2.18 (Domain of dependence for isotropic wave equation). $\rightarrow [15, 2.5, \text{Thm. 6}]$

Let $u : \tilde{\Omega} \mapsto \mathbb{R}$ be a (classical) solution of $\frac{\partial^2 u}{\partial t^2} - c\Delta u = 0$. Then

$$\left(|\mathbf{x} - \mathbf{x}_0| \geq R \Rightarrow \begin{array}{l} u(\mathbf{x}, 0) = 0, \\ \frac{\partial u}{\partial t}(\mathbf{x}, 0) = 0 \end{array} \right) \Rightarrow u(\mathbf{x}, t) = 0, \text{ if } |\mathbf{x} - \mathbf{x}_0| \geq R + ct.$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

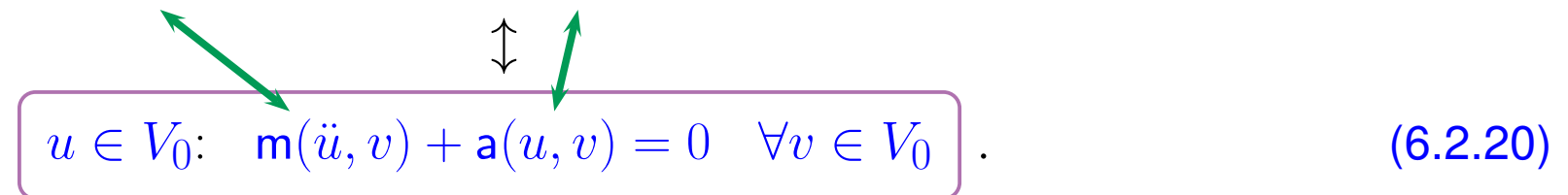
SAM, ETHZ

The solution formula (6.2.16) clearly indicates that in 1D and in the absence of boundary conditions the solution of the wave equation will persist undamped for all times.

This absence of damping corresponds to a *conservation of total energy*, which is a distinguishing feature of conservative wave propagation phenomena.

Now, we examine this for the model problem

$$u \in H_0^1(\Omega): \int_{\Omega} \rho(\mathbf{x}) \cdot \frac{\partial^2 u}{\partial t^2} v \, d\mathbf{x} + \int_{\Omega} \sigma(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = 0 \quad \forall v \in H_0^1(\Omega) \quad (6.2.19)$$

$$u \in V_0: \mathbf{m}(\ddot{u}, v) + \mathbf{a}(u, v) = 0 \quad \forall v \in V_0 \quad (6.2.20)$$


R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Here we do not include the case of non-homogeneous spatial Dirichlet boundary conditions through an affine trial space. This can always be taken into account by offset functions, see the remark after (6.1.5).

Theorem 6.2.21 (Energy conservation in wave propagation).

If $u : \tilde{\Omega} \mapsto \mathbb{R}$ solves (6.2.20), then

$$t \mapsto \frac{1}{2}m\left(\frac{\partial u}{\partial t}, \frac{\partial u}{\partial t}\right) + \frac{1}{2}a(u, u) \equiv \text{const} .$$

kinetic energy
elastic (potential) energy, see (2.1.3)

Proof. A “formal proof” boils down to a straightforward application of the product rule (\rightarrow Rem. 6.1.13) together with the symmetry of the bilinear forms m and a .

Introduce the **total energy**

$$E(t) := \frac{1}{2}m\left(\frac{\partial u}{\partial t}, \frac{\partial u}{\partial t}\right) + \frac{1}{2}a(u, u) .$$

▶ $\frac{dE}{dt}(t) = m(\ddot{u}, \dot{u}) + a(\dot{u}, u) = 0$ for solution u of (6.2.20) ,

because this is what we conclude from (6.2.20) for the special test function $v(\mathbf{x}) = \dot{u}(\mathbf{x}, t)$ for any $t \in]0, T[$. □

6.2.3 Method of lines

The method of lines approach to the wave equation (6.2.19), (6.2.20) is exactly the same as for the heat equation, see Sect. 6.1.3.

Idea: Apply **Galerkin discretization** (\rightarrow Sect. 3.1) to abstract linear parabolic variational problem (6.1.11).

$$t \in]0, T[\mapsto u(t) \in V_0 \quad : \quad \begin{cases} m\left(\frac{d^2 u}{dt^2}(t), v\right) + a(u(t), v) = 0 \quad \forall v \in V_0, \\ u(0) = u_0 \in V_0 \quad , \quad \frac{du}{dt}(0) = v_0 \in V_0. \end{cases} \quad (6.2.22)$$

1st step: replace V_0 with a finite dimensional subspace $V_{0,N}$, $N := \dim V_{0,N} < \infty$

► Discrete hyperbolic evolution problem

$$t \in]0, T[\mapsto u(t) \in V_{0,N} \quad : \quad \begin{cases} m\left(\frac{d^2 u_N}{dt^2}(t), v_N\right) + a(u_N(t), v_N) = 0 \quad \forall v_N \in V_{0,N}, \\ u_N(0) = \text{projection/interpolant of } u_0 \text{ in } V_{0,N}, \\ \frac{du_N}{dt}(0) = \text{projection/interpolant of } v_0 \text{ in } V_{0,N}. \end{cases} \quad (6.2.23)$$

2nd step: introduce (ordered) basis $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ of $V_{0,N}$

$$(6.2.23) \quad \Rightarrow \quad \begin{cases} \mathbf{M} \left\{ \frac{d^2}{dt^2} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = 0 & \text{for } 0 < t < T, \\ \vec{\mu}(0) = \vec{\mu}_0, \quad \frac{d\vec{\mu}}{dt}(0) = \vec{\nu}_0. \end{cases} \quad (6.2.24)$$

- ▷ s.p.d. stiffness matrix $\mathbf{A} \in \mathbb{R}^{N,N}$, $(\mathbf{A})_{ij} := \mathbf{a}(b_N^j, b_N^i)$ (independent of time),
- ▷ s.p.d. **mass matrix** $\mathbf{M} \in \mathbb{R}^{N,N}$, $(\mathbf{M})_{ij} := \mathbf{m}(b_N^j, b_N^i)$ (independent of time),
- ▷ source (load) vector $\vec{\varphi}(t) \in \mathbb{R}^N$, $(\vec{\varphi}(t))_i := \ell(t)(b_N^i)$ (time-dependent),
- ▷ $\vec{\mu}_0 \hat{=}$ coefficient vector of a projection of u_0 onto $V_{0,N}$.
- ▷ $\vec{\nu}_0 \hat{=}$ coefficient vector of a projection of v_0 onto $V_{0,N}$.

Note:

(6.2.24) is a 2nd-order ordinary differential equation (ODE) for $t \mapsto \vec{\mu}(t) \in \mathbb{R}^N$

Remark 6.2.25 (First-order semidiscrete hyperbolic evolution problem).

Completely analogous to Rem. 6.2.12:

$$\begin{aligned}
 & \mathbf{M} \left\{ \frac{d^2}{dt^2} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = 0 \\
 & \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \leftarrow \text{auxiliary unknown } \nu = \dot{\mu} \\
 & \left\{ \begin{array}{l} \frac{d}{dt} \vec{\mu}(t) = \vec{\nu}(t) , \\ \mathbf{M} \frac{d}{dt} \vec{\nu}(t) = -\mathbf{A} \vec{\mu}(t) , \end{array} \right. , \quad 0 < t < T .
 \end{aligned}
 \tag{6.2.26}$$

with initial conditions

$$\vec{\mu}(0) = \vec{\mu}_0 \quad , \quad \vec{\nu}(0) = \vec{\nu}_0 .
 \tag{6.2.27}$$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

6.2.4 Timestepping

The method of lines approach gives us the semi-discrete hyperbolic evolution problem = 2nd-order ODE:

$$\mathbf{M} \left\{ \frac{d^2}{dt^2} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = 0 \quad , \quad \vec{\mu}(0) = \vec{\mu}_0 \quad , \quad \frac{d\vec{\mu}}{dt}(0) = \vec{\eta}_0 \quad . \quad (6.2.28)$$

Key features of (6.2.28) \Rightarrow to be respected “approximately” by timestepping:

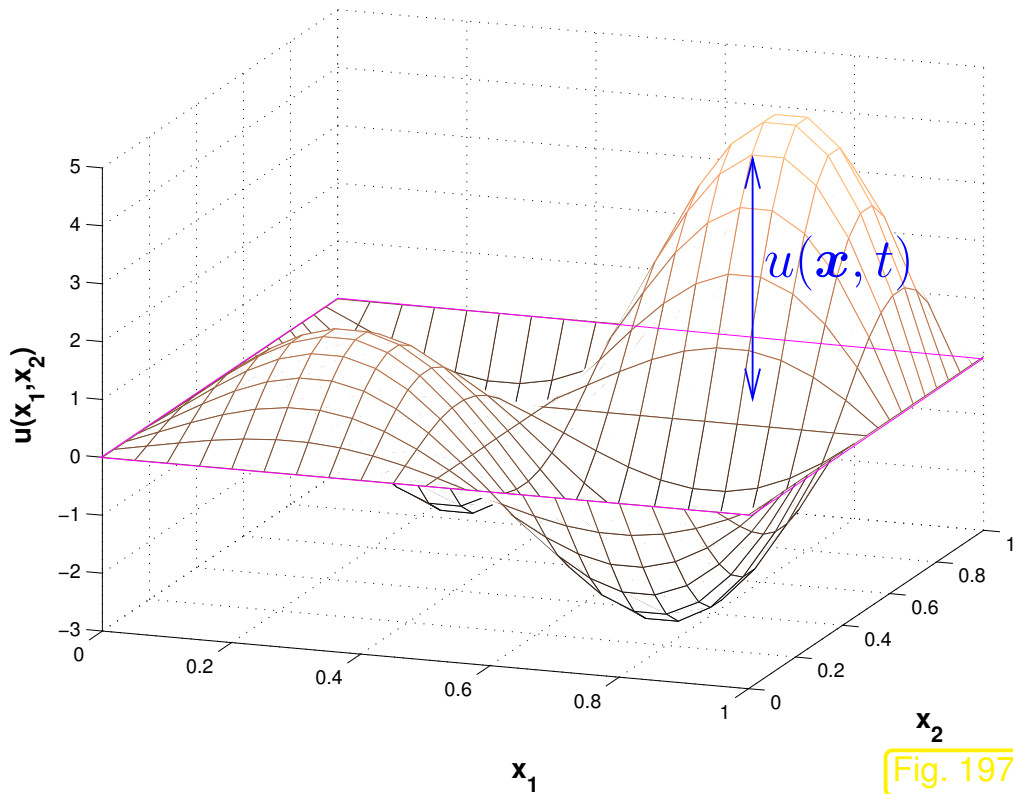
- **reversibility:** (6.2.28) invariant under time-reversal $t \leftarrow -t$

- **energy conservation**, cf. Thm. 6.2.21:
$$E_N(t) := \frac{1}{2} \frac{d\vec{\mu}}{dt} \cdot \mathbf{M} \frac{d\vec{\mu}}{dt} + \frac{1}{2} \vec{\mu} \cdot \mathbf{A} \vec{\mu} = \text{const}$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Example 6.2.29 (Euler timestepping for 1st-order form of semi-discrete wave equation).



Model problem: wave propagation on a square membrane

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} - \Delta u &= 0 \quad \text{on }]0, 1[^2 \times]0, 1[, \\ u(\mathbf{x}, t) &= 0 \quad \text{on } \partial\Omega \times]0, T[, \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) \quad , \quad \frac{\partial u}{\partial t}(\mathbf{x}, 0) = 0 . \end{aligned}$$

- Initial data $u_0(\mathbf{x}) = \max\{0, \frac{1}{5} - \|\mathbf{x}\|\}$, $v_0(\mathbf{x}) = 0$,
- $\mathcal{M} \hat{=}$ “structured triangular tensor product mesh”, see Fig. 124, n squares in each direction,
- linear finite element space $V_{N,0} = \mathcal{S}_{1,0}^0(\mathcal{M})$, $N := \dim \mathcal{S}_{1,0}^0(\mathcal{M}) = (n - 1)^2$,
- All local computations (\rightarrow Sect. 3.5.4) rely on 3-point vertex based local quadrature formula “2D trapezoidal rule” (3.2.18). More explanations will be given in Rem. 6.2.34 below.

• $\mathbf{A} = N \times N$ Poisson matrix, see (4.1.3), scaled with $h := n^{-1}$,

• mass matrix $\mathbf{M} = h\mathbf{I}$, thanks to quadrature formula, see Rem. 6.2.34.

Timestepping: implicit and explicit Euler method (\rightarrow Ex. 6.1.28, [21, Sect. 12.2]) for 1st-order ODE (6.2.26), timestep $\tau > 0$:

$$\begin{aligned} \vec{\mu}^{(j)} - \vec{\mu}^{(j-1)} &= \tau \vec{\nu}^{(j-1)}, \\ \mathbf{M}(\vec{\nu}^{(j)} - \vec{\nu}^{(j-1)}) &= -\tau \mathbf{A} \vec{\mu}^{(j-1)}. \end{aligned}$$

explicit Euler

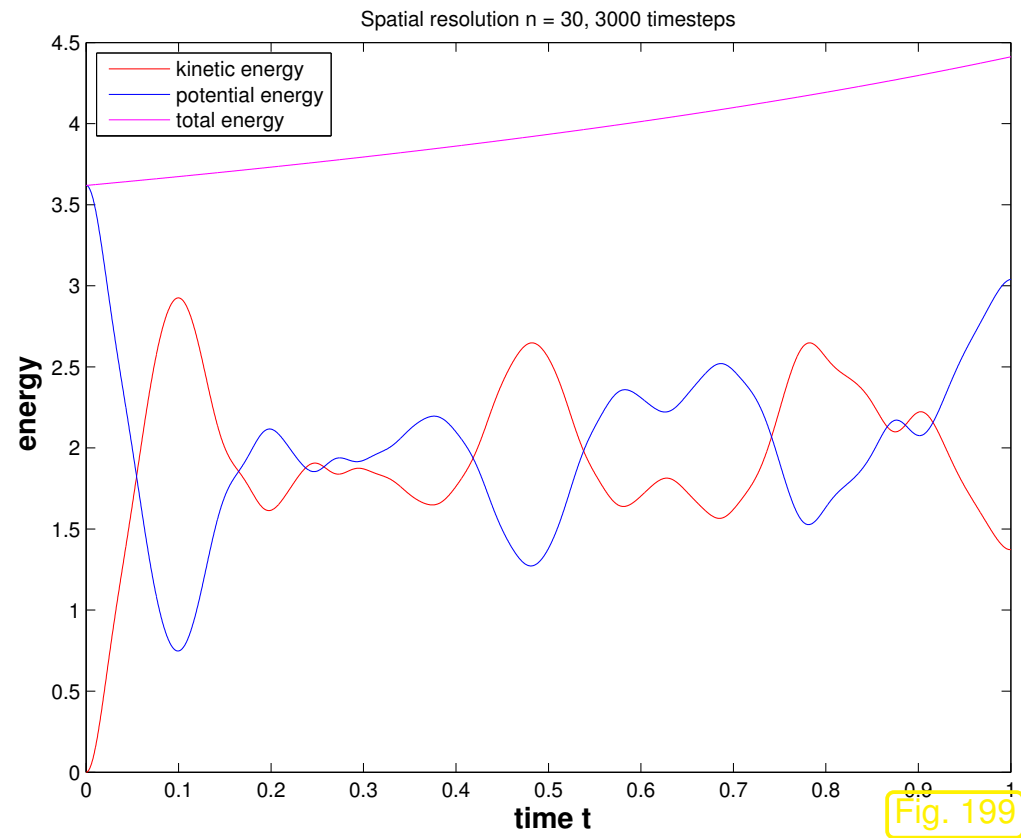
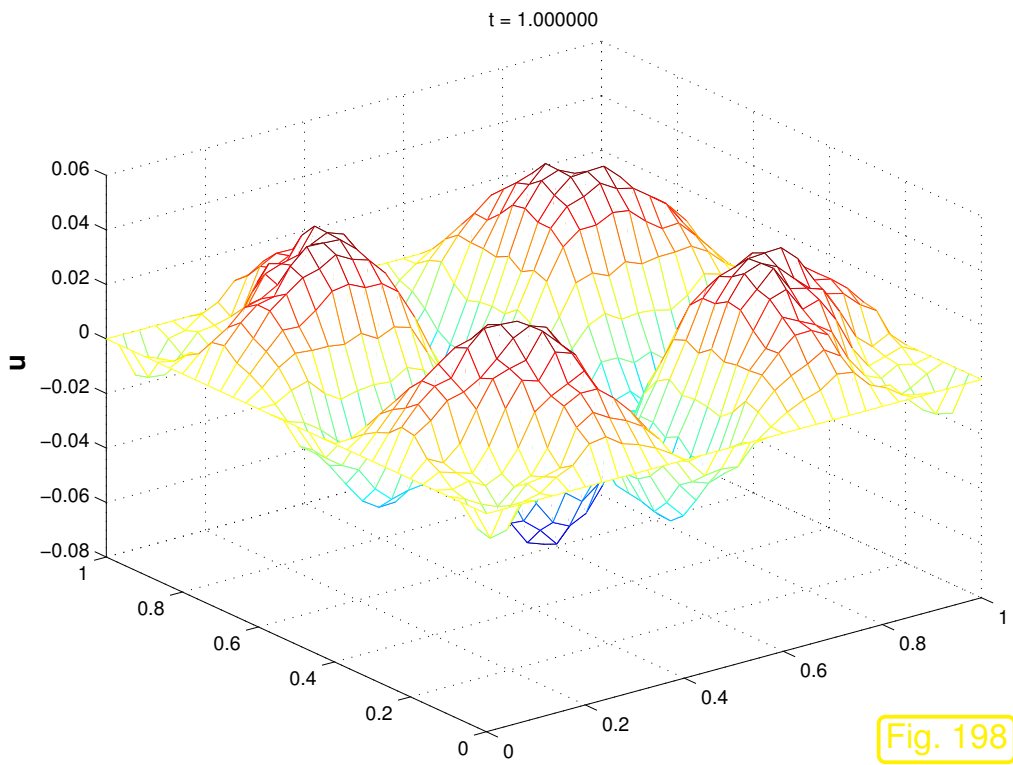
$$\begin{aligned} \vec{\mu}^{(j)} - \vec{\mu}^{(j-1)} &= \tau \vec{\nu}^{(j)}, \\ \mathbf{M}(\vec{\nu}^{(j)} - \vec{\nu}^{(j-1)}) &= -\tau \mathbf{A} \vec{\mu}^{(j)}. \end{aligned}$$

implicit Euler

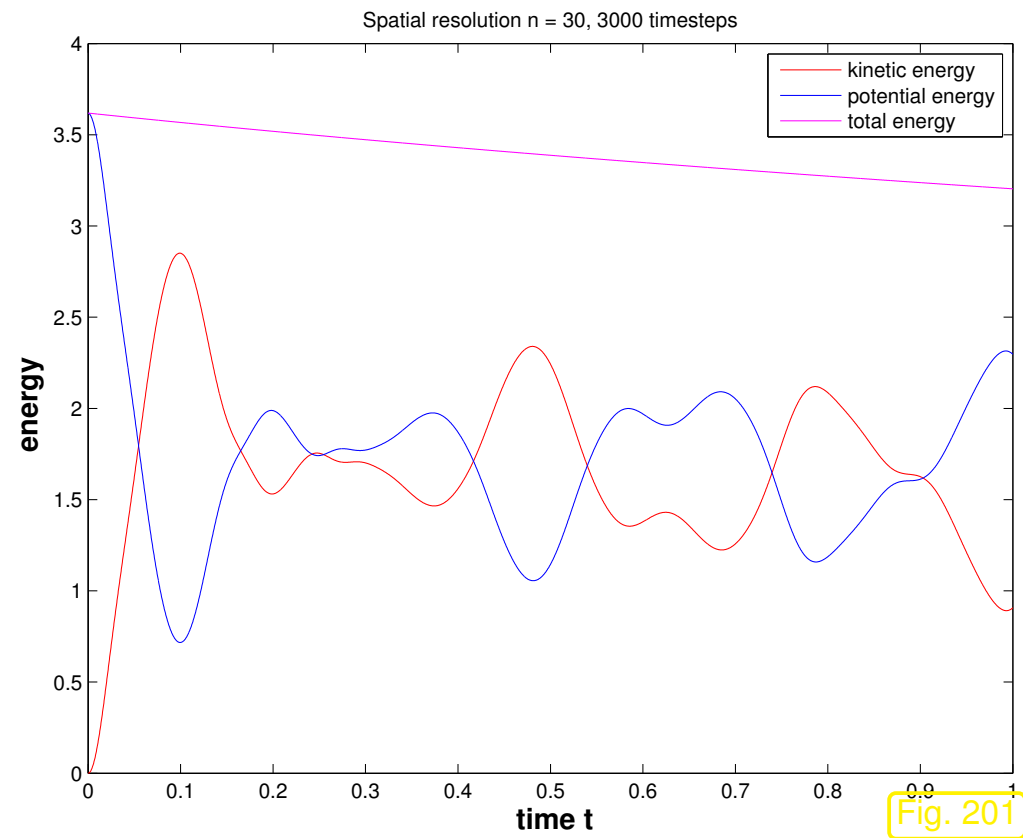
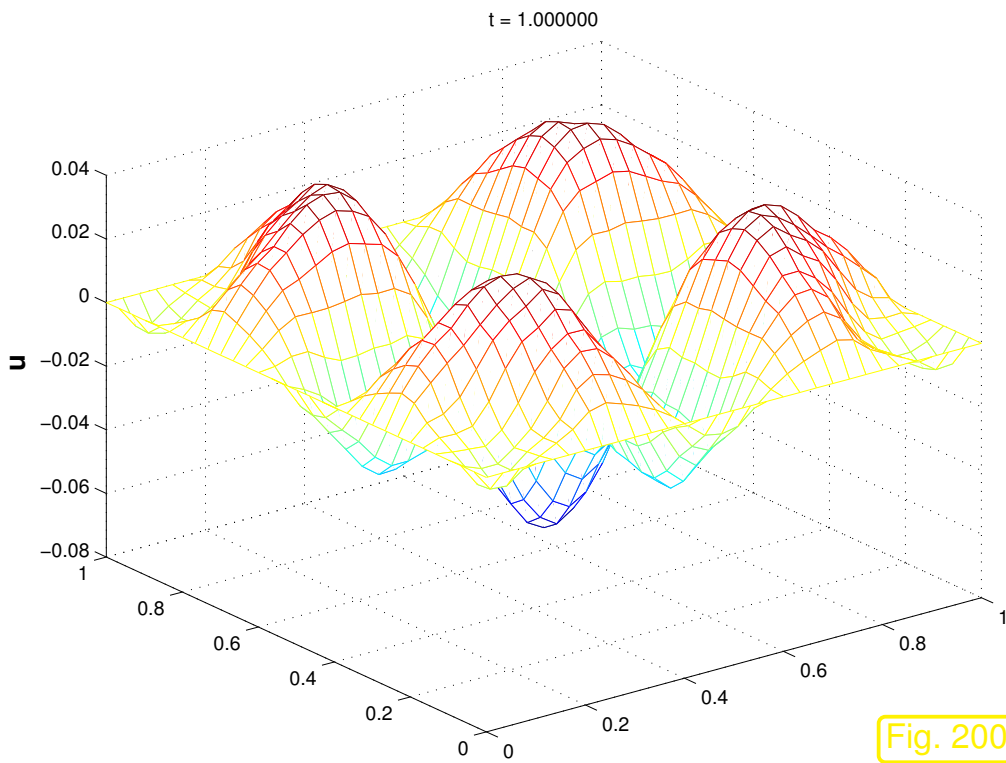
Monitored: behavior of (discrete) kinetic, potential, and total **energy**

$$E_{\text{kin}}^{(j)} = (\vec{\nu}^{(j)})^T \mathbf{M} \vec{\nu}^{(j)}, \quad E_{\text{pot}}^{(j)} = (\vec{\mu}^{(j)})^T \mathbf{A} \vec{\mu}^{(j)}, \quad j = 0, 1, \dots$$

Explicit Euler timestepping:



Implicit Euler timestepping:



Observation: neither method conserves energy,

☞ explicit Euler timestepping ➤ steady increase of total energy

☞ implicit Euler timestepping ➤ steady decrease of total energy

Ex. 6.2.29 ➤ Euler methods violate energy conservation!

(The same is true of all explicit Runge-Kutta methods, which lead to an increase of the total energy over time, and L(π)-stable implicit Runge-Kutta method, which make the total energy decay.)

Let us try another simple idea for the 2nd-order ODE (6.2.24):

Replace $\frac{d^2}{dt^2}\vec{\mu}$ with symmetric difference quotient (1.5.137)

$$\mathbf{M} \left\{ \frac{d^2}{dt^2} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = 0 \quad (6.2.28)$$

$$\mathbf{M} \frac{\vec{\mu}^{(j+1)} - 2\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)}}{\tau^2} = -\mathbf{A} \vec{\mu}^{(j)}, \quad j = 0, 1, \dots \quad (6.2.30)$$

This is a **two-step method**, the **Störmer scheme/explicit trapezoidal rule**

By Taylor expansion:

Störmer scheme is a **2nd-order** method

However, from where do we get $\vec{\mu}^{(-1)}$? Two-step methods need to be kick-started by a *special initial step*: This is constructed by approximating the second initial condition by a symmetric difference quotient:

$$\frac{d}{dt}\vec{\mu}(0) = \vec{\nu}_0 \quad \blacktriangleright \quad \frac{\vec{\mu}^{(1)} - \vec{\mu}^{(-1)}}{2\tau} = \vec{\nu}_0 . \quad (6.2.31)$$

Example 6.2.32 (Leapfrog timestepping).

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

For the semi-discrete wave equation we again consider the explicit trapezoidal rule (Störmer scheme):

$$\mathbf{M} \frac{\vec{\mu}^{(j+1)} - 2\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)}}{\tau^2} = -\mathbf{A}\vec{\mu}^{(j)}, \quad j = 1, \dots . \quad (6.2.30)$$

Inspired by Rem. 6.2.25 we introduce the auxiliary variable

$$\vec{\nu}^{(j+1/2)} := \frac{\vec{\mu}^{(j+1)} - \vec{\mu}^{(j)}}{\tau},$$

which can be read as an approximation of the velocity $v := \dot{u}$.

This leads to a timestepping scheme, which is *algebraically equivalent* to the explicit trapezoidal rule:

leapfrog timestepping (with uniform timestep $\tau > 0$):

$$\mathbf{M} \frac{\vec{v}^{(j+\frac{1}{2})} - \vec{v}^{(j-\frac{1}{2})}}{\tau} = -\mathbf{A}\vec{\mu}^{(j)}, \quad j = 0, 1, \dots, \quad (6.2.33)$$

$$\frac{\vec{\mu}^{(j+1)} - \vec{\mu}^{(j)}}{\tau} = \vec{v}^{(j+\frac{1}{2})},$$

+ initial step $\vec{v}^{(-\frac{1}{2})} + \vec{v}^{(\frac{1}{2})} = 2\vec{v}_0$.

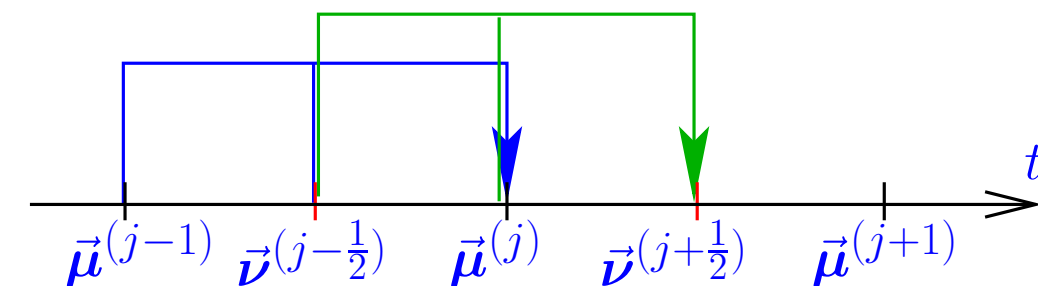
R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

work per step:

1 × evaluation $\mathbf{A} \times$ vector,

1 × solution of linear system for \mathbf{M}



Remark 6.2.34 (Mass lumping).

Required in each step of leapfrog timestepping: solution of linear system of equations with (large sparse) system matrix $\mathbf{M} \in \mathbb{R}^{N,N}$ \triangleright **expensive!**

Trick for (bi-)linear finite element Galerkin discretization: $V_{0,N} \subset \mathcal{S}_1^0(\mathcal{M})$:

use *vertex based local quadrature rule*

(e.g. “2D trapezoidal rule” (3.2.18) on triangular mesh)

$$\int_K f(\mathbf{x}) \, d\mathbf{x} \approx \frac{|K|}{\#\mathcal{V}(K)} \sum_{\mathbf{p} \in \mathcal{V}(K)} f(\mathbf{p}), \quad \mathcal{V}(K) := \text{set of vertices of } K.$$

(For a comprehensive discussion of local quadrature rules see Sect. 3.5.4)

► Mass matrix \mathbf{M} will become a *diagonal* matrix (due to defining equation (3.2.4) for nodal basis functions, which are associated with nodes of the mesh).

This so-called mass lumping trick was used in the finite element discretization of Ex. 6.2.29.



Example 6.2.35 (Energy conservation for leapfrog).

Model problem and discretization as in Ex. 6.2.29.

Leapfrog timestepping with constant timestep size $\tau = 0.01$

Code 6.2.36: Computing behavior of energies for Störmer timestepping

```
1 function lfen(n,m)
2 % leapfrog timestepping for 2D wave equation, computation of energies
3 % n: spatial resolution (no. of cells in one direction)
4 % m: number of timesteps
5
6 % Assemble stiffness matrix, see Sect. 4.1, (4.1.3)
7 N = (n-1)^2; h = 1/n; A = gallery('poisson',n-1)/(h*h);
8
9 % initial displacement  $u_0(\mathbf{x}) = \max\{0, \frac{1}{5} - \|\mathbf{x}\|\}$ 
10 [X,Y] = meshgrid(0:h:1,0:h:1);
11 U0 = 0.2 - sqrt((X-0.5).^2 + (Y-0.5).^2);
```

```
12 U0 ( find (U0 < 0) ) = 0.0;
13 u0 = reshape (U0 (2:end-1, 2:end-1) , N, 1) ;
14 v0 = zeros (N, 1) ;   % initial velocity
15
16 % loop for Störmer timestepping, see (6.2.30)
17 tau = 1/m;           % uniform timestep size
18 u = u0+tau*v0-0.5*tau^2*A*u0; % special initial step
19 u_old = u0;
20 [pen,ken] = geten (A,tau,u0,u) ; % compute potential and kinetic energy
21 E = [0.5*tau,pen,ken,pen+ken];
22 for k=1:m-1
23     u_new = -(tau^2)*(A*u) + 2*u - u_old;
24     [pen,ken] = geten (A,tau,u,u_new) ;
25     E = [E; (k+0.5)*tau,pen,ken,pen+ken];
26     u_old = u; u = u_new;
27 end
28
29 figure ('name', 'Leapfrog energies') ;
30 plot (E (:, 1) , E (:, 3) , 'r-' , E (:, 1) , E (:, 2) , 'b-' , E (:, 1) , E (:, 4) , 'm-' ) ;
31 xlabel ('{\bf time t}' , 'fontsize' , 14) ;
32 ylabel ('{\bf energies}' , 'fontsize' , 14) ;
33 legend ('kinetic energy' , 'potential energy' , 'total
    energy' , 'location' , 'south' ) ;
```

```

34 title (sprintf ('Spatial resolution n = %i, %i timesteps', n, m) );
35
36 print ('-depsc', sprintf ('../../../../rw/Slides/NPDEpics/leapfrogend

```

Code 6.2.37: Computing potential and kinetic energy for Störmer timestepping

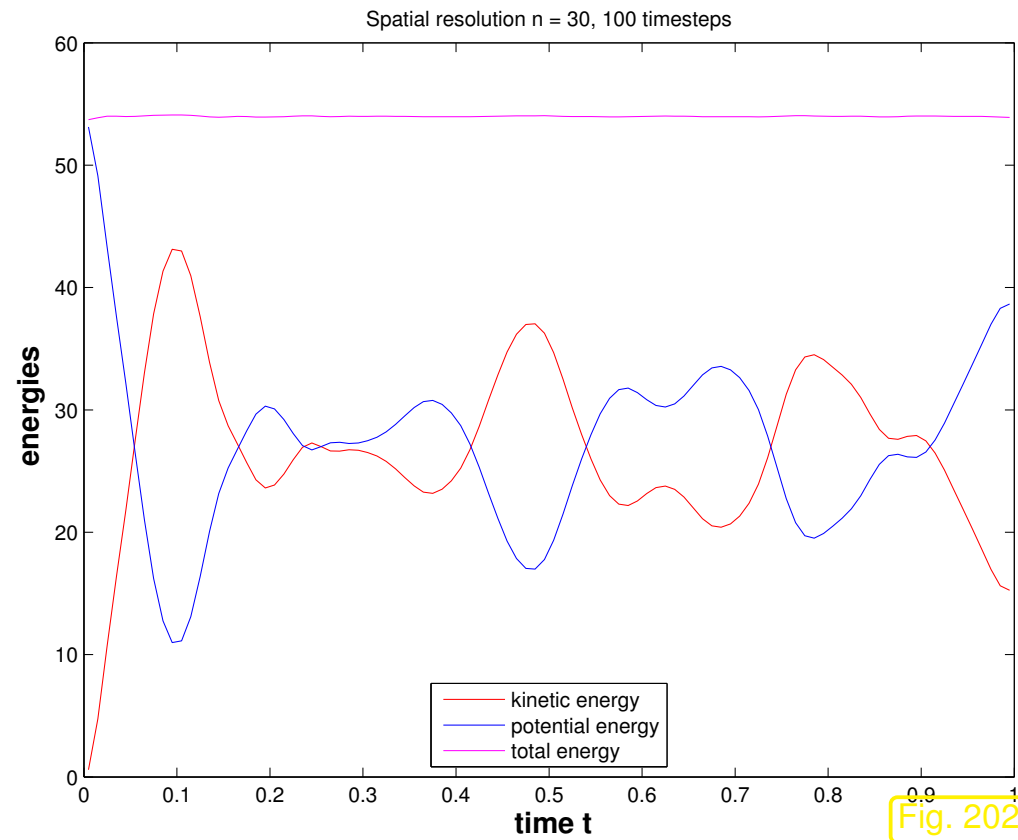
```

1 function [pen, ken] = geten(A, ts, u_old, u_new)
2 % Compute the current approximate potential and kinetic energies for u_old
3 % and u_new from Sörmer timestepping
4 %  $E_{\text{kin}}^{(j)} = \tau^{-2}(\vec{\mu}^{(j)} - \vec{\mu}^{(j-1)})^T \mathbf{M}(\vec{\mu}^{(j)} - \vec{\mu}^{(j-1)})$  ,  $E_{\text{pot}}^{(j)} = \frac{1}{4}(\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)})^T \mathbf{A}(\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)})$  ,  $j = 0, 1, \dots$ 
5 meanv = 0.5*(u_old+u_new); pen = dot(meanv, A*meanv); % potential
   energy
6 dtemp = (u_new-u_old)/ts; ken = dot(dtemp, dtemp); % kinetic energy

```

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Leapfrog is (nearly) energy conserving
(no energy drift, only small oscillations)

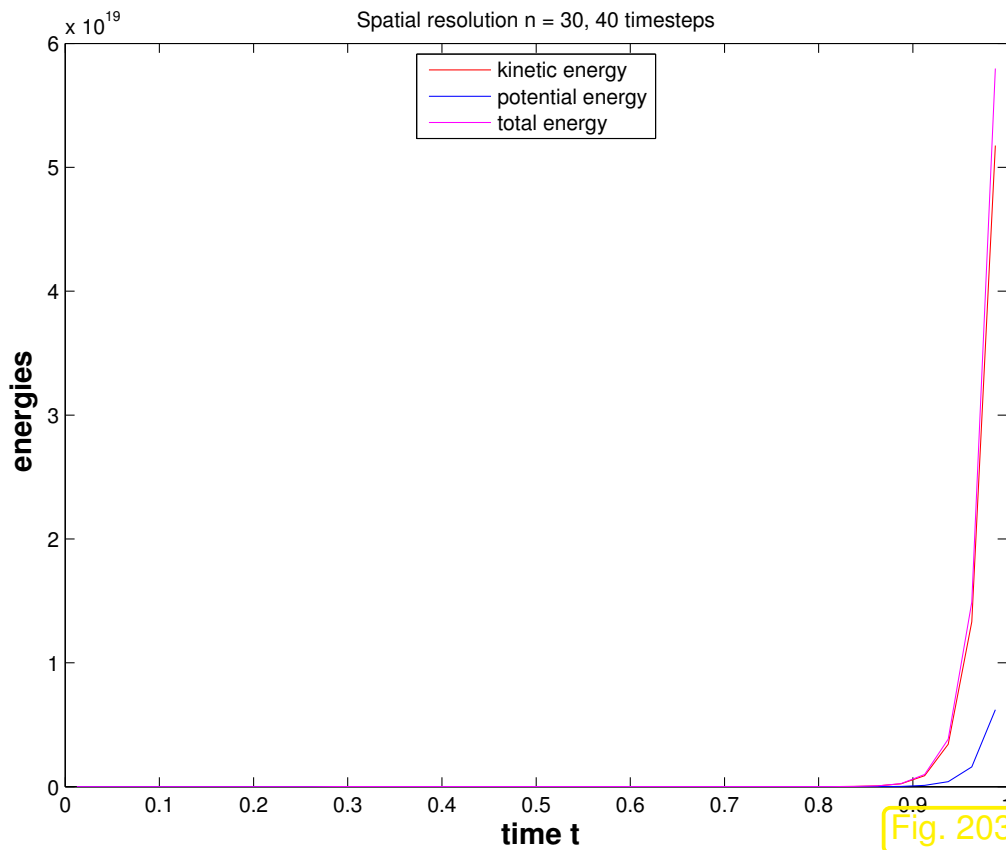
This behavior is explained by the deep mathematical theory of **symplectic integrators**, see [20].

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

6.2.5 CFL-condition

Example 6.2.38 (Blow-up for leapfrog timestepping).



◁ Ex. 6.2.35 repeated with $\tau = 0.04$

Observation:

Leapfrog suffers a **blow-up**: exponential increase of energies!

A similar behavior is observed with the explicit Euler scheme for the semi-discrete heat equation, in case the timestep constraint is violated, see Sect. 6.1.4.2.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



➤ (as in Sect. 6.1.4.2) Stability analysis of leapfrog timestepping based on **diagonalization**:

$$\exists \text{ orthogonal } \mathbf{T} \in \mathbb{R}^{N,N}: \quad \mathbf{T}^\top \mathbf{M}^{-1/2} \mathbf{A} \mathbf{M}^{-1/2} \mathbf{T} = \mathbf{D} := \text{diag}(\lambda_1, \dots, \lambda_N).$$

where the $\lambda_i > 0$ are *generalized eigenvalues* for $\mathbf{A}\vec{\xi} = \lambda\mathbf{M}\vec{\xi}$ ➤ $\lambda_i \geq \gamma$ for all i (γ is the constant introduced in (6.1.12)).

Next, apply transformation $\vec{\eta} := \mathbf{T}^T \mathbf{M}^{1/2} \vec{\mu}$ to the 2-step formulation (6.2.30)

$$(6.2.30) \quad \vec{\eta} := \mathbf{T}^T \mathbf{M}^{1/2} \vec{\mu} \quad \vec{\eta}^{(j+1)} - 2\vec{\eta}^{(j)} + \vec{\eta}^{(j-1)} = -\tau^2 \mathbf{D} \vec{\eta}^{(j)} .$$

Again, we have achieved a complete decoupling of the timestepping for the eigencomponents.

$$\eta_i^{(j+1)} - 2\eta_i^{(j)} + \eta_i^{(j-1)} = -\tau^2 \lambda_i \eta_i^{(j)} , \quad i = 1, \dots, N , \quad j = 1, 2, \dots . \quad (6.2.39)$$

In fact, (6.2.39) is what we end up with then applying Störmer's scheme to the *scalar* linear 2nd-order ODE $\ddot{\eta}_i = -\lambda_i \eta_i$. In a sense, the commuting diagram (6.1.61) remains true for 2-step methods and second-order ODEs.

(6.2.39) is a **linear two-step recurrence** formula for the sequences $(\eta_i^{(j)})_j$.

Try: $\eta_i^{(j)} = \xi^j$ for some $\xi \in \mathbb{C} \setminus \{0\}$

Plug this into (6.2.39)

$$\begin{aligned} \blacktriangleright \quad & \xi^2 - 2\xi + 1 = -\tau^2 \lambda_i \xi \quad \Leftrightarrow \quad \xi^2 - (2 - \tau^2 \lambda_i) \xi + 1 = 0 . \\ \Rightarrow \quad & \text{two solutions} \quad \xi_{\pm} = \frac{1}{2} \left(2 - \tau^2 \lambda_i \pm \sqrt{(2 - \tau^2 \lambda_i)^2 - 4} \right) . \end{aligned}$$

We can get a blow-up of some solutions of (6.2.39), if $|\xi_+| > 1$ or $|\xi_-| > 1$. From secondary school we know Vieta's formula

$$\xi_+ \cdot \xi_- = 1 \Rightarrow \left\{ \begin{array}{l} \xi_{\pm} \in \mathbb{R} \text{ and } \xi_+ \neq \xi_- \Rightarrow |\xi_+| > 1 \text{ or } |\xi_-| > 1 \\ \xi_- = \xi_+^* \Rightarrow |\xi_-| = |\xi_+| = 1 \end{array} \right\},$$

where ξ_+^* designates complex conjugation. So the recurrence (6.2.39) has only bounded solution, if and only if

$$\text{discriminant } D := (2 - \tau^2 \lambda_i)^2 - 4 \leq 0 \Leftrightarrow \tau < \frac{2}{\sqrt{\lambda_i}}. \quad (6.2.40)$$

⟷

stability induced timestep constraint for leapfrog timestepping

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Special setting: spatial finite element Galerkin discretization based on fixed degree Lagrangian finite element spaces (\rightarrow Sect. 3.4), meshes created by uniform regular refinement.

Under these conditions a generalization of Lemma 6.1.55 shows

Stability of leapfrog timestepping entails $\tau \leq O(h_{\mathcal{M}})$ for $h_{\mathcal{M}} \rightarrow 0$

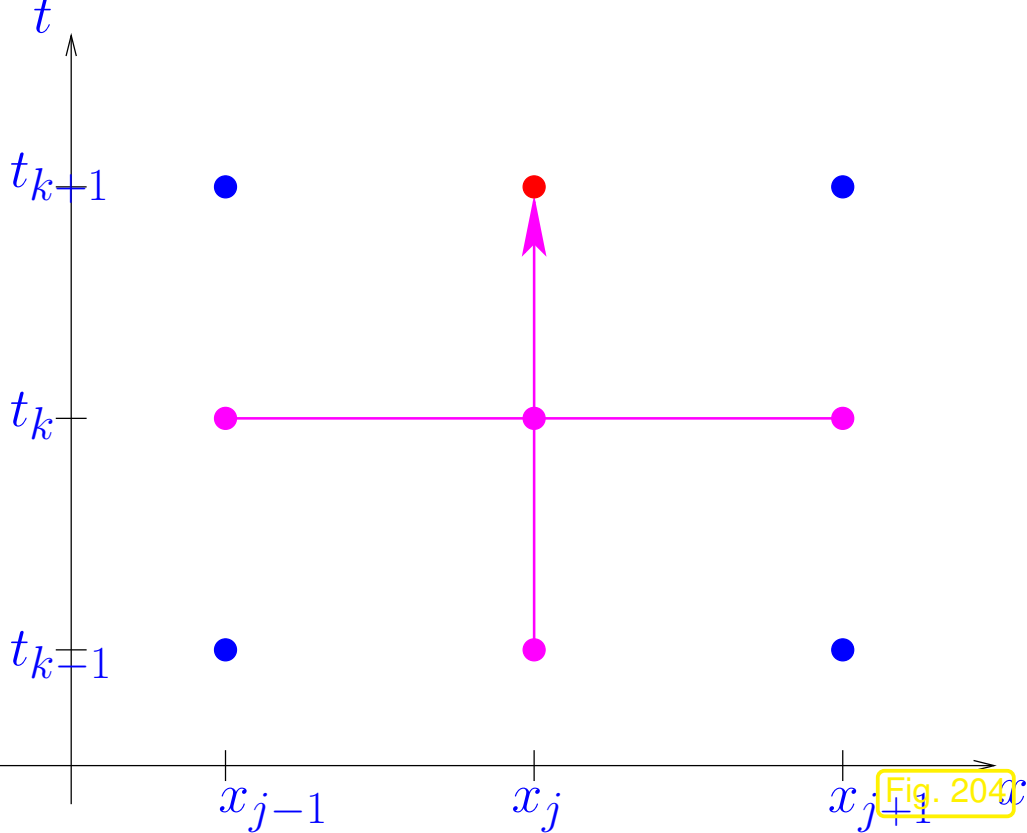
Remark 6.2.41 (Geometric interpretation of CFL condition in 1D).

Setting:

- 1D wave equation, (spatial) boundary conditions ignored (“Cauchy problem”),

$$c > 0: \quad \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad , \quad u(x, 0) = u_0(x) \quad , \quad \frac{\partial u}{\partial t}(x, 0) = v_0(x) \quad , \quad x \in \mathbb{R} . \quad (6.2.15)$$

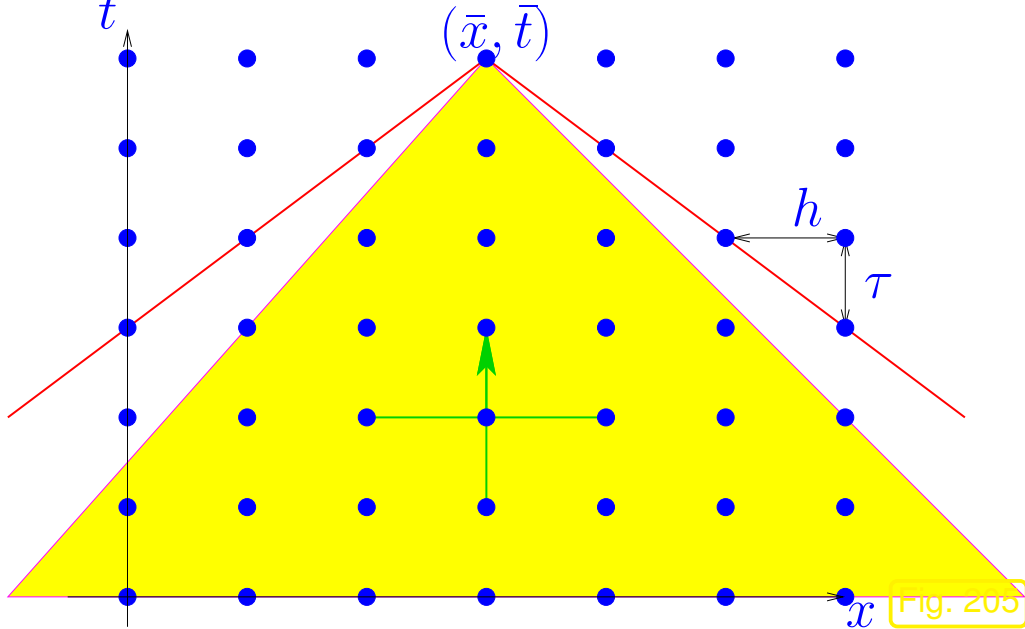
- Linear finite element Galerkin discretization on equidistant spatial mesh $\mathcal{M} := \{[x_{j-1}, x_j]: j \in \mathbb{Z}\}$, $x_j := hj$ (meshwidth h), see Sect. 1.5.1.2.
- Mass lumping for computation of mass matrix, which will become $h \cdot \mathbf{I}$, see Rem 6.2.34.
- Timestepping by Sörmer scheme (6.2.30) with constant timestep $\tau > 0$.



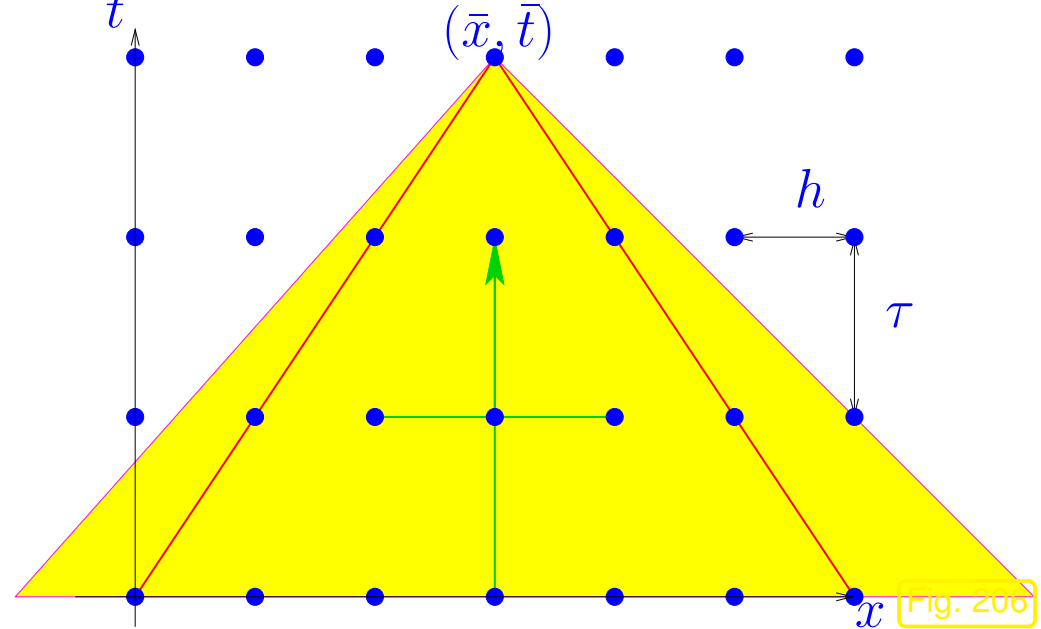
◁ flow of information in one step of Störmer scheme

Since the method is a two-step method, information from time-slices t_k and t_{k-1} is needed.

Below: yellow region $\hat{=}$ domain of dependence (d.o.d.) of (\bar{x}, \bar{t})



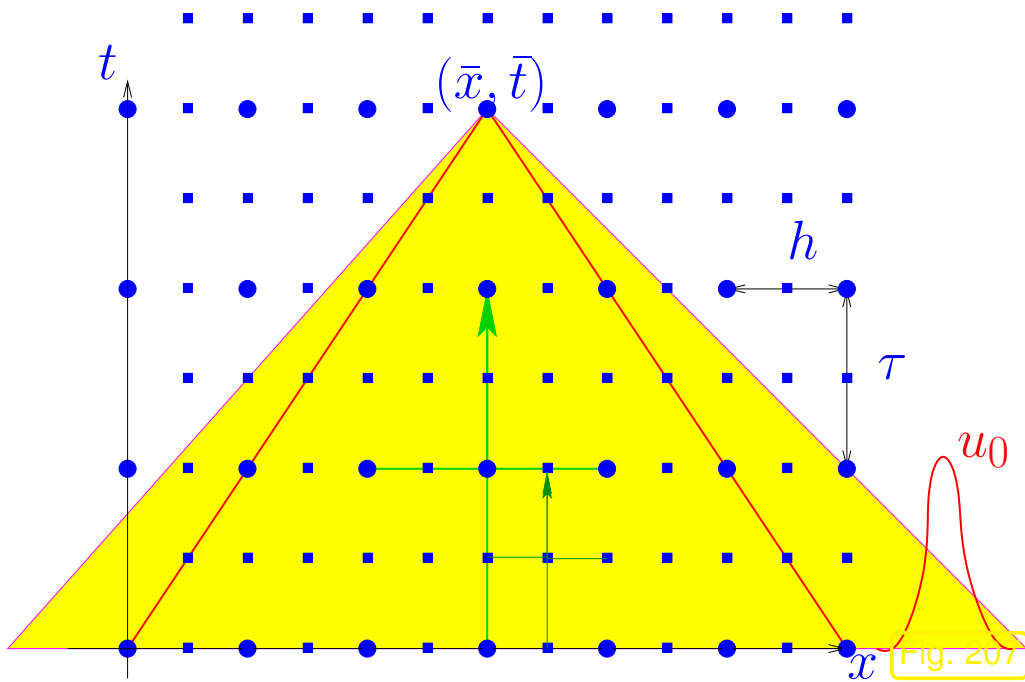
$c\tau < h$: numerical domain of dependence (marked —) contained in d.o.d.
 \Rightarrow CFL-condition met



$c\tau > h$: numerical domain of dependence (marked —) **not** contained in d.o.d.
 \Rightarrow CFL-condition violated

R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ



($\bullet \hat{=}$ coarse grid, $\blacksquare \hat{=}$ fine grid, $\blacksquare \hat{=}$ d.o.d)

\triangleleft 1D consideration:

sequence of equidistant space-time grids of $\tilde{\Omega}$ with $\tau = \gamma h$ ($\tau/h =$ meshwidth in time/space)

If $\gamma >$ CFL-constraint (here $\gamma > c^{-1}$), then analytical domain of dependence $\not\subset$ numerical domain of dependence

▲ initial data u_0 outside numerical domain of dependence cannot influence approximation at grid point (\bar{x}, \bar{t}) on *any* mesh ► no convergence !

CFL-condition \Leftrightarrow analytical domain of dependence \subset numerical domain of dependence



Will the CFL-condition thwart the efficient use of leapfrog, see Rem. 6.1.76 ?

To this end we need an idea about the convergence of the solutions of the fully discrete method:

“Meta-theorem” 6.2.42 (Convergence of fully discrete solutions of the wave equation).

Assume that

- the solution of the IBVP for the wave equation (6.2.19) is “sufficiently smooth”,
- its spatial Galerkin finite element discretization relies on degree p Lagrangian finite elements (\rightarrow Sect. 3.4) on uniformly shape-regular families of meshes,
- timestepping is based on the leapfrog method (6.2.33) with uniform timestep $\tau > 0$.

Then we can expect an asymptotic behavior of the total discretization error according to

$$\left(\tau \sum_{j=1}^M \|u - u_N(\tau j)\|_{H^1(\Omega)}^2 \right)^{\frac{1}{2}} \leq C (h_{\mathcal{M}}^p + \tau^2), \quad (6.2.43)$$

$$\left(\tau \sum_{j=1}^M \|u - u_N(\tau j)\|_{L^2(\Omega)}^2 \right)^{\frac{1}{2}} \leq C (h_{\mathcal{M}}^{p+1} + \tau^2), \quad (6.2.44)$$

where $C > 0$ must not depend on $h_{\mathcal{M}}, \tau$.

L.F. is 2nd-order !

As in the case of Metatheorem 6.1.70 (👉 nothing new!) we find:

$$\text{total discretization error} = \text{spatial error} + \text{temporal error}$$

Rem. 5.3.45 still applies: (6.2.43) does not give information about actual error, but only about the **trend** of the error, when discretization parameters $h_{\mathcal{M}}$ and τ are varied.

► Nevertheless, as in the case of the a priori error estimates of Sect. 5.3.5, we can draw conclusions about optimal γ refinement strategies in order to achieve prescribed *error reduction*.

As in Sect. 5.3.5 we make the **assumption** that the estimates (6.2.43) are sharp for all contributions to the total error and that the constants are the same (!)

$$\begin{aligned} \text{contribution of spatial (energy) error} &\approx Ch_{\mathcal{M}}^p, \quad h_{\mathcal{M}} \hat{=} \text{mesh width } (\rightarrow \text{Def. 5.2.3}), \\ \text{contribution of temporal error} &\approx C\tau^2, \quad \tau \hat{=} \text{timestep size}. \end{aligned} \quad (6.2.45)$$

This suggests the following change of $h_{\mathcal{M}}, \tau$ in order to achieve *error reduction* by a factor of $\rho > 1$:

$$\begin{aligned} \text{reduce mesh width by factor } \rho^{1/p} &\quad (6.1.74) \\ \text{reduce timestep by factor } \rho^{1/2} &\quad \implies \text{(energy) error reduction by } \rho > 1. \end{aligned} \quad (6.2.46)$$

Guideline: spatial and temporal resolution have to be adjusted in tandem

Parallel zu Rem. 6.1.76 we may wonder whether the timestep constraint $\tau < O(h_{\mathcal{M}})$ (asymptotically) enforces small timesteps not required for accuracy:

When interested in error in *energy norm* ($\leftrightarrow H^1(\Omega)$ -norm):

Only for $p = 1$ (linear Lagrangian finite elements) the requirement $\tau < O(h_{\mathcal{M}})$ stipulates the use of a smaller timestep than accuracy balancing according to (6.2.46).

When interested in $L^2(\Omega)$ -norm:

No undue timestep constraint enforced by CFL-condition for any (h -version) of Lagrangian finite element Galerkin discretization.

The leapfrog timestep constraint $\tau \leq O(h_{\mathcal{M}})$ does not compromise (asymptotic) efficiency, if $p \geq 2$ ($p \hat{=}$ degree of spatial Lagrangian finite elements).

7

Convection-Diffusion Problems



Supplementary and further reading:

[28] offers comprehensive about theory and algorithms for the numerical approximation of singularly perturbed problems of the type treated in this chapter.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

7.1 Heat conduction in a fluid

$\Omega \subset \mathbb{R}^d \hat{=}$ bounded computational domain, $d = 1, 2, 3$

To begin with we want to develop a mathematical model for stationary fluid flow, for instance, the steady streaming of water.

7.1.1 Modelling fluid flow

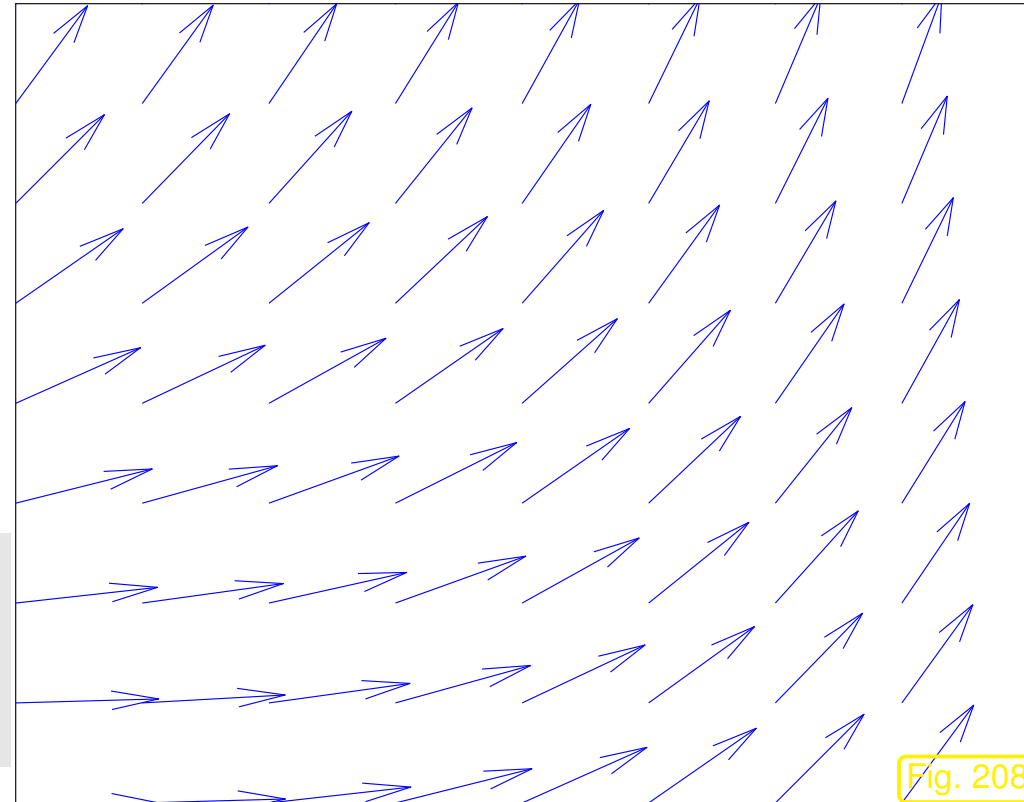
Flow field:

$$\mathbf{v} : \Omega \mapsto \mathbb{R}^d$$

Assumption:

$$\mathbf{v} \text{ is } \textit{continuous}, \mathbf{v} \in (C^0(\bar{\Omega}))^d$$

In fact, we will require that \mathbf{v} is uniformly Lipschitz continuous, but this is a mere technical assumption.



Clearly:

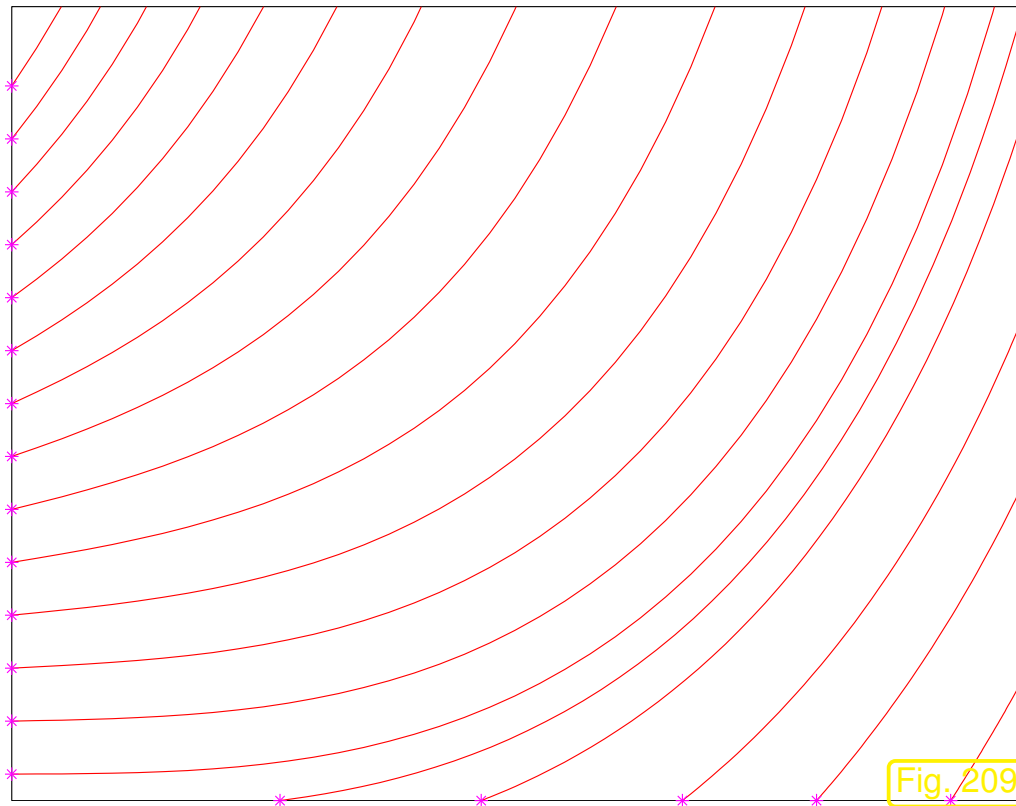
$$\mathbf{v}(\mathbf{x}) \hat{=} \text{fluid velocity at point } \mathbf{x} \in \Omega$$

➤ \mathbf{v} corresponds to a **velocity field**!

Given a flow field $\mathbf{v} \in (C^0(\overline{\Omega}))^d$ we can consider the autonomous initial value problems

$$\frac{d}{dt}\mathbf{y} = \mathbf{v}(\mathbf{y}) \quad , \quad \mathbf{y}(0) = \mathbf{x}_0 . \quad (7.1.1)$$

Its solution $t \mapsto \mathbf{y}(t)$ defines the path travelled by a particle carried along by the fluid, a **particle trajectory**, also called a **streamline**.

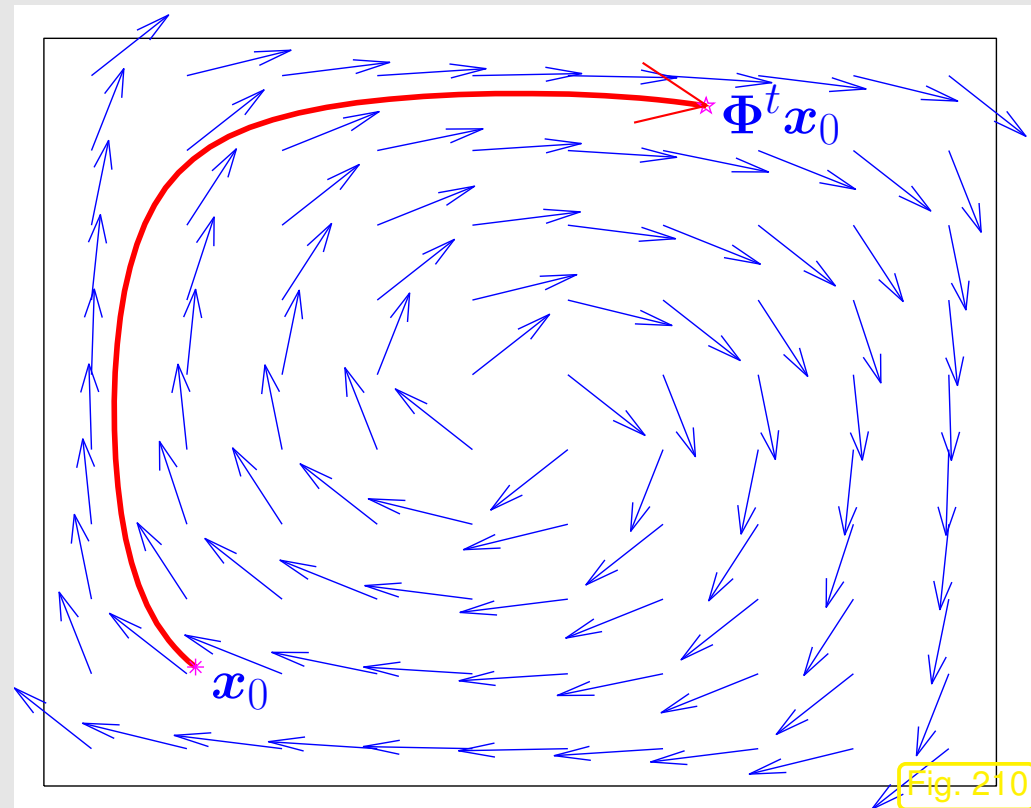


◁ particle trajectories (streamlines) in flow field of Fig. 208.

(* $\hat{=}$ initial particle positions)

A flow field induces a transformation (mapping) of space! to explain this, let us temporarily make the assumption that

the flow does neither enter nor leave Ω ,
(this applies to fluid flow in a close container)



which can be modelled by

$$\mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \partial\Omega, \quad (7.1.2)$$

that is, the flow is always parallel to the boundary of Ω : all particle trajectories stay inside Ω .

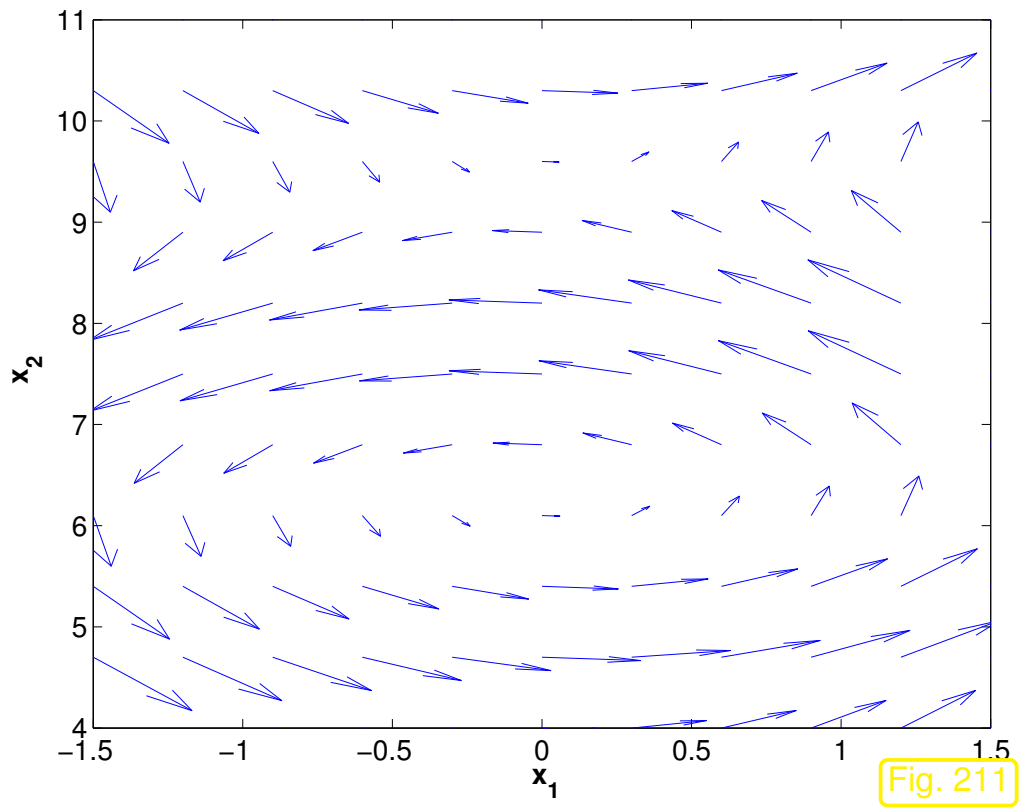
Now we fix some “time of interest” $t > 0$.

$$\text{➤ mapping } \Phi^t : \begin{cases} \Omega \mapsto \Omega \\ \mathbf{x}_0 \mapsto \mathbf{y}(t) \end{cases}, \quad t \mapsto \mathbf{y}(t) \text{ solution of IVP (7.1.1)}, \quad (7.1.3)$$

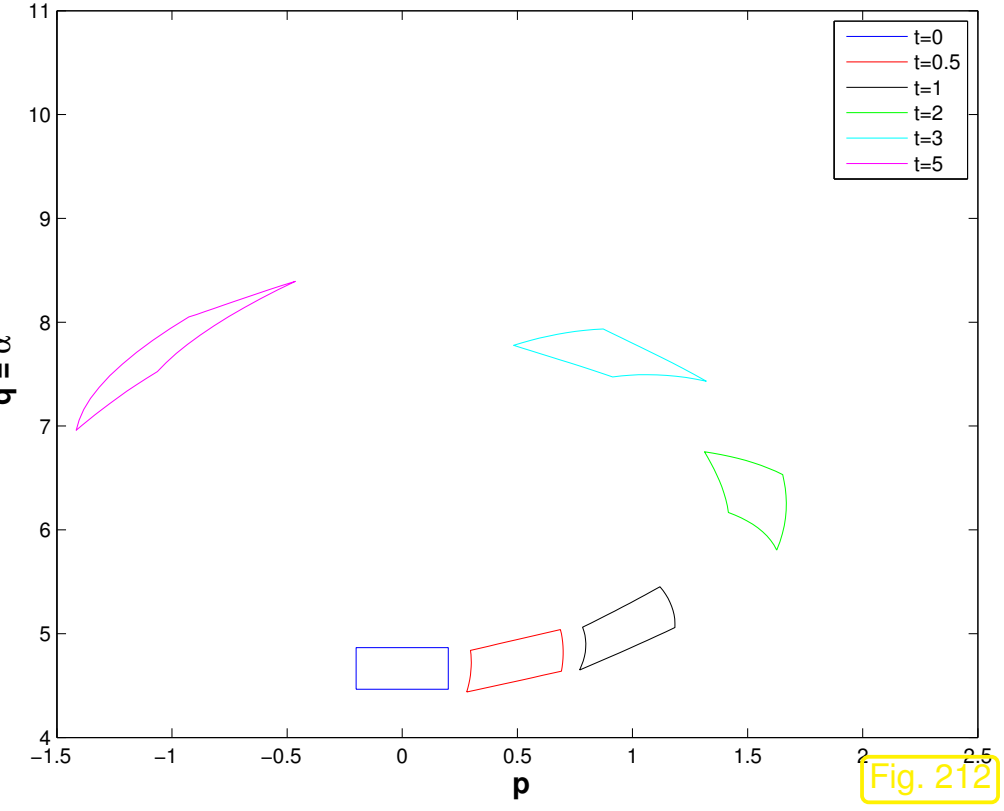
is well-defined mapping of Ω to itself, the **flow map**. Obviously, it satisfies

$$\Phi^0 x_0 = x_0 \quad \forall x_0 \in \Omega . \tag{7.1.4}$$

In [21, Def. 12.1.25] the more general concept of an **evolution operator** was introduced, which agrees with the flow map in the current setting.



flow field $v : \Omega \mapsto \mathbb{R}^2$



snapshots of $\Phi^t(V)$ for control volume V

$\Phi^\tau(V) \hat{=}$ volume occupied at time $t = \tau$ by particles that occupied $V \subset \Omega$ at time $t = 0$.

7.1.2 Heat convection and diffusion

$u : \Omega \mapsto \mathbb{R} \hat{=}$ stationary temperature distribution in fluid *moving* according to a stationary flow field
 $\mathbf{v} : \Omega \mapsto \mathbb{R}^d$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

We adapt the considerations of Sect. 2.5 that led to the stationary heat equation. Recall

Conservation of energy

$$\int_{\partial V} \mathbf{j} \cdot \mathbf{n} \, dS = \int_V f \, d\mathbf{x} \quad \text{for all "control volumes" } V. \quad (2.5.2)$$

power flux through surface of V

heat production inside V

From 2.5.2 by Gauss' theorem Thm. 2.4.9

$$\int_V \operatorname{div} \mathbf{j}(\mathbf{x}) \, d\mathbf{x} = \int_V f(\mathbf{x}) \, d\mathbf{x} \quad \text{for all "control volumes" } V \subset \Omega .$$

Now appeal to another version of the fundamental lemma of the calculus of variations, see Lemma 2.4.15, this time sporting piecewise constant test functions.

▶ **local form of energy conservation:**

$$\operatorname{div} \mathbf{j} = f \quad \text{in } \Omega . \tag{2.5.5}$$

However, in a moving fluid a power flux through a fixed surface is already caused by the sheer fluid flow carrying along thermal energy. This is reflected in a modified Fourier's law (2.5.3):

Fourier's law in moving fluid

$$\mathbf{j}(\mathbf{x}) = -\kappa \mathbf{grad} u(\mathbf{x}) + \mathbf{v}(\mathbf{x})\rho u(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (7.1.5)$$

diffusive heat flux

(due to spatial variation of temperature)

convective heat flux

(due to fluid flow)

$\kappa > 0 \hat{=}$ heat conductivity ($[\kappa] = 1 \frac{\text{W}}{\text{K m}}$), $\rho > 0 \hat{=}$ heat capacity ($[\rho] = \frac{\text{J}}{\text{K m}^3}$), both assumed to be constant (in contrast to the models of Sect. 2.5 and Sect. 6.1.1).

Combine equations (2.5.5) & (7.1.5):

$$\begin{aligned} \operatorname{div} \mathbf{j} = f \quad + \quad \mathbf{j}(\mathbf{x}) = -\kappa \mathbf{grad} u(\mathbf{x}) + \mathbf{v}(\mathbf{x})\rho u(\mathbf{x}) \\ \Downarrow \\ -\operatorname{div}(\kappa \mathbf{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x})u) = f \quad \text{in } \Omega. \end{aligned} \quad (7.1.6)$$

$$-\operatorname{div}(\kappa \mathbf{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x})u) = f .$$

Terminology :

$$\begin{array}{ccc} \downarrow & & \downarrow \\ \text{diffusive term} & & \text{convective term} \\ \text{(2nd-order)} & & \text{(1st-order)} \end{array}$$

The 2nd-order elliptic PDE (7.1.6) has to be supplemented with exactly one *boundary condition* on any part of $\partial\Omega$, see Sect. 2.6, Ex. 2.6.7. This can be any of the (“elliptic”) boundary conditions introduced in Sect. 2.6:

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

- Dirichlet boundary conditions: $u = g \in C^0(\partial\Omega)$ on $\partial\Omega$ (fixed surface temperatur),
- Neumann boundary conditions: $\mathbf{j} \cdot \mathbf{n} = -h$ on $\partial\Omega$ (fixed heat flux),
- (non-linear) radiation boundary conditions: $\mathbf{j} \cdot \mathbf{n} = \Psi(u)$ on $\partial\Omega$ (temperature dependent heat flux, radiative heat flux).

Guideline: Required boundary conditions determined by highest-order term

For the sake of simplicity we will mainly consider **incompressible fluids**.

Definition 7.1.7 (Incompressible flow field).

A fluid flow is called **incompressible**, if the associated flow map Φ^t is **volume preserving**,

$$|\Phi^t(V)| = |\Phi^0(V)| \quad \text{for all sufficiently small } t > 0, \text{ for all control volumes } V .$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Can incompressibility be read off the velocity field \mathbf{v} of the flow?

To investigate this issue, again assume the “no flow through the boundary condition” (7.1.2) and recall that the flowmap Φ^t from (7.1.3) satisfies

$$\frac{\partial}{\partial t} \Phi(t, \mathbf{x}) = \mathbf{v}(\Phi(t, \mathbf{x})) , \quad \mathbf{x} \in \Omega , t > 0 . \quad (7.1.8)$$

Here, in order to make clear the dependence on independent variables, time occurs as an argument of Φ in brackets, on par with \mathbf{x} .

Next, formal differentiation w.r.t. \mathbf{x} and change of order of differentiation yields a differential equation for the Jacobian $D_{\mathbf{x}}\Phi^t$,

$$(7.1.8) \Rightarrow \frac{\partial}{\partial t}(D_{\mathbf{x}}\Phi)(t, \mathbf{x}) = D_{\mathbf{v}}(\Phi(t, \mathbf{x}))(D_{\mathbf{x}}\Phi)(t, \mathbf{x}) . \quad (7.1.9)$$

Jacobian $\in \mathbb{R}^{d,d}$
Jacobian $\in \mathbb{R}^{d,d}$

Second strand of thought: apply transformation formula for integrals (3.5.31), [32, Satz 8.5.2]: for fixed $t > 0$

$$|\Phi(t, V)| = \int_{\Phi(t, V)} 1 \, d\mathbf{x} = \int_V |\det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}})| \, d\hat{\mathbf{x}} . \quad (7.1.10)$$

Volume preservation by the flow map is equivalent to

$$t \mapsto |\Phi(t, V)| = \text{const.} \iff \frac{d}{dt} |\Phi(t, V)| = 0 ,$$

for any control volume $V \subset \Omega$.

$$(7.1.10) \Rightarrow \frac{d}{dt} |\Phi(t, V)| = \int_V \frac{\partial}{\partial t} |\det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}})| \, d\hat{\mathbf{x}} .$$

Theorem 7.1.11 (Differentiation formula for determinants).

Let $\mathbf{S} : I \subset \mathbb{R} \mapsto \mathbb{R}^{n,n}$ be a smooth matrix-valued function. If $\mathbf{S}(t_0)$ is regular for some $t_0 \in I$, then

$$\frac{d}{dt}(\det \circ \mathbf{S})(t_0) = \det(\mathbf{S}(t_0)) \operatorname{tr}\left(\frac{d\mathbf{S}}{dt}(t_0)\mathbf{S}^{-1}(t_0)\right).$$

►
$$\begin{aligned} \frac{\partial}{\partial t} \det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}}) &\stackrel{(7.1.9)}{=} \det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}}) \operatorname{tr}\left(D_{\mathbf{v}}(\Phi(t, \hat{\mathbf{x}})) \underbrace{(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}})(D_{\mathbf{x}}\Phi)^{-1}(t, \hat{\mathbf{x}})}_{=\mathbf{I}}\right) \\ &= \det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}}) \operatorname{div} \mathbf{v}(\Phi(t, \hat{\mathbf{x}})), \end{aligned}$$

 R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

because the divergence of a vector field \mathbf{v} is just the trace of its Jacobian $D_{\mathbf{v}}$! From (7.1.4) we know that for small $t > 0$ the Jacobian $D_{\mathbf{x}}\Phi(t, \hat{\mathbf{x}})$ will be close to \mathbf{I} and, therefore, $\det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}}) \neq 0$ for $t \approx 0$. Thus, for small $t > 0$ we conclude

$$\frac{d}{dt}|\Phi(t, V)| = 0 \iff \operatorname{div} \mathbf{v}(\Phi(t, \hat{\mathbf{x}})) = 0 \quad \forall \hat{\mathbf{x}} \in V.$$

Since this is to hold for **any** control volume V , the final equivalence is

$$\frac{d}{dt}|\Phi(t, V)| = 0 \quad \forall \text{ control volumes } V \iff \operatorname{div} \mathbf{v} = 0 \quad \text{in } \Omega.$$

Theorem 7.1.12 (Divergence-free velocity fields for incompressible flows).

A stationary fluid flow in Ω is incompressible (\rightarrow Def. 7.1.7), if and only if its associated velocity field \mathbf{v} satisfies $\operatorname{div} \mathbf{v} = 0$ everywhere in Ω .

In the sequel we make the **assumption**:

$$\operatorname{div} \mathbf{v} = \sum_{j=1}^d \frac{\partial v_j}{\partial x_j} = 0 .$$

(Note: for $d = 1$ this boils down to $\frac{dv}{dx} = 0$ and implies $v = \text{const.}$)

Then we can use the product rule in higher dimensions of Lemma 2.4.7:

$$\operatorname{div}(\rho \mathbf{v} u) \stackrel{\text{Lemma 2.4.7}}{=} \rho(u \operatorname{div} \mathbf{v} + \mathbf{v} \cdot \mathbf{grad} u) \stackrel{\operatorname{div} \mathbf{v}=0}{=} \rho \mathbf{v} \cdot \mathbf{grad} u . \quad (7.1.13)$$

Thus, we can rewrite the scalar convection-diffusion equation (7.1.6) for an incompressible flow field

$$-\operatorname{div}(\kappa \mathbf{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x})u) = f \quad \text{in } \Omega$$

$$\leftarrow \operatorname{div} \mathbf{v} = 0$$

$$-\kappa \Delta u + \rho \mathbf{v} \cdot \operatorname{grad} u = f \quad \text{in } \Omega . \quad (7.1.14)$$

When carried along by the flow of an incompressible fluid, the temperature cannot be increased by local compression, the effect that you can witness when pumping air. Hence, only sources/sinks can lead to local extrema of the temperature.

Now recall the discussion of the physical intuition behind the **maximum principle** of Thm. 5.7.2. These considerations still apply to stationary heat flow in a moving incompressible fluid.

Theorem 7.1.15 (Maximum principle for scalar 2nd-order convection diffusion equations). \rightarrow

[15, 6.4.1, Thm. I]

Let $\mathbf{v} : \Omega \mapsto \mathbb{R}^d$ be a continuously differentiable vector field. Then there holds the **maximum principle**

$$-\Delta u + \mathbf{v} \cdot \operatorname{grad} u \geq 0 \quad \Longrightarrow \quad \min_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) = \min_{\mathbf{x} \in \Omega} u(\mathbf{x}) ,$$

$$-\Delta u + \mathbf{v} \cdot \operatorname{grad} u \leq 0 \quad \Longrightarrow \quad \max_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) = \max_{\mathbf{x} \in \Omega} u(\mathbf{x}) .$$

7.1.4 Transient heat conduction

In Sect. 6.1.1 we generalized the laws of stationary heat conduction derived in Sect. 2.5 to time-dependent temperature distributions $u = u(\mathbf{x}, t)$ sought on a space-time cylinder $\tilde{\Omega} := \Omega \times]0, T[$.

The same ideas apply to heat conduction in a fluid:

- Start from energy balance law (6.1.1) and convert it into local form (6.1.2).
- Combine it with the extended Fourier's law (7.1.5).



$$\frac{\partial}{\partial t}(\rho u) - \operatorname{div}(\kappa \mathbf{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x}, t)u) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[. \quad (7.1.16)$$

For details and notations refer to Sect. 6.1.1.

This PDE has to be supplemented with

- boundary conditions (as in the stationary case, see Sect. 2.6),
- initial conditions (same as for pure diffusion, see Sect. 6.1.1).

Under the assumption $\operatorname{div}_{\mathbf{x}} \mathbf{v}(\mathbf{x}, t) = 0$ of incompressibility (\rightarrow Def. 7.1.7 and Thm. 7.1.12) and in the case of constant (in space) coefficients (7.1.16) is equivalent to, *cf.* (7.1.13),

$$\frac{\partial}{\partial t}(\rho u) - \kappa \Delta u + \rho \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[\quad . \quad (7.1.17)$$

7.2 Stationary convection-diffusion problems

Model problem, *cf.* (7.1.14), modelling stationary heat flow in an incompressible fluid with prescribed temperature at “walls of the container” (\Leftrightarrow Dirichlet boundary conditions).

$$-\kappa\Delta u + \rho\mathbf{v}(\mathbf{x}) \cdot \mathbf{grad} u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega.$$

Perform **scaling** $\hat{=}$ choice of physical units: makes equation non-dimensional by fixing “reference length”, “reference time interval”, “reference temperature”, “reference power”.

A suitable choice of physical units leads to rescaled physical constants $\kappa \rightarrow \epsilon$, $\rho \rightarrow 1$, $\|\mathbf{v}\|_{L^\infty(\Omega)} \rightarrow 1$.

After scaling we deal with the non-dimensional boundary value problem

$$-\epsilon\Delta u + \mathbf{v}(\mathbf{x}) \cdot \mathbf{grad} u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega, \quad (7.2.1)$$

diffusion term
(2nd-order term)
convection term
(1st-order term)

with $\epsilon > 0$, $\|\mathbf{v}\|_{L^\infty(\Omega)} = 1$, $\mathbf{div} \mathbf{v} = 0 \rightarrow$ incompressible fluid, see Def. 7.1.7.

Remark 7.2.2 (Variational formulation for convection-diffusion BVP).

Standard “4-step approach” of Sect. 2.8 can be directly applied to BVP (7.2.1) with one new twist:

Do not use integration by parts (Green’s formula, Thm. 2.4.11) on convective terms!

► variational formulation for BVP (7.2.1):

$$u \in H_0^1(\Omega): \quad \underbrace{\epsilon \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} + \int_{\Omega} (\mathbf{v} \cdot \mathbf{grad} u) v \, d\mathbf{x}}_{\text{bilinear form } \mathbf{a}(u,v)} = \underbrace{\int_{\Omega} f(\mathbf{x}) \, d\mathbf{x}}_{\text{linear form } \ell(v)} \quad \forall v \in H_0^1(\Omega) .$$

$\hat{=}$ a linear variational problem, see Sect. 2.3.1.

Obvious: \mathbf{a} is **not** symmetric, see (2.1.21).

➔ \mathbf{a} does not induce an energy norm (\rightarrow Def. 2.1.35)

As replacement for the energy norm use $H^1(\Omega)$ -(semi)norm (\rightarrow Def. 2.2.15)

In this case we have to make sure that \mathbf{a} fits the chosen norm in the sense that

$$\exists C > 0: \quad |\mathbf{a}(u, v)| \leq C |u|_{H^1(\Omega)} |v|_{H^1(\Omega)} \quad \forall u, v \in H_0^1(\Omega) . \quad (7.2.3)$$

\longleftrightarrow Terminology: (7.2.3) $\hat{=}$ \mathbf{a} is **continuous** on $H^1(\Omega)$, cf. (3.1.2).

By Cauchy-Schwarz inequality for integrals (2.1.37): for all $u, v \in H_0^1(\Omega)$

$$|\mathbf{a}(u, v)| \leq \|v\|_{L^\infty(\Omega)} |u|_{H^1(\Omega)} \|v\|_{L^2(\Omega)} \stackrel{\text{Thm. 2.2.25}}{\leq} \text{diam}(\Omega) \|v\|_{L^\infty(\Omega)} |u|_{H^1(\Omega)} |v|_{H^1(\Omega)} ,$$

which confirms (7.2.3)

Surprise: \mathbf{a} is **positive definite** (\rightarrow Def. 2.1.32), because

$$\int_{\Omega} (\mathbf{v} \cdot \mathbf{grad} u) u \, d\mathbf{x} = \int_{\Omega} (\mathbf{v} u) \cdot \mathbf{grad} u \, d\mathbf{x}$$

$$\stackrel{\text{Green's formula}}{=} - \int_{\Omega} \text{div}(\mathbf{v} u) u \, d\mathbf{x} + \int_{\partial\Omega} \underbrace{u^2}_{=0} \mathbf{v} \cdot \mathbf{n} \, dS$$

$$\stackrel{(2.4.8) \ \& \ \text{div } \mathbf{v}=0}{=} - \int_{\Omega} (\mathbf{v} \cdot \mathbf{grad} u) u \, d\mathbf{x} .$$

$$\blacktriangleright \quad \mathbf{a}(u, u) = \epsilon \int_{\Omega} \|\mathbf{grad} u\|^2 \, d\mathbf{x} > 0 \quad \forall u \in H_0^1(\Omega) \setminus \{0\} . \quad (7.2.4)$$

From this and (7.2.3) we conclude existence and uniqueness of solutions of the BVP (7.2.1) in the Sobolev space $H_0^1(\Omega)$.

7.2.1 Singular perturbation

Setting: fast-moving fluid \leftrightarrow convection dominates diffusion $\leftrightarrow \epsilon \ll 1$ in (7.2.1)

Example 7.2.5 (1D convection-diffusion boundary value problem).

$$-\epsilon \frac{d^2 u_\epsilon}{dx^2} + \frac{du_\epsilon}{dx} = 1 \quad \text{in } \Omega ,$$

$$u_\epsilon(0) = 0 \quad , \quad u_\epsilon(1) = 0 ,$$

►
$$u_\epsilon(x) = x + \frac{\exp(-x/\epsilon) - 1}{1 - \exp(-1/\epsilon)} .$$

For $\epsilon \ll 1$:

boundary layer at $x = 1$

Pointwise limit:

$$\lim_{\epsilon \rightarrow 0} u_\epsilon(x) \rightarrow x \quad \forall 0 < x < 1 .$$

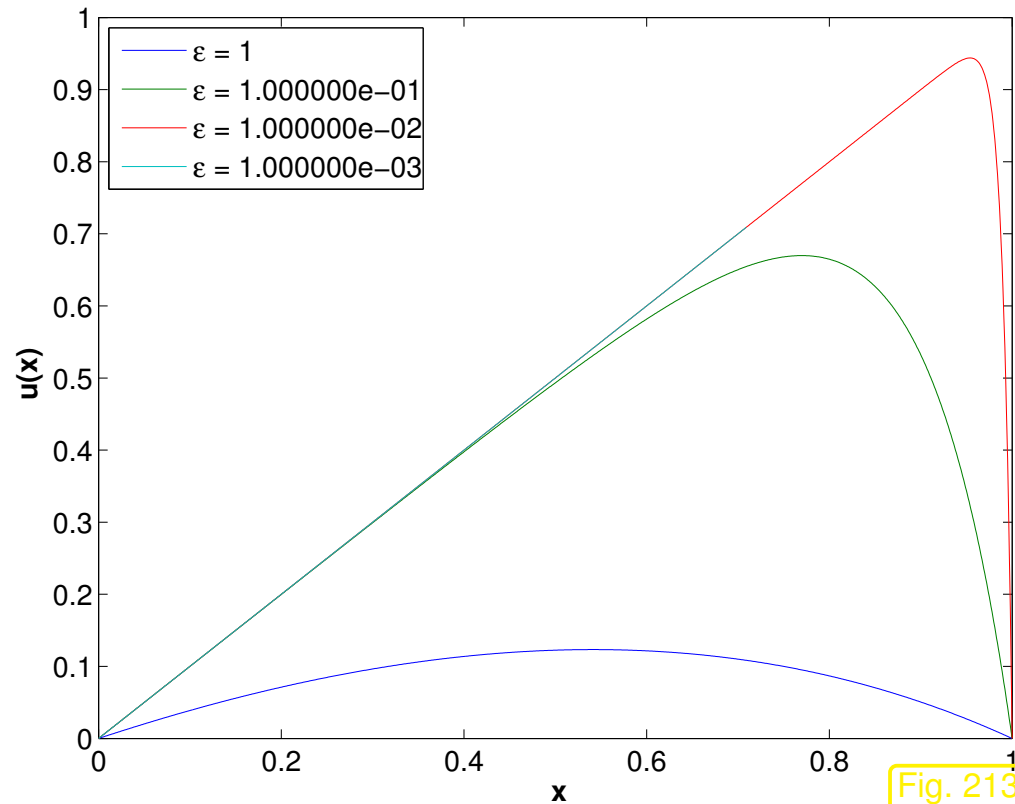


Fig. 213



“Limit problem”: ignore diffusion ➤ set $\epsilon = 0$

$$(7.2.1) \quad \text{►} \quad \overset{\epsilon=0}{\mathbf{v}(\mathbf{x}) \cdot \mathbf{grad} u = f(\mathbf{x})} \quad \text{in } \Omega . \quad (7.2.6)$$

Case $d = 1$ ($\Omega =]0, 1[$, $v = \pm 1$)

$$(7.2.6) \quad \blacktriangleright^{d=1} \quad \pm \frac{du}{dx}(x) = f(x) \quad \Rightarrow \quad u(x) = \int f \, dx + C . \quad (7.2.7)$$

What about this constant C ?

If $v = 1 \Leftrightarrow$ fluid flows “from left to right”, so we should integrate the source from 0 to x :

$$u(x) = u(0) + \int_0^x f(s) \, ds = \int_0^x f(s) \, ds , \quad (7.2.8)$$

because $u(0) = 0$ by the boundary condition $u = 0$ on $\partial\Omega$. If $v = -1$ we start the integration at $x = 1$. Note that this makes the maximum principle of Thm. 7.1.15 hold.

For $d > 1$ we can solve (7.2.6) by **the method of characteristics**:

To motivate it, be aware that (7.2.6) describes **pure transport** of a temperature distribution in the velocity field \mathbf{v} , that is, the heat/temperature is just carried along particle trajectories and changes only under the influence of heat sources/sinks along that trajectory.

Denote by u the solution of (7.2.6) and recall the differential equation (7.1.1) for a particle trajectory

$$\frac{d\mathbf{y}}{dt}(t) = \mathbf{v}(\mathbf{y}(t)) \quad , \quad \mathbf{y}(0) = \mathbf{x}_0 \quad . \quad (7.1.1)$$

$$\blacktriangleright \quad \frac{d}{dt}u(\mathbf{y}(t)) = \mathbf{grad} u(\mathbf{y}(t)) \cdot \frac{d}{dt}\mathbf{y}(t) = \mathbf{grad} u \cdot \mathbf{v}(\mathbf{y}(t)) \stackrel{(7.2.6)}{=} f(\mathbf{y}(t)) \quad .$$

➤ Compute $\mathbf{u}(\mathbf{y}(t))$ by integrating source f along particle trajectory!

$$u(\mathbf{y}(t)) = u(\mathbf{x}_0) + \int_0^t f(\mathbf{y}(s)) \, ds \quad (7.2.9)$$

Taking the cue from $d = 1$ we choose \mathbf{x}_0 as “the point on the boundary where the particle enters Ω ”.

These points form the part of the boundary through which the flow enters Ω , the **inflow boundary**

$$\Gamma_{\text{in}} := \{ \mathbf{x} \in \partial\Omega : \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0 \} \quad . \quad (7.2.10)$$

Its complement in $\partial\Omega$ contains the **outflow boundary**

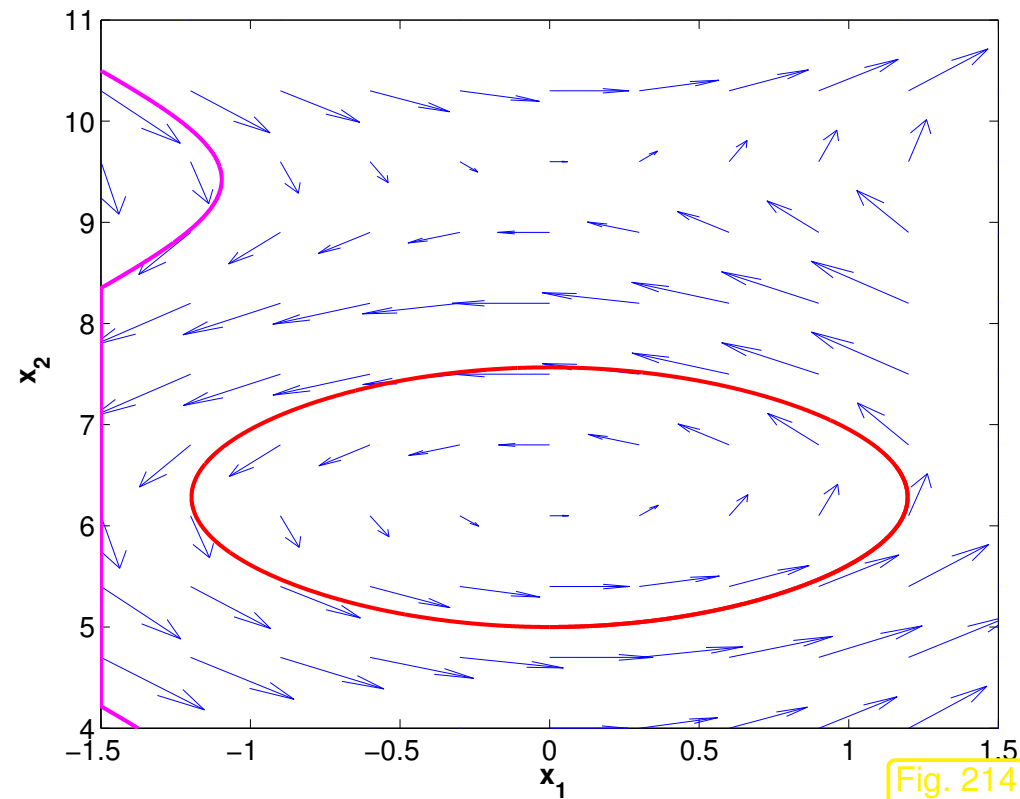
$$\Gamma_{\text{out}} := \{ \mathbf{x} \in \partial\Omega : \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) > 0 \} \quad . \quad (7.2.11)$$

Remark 7.2.12 (Streamlines).

→ velocity field

—: Streamline connecting Γ_{in} and Γ_{out}

—: Closed streamline
(recirculating flow)



In the case of closed streamlines the stationary pure transport problem fails to have a unique solution: on a closed streamline u can attain “any” value, because there is no boundary value to fix u .



Return to case $d = 1$. In general solution $u(x)$ from (7.2.7) will **not** satisfy the boundary condition $u(1) = 0$! Also for $u(x)$ from (7.2.9) the homogeneous boundary conditions may be violated where the particle trajectory leaves Ω !

In the limit case $\epsilon = 0$ not all boundary conditions of (7.2.1) can be satisfied.

Notion 7.2.13 (Singularly perturbed problem).

*A boundary value problem depending on parameter $\epsilon \approx \epsilon_0$ is called **singularly perturbed**, if the limit problem for $\epsilon \rightarrow \epsilon_0$ is not compatible with the boundary conditions.*

Especially in the case of 2nd-order elliptic boundary value problems:

Singular perturbation = 1st-order terms become dominant for $\epsilon \rightarrow \epsilon_0$

In mathematical terms, singular perturbation for boundary values for PDEs is defined as a *change of type* of the PDE for $\epsilon = 0$: in the case of (7.2.1) the type changes from elliptic to hyperbolic, see Rem. 2.0.2.

7.2.2 Upwinding

Focus: linear finite element Galerkin discretization for 1D model problem, *cf.* Ex. 7.2.5

$$-\epsilon \frac{d^2 u}{dx^2} + \frac{du}{dx} = f(x) \quad \text{in } \Omega, \quad u(0) = 0, \quad u(1) = 0. \quad (7.2.14)$$

Variational formulation, see Rem. 7.2.2:

$$u \in H_0^1(]0, 1[): \quad \underbrace{\epsilon \int_0^1 \frac{du}{dx}(x) \frac{dv}{dx}(x) dx + \int_0^1 \frac{du}{dx}(x) v(x) dx}_{=: a(u,v)} = \underbrace{\int_0^1 f(x) v(x) dx}_{=: \ell(v)} \quad \forall v \in H_0^1(]0, 1[).$$

As in Sect. 1.5.1.2: use equidistant mesh \mathcal{M} (mesh width $h > 0$), composite trapezoidal rule (1.5.85) for right hand side linear form, standard “tent function basis”, see (1.5.76).

► linear system of equations for coefficients μ_i , $i = 1, \dots, M - 1$, providing approximations for point values $u(ih)$ of exact solution u .

$$\left(-\frac{\epsilon}{h} - \frac{1}{2}\right) \mu_{i-1} + \frac{2\epsilon}{h} \mu_i + \left(-\frac{\epsilon}{h} + \frac{1}{2}\right) \mu_{i+1} = hf(ih), \quad i = 1, \dots, M - 1, \quad (7.2.15)$$

where the homogeneous Dirichlet boundary conditions are taken into account by setting $\mu_0 = \mu_M = 0$.

Remark 7.2.16 (Finite differences for convection-diffusion equation in 1D).

As in Sect. 1.5.3 on the finite difference in 1D, we can also obtain (7.2.15) by replacing the derivatives

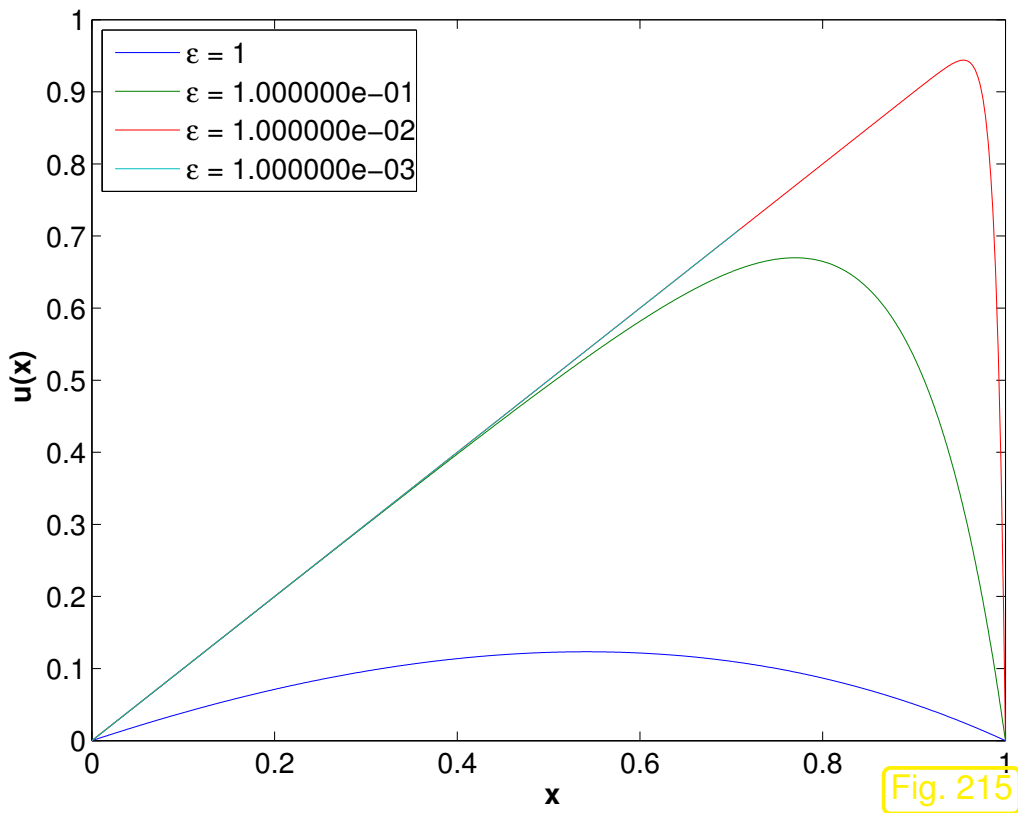
by suitable difference quotients:

$$\begin{array}{ccc}
 -\epsilon \frac{d^2 u}{dx^2} & + & \frac{du}{dx} & = & f(x) \\
 \updownarrow & & \updownarrow & & \updownarrow \\
 \epsilon \underbrace{\frac{-\mu_{i+1} + 2\mu_i - \mu_{i-1}}{h^2}}_{\text{difference quotient for } \frac{d^2 u}{dx^2}} & + & \underbrace{\frac{\mu_{i+1} - \mu_{i-1}}{2h}}_{\text{symmetric d.q. for } \frac{du}{dx}} & = & f(ih) .
 \end{array} \tag{7.2.15}$$

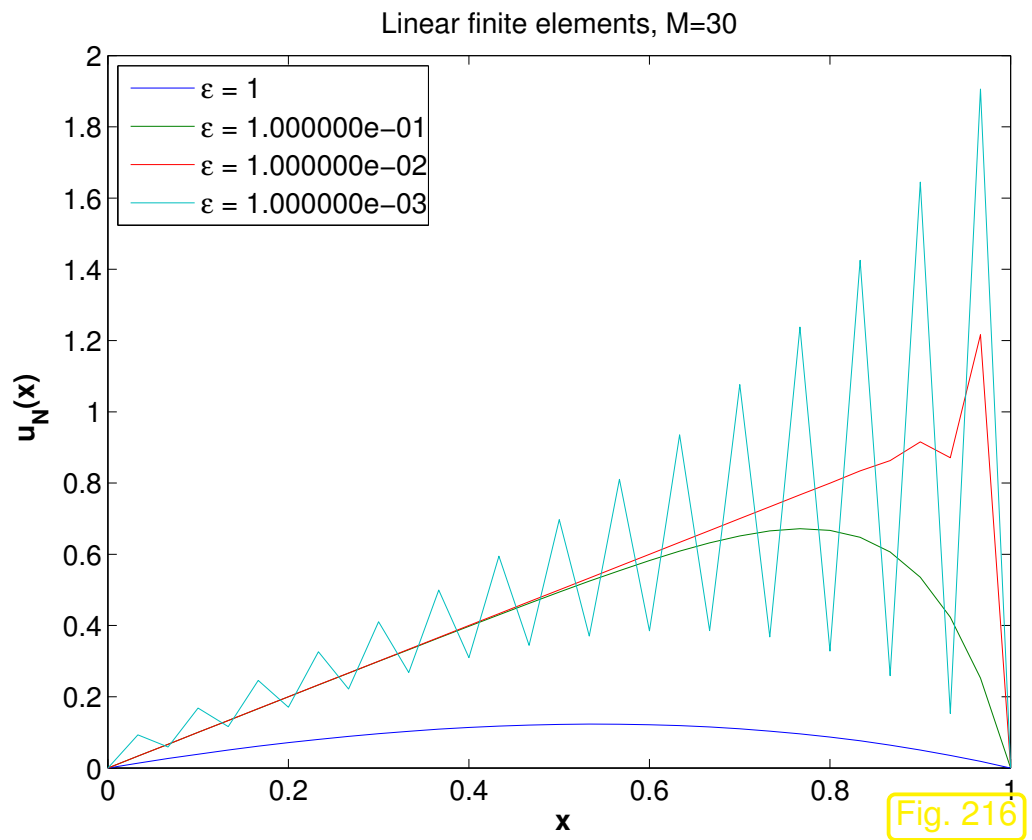


Example 7.2.17 (Linear FE discretization of 1D convection-diffusion problem).

- Model boundary value problem (7.2.14)
- linear finite element Galerkin discretization as described above
- As in Ex. 7.2.5: $f \equiv 1$



exact solutions



FE solutions

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

For very small ϵ : spurious *oscillations* of linear FE Galerkin solution.



In order to understand this observation, study the linear finite element Galerkin discretization in the limit case $\epsilon = 0$

$$(7.2.15) \quad \blacktriangleright^{\epsilon=0} \quad \mu_{i+1} - \mu_{i-1} = 2hf(ih), \quad i = 1, \dots, M. \quad (7.2.18)$$

\blacktriangleright (7.2.18) $\hat{=}$ Linear system of equations with *singular* system matrix!

For $\epsilon > 0$ the Galerkin matrix will always be regular due to (7.2.4), but the linear relationship (7.2.18) will become more and more dominant as $\epsilon > 0$ becomes smaller and smaller. In particular, (7.2.18) sends the message that values at even and odd numbered nodes will become decoupled, which accounts for the oscillations.

Desired: **robust** discretization of (7.2.14)

= discretization that produces qualitatively correct (*) solutions for **any** $\epsilon > 0$

(*): “qualitatively correct”, e.g., satisfaction of maximum principle, Thm. 7.1.15]

Guideline:

What is a meaningful scheme for limit problem $u' = f$ on an equidistant mesh of $\Omega :=]0, 1[$?

Explicit Euler method: $\mu_{i+1} - \mu_i = hf(\xi_i) \quad i = 0, \dots, N,$

Implicit Euler method: $\mu_{i+1} - \mu_i = hf(\xi_{i+1}) \quad i = 0, \dots, N.$

► Use **one-sided difference quotients** for discretization of convective term !

Which type ? (Explicit or implicit Euler ?)

Linear system arising from *use of backward difference quotient* $\frac{du}{dx}|_{x=x_i} = \frac{\mu_i - \mu_{i-1}}{h} :$

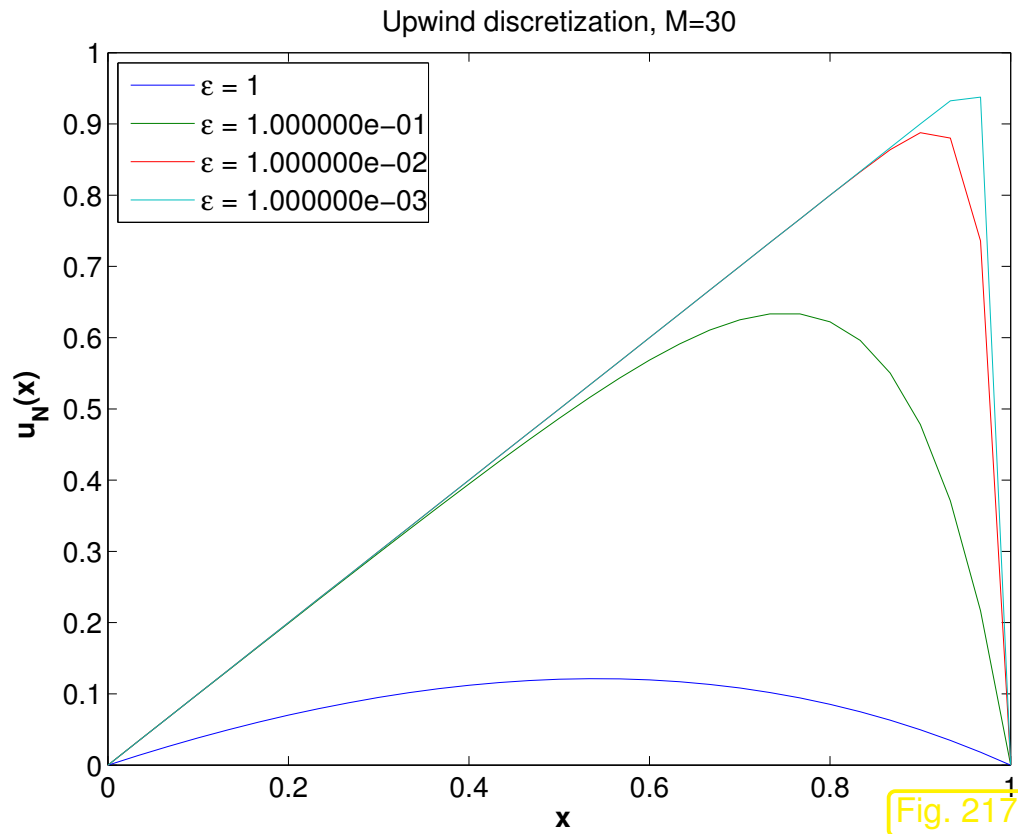
$$\left(-\frac{\epsilon}{h} - 1\right) \mu_{i-1} + \left(\frac{2\epsilon}{h} + 1\right) \mu_i + -\frac{\epsilon}{h} \mu_{i+1} = hf(ih), \quad i = 1, \dots, M - 1, \quad (7.2.19)$$

Linear system arising from *use of forward difference quotient* $\frac{du}{dx}|_{x=x_i} = \frac{\mu_{i+1} - \mu_i}{h} :$

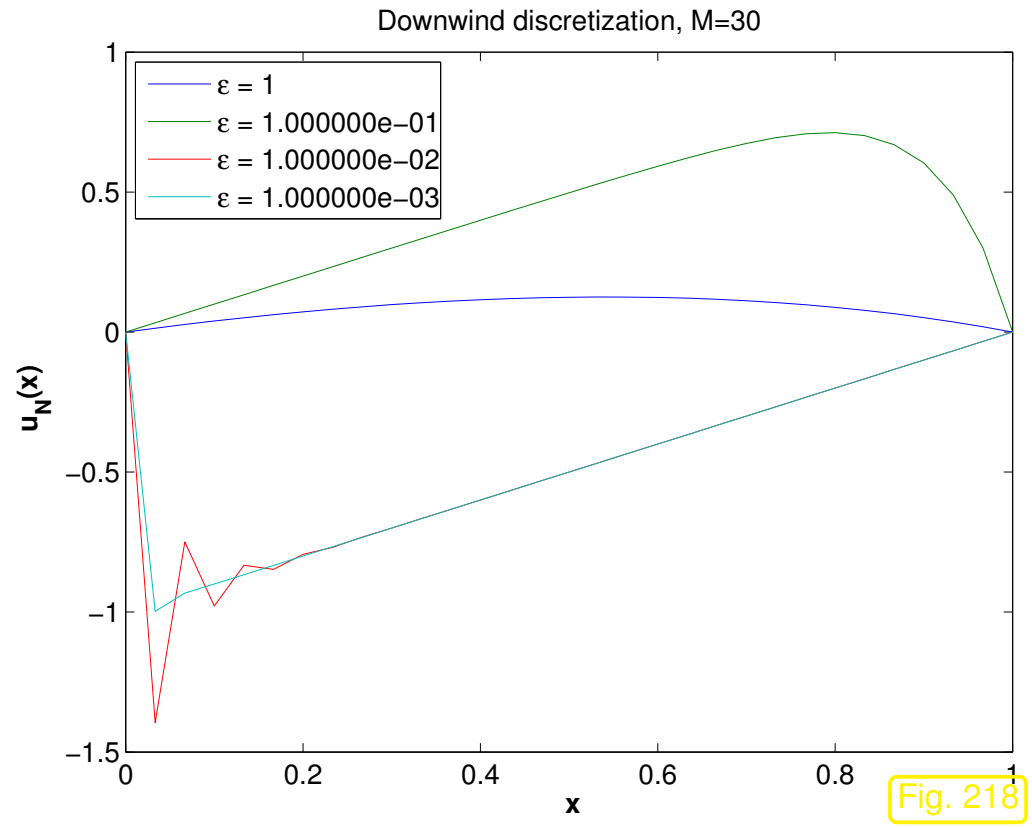
$$-\frac{\epsilon}{h} \mu_{i-1} + \left(\frac{2\epsilon}{h} - 1\right) \mu_i + \left(-\frac{\epsilon}{h} + 1\right) \mu_{i+1} = hf(ih), \quad i = 1, \dots, M - 1, \quad (7.2.20)$$

Example 7.2.21 (One-sided difference approximation of convective terms).

Model problem of Ex. 7.2.17, discretizations (7.2.19) and (7.2.20).



backward difference quotient



forward difference quotient

Only the discretization of $\frac{du}{dx}$ based on the backward difference quotient generates qualitatively correct (piecewise linear) discrete solutions (a “good method”).

If the forward difference quotient is used, the discrete solutions may violate the maximum principle of Thm. 7.1.15 (a “bad method”).

How can we tell a good method from a bad method by merely examining the system matrix?



Heuristic criterion for $\epsilon \rightarrow 0$ -robust stability of nodal finite element Galerkin discretization/finite difference discretization of *singularly perturbed* scalar linear convection-diffusion BVP (7.2.1) (with Dirichlet b.c.):

(Linearly interpolated) discrete solution satisfies **maximum principle** (5.7.3).



System matrix complies with sign-conditions (5.7.9)–(5.7.11).

Nodal finite element Galerkin discretization $\hat{=}$ basis expansion coefficients μ_i of Galerkin solution $u_N \in V_N$ double as point values of u_N at interpolation nodes. This is satisfied for Lagrangian finite element methods (\rightarrow Sect. 3.4) when standard nodal basis functions according to (3.4.3) are used.

Recall the sign-conditions (5.7.9)–(5.7.11) for the system matrix \mathbf{A} arising from nodal finite element Galerkin discretization or finite difference discretization:

- (5.7.9): positive diagonal entries,

$$(\mathbf{A})_{ii} > 0,$$

- (5.7.10): non-positive off-diagonal entries,

$$(\mathbf{A})_{ij} \leq 0, \text{ if } i \neq j,$$

- “(5.7.11)”: diagonal dominance,

$$\sum_j (\mathbf{A})_{ij} \geq 0.$$

These conditions are met for *equidistant meshes in 1D*

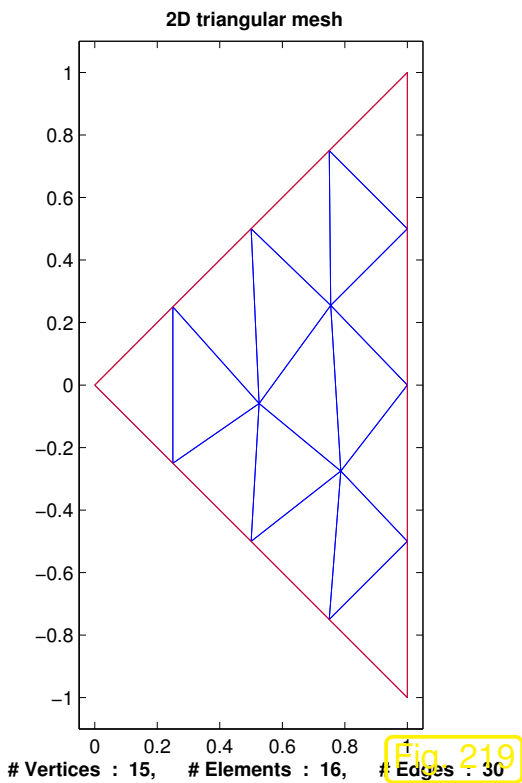
- for the standard $\mathcal{S}_1^0(\mathcal{M})$ -Galerkin discretization (7.2.15), **provided that** $|\epsilon h^{-1}| \geq \frac{1}{2}$,
- when using *backward* difference quotients for the convective term (7.2.19) for **any** choice of $\epsilon \geq 0$, $h > 0$,
- when using *forward* difference quotients for the convective term (7.2.20), **provided that** $|\epsilon h^{-1}| \geq 1$.

► Only the use of a *backward* difference quotient for the convective term guarantees the (discrete) maximum principle in an $\epsilon \rightarrow 0$ -robust fashion!

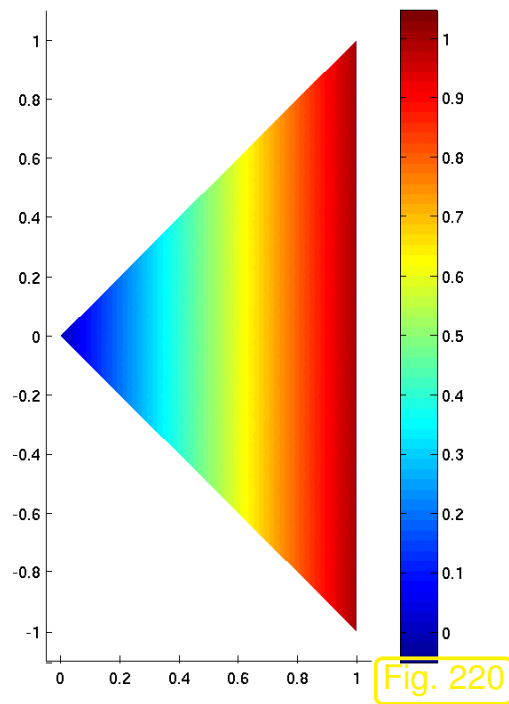
Terminology: Approximation of $\frac{du}{dx}$ by *backward* difference quotients $\hat{=}$ **upwinding**

Example 7.2.22 (Spurious Galerkin solution for 2D convection-diffusion BVP).

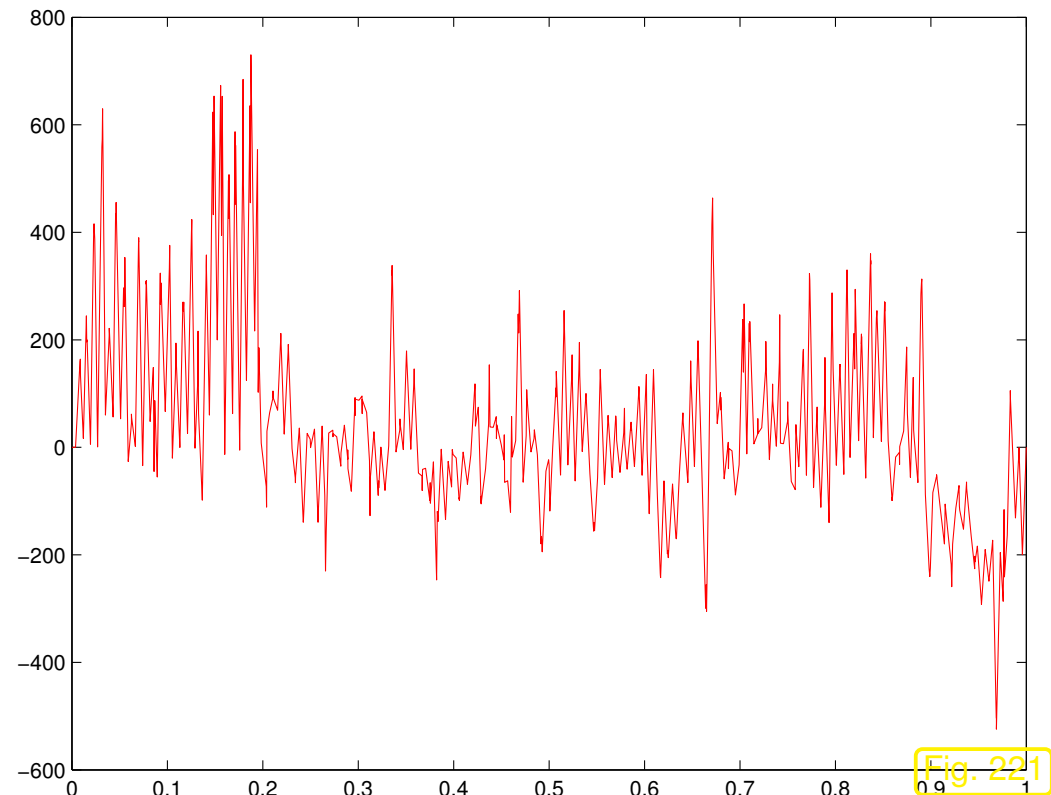
- Triangle domain $\Omega = \{(x, y) : 0 \leq x \leq 1, -x \leq y \leq x\}$.
- Velocity $\mathbf{v}(\mathbf{x}) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ \triangleright (7.2.1) becomes $-\epsilon \Delta u + u_x = 1$.
- Exact solution: $u_\epsilon(x_1, x_2) = x - \frac{1}{1-e^{-1/\epsilon}}(e^{-(1-x_1)/\epsilon} - e^{-1/\epsilon})$, Dirichlet boundary conditions set accordingly
- Standard Galerkin discretization by means of linear finite elements on sequence of triangular mesh created by regular refinement.



Coarse initial mesh



Exact solution
 $(\epsilon = 10^{-10})$



Standard Galerkin solution on $x_2 = 0$ -line

As expected: spurious oscillations mar Galerkin solution

➤ Difficulty observed in 1D also haunts discretization in higher dimensions.



Issue: extension of upwinding idea to $d > 1$

7.2.2.1 Upwind quadrature

Revisit 1D model problem

$$-\epsilon \frac{d^2 u}{dx^2} + \frac{du}{dx} = f(x) \quad \text{in } \Omega, \quad u(0) = 0, \quad u(1) = 0, \quad (7.2.14)$$

with variational formulation, see Rem. 7.2.2:

$$u \in H_0^1(]0, 1[): \quad \underbrace{\epsilon \int_0^1 \frac{du}{dx}(x) \frac{du}{dx}(x) dx + \int_0^1 \frac{du}{dx}(x) v(x) dx}_{=: a(u, v)} = \underbrace{\int_0^1 f(x) v(x) dx}_{=: \ell(v)} \quad \forall v \in H_0^1(]0, 1[). \quad \text{convective term}$$

Linear finite element Galerkin discretization on equidistant mesh \mathcal{M} with M cells, meshwidth $h = \frac{1}{M}$,
cf. Sect. 1.5.1.2.

We opt for the composite trapezoidal rule

$$\int_0^1 \psi(x) dx \approx h \sum_{j=1}^{M-1} \psi(jh), \quad \text{for } \psi \in C^0([0, 1]), \psi(0) = \psi(1) = 0.$$

for evaluation of convective term in bilinear form **a**:

$$\int_0^1 \frac{du_N}{dx}(x) v_N(x) dx \approx h \sum_{j=1}^{M-1} \frac{du_N}{dx}(jh) v(hj), \quad v_N \in \mathcal{S}_{1,0}^0(\mathcal{M}). \quad (7.2.23)$$

Note: this is not a valid formula, because $\frac{du_N}{dx}(jh)$ is *ambiguous*, since $\frac{du_N}{dx}$ is discontinuous at nodes of the mesh for $u_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$!

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Up to now we resolved this ambiguity by the policy of *local* quadrature, see Sect. 3.5.4: quadrature rule applied locally on each cell with all information taken from that cell.

However:

Convection transports information in the direction of **v**!

Idea:

Use **upstream/upwind** information to evaluate $\frac{du_N}{dx}(jh)$ in (7.2.23)

$$\frac{du_N}{dx}(jh) := \lim_{\delta \rightarrow 0} \frac{du_N}{dx}(jh - \delta) = \frac{du_N}{dx} \Big|_{x_{j-1}, x_j} .$$

$\hat{=}$ **upwind quadrature**

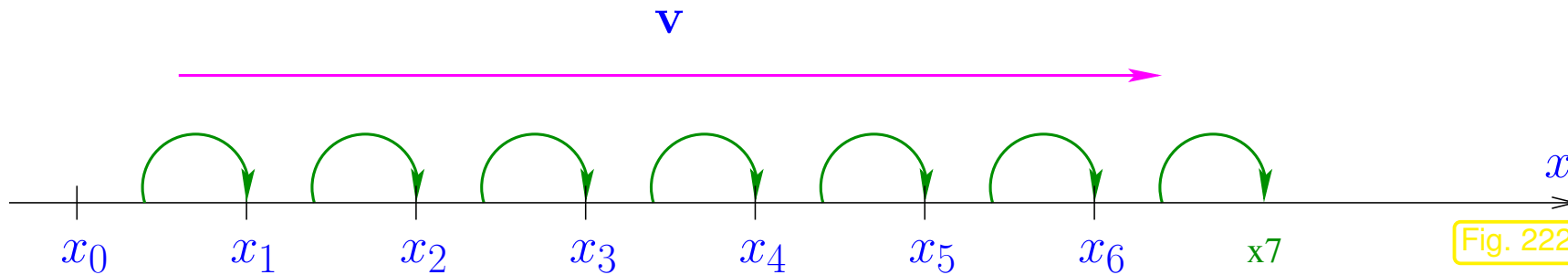
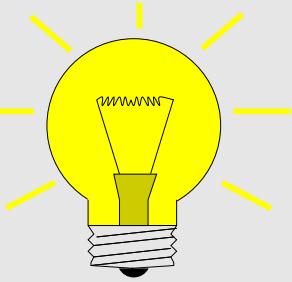


Fig. 222

Upwind quadrature yields the following contribution of the discretized convective term to the linear system using the basis expansion $u_N = \sum_{l=1}^{M-1} \mu_l b_N^l$ into *locally supported* nodal basis functions (“tent functions”)

$$\int_0^1 \sum_{l=1}^{M-1} \mu_l \frac{db_N^l}{dx}(x) b_N^i(x) dx \stackrel{(7.2.23)}{\approx} h \frac{\mu_i - \mu_{i-1}}{h} ,$$

where we used

- $b_N^i(jh) = \delta_{ij}$, see (1.5.77),
- $\frac{du_N}{dx} \Big|_{x_{j-1}, x_j} = \frac{\mu_i - \mu_{i-1}}{h}$ from (1.5.79).

► Linear system from upwind quadrature:

$$\left(-\frac{\epsilon}{h} - 1\right) \mu_{i-1} + \left(\frac{2\epsilon}{h} + 1\right) \mu_i + -\frac{\epsilon}{h} \mu_{i+1} = hf(ih), \quad i = 1, \dots, M-1, \quad (7.2.19)$$

which is the **same** as that obtained from a backward finite difference discretization of $\frac{du}{dx}$!

The idea of upwind quadrature can be generalized to $d > 1$: we consider $d = 2$ and linear Lagrangian finite element Galerkin discretization on triangular meshes, see Sect. 3.2.

- 1 Approximation of contribution of convective terms to bilinear form by means of *global trapezoidal rule*:

$$\int_{\Omega} (\mathbf{v} \cdot \mathbf{grad} u) v \, d\mathbf{x} \approx \sum_{\mathbf{p} \in \mathcal{N}(\mathcal{M})} \left(\frac{1}{3} \sum_{K \in \mathcal{U}_{\mathbf{p}}} |K| \right) \cdot \mathbf{v}(\mathbf{p}) \cdot \mathbf{grad} u(\mathbf{p}) v(\mathbf{p}). \quad (7.2.24)$$

ambiguous for $u \in \mathcal{S}_1^0(\mathcal{M})$!

notation: $\mathcal{U}_{\mathbf{p}} := \{K \in \mathcal{M} : \mathbf{p} \in \overline{K}\}$

For a continuous function $f : \Omega \mapsto \mathbb{R}$ the trapezoidal rule can easily be derived from the 2D *composite* trapezoidal rule based on

$$\int_K f(\mathbf{x}) \, d\mathbf{x} \approx \frac{|K|}{3} (f(\mathbf{a}^1) + f(\mathbf{a}^2) + f(\mathbf{a}^3)), \quad (3.2.18)$$

where the \mathbf{a}^i , $i = 1, 2, 3$, are the vertices of the triangle K .

$$\begin{aligned} \blacktriangleright \int_{\Omega} f(\mathbf{x}) \, d\mathbf{x} &= \sum_{K \in \mathcal{M}} \int_K f(\mathbf{x}) \, d\mathbf{x} \approx \sum_{K \in \mathcal{M}} \frac{|K|}{3} (f(\mathbf{a}_K^1) + f(\mathbf{a}_K^2) + f(\mathbf{a}_K^3)) \\ &= \sum_{\mathbf{p} \in \mathcal{N}(\mathcal{M})} \left(\frac{1}{3} \sum_{K \in \mathcal{U}_{\mathbf{p}}} |K| \right) f(\mathbf{p}), \end{aligned}$$

by changing the order of summation.

- ② Fix the ambiguous value of $\mathbf{v}(\mathbf{p}) \cdot \mathbf{grad} u_N(\mathbf{p})$, $u_N \in \mathcal{S}_1^0(\mathcal{M})$, by taking the gradient from the triangle upstream to the node \mathbf{p} :

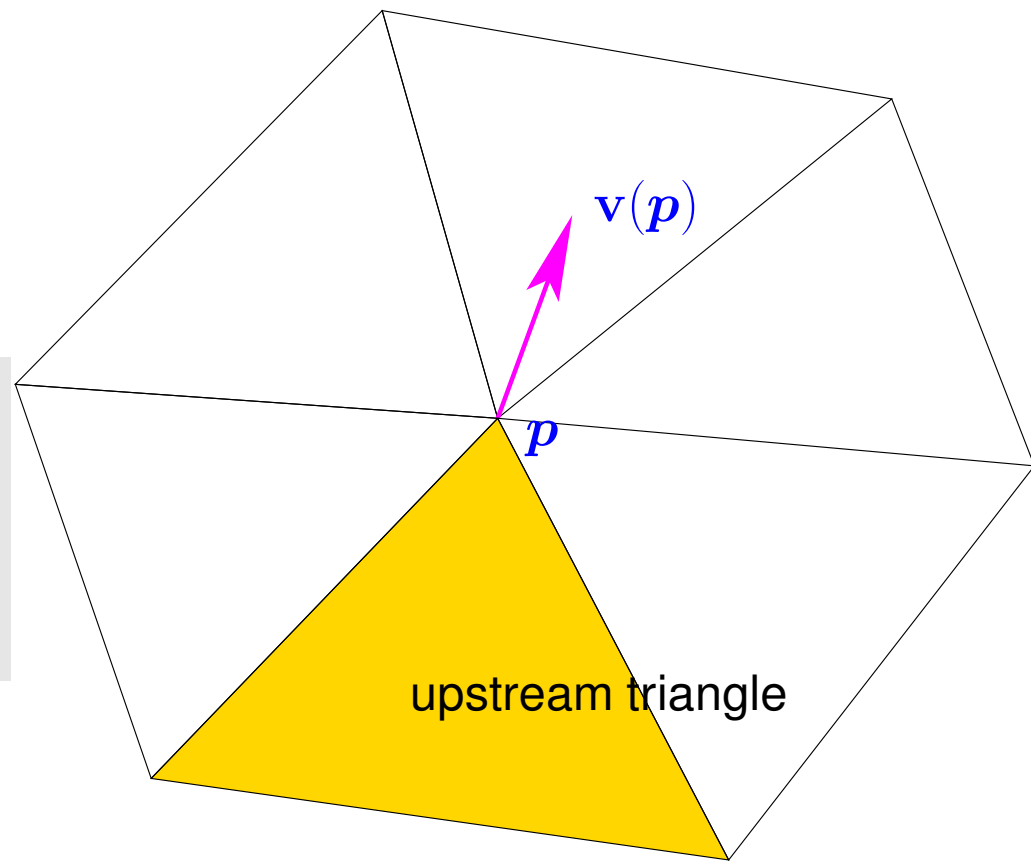
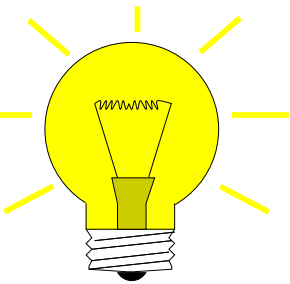


Fig. 223

Idea: Use **upstream/upwind** information to evaluate $\mathbf{grad} u_N(\mathbf{p})$ in (7.2.24)

$$\mathbf{v}(\mathbf{p}) \cdot \mathbf{grad} u_N(\mathbf{p}) := \lim_{\delta \rightarrow 0} \mathbf{v}(\mathbf{p}) \cdot \mathbf{grad} u_N(\mathbf{p} - \delta \mathbf{v}(\mathbf{p})) . \quad (7.2.25)$$

$\hat{=}$ general **upwind quadrature**



Note: By (7.1.1) the vector $\mathbf{v}(\mathbf{p})$ supplies the direction of the streamline through \mathbf{p} . Hence, $-\mathbf{v}(\mathbf{p})$ is the direction from which information is “carried into \mathbf{p} ” by the flow.

Contribution of convective term to the i -th row of the final linear system of equations (test function = tent function b_N^i)

$$\underbrace{\left(\frac{1}{3} \sum_{K \in \mathcal{U}_i} |K|\right)}_{=: U_i} \mathbf{v}(\mathbf{x}^i) \cdot \mathbf{grad} u_N|_{K_u},$$

where K_u is the upstream triangle of \mathbf{p} . ▷

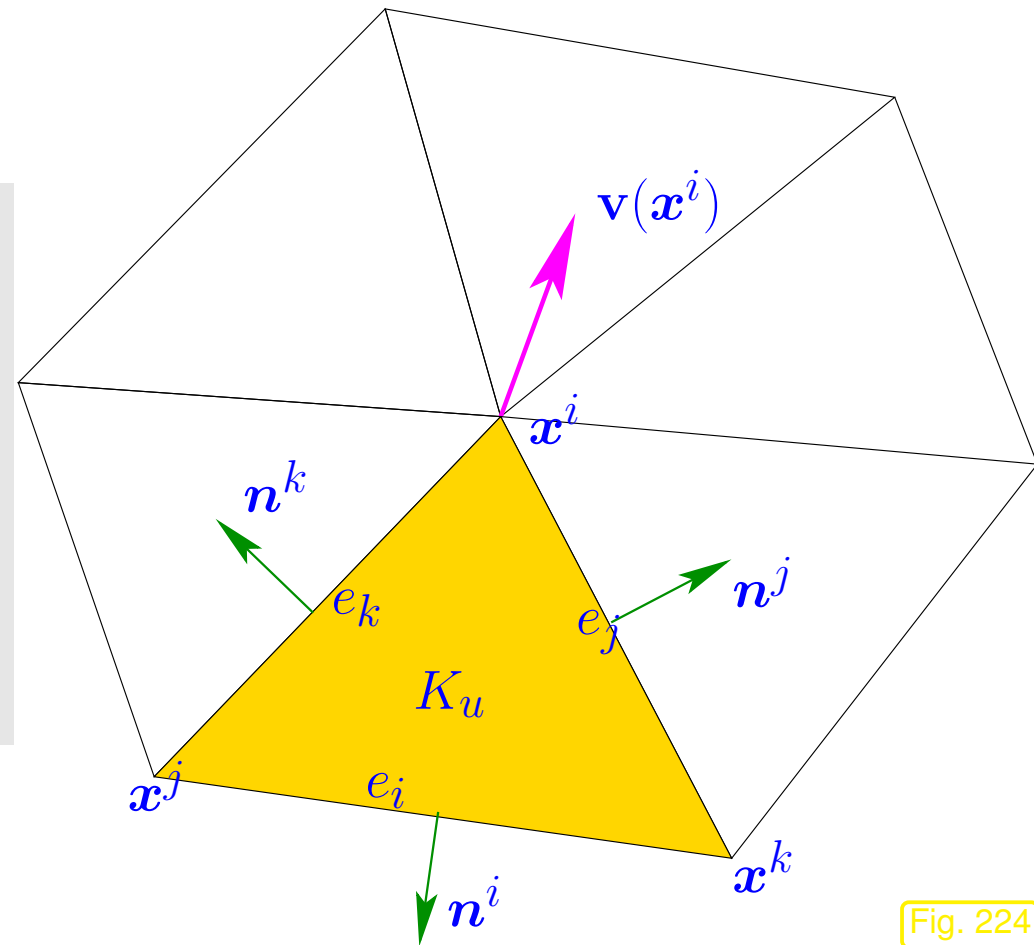


Fig. 224

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Using the expressions for the gradients of barycentric coordinate functions from Sect. 3.2.5

$$\mathbf{grad} \lambda_* = -\frac{|e_i|}{2|K|} \mathbf{n}^*, \quad * = i, j, k, \quad \text{see Fig. 224,}$$

and the nodal basis expansion of u_N , we obtain for the convective contribution to the i -th line of the final linear system

$$\frac{U_i}{2|K_u|} \left(\underbrace{-\|\mathbf{x}^j - \mathbf{x}^k\| \mathbf{n}^i \cdot \mathbf{v}(\mathbf{x}^i) \mu_i - \|\mathbf{x}^i - \mathbf{x}^j\| \mathbf{n}^k \cdot \mathbf{v}(\mathbf{x}^i) \mu_k - \|\mathbf{x}^i - \mathbf{x}^k\| \mathbf{n}^j \cdot \mathbf{v}(\mathbf{x}^i) \mu_j}_{\leftrightarrow \text{diagonal entry}} \right)$$

By the very definition of the upstream triangle K_u we find

$$\mathbf{n}^i \cdot \mathbf{v}(\mathbf{x}^i) \leq 0 \quad , \quad \mathbf{n}^k \cdot \mathbf{v}(\mathbf{x}^i) \geq 0 \quad , \quad \mathbf{n}^j \cdot \mathbf{v}(\mathbf{x}^i) \geq 0 \quad .$$

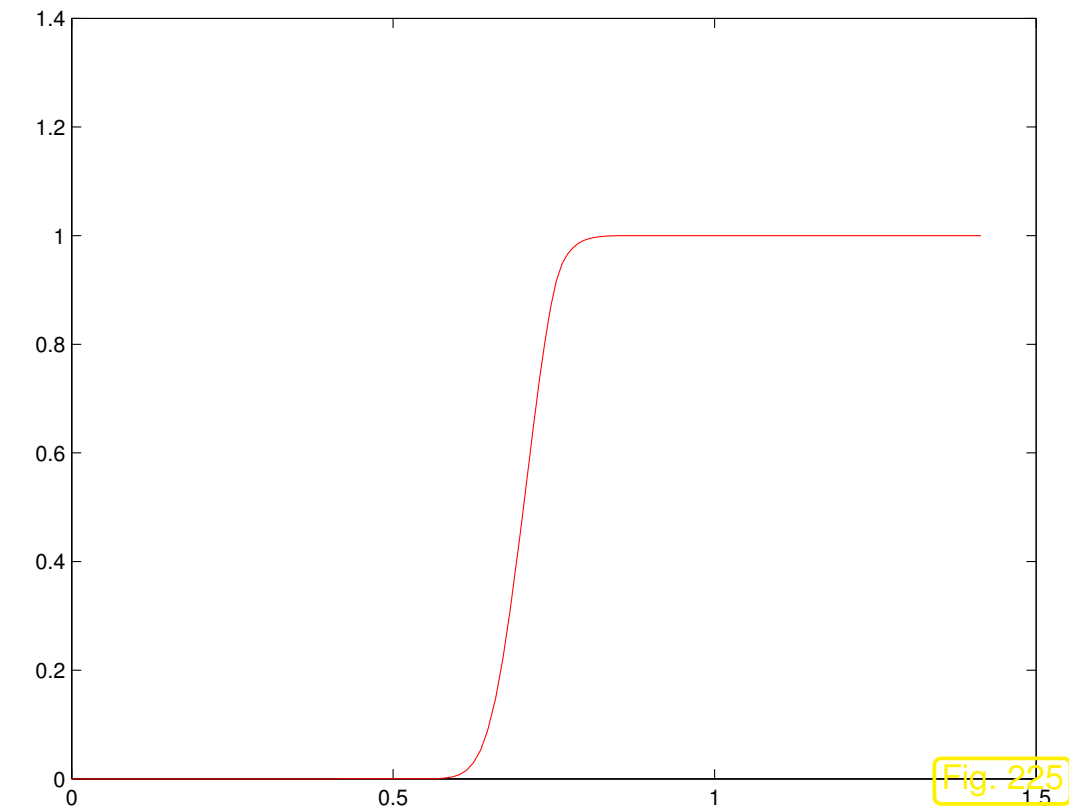
➤ sign conditions (5.7.9), (5.7.10) are satisfied, (5.7.11) is obvious from $\lambda_i + \lambda_j + \lambda_k = 1$, which means $\mathbf{grad}(\lambda_i + \lambda_j + \lambda_k) = 0$.

Example 7.2.26 (Upwind quadrature discretization).

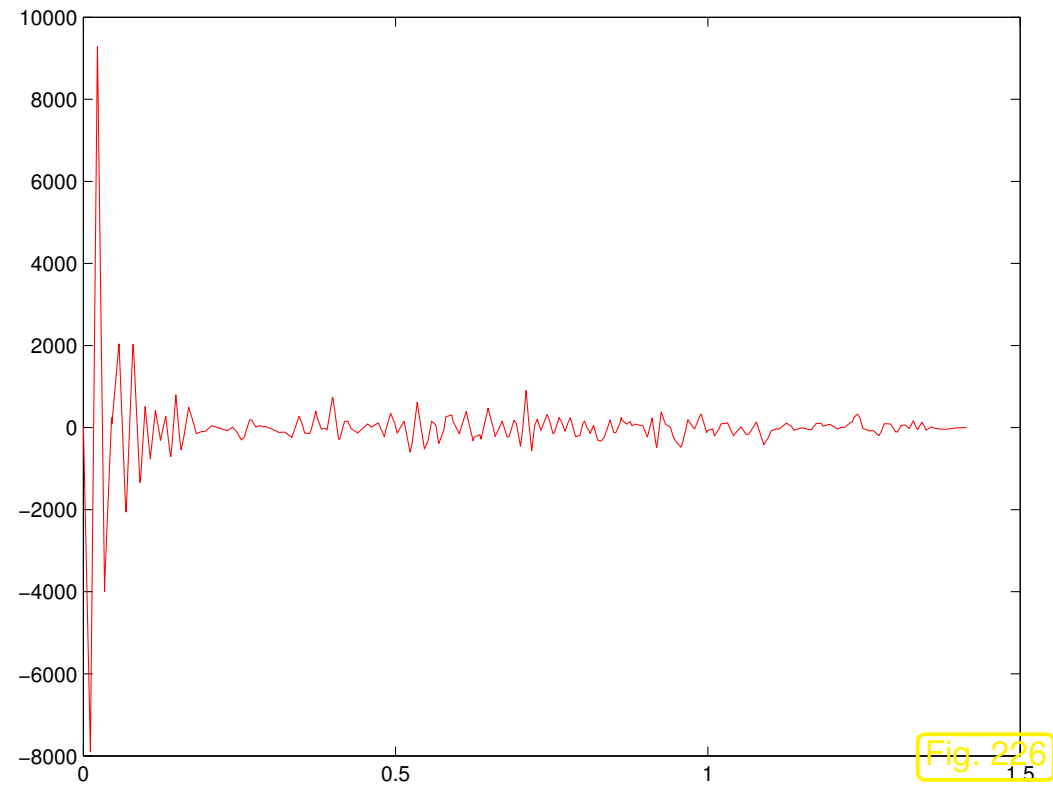
- $\Omega = [0, 1]^2$
- $-\epsilon \Delta u + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot \mathbf{grad} u = 0$
- Dirichlet boundary conditions: $u(x, y) = 1$ for $x > y$ and $u(x, y) = 0$ for $x \leq y$
- Limiting case ($\epsilon \rightarrow 0$): $u(x, y) = 1$ for $x > y$ and $u(x, y) = 0$ for $x \leq y$

- layer along the diagonal from $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ to $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ in the limit $\epsilon \rightarrow 0$
- 2D triangular Delaunay triangulation, see Rem. 4.2.4
- linear finite element upwind quadrature discretization

► Monitored: discrete solutions along diagonal from $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ to $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ for $\epsilon = 10^{-10}$.



upwind quadrature solution



standard Galerkin solution

Upwind quadrature scheme respects maximum principle, whereas the standard Galerkin solution is rendered useless by spurious oscillations.



7.2.2.2 Streamline diffusion

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

We take another look at the 1D upwind discretization of (7.2.14) and view it from a different perspective.

1D **upwind** (finite difference) discretization of (7.2.14):

$$\left(-\frac{\epsilon}{h} - 1\right) \mu_{i-1} + \left(\frac{2\epsilon}{h} + 1\right) \mu_i + -\frac{\epsilon}{h} \mu_{i+1} = hf(ih) \quad .i = 1, \dots, M - 1 . \quad (7.2.19)$$



$$(\epsilon+h/2) \underbrace{\frac{-\mu_{i-1} + 2\mu_i - \mu_{i+1}}{h^2}}_{\hat{=} \text{ difference quotient for } \frac{d^2u}{dx^2}} + \underbrace{\frac{-\mu_{i-1} + \mu_{i+1}}{2h}}_{\hat{=} \text{ difference quotient for } \frac{du}{dx}} = f(ih),$$

for $i = 1, \dots, M - 1$.

Upwinding = h -dependent enhancement of diffusive term

artificial diffusion/viscosity

We also observe that the upwinding strategy just adds the *minimal amount of diffusion* to make the resulting system matrix comply with the conditions (5.7.9)–(5.7.11), which ensure that the discrete solution satisfies the maximum principle.

Issue: How to extend the trick of adding artificial diffusion to $d > 1$?

Well, just add an extra h -dependent multiple of $-\Delta$! Let's try.

Example 7.2.27 (Effect of added diffusion).

Convection-diffusion boundary value problem ((7.2.1) with $\mathbf{v} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$)

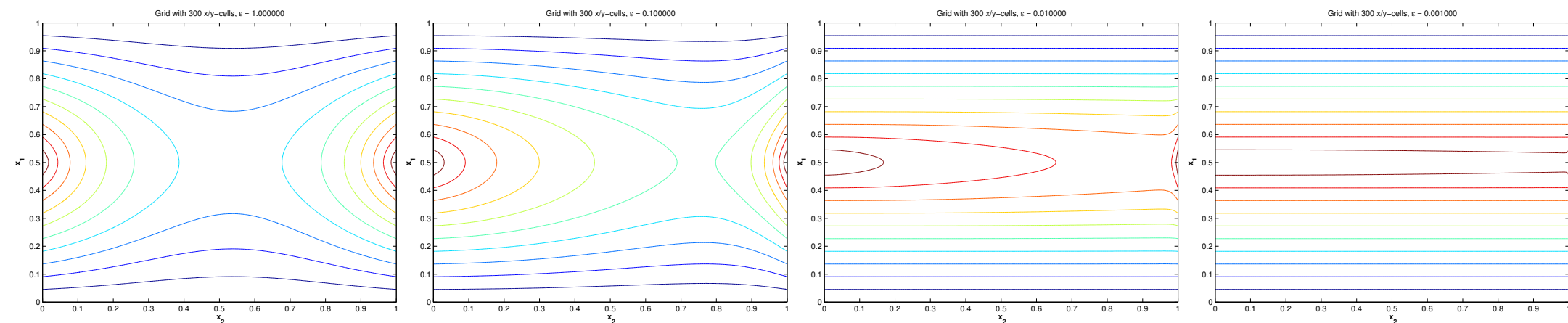
$$-\epsilon \Delta u + \frac{\partial u}{\partial x_1} = 0 \quad \text{in } \Omega =]0, 1[^2, \quad u = g \quad \text{on } \partial\Omega.$$

Here, Dirichlet data $g(\mathbf{x}) = 1 - 2|x_2 - \frac{1}{2}|$.

Thus, for $\epsilon \approx 0$ we expect $u \approx g$, because the Dirichlet data are just transported in x_1 -direction and there are no boundary layers.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



$\epsilon = 1$

$\epsilon = 0.1$

$\epsilon = 0.01$

$\epsilon = 0.001$

7.2

p. 764

Stronger diffusion leads to “smearing” of features that the flow field transports into the interior of the domain.



(Too much) artificial diffusion \Rightarrow smearing of internal layers

(We are no longer solving the right problem!)

Remark 7.2.28 (Internal layers).

Pure transport problem:

$$\mathbf{v} \cdot \text{grad } u = 0 \quad \text{in } \Omega,$$

where $\Omega =]0, 1[^2$, $\mathbf{v} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, $\epsilon = 10^{-4}$,

Dirichlet b.c. that can only partly be fulfilled: $u = 1$ on $\{x_1 = 0\} \cup \{x_2 = 1\}$, $u = 0$ on $\{x_1 = 1\} \cup \{x_2 = 0\}$.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

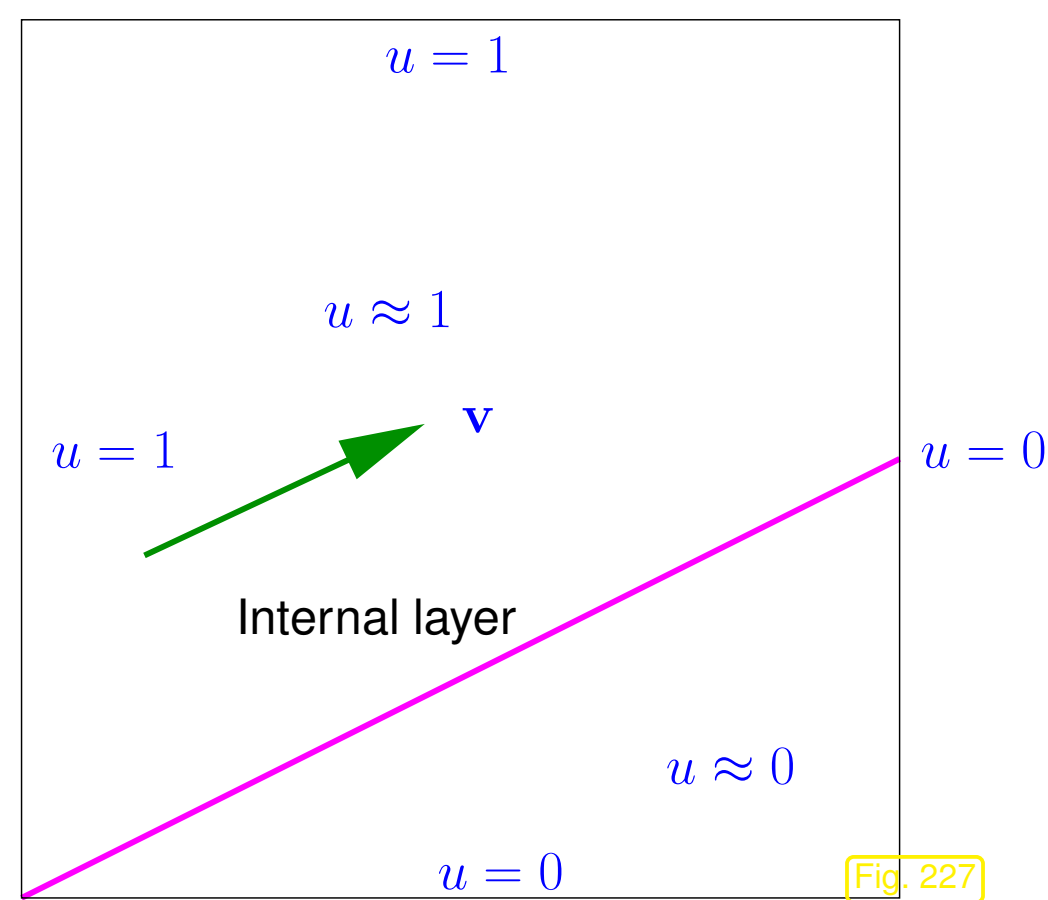


Fig. 227

Solution of pure transport problem with discontinuous boundary data

- displays a discontinuity across the streamline emanating from the point of discontinuity on $\partial\Omega$,
- is *smooth along streamlines*.

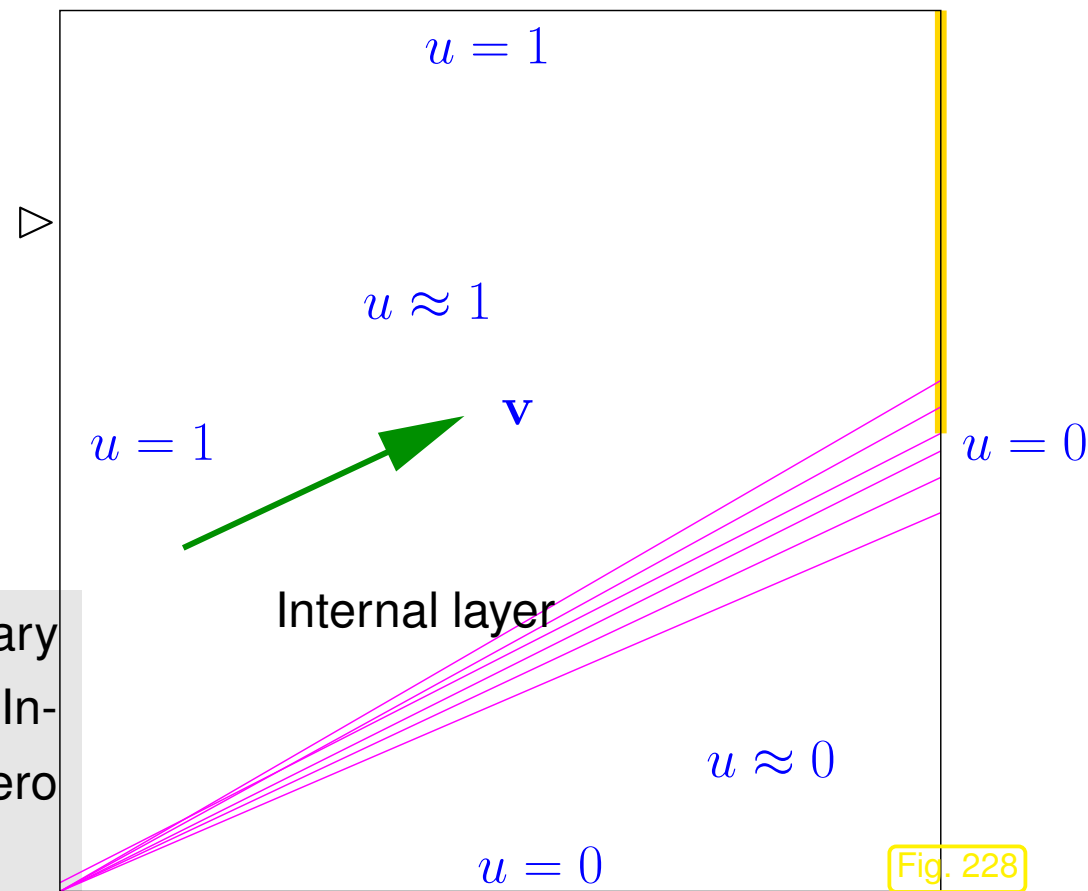
Qualitative solution of

$$-\delta \Delta + \mathbf{v} \cdot \mathbf{grad} u = 0 \quad \text{in } \Omega,$$

with $\delta > 0$, the same boundary data

➤ Smearing of internal layer !

As in Ex. 7.2.5, we would also find a boundary layer which is marked in gold in the figure. Inside this boundary layer the solution drops to zero abruptly.



Heuristics: If the solution is smooth along streamlines, then adding diffusion in the direction of streamlines cannot do much harm.

What does “diffusion in a direction” mean?

☞ Think of a generalized Fourier's law (2.5.3) for $d = 2$, e.g.,

$$\mathbf{j}(\mathbf{x}) = - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{grad} u(\mathbf{x}) .$$

This means, only a temperature variation in x_1 -direction triggers a heat flow.

☞ diffusion in a direction $\mathbf{v} \in \mathbb{R}^2$

$$\mathbf{j}(\mathbf{x}) = -\mathbf{v}\mathbf{v}^T \mathbf{grad} u(\mathbf{x}) \quad (7.2.29)$$

Such an extended Fourier's law is an example of **anisotropic diffusion**.

Anisotropic diffusion can simply be taken into account in variational formulations and Galerkin discretization by replacing the heat conductivity κ /stiffness σ with a symmetric, positive (semi-)definite matrix, the **diffusion tensor**.

 R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

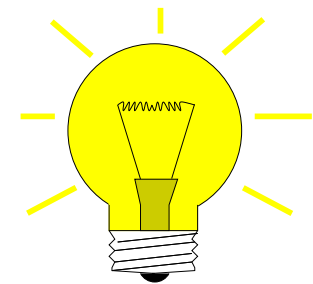
SAM, ETHZ

Idea: **Anisotropic artificial diffusion** in streamline direction

On cell K replace: $\epsilon \leftarrow \underbrace{\epsilon \mathbf{I} + \delta_K \mathbf{v}_K \mathbf{v}_K^T}_{\text{new diffusion tensor}} \in \mathbb{R}^{2,2} .$

$\mathbf{v}_K \hat{=}$ local velocity (e.g., obtained by averaging)

$\delta_K > 0 \hat{=}$ method parameter controlling the strength of anisotropic diffusion



This idea underlies the so-called **streamline-diffusion method**.

Thus, (for the model problem) Galerkin discretization may target the variational problem

$$\int_{\Omega} (\epsilon \mathbf{I} + \delta_K \mathbf{v}_K \mathbf{v}_K^T) \mathbf{grad} u \cdot \mathbf{grad} v + \mathbf{v}(\mathbf{x}) \cdot \mathbf{grad} u v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega). \quad (7.2.30)$$



This tampering affects the solution u
(solution of (7.2.30) \neq solution of (7.2.1))

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Desirable:

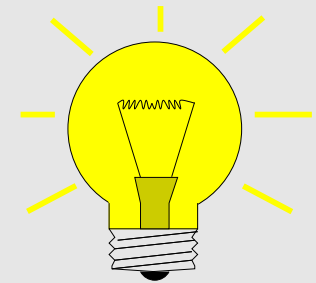
Maintain **consistency** of variational problem!

Definition 7.2.33 (Consistent modifications of variational problems).

A variational problem is called a **consistent modification** of another, if both possess the same (unique) solution(s).

Note: the variational crimes investigated in Sect. 5.5 represent non-consistent modifications.

Ensuring consistency for streamline-upwind variational problem:



Idea: Add anisotropic diffusion through a **residual term** that vanishes for the exact solution u

► streamline-upwind variational problem: given mesh \mathcal{M} seek $u \in H_0^1(\Omega) \cap H^2(\mathcal{M})$

$$\int_{\Omega} \epsilon \mathbf{grad} u \cdot \mathbf{grad} v + \mathbf{v}(\mathbf{x}) \cdot \mathbf{grad} u v \, d\mathbf{x} + \underbrace{\sum_{K \in \mathcal{M}} \delta_K \int_K (-\epsilon \Delta u + \mathbf{v} \cdot \mathbf{grad} u - f) \cdot (\mathbf{v} \cdot \mathbf{grad} v) \, d\mathbf{x}}_{\text{stabilization term}} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega). \quad (7.2.34)$$

☞ Note that enhanced smoothness of u , namely in addition $u \in H^2(K)$ for all $K \in \mathcal{M}$, is required to render (7.2.34) meaningful (\rightarrow Sobolev space $H^2(\mathcal{M})$).

Note: in the case of Galerkin discretization based on $V_{N,0} = \mathcal{S}_1^0(\mathcal{M})$, we find $\Delta u_N = 0$ in each $K \in \mathcal{M}$.

For Galerkin discretization of (7.2.34) by means of linear Lagrangian finite elements, the local control parameters δ_K are usually chosen according to the rule

$$\delta_K := \begin{cases} \epsilon^{-1} h_K^2 & , \text{ if } \frac{\|\mathbf{v}\|_{K,\infty} h_K}{2\epsilon} \leq 1 , \\ h_K & , \text{ if } \frac{\|\mathbf{v}\|_{K,\infty} h_K}{2\epsilon} > 1 . \end{cases}$$

which is suggested by theoretical investigations and practical experience, *cf.* 1D artificial diffusion (7.2.19) for a reason why to choose $\delta_K \sim h_K$ for small ϵ .

Example 7.2.35 (Streamline-diffusion discretization).

Exactly the same setting as in Ex. 7.2.26 with the upwind quadrature approach replaced with the streamline diffusion method.

```
1 function Aloc = STIMASUPGLFE (Vertices, flag, QuadRule, VHandle,  
   a,d1,d2, varargin)  
2 % ALOC = STIMA_SUPG_LFE (VERTICES) provides the extra terms for SUPG  
   stabilization to be  
3 % added to the Galerkin element matrix for linear finite elements  
4 %  
5 % VERTICES is 3-by-2 matrix specifying the vertices of the current element  
6 % in a row wise orientation.  
7 %  
8 % a: diffusivity  
9 % d1 d2: apriori chosen constants for SUPG-modification  
10 %  
11 % Flag not used, needed for interface to assemMat_LFE  
12 %  
13 % QUADRULE is a struct, which specifies the Gauss quadrature that is used  
14 % to do the numerical integration:  
15 % W Weights of the Gauss quadrature.  
16 % X Abscissae of the Gauss quadrature.e:  
17 %  
18 % VHANDLE is function handle for velocity field  
19  
20 % Preallocate memory for element matrix  
21 Aloc = zeros (3,3);  
22
```

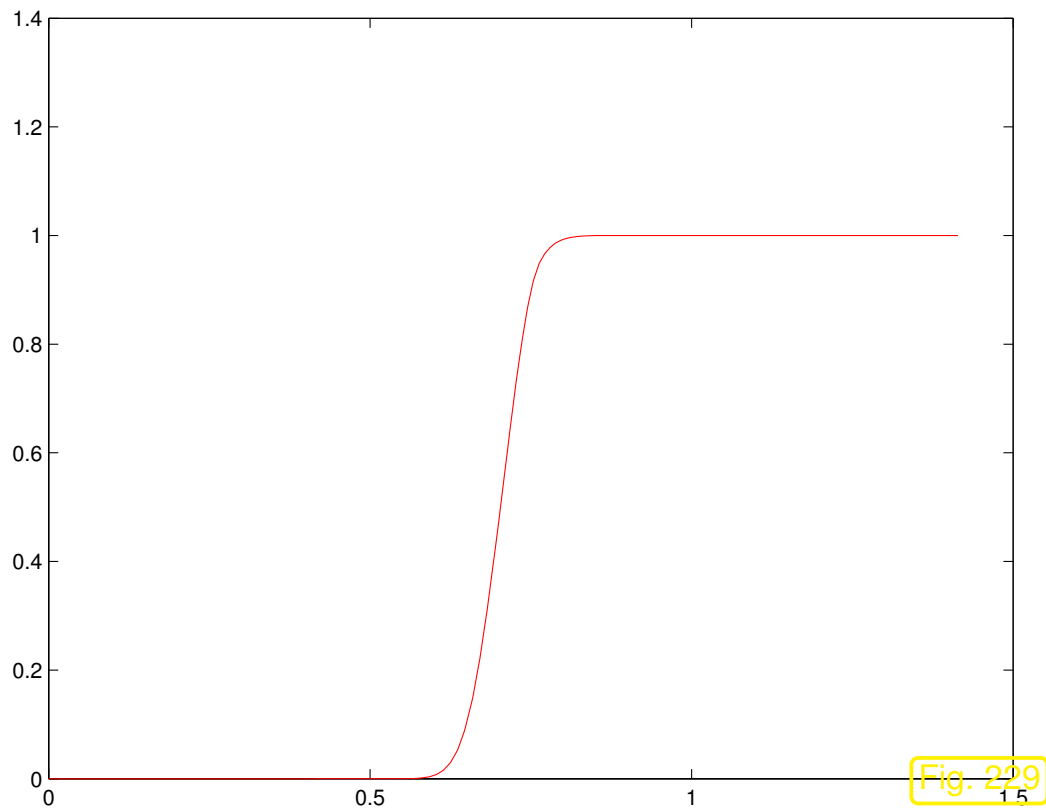
```
23 % Analytic computation of entries of element matrix using barycentric
24 % coordinates, see Sect. 3.2.5
25 l1x = Vertices(2,2)-Vertices(3,2);
26 l1y = Vertices(3,1)-Vertices(2,1);
27 l2x = Vertices(3,2)-Vertices(1,2);
28 l2y = Vertices(1,1)-Vertices(3,1);
29 l3x = Vertices(1,2)-Vertices(2,2);
30 l3y = Vertices(2,1)-Vertices(1,1);
31
32 % Compute element mapping
33
34 P1 = Vertices(1,:);
35 P2 = Vertices(2,:);
36 P3 = Vertices(3,:);
37
38 BK = [ P2 - P1 ; P3 - P1 ];           % transpose of transformation
      matrix                             % twice the area of the triangle
39 det_BK = abs(det(BK));
40
41 nPoints = size(QuadRule.w,1);
42
43 % Quadrature points in actual element stored as rows of a matrix
44 x = QuadRule.x*BK + ones(nPoints,1)*P1;
45
```

```
46 % Evaluate coefficient function (velocity) at quadrature nodes
47 c =VHandle(x,varargin{:});
48 % Entries of anisotropic diffusion tensor
49 FHandle=[c(:,1).*c(:,1) c(:,1).*c(:,2) c(:,2).*c(:,1)
50          c(:,2).*c(:,2)];
51 % Compute local PecletNumber for SUPG control parameter
52 hK=max([norm(P2-P1), norm(P3-P1), norm(P2-P3)]);
53 v_infK=max(abs(c(:))); PK=v_infK*hK/(2*a);
54 % Apply quadrature rule and fix constant part
55 w = QuadRule.w; e = sum((FHandle.*[w w w w]), 1);
56 te = (reshape(e,2,2)')/det_BK;
57
58 % Compute Aloc values
59 Aloc(1,1) = (te*[11x 11y]')'*[11x 11y]';
60 Aloc(1,2) = (te*[11x 11y]')'*[12x 12y]';
61 Aloc(1,3) = (te*[11x 11y]')'*[13x 13y]';
62 Aloc(2,2) = (te*[12x 12y]')'*[12x 12y]';
63 Aloc(2,3) = (te*[12x 12y]')'*[13x 13y]';
64 Aloc(3,3) = (te*[13x 13y]')'*[13x 13y]';
65 Aloc(2,1) = (te*[12x 12y]')'*[11x 11y]';
66 Aloc(3,1) = (te*[13x 13y]')'*[11x 11y]';
67 Aloc(3,2) = (te*[13x 13y]')'*[12x 12y]';
```

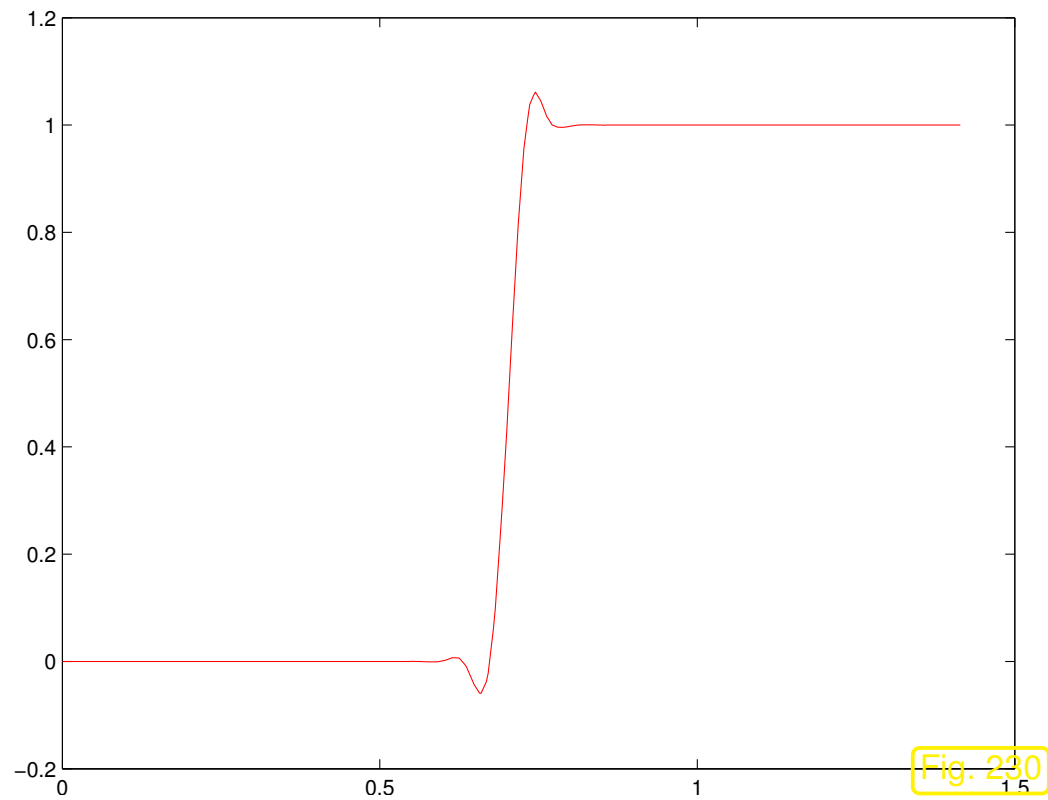
```

68
69 if (PK<=1), Aloc=d1*hK^2/a*Aloc;
70 else Aloc=d2*hK*Aloc; end
71
72 return

```



upwind quadrature solution



streamline-diffusion solution

Observations:

- The streamline upwind method does not exactly respect the maximum principle, but offers a better resolution of the internal layer compared with upwind quadrature (Parlance: streamline diffusion method is “less diffusive”).

◇ R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

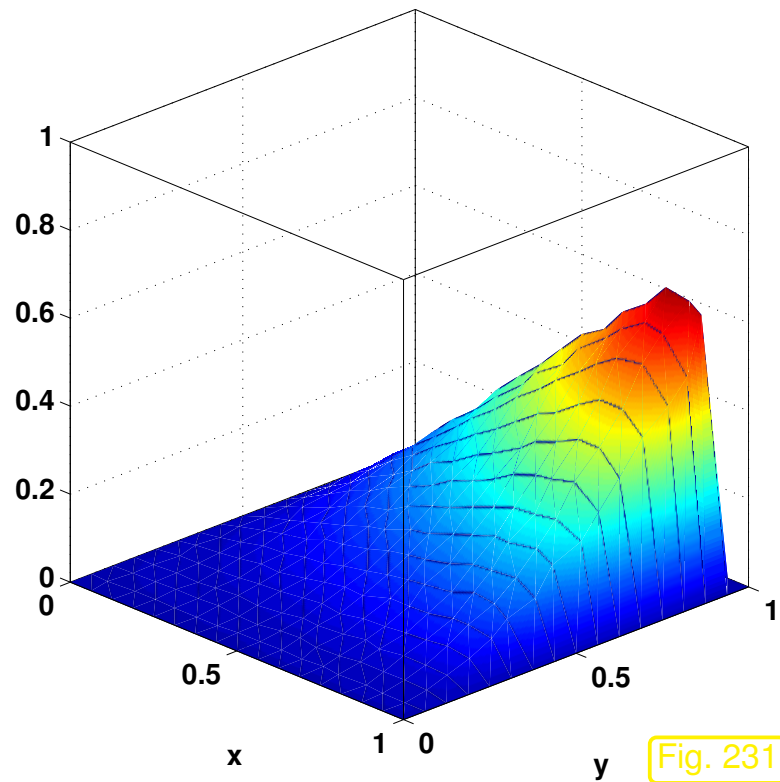
Example 7.2.37 (Convergence of SUPG and upwind quadrature FEM).

- $\Omega =]0, 1[^2$, model problem (7.2.1), $\mathbf{v}(\mathbf{x}) = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$, right hand side f such that

$$u_\varepsilon(x, y) = xy^2 - y^2 e^{2\frac{x-1}{\varepsilon}} - x e^{3\frac{y-1}{\varepsilon}} + e^{2\frac{x-1}{\varepsilon} + 3\frac{y-1}{\varepsilon}}.$$

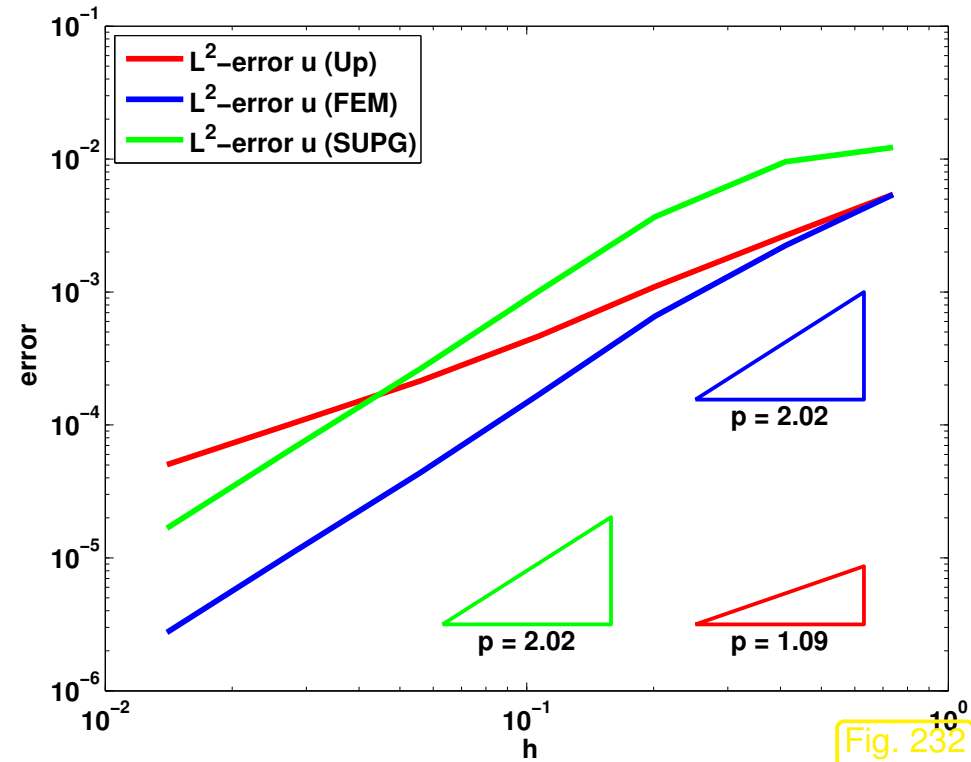
- Finite element discretization, $V_{0,N} = \mathcal{S}_1^0(\mathcal{M})$ und sequence of unstructured triangular “uniform” meshes, with
 - upwind quadrature stabilization from Sect. 7.2.2.1,
 - SUPG stabilization according to (7.2.34).

- Monitored: (Approximate) $L^2(\Omega)$ -norm of discretization error (computed with high-order local quadrature)



u_ϵ for $\epsilon = 1$

Fig. 231



Convergence for $\epsilon = 1$

Fig. 232

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Observation: SUPG stabilization does not affect $O(h_M^2)$ -convergence of $\|u - u_N\|_{L^2(\Omega)}$ for h -refinement and $h_M \rightarrow 0$, whereas upwind quadrature leads to worse $O(h_M)$ convergence of the L^2 -error norm.

7.3 Transient convection-diffusion BVP

Sect. 7.1.4 introduced the transient heat conduction model in a fluid, whose motion is described by a non-stationary velocity field (\rightarrow Sect. 7.1.1) $\mathbf{v} : \Omega \times]0, T[\mapsto \mathbb{R}^d$

$$\frac{\partial}{\partial t}(\rho u) - \operatorname{div}(\kappa \mathbf{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x}, t)u) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[, \quad (7.1.16)$$

where $u = u(\mathbf{x}, t) : \tilde{\Omega} \mapsto \mathbb{R}$ is the unknown temperature.

Assuming $\operatorname{div} \mathbf{v}(\mathbf{x}, t) = 0$, as in Sect. 7.2, by scaling we arrive at the model equation for transient convection-diffusion

$$\frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[, \quad (7.3.1)$$

supplemented with

- Dirichlet boundary conditions: $u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \forall \mathbf{x} \in \partial\Omega, \quad 0 < t < T,$
- initial conditions: $u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega.$

7.3.1 Method of lines

For the solution of IBVP (7.3.1) follow the general policy introduced in Sect. 6.1.3:

- ➊ Discretization in space on a *fixed* mesh \Rightarrow initial value problem for ODE
- ➋ Discretization in time (by suitable numerical integrator = timestepping)

For instance, in the case of Dirichlet boundary conditions,

$$\begin{cases} \frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f & \text{in } \tilde{\Omega} := \Omega \times]0, T[, \\ u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \forall \mathbf{x} \in \partial\Omega, 0 < t < T \quad , \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega . \end{cases} \quad (7.3.2)$$

 ← spatial discretization

$$\mathbf{M} \frac{d\vec{\mu}}{dt}(t) + \epsilon \mathbf{A} \vec{\mu}(t) + \mathbf{B} \vec{\mu}(t) = \vec{\varphi}(t) , \quad (7.3.3)$$

- where
- $\vec{\mu} = \vec{\mu}(t) :]0, T[\mapsto \mathbb{R}^N \hat{=}$ coefficient vector describing approximation $u_N(t)$ of $u(\cdot, t)$,
 - $\mathbf{A} \in \mathbb{R}^{N,N} \hat{=}$ s.p.d. matrix of discretized $-\Delta$, e.g., (finite element) Galerkin matrix,
 - $\mathbf{M} \in \mathbb{R}^{N,N} \hat{=}$ (lumped \rightarrow Rem. 6.2.34) mass matrix
 - $\mathbf{B} \in \mathbb{R}^{N,N} \hat{=}$ matrix for discretized convective term, e.g., Galerkin matrix, upwind quadrature matrix (\rightarrow Sect. 7.2.2.1), streamline diffusion matrix (\rightarrow Sect. 7.2.2.2).

Example 7.3.4 (Implicit Euler method of lines for transient convection-diffusion).

1D convection-diffusion IBVP:

$$\frac{\partial u}{\partial t} - \epsilon \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x} = 0, \quad u(x, 0) = \max(1 - 3|x - \frac{1}{3}|, 0), \quad u(0) = u(1) = 0. \quad (7.3.5)$$

- Spatial discretization on equidistant mesh with meshwidth $h = 1/N$:
 1. central finite difference scheme, see (7.2.15) (\leftrightarrow linear FE Galerkin discretization),
 2. upwind finite difference discretization, see (7.2.19),
- $\mathbf{M} = h\mathbf{I}$ (“lumped” mass matrix, see Rem. 6.2.34),
- Temporal discretization with uniform timestep $\tau > 0$:
 1. implicit Euler method, see (6.1.30),
 2. explicit Euler method, see (6.1.29),

Computations with $\epsilon = 10^{-5}$, implicit Euler discretization, $h = 0.01$, $\tau = 0.00125$:

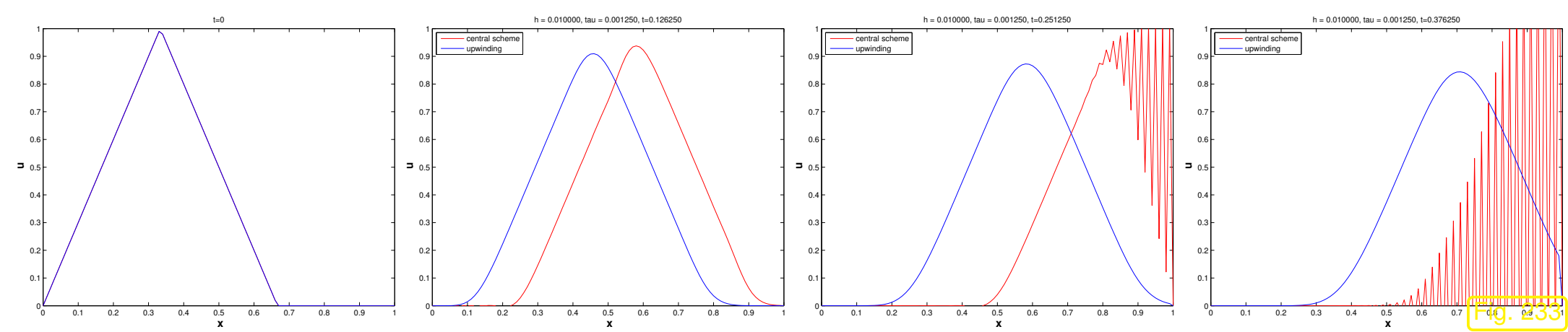
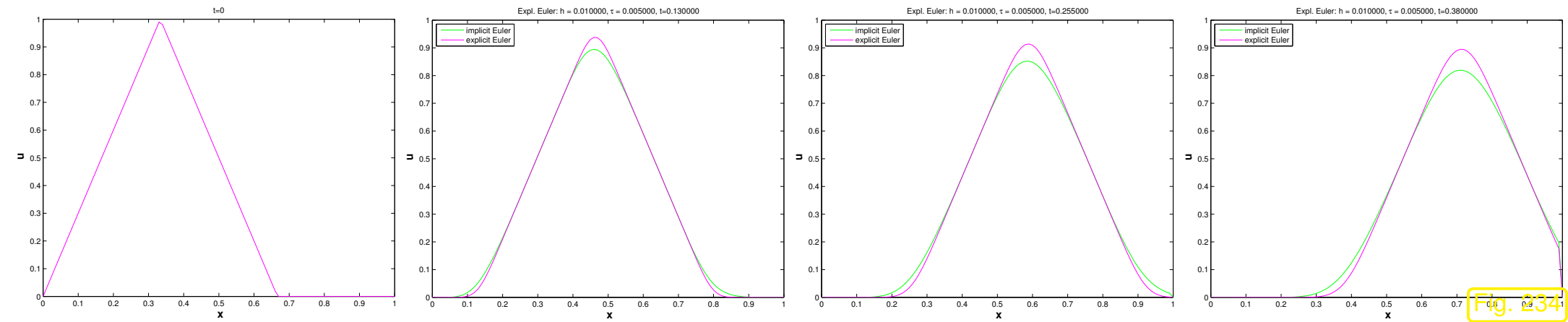


Fig. 4.33

Observation:

- Central finite differences display spurious oscillations as in Ex. 7.2.17.
- Upwinding suppresses spurious oscillations, but introduces *spurious damping*.

Computations with $\epsilon = 10^{-5}$, spatial upwind discretization, $h = 0.01$, $\tau = 0.005$:



Observation: implicit Euler timestepping causes stronger spurious damping than explicit Euler timestepping.

However, explicit Euler subject to tight stability induced timestep constraint for larger values of ϵ , see Sect. 6.1.4.2.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Use ϵ -robustly stable spatial discretization of convective term.

Remark 7.3.6 (Choice of timestepping for m.o.l. for transient convection-diffusion).

If ϵ -robustness *for all* $\epsilon > 0$ (including $\epsilon > 1$) desired \triangleright Arguments of Sect. 6.1.4.2 stipulate use of L(π)-stable (\rightarrow Def. 6.1.63) timestepping methods (implicit Euler (6.1.30), RADAU-3 (6.1.65), SDIRK-2 (6.1.66))

In the *singularly perturbed case* $0 < \epsilon \ll 1$ conditionally stable explicit timestepping is an option, due to a timestep constraint of the form “ $\tau < O(h_{\mathcal{M}})$ ”, which does not interfere with efficiency, *cf.* the discussion in Sect. 6.1.5.

7.3.2 Transport equation

Focus on the situation of **singular perturbation** (\rightarrow Def. 7.2.13): $0 < \epsilon \ll 1$

➤ study limit problem (as in Sect. 7.2.1)

$$\frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[,$$

$$\leftarrow \epsilon = 0$$

$$\frac{\partial u}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[. \quad (7.3.7)$$

=

transport equation

Now: focus on case $f \equiv 0$ (no sources)

Let $u = u(\mathbf{x}, t)$ be a C^1 -solution of

$$\frac{\partial u}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = 0 \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[. \quad (7.3.8)$$

Recall: for the stationary pure transport problem (7.2.6) we found solutions by integrating the source term along streamlines (following the flow direction).

➤ study the behavior of u “as seen from a moving fluid particle”

$$t \mapsto u(\mathbf{y}(t), t), \quad \text{where } \mathbf{y}(t) \text{ solves } \frac{d\mathbf{y}}{dt}(t) = \mathbf{v}(\mathbf{y}(t), t), \quad \text{see (7.1.1) .}$$

By the chain rule

$$\begin{aligned} \blacktriangleright \quad \frac{d}{dt} u(\mathbf{y}(t), t) &= \mathbf{grad} u(\mathbf{y}(t), t) \cdot \frac{d\mathbf{y}}{dt}(t) + \frac{\partial u}{\partial t}(\mathbf{y}(t), t) \\ &= \mathbf{grad} u(\mathbf{y}(t), t) \cdot \mathbf{v}(\mathbf{y}(t), t) + \frac{\partial u}{\partial t}(\mathbf{y}(t), t) \stackrel{(7.3.8)}{=} 0 . \end{aligned} \quad (7.3.9)$$

▶ A fluid particle “sees” a constant temperature!

Remark 7.3.10 (Solution formula for sourceless transport).

Situation: *no inflow/outflow* (e.g., fluid in a container)

$$\mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \partial\Omega, 0 < t < T. \quad (7.1.2)$$

➤ all streamlines will “stay inside Ω ”, flow map Φ^t (7.1.3) defined for all times $t \in \mathbb{R}$.

Initial value problem:

$$\mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = 0 \quad \text{in } \tilde{\Omega}, \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega.$$

Exact solution

$$u(\mathbf{x}, t) = u_0(\mathbf{x}_0(\mathbf{x}, t)), \quad (7.3.11)$$

where $\mathbf{x}_0(\mathbf{x}, t)$ is the position at time 0 of the fluid particle that is located at \mathbf{x} at time t .

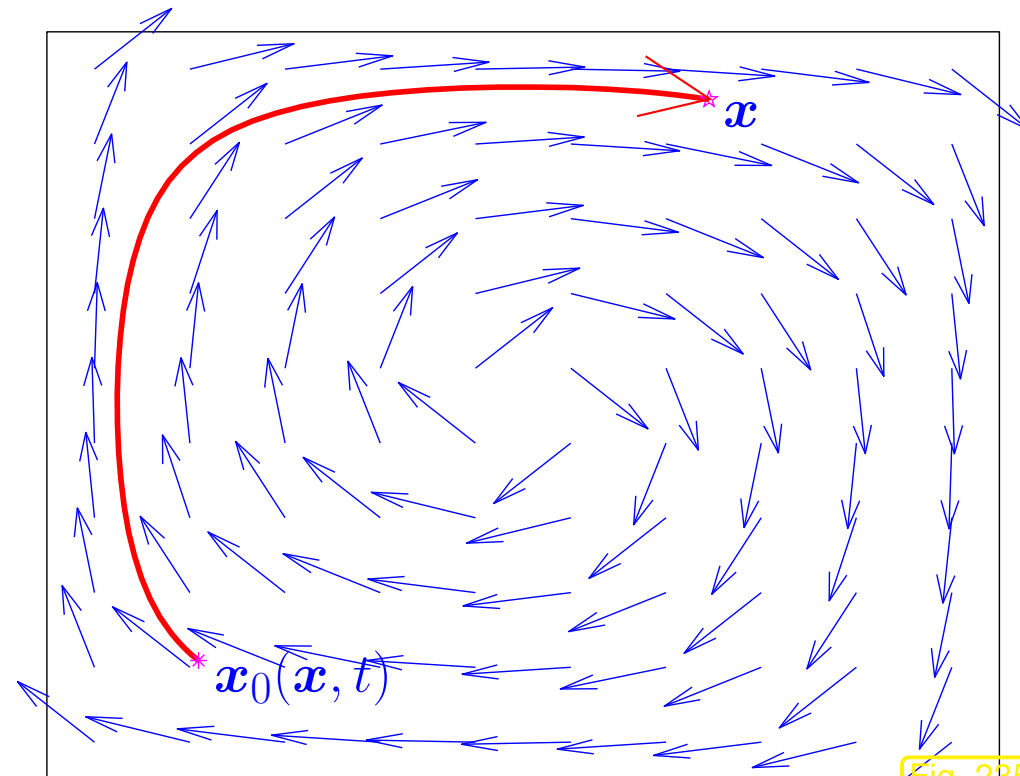


Fig. 285

This solution formula can be generalized to any divergence free velocity field $\mathbf{v} : \Omega \mapsto \mathbb{R}^d$ and $f \neq 0$. The new aspect is that streamlines can *enter* and *leave* the domain Ω . In the former case the solution value is given by a “transported boundary value”:

$$\begin{aligned} & \frac{d}{dt}u(\mathbf{y}(t)) = f(\mathbf{y}(t), t) \\ \blacktriangleright \quad u(\mathbf{x}, t) = & \begin{cases} u_0(\mathbf{x}_0) + \int_0^t f(\mathbf{y}(s), s) \, ds & , \text{ if } \mathbf{y}(s) \in \Omega \quad \forall 0 < s < t , \\ g(\mathbf{y}(s_0), s_0) + \int_{s_0}^t f(\mathbf{y}(s), s) \, ds & , \text{ if } \mathbf{y}(s_0) \in \partial\Omega, \mathbf{y}(s) \in \Omega \quad \forall s_0 < s < t , \end{cases} \end{aligned} \tag{7.3.12}$$

where we have assumed Dirichlet boundary conditions on the inflow boundary

$$u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \text{for } \mathbf{x} \in \Gamma_{\text{in}} := \{\mathbf{x} \in \partial\Omega : \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\} \quad , \text{ cf. (7.2.10).}$$

7.3.3 Lagrangian split-step method

Lagrangian discretization schemes for the IBVP (7.3.2) are inspired by insight into the traits of solutions of pure transport problems.

The variant that we are going to study separates the transient convection-diffusion problem into a pure diffusion problem (heat equation \rightarrow Sect. 6.1.1) and a pure transport problem (7.3.7). This is achieved by means of a particular approach to timestepping.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

7.3.3.1 Split-step timestepping

Abstract perspective: consider ODE, whose right hand side is the sum of two (smooth) functions

$$\dot{\mathbf{y}} = \mathbf{g}(t, \mathbf{y}) + \mathbf{h}(t, \mathbf{y}), \quad \mathbf{g}, \mathbf{h} : \mathbb{R}^m \mapsto \mathbb{R}^m. \quad (7.3.13)$$

There is an abstract timestepping scheme that offers great benefits if one commands efficient methods to solve initial value problems for both $\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z})$ and $\dot{\mathbf{w}} = \mathbf{h}(\mathbf{w})$.

Strang splitting single step method for (7.3.13), timestep $\tau := t_j - t_{j-1} > 0$: compute $\mathbf{y}^{(j)} \approx \mathbf{y}(t_j)$ from $\mathbf{y}^{(j-1)} \approx \mathbf{y}(t_{j-1})$ according to

$$\tilde{\mathbf{y}} := \mathbf{z}(t_{j-1} + \frac{1}{2}\tau), \quad \text{where } \mathbf{z}(t) \text{ solves } \dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{z}), \quad \mathbf{z}(t_{j-1}) = \mathbf{y}^{(j-1)}, \quad (7.3.14)$$

$$\hat{\mathbf{y}} := \mathbf{w}(t_j) \quad \text{where } \mathbf{w}(t) \text{ solves } \dot{\mathbf{w}} = \mathbf{h}(t, \mathbf{w}), \quad \mathbf{w}(t_{j-1}) = \tilde{\mathbf{y}}, \quad (7.3.15)$$

$$\mathbf{y}^{(j)} := \mathbf{z}(t_j), \quad \text{where } \mathbf{z}(t) \text{ solves } \dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{z}), \quad \mathbf{z}(t_{j-1} + \frac{1}{2}\tau) = \hat{\mathbf{y}}. \quad (7.3.16)$$

One timestep involves three sub-steps:

- ① Solve $\dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{z})$ over time $[t_{j-1}, t_{j-1} + \frac{1}{2}\tau]$ using the result of the previous timestep as initial value \leftrightarrow (7.3.14).
- ② Solve $\dot{\mathbf{w}} = \mathbf{h}(t, \mathbf{w})$ over time τ using the result of ① as initial value \leftrightarrow (7.3.15).
- ③ Solve $\dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{z})$ over time $[t_{j-1} + \frac{1}{2}\tau, t_j]$ using the result of ② as initial value \leftrightarrow (7.3.16).

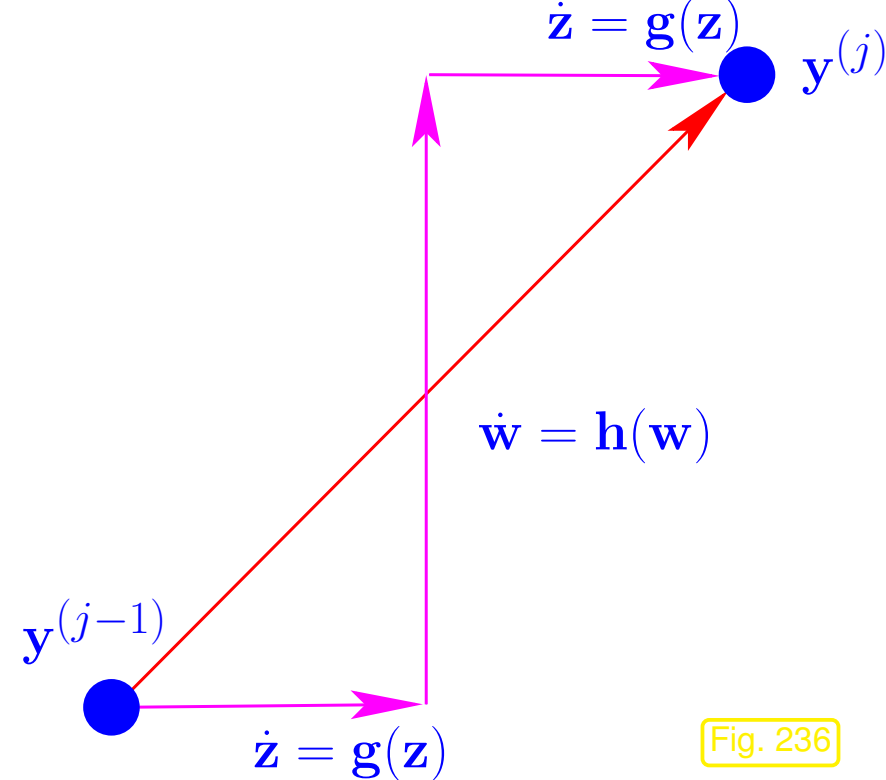


Fig. 236

Theorem 7.3.17 (Order of Strang splitting single step method).

Assuming exact solution of the initial value problems of the sub-steps, the Strang splitting single step method for (7.3.13) is of *second order*.

This applies to Strang splitting timestepping for initial value problems for ODEs. Now we boldly regard (7.3.2) as an “*ODE in function space*” for the unknown “function space valued function” $u = u(t)$:

$$[0, T] \mapsto H^1(\Omega)$$

$$\begin{aligned} \frac{du}{dt} &= \epsilon \Delta u + f - \mathbf{v} \cdot \mathbf{grad} u \\ \updownarrow & \quad \quad \quad \updownarrow \quad \quad \quad \updownarrow \\ \dot{\mathbf{y}} &= \mathbf{g}(\mathbf{y}) + \mathbf{h}(\mathbf{y}) \end{aligned}$$

Formally, we arrive at the following “timestepping scheme in function space” on a temporal mesh $0 = t_0 < t_1 < \dots < t_M := T$ for (7.3.1):

Given approximation $u^{(j-1)} \approx u(t_{j-1})$,

① Solve (autonomous) parabolic IBVP for *pure diffusion* from t_{j-1} to $t_{j-1} + \frac{1}{2}\tau$

$$\begin{aligned} (7.3.14) \quad \Leftrightarrow \quad & \frac{\partial w}{\partial t} - \epsilon \Delta w = 0 \quad \text{in } \Omega \times]t_{j-1}, t_{j-1} + \frac{1}{2}\tau[, \\ & w(\mathbf{x}, t) = g(\mathbf{x}, t_{j-1}) \quad \forall \mathbf{x} \in \partial\Omega, t_{j-1} < t < t_{j-1} + \frac{1}{2}\tau, \\ & w(\mathbf{x}, t_{j-1}) = u^{(j-1)}(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega. \end{aligned} \tag{7.3.18}$$

② Solve IBVP for *pure transport* (= **advection**), see Sect. 7.3.2,

$$(7.3.15) \quad \Leftrightarrow \quad \begin{aligned} \frac{\partial z}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} z &= f(\mathbf{x}, t) \quad \text{in } \Omega \times]t_{j-1}, t_j[, \\ z(\mathbf{x}, t) &= g(\mathbf{x}, t) \quad \text{on inflow boundary } \Gamma_{\text{in}} , t_{j-1} < t < t_j , \\ z(\mathbf{x}, t_{j-1}) &= w(\mathbf{x}, t_{j-1} + \frac{1}{2}\tau) \quad \forall \mathbf{x} \in \Omega . \end{aligned} \quad (7.3.19)$$

③ Solve (autonomous) parabolic IBVP for *pure diffusion* from $t_{j-1} + \frac{1}{2}\tau$ to t_j

$$(7.3.16) \quad \Leftrightarrow \quad \begin{aligned} \frac{\partial w}{\partial t} - \epsilon \Delta w &= 0 \quad \text{in } \Omega \times]t_{j-1} + \frac{1}{2}\tau, t_j[, \\ w(\mathbf{x}, t) &= g(\mathbf{x}, t_j) \quad \forall \mathbf{x} \in \partial\Omega , t_{j-1} + \frac{1}{2}\tau < t < t_j , \\ w(\mathbf{x}, t_{j-1} + \frac{1}{2}\tau) &= z(\mathbf{x}, t_j) \quad \forall \mathbf{x} \in \Omega . \end{aligned} \quad (7.3.20)$$

Then set $u^{(j)}(\mathbf{x}) := w(\mathbf{x}, t_j), \mathbf{x} \in \Omega.$

Efficient “implementation” of Strang splitting timestepping, if $\mathbf{g} = \mathbf{g}(\mathbf{y})$:

combine last sub-step ③ with first sub-step ① of the next timestep

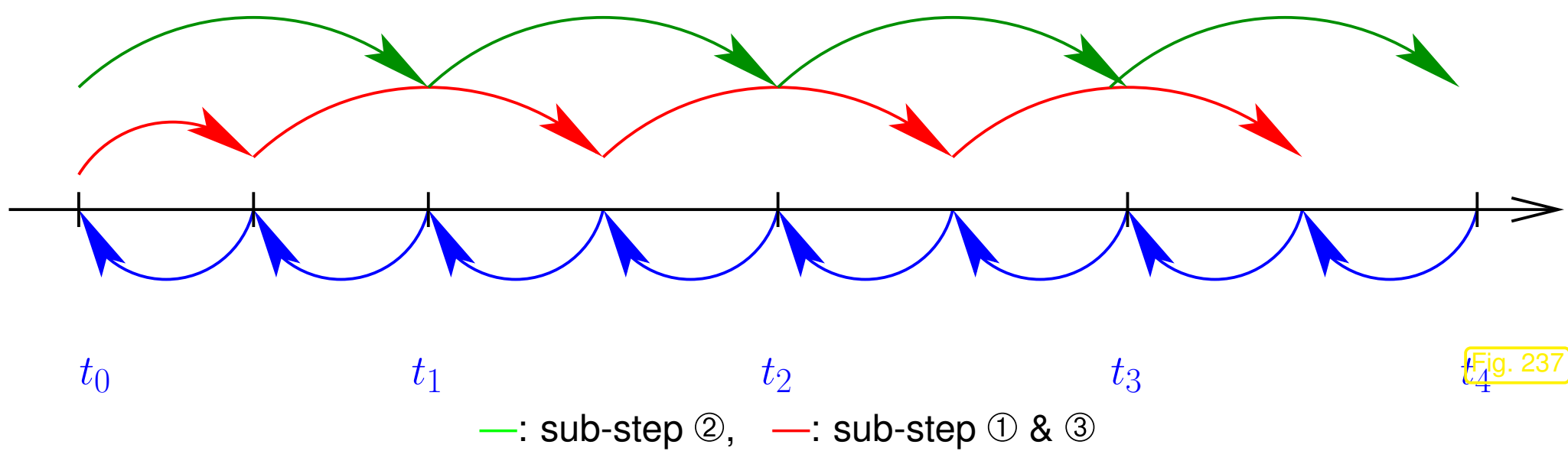


Fig. 237

Remark 7.3.21 (Approximate sub-steps for Strang splitting time).

R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

The solutions of the initial value problems in the sub-steps of Strang splitting time-stepping may be computed *only approximately*.

If this is done by one step of a 2nd-order time-stepping method in each case, then the resulting approximate Strang splitting time-stepping will still be of second order, cf. Thm. 7.3.17.

Recall the discussion of the IBVP for the pure transport (= **advection**) equation from Sect. 7.3.2

$$\begin{aligned} \frac{\partial u}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u &= f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[, \\ u(\mathbf{x}, t) &= g(\mathbf{x}, t) \quad \text{on } \Gamma_{\text{in}} \times]0, T[, \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) \quad \text{in } \Omega , \end{aligned} \tag{7.3.22}$$

with **inflow boundary**

$$\Gamma_{\text{in}} := \{ \mathbf{x} \in \partial\Omega : \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0 \} . \tag{7.2.10}$$

Case $f \equiv 0$: a travelling fluid particle sees a constant solution, see (7.3.9)

$$\blacktriangleright u(\mathbf{x}, t) = \begin{cases} u_0(\mathbf{x}_0) & , \text{ if } \mathbf{y}(s) \in \Omega \quad \forall 0 < s < t , \\ g(\mathbf{y}(s_0), s_0) & , \text{ if } \mathbf{y}(s_0) \in \partial\Omega, \mathbf{y}(s) \in \Omega \quad \forall s_0 < s < t , \end{cases} \tag{7.3.23}$$

where $s \mapsto \mathbf{y}(s)$ solves the initial value problem $\frac{d\mathbf{y}}{ds}(s) = \mathbf{v}(\mathbf{y}(s), s)$, $\mathbf{y}(t) = \mathbf{x}$ (“backward particle trajectory”).

Case of general f , see Rem. 7.3.10: Since $\frac{d}{dt}u(\mathbf{y}(t)) = f(\mathbf{y}(t), t)$

$$\blacktriangleright u(\mathbf{x}, t) = \begin{cases} u_0(\mathbf{x}_0) + \int_0^t f(\mathbf{y}(s), s) ds & , \text{ if } \mathbf{y}(s) \in \Omega \quad \forall 0 < s < t , \\ g(\mathbf{y}(s_0), s_0) + \int_{s_0}^t f(\mathbf{y}(s), s) ds & , \text{ if } \mathbf{y}(s_0) \in \partial\Omega, \mathbf{y}(s) \in \Omega \quad \forall s_0 < s < t . \end{cases} \quad (7.3.12)$$

The solution formula (7.3.12) suggests an approach for solving (7.3.22) approximately.

We first consider the simple situation of no inflow/outflow (e.g., fluid in a container, see Rem. 7.3.10)

$$\mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \partial\Omega , 0 < t < T . \quad (7.1.2)$$

① Pick suitable **interpolation nodes** $\{\mathbf{p}_i\}_{i=1}^N \subset \Omega$ (initial ‘particle positions’)

② “Particle pushing”: Solve initial value problems (*cf.* ODE (7.1.1) for particle trajectories)

$$\dot{\mathbf{y}}(t) = \mathbf{v}(\mathbf{y}(t), t) \quad , \quad \mathbf{y}(0) = \mathbf{p}_i \quad , \quad i = 1, \dots, N \quad ,$$

by means of a suitable single-step method with uniform timestep $\tau := T/M$, $M \in \mathbb{N}$.

➤ sequences of solution points $\mathbf{p}_i^{(j)}$, $j = 0, \dots, M$, $i = 1, \dots, N$

③ **Reconstruct** approximation $u_N^{(j)} \approx u(\cdot, t_j)$, $t_j := j\tau$, by interpolation:

$$u_N^{(j)}(\mathbf{p}_i^{(j)}) := u_0(\mathbf{p}_i) + \tau \sum_{l=1}^{j-1} f\left(\frac{1}{2}(\mathbf{p}_i^{(l)} + \mathbf{p}_i^{(l-1)}), \frac{1}{2}(t_l + t_{l-1})\right) \quad , \quad i = 1, \dots, N$$

where the composite midpoint quadrature rule was used to approximate the source integral in (7.3.12).

This method falls into the class of

- **particle methods**, because the interpolation nodes can be regarded fluid particles tracked by the method,
- **Lagrangian methods**, which treat the IBVP in coordinate systems moving with the flow,
- **characteristic methods**, which reconstruct the solution from knowledge about its behavior along streamlines.

For general velocity field $\mathbf{v} : \Omega \mapsto \mathbb{R}^d$:

- Stop tracking i -th trajectory as soon as an interpolation nodes $\mathbf{p}_i^{(j)}$ lies outside spatial domain Ω .
- In each timestep start new trajectories from fixed locations on inflow boundary Γ_{in} (“particle injection”). These interpolation nodes will carry the boundary value.

Example 7.3.24 (Point particle method for pure advection).

- IBVP (7.3.22) on $\Omega =]0, 1[{}^2$, $T = 2$, with $f \equiv 0$, $g \equiv 0$.
- Initial locally supported bump $u_0(\mathbf{x}) = \max\{0, 1 - 4 \left\| \mathbf{x} - \begin{pmatrix} 1/2 \\ 1/4 \end{pmatrix} \right\|\}$.
- Two stationary divergence-free velocity fields
 - $\mathbf{v}_1(\mathbf{x}) = \begin{pmatrix} -\sin(\pi x_1) \cos(\pi x_2) \\ \cos(\pi x_1) \sin(\pi x_2) \end{pmatrix}$ satisfying (7.1.2),
 - $\mathbf{v}_2(\mathbf{x}) = \begin{pmatrix} -x_2 \\ x_1 \end{pmatrix}$.

- Initial positions of interpolation points on regular tensor product grid with meshwidth $h = \frac{1}{40}$.
- Approximation of trajectories by means of explicit trapezoidal rule [21, Eq. 12.4.5] (method of Heun).

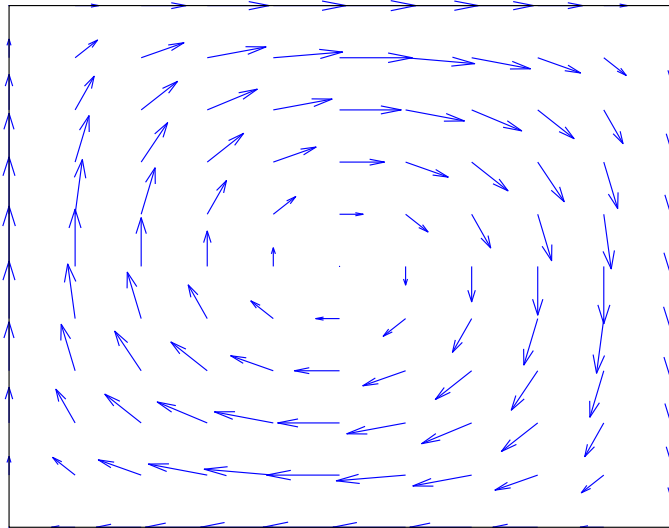


Fig. 238

velocity field \mathbf{v}_1 (circvel)

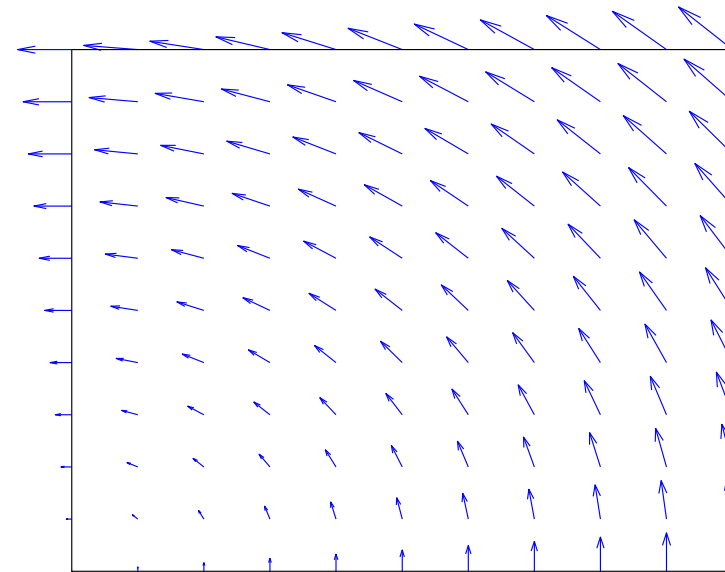


Fig. 239

velocity field \mathbf{v}_2 (rotvel)

Code 7.3.28: Confined velocity field

```
1 function V = circvel(P)
2 % Circular velocity (divergence free, zero normal component on unit square).
3 % P: 2xN matrix of point coordinates
4 % return value: velocity vectors at points in P
5
6 v = @(p) [-sin(pi*p(1))*cos(pi*p(2)); sin(pi*p(2))*cos(pi*p(1))];
7
8 V = [];
9 for p=P
10     V = [V, v(p)];
11 end
```

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Code 7.3.29: Pass-through velocity field

```
1 function V = rotvel(P)
2 % Circular velocity
3
4 v = @(p) [-p(2); p(1)];
5
6 V = [];
7 for p=P
8     V = [V, v(p)];
9 end
```


Code 7.3.30: Point particle method for pure advection

```

1 function partadv(v,u0,g,n,tau,m)
2 % Point particle method for pure advection problem
3 % on the unit square
4 % v: handle to a function returning the velocity field for (an array) of
   points
5 % u0: handle to a function returning the initial value  $u_0$  for (an array)
6 % of points
7 % g: handle to a function  $g = g(\mathbf{x})$  returning the Dirichlet boundary values
8 % n:  $h = 1/n$  is the grid spacing of the initial point distribution
9 % tau: timestep size, m: number of timesteps, that is,  $T = m\tau$ 
10
11 % Initialize points
12 h = 1/n; [Xp,Yp] = meshgrid(0:h:1,0:h:1);
13 P = [reshape(Xp,1,(n+1)^2);reshape(Yp,1,(n+1)^2)];
14 % Initialize points on the boundary
15 BP = [[(0:h:1);zeros(1,n+1)], [ones(1,n+1);(0:h:1)], ...
16       [(0:h:1);ones(1,n+1)], [zeros(1,n+1);(0:h:1)]];
17 U = u0(P); % Initial values
18
19 % Plot velocity field
20 hp = 1/10; [Xp,Yp] = meshgrid(0:hp:1,0:hp:1);
21 Up = zeros(size(Xp)); Vp = zeros(size(Xp));
22 for i=0:10, for j=0:10

```

```
23 x = v([Xp(i+1,j+1);Yp(i+1,j+1)]);
24 Up(i+1,j+1) = x(1); Vp(i+1,j+1) = x(2);
25 end; end
26 figure('name','velocity field','renderer','painters');
27 quiver(Xp,Yp,Up,Vp,'b-'); set(gca,'fontsize',14); hold on;
28 plot([0 1 1 0 0],[0 0 1 1 0],'k-');
29 axis([-0.1 1.1 -0.1 1.1]);
30 xlabel('\bf x_1'); ylabel('\bf x_2');
31 axis off;
32
33 fp = figure('name','particles','renderer','painters');
34 fs = figure('name','solution','renderer','painter');
35
36 % Visualize points (interior points in red, boundary points in blue)
37 figure(fp); plot(P(1,:),P(2),'r+',BP(1,:),BP(2),'b*');
38 title(sprintf('n = %i, t = %f, \tau = %f, %i
    points',n,0,tau,size(P,2)));
39 drawnow; pause;
40
41 % Visualize solution
42 figure(fs); plotpartsol(P,U); drawnow;
43
44 t = 0;
```

```
45 for l=1:m
46 % Advect points (explicit trapezoidal rule)
47 P1 = P + tau/2*v(P); P = P + tau*v(P1);
48
49 % Remove points on the boundary or outside the domain
50 Pnew = []; Unew = []; l = 1;
51 for p=P
52     if ((p(1) > eps) (p(1) < 1-eps) (p(2) > eps) (p(2) <
53         1-eps))
54         Pnew = [Pnew,p]; Unew = [Unew; U(l)];
55     end
56     l = l+1;
57 end
58 % Add points on the boundary (particle injection)
59 P = [Pnew, BP]; U = [Unew; g(BP)];
60
61 % Visualize points
62 figure (fp); plot (P(1,:), P(2,:), 'r+', BP(1,:), BP(2,:), 'b*');
63 title (sprintf ('n = %i, t = %f, \\\tau = %f, %i
64     points', n, t, tau, size (P, 2)));
65 drawnow;
66 % Visualize solution
```

```
66 figure (fs); plotpartsol (P,U); drawnow;  
67  
68 t = t+tau;  
69 end
```

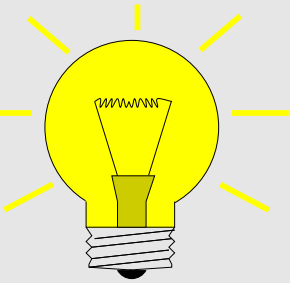


7.3.3.3 Particle mesh method

The method introduced in the previous section, can be used to tackle the pure advection problem (7.3.19) in the 2nd sub-step of the Strang splitting timestepping.

Issue: How to combine Lagrangian advection with a method for the pure diffusion problem (7.3.18) faced in the other sub-steps of the Strang splitting timestepping?

Idea: two views


 “particle temperatures” $u(\mathbf{p}_i^{(j)})$

 Nodal values of finite element function $u_N^{(j)} \in \mathcal{S}_1^0(\mathcal{M})$

► Outline: algorithm for one step of size $\tau > 0$ of Strang splitting timestepping for transient convection-diffusion problem

$$\begin{cases} \frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f & \text{in } \tilde{\Omega} := \Omega \times]0, T[, \\ u(\mathbf{x}, t) = 0 \quad \forall \mathbf{x} \in \partial\Omega, 0 < t < T \quad , \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega . \end{cases} \quad (7.3.2)$$

- ① Given
- triangular mesh $\mathcal{M}^{(j-1)}$ of Ω ,
 - $u_N^{(j-1)} \in \mathcal{S}_{1,0}^0(\mathcal{M}^{(j-1)}) \leftrightarrow$ coefficient vector $\vec{\mu}^{(j-1)} \in \mathbb{R}^{N_{j-1}}$,
- approximately solve (7.3.18) by a single step of implicit Euler (6.1.30) (size $\frac{1}{2}\tau$)

$$\vec{\nu} = (\mathbf{M} + \frac{1}{2}\tau\epsilon\mathbf{A})^{-1} \vec{\mu}^{(j-1)} ,$$

where $\mathbf{A} \in \mathbb{R}^{N_{j-1}, N_{j-1}} \hat{=} \mathcal{S}_{1,0}^0(\mathcal{M})$ -Galerkin matrix for $-\Delta$, $\mathbf{M} \hat{=} (\text{possibly lumped}) \mathcal{S}_{1,0}^0(\mathcal{M})$ -mass matrix.

More advisable to maintain 2nd-order timestepping: 2nd-order $L(\pi)$ -stable single step method, e.g., SDIRK-2 (6.1.66).

② Lagrangian advection step (of size τ) for (7.3.19) with

- initial “particle positions” \mathbf{p}_i given by nodes of $\mathcal{M}^{(j-1)}$, $i = 1, \dots, N_j$,
- initial “particle temperatures” given by corresponding coefficients ν_i .

③ *Remeshing*: advection step has moved nodes to new positions $\tilde{\mathbf{p}}_i$ (and, maybe, introduced new nodes by “particle injection”, deleted nodes by “particle removal”).

- Create **new** triangular mesh $\mathcal{M}^{(j)}$ with nodes $\tilde{\mathbf{p}}_i$ (+ boundary nodes), $i = 1, \dots, N_j$

④ Repeat diffusion step ① starting with $w_N \in \mathcal{S}_{1,0}^0(\mathcal{M}^{(j)}) = \text{linear interpolant} (\rightarrow \text{Def. 5.3.13})$ of “particle temperatures” on $\mathcal{M}^{(j)}$.

- new approximate solution $u_N^{(j)}$

Example 7.3.31 (Delaunay-remeshing in 2D).

Delaunay algorithm for creating a 2D triangular mesh *with prescribed nodes*:

- ① Compute Voronoi cells, see (4.2.3) & <http://www.qhull.org/>.
- ② Connect two nodes, if their associated Voronoi dual cells have an edge in common.

➔ MATLAB `TRI = delaunay(x,y)`

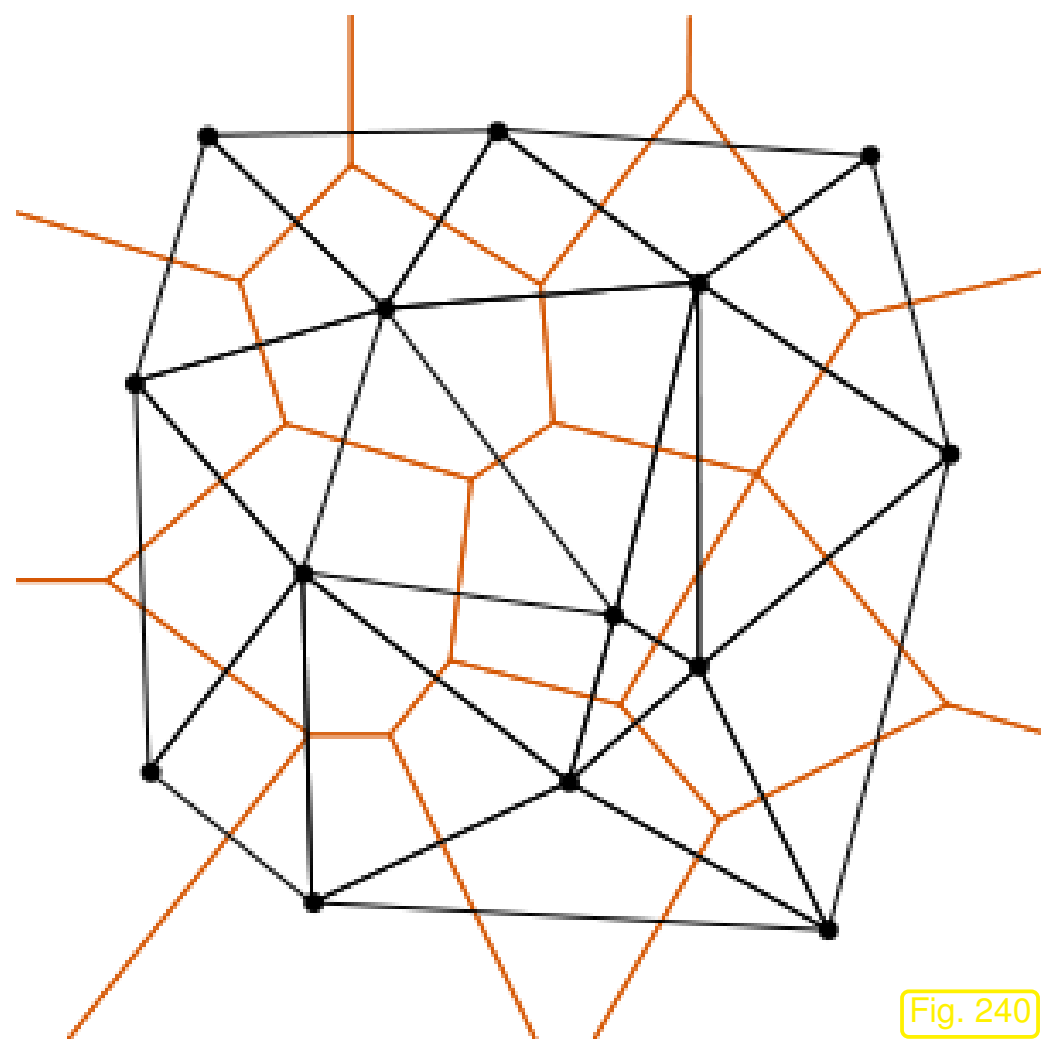


Fig. 240

Code 7.3.32: Demonstration of Delaunay-remeshing

```

1 function meshadv(v,n,tau,m)
2 % Point advaction and remeshing for Lagrangian method
3 % v: handle to a function returning the velocity field for (an array) of
  points
4 % n:  $h=1/n$  is the grid spacing of the inintial point distribution
5
6 % Initialize points
7 h = 1/n; [Xp,Yp] = meshgrid(0:h:1,0:h:1);

```

```
8 P = [reshape (Xp,1, (n+1)^2); reshape (Yp,1, (n+1)^2)];
9 % Initialize points on the boundary
10 BP = [[(0:h:1); zeros (1,n+1)], [ones (1,n+1); (0:h:1)], ...
11       [(0:h:1); ones (1,n+1)], [zeros (1,n+1); (0:h:1)]];
12
13 % Plot triangulation
14 fp = figure ('name', 'evolving meshes', 'renderer', 'painters');
15 TRI = delaunay (P (1, :), P (2, :));
16 plot (P (1, :), P (2, :), 'r+'); hold on;
17     triplot (TRI, P (1, :), P (2, :), 'blue'); hold off;
18 title (sprintf ('n = %i, t = %f, \tau = %f, %i
19     points', n, 0, tau, size (P, 2)));
20 drawnow; pause;
21
22 t = 0;
23 for l=1:m
24     % Advect points (explicit trapezoidal rule)
25     P1 = P + tau/2*v (P); P = P + tau*v (P1);
26
27     % Remove points on the boundary or outside the domain
28     Pnew = []; l = 1;
29     for p=P
30         if ((p (1) > eps) (p (1) < 1-eps) (p (2) > eps) (p (2) <
31             1-eps))
```



```
29     Pnew = [Pnew,p];
30     end
31     l = l+1;
32 end
33
34 P = [Pnew, BP]; % Add points on the boundary (particle injection)
35
36 % Plot triangulation
37 TRI = delaunay(P(1,:),P(2,:));
38 plot(P(1,:),P(2,:), 'r+'); hold on;
39     triplot(TRI,P(1,:),P(2,:), 'blue'); hold off;
40     title(sprintf('n = %i, t = %f, \\\tau = %f, %i
41     points',n,t,tau, size(P,2)));
42 drawnow;
43
44 t = t+tau;
45 end
```

$\Omega =]0, 1[^2$, velocity fields like in Ex. 7.3.24. Advection of interpolation nodes by means of explicit trapezoidal rule.

Start animations:

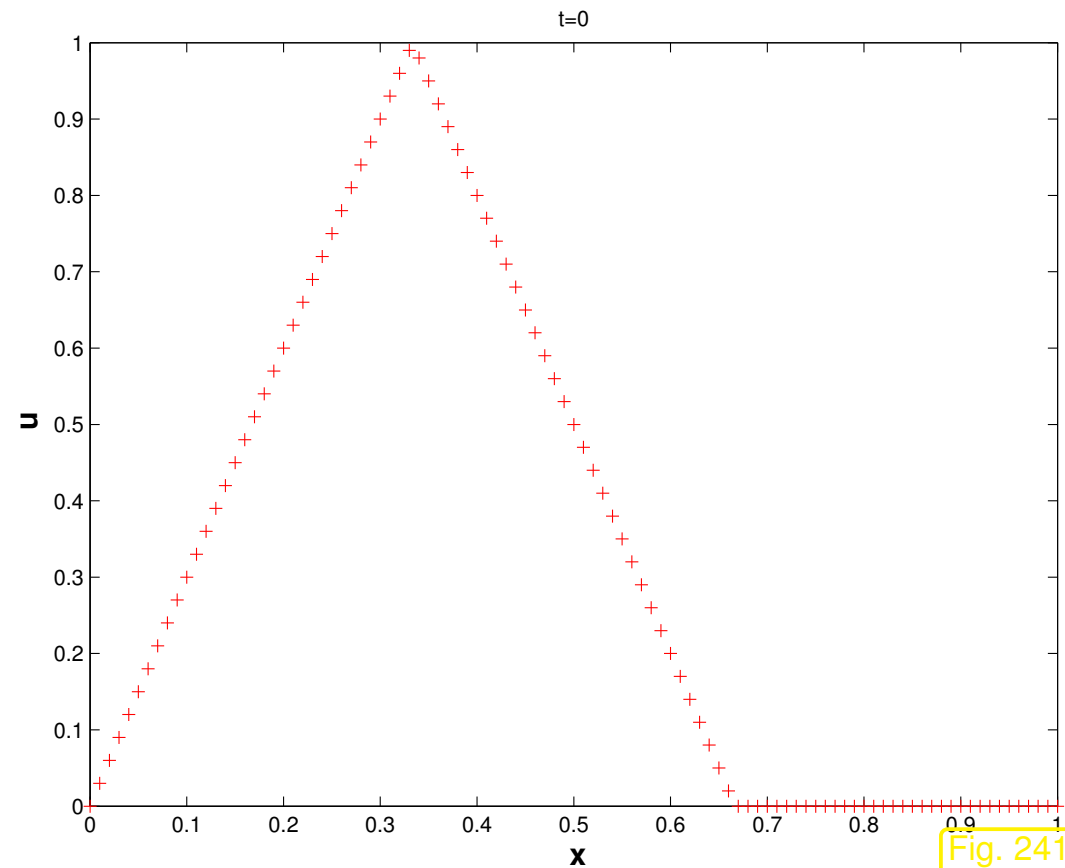
```
meshadv (@circvel, 20, 0.05, 40);  
meshadv (@rotvel, 20, 0.05, 40);
```



Example 7.3.33 (Lagrangian method for convection-diffusion in 1D).

Same IBVP as in Ex. 7.3.4

- Linear finite element Galerkin discretization with mass lumping in space
- Strang splitting applied to diffusive and convective terms
- Implicit Euler timestepping for diffusive partial timestep



Code 7.3.34: Lagrangian method for (7.3.5)

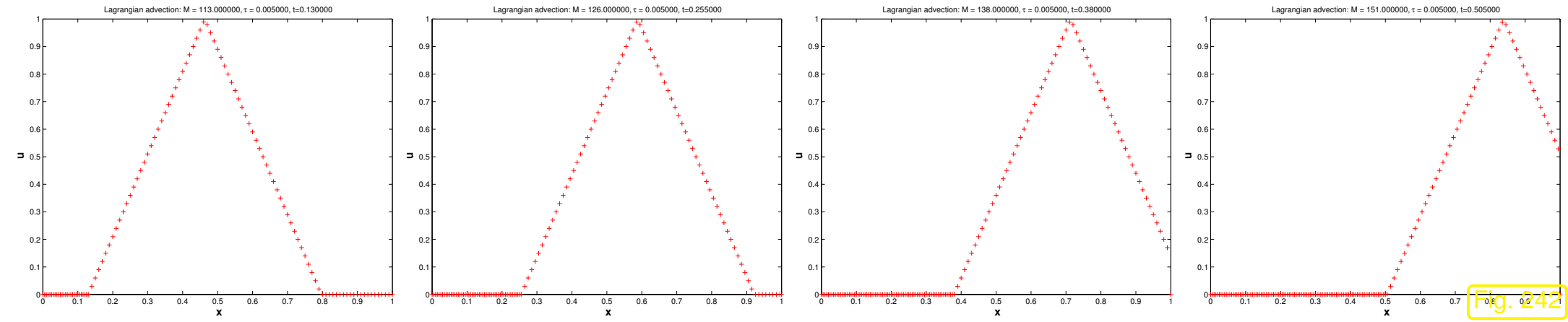
```

1 function lagr(epsilon,N,M)
2 % This function implements a simple Lagrangian advection scheme for the 1D
  convection-diffusion
3 % IBVP  $-\epsilon \frac{d^2 u}{dx^2} + \frac{du}{dx} = 0$ ,  $u(x,0) = \max(1 - 3|x - \frac{1}{3}|, 0)$ ,
4 % and homogeneous Dirichlet boundary conditions  $u(0) = u(1) = 0$ . Timestepping
  employs Strang splitting
5 % applied to diffusive and convective spatial operators.
6 % epsilon: strength of diffusion
7 % N: number of cells of spatial mesh
8 % M: number of timesteps

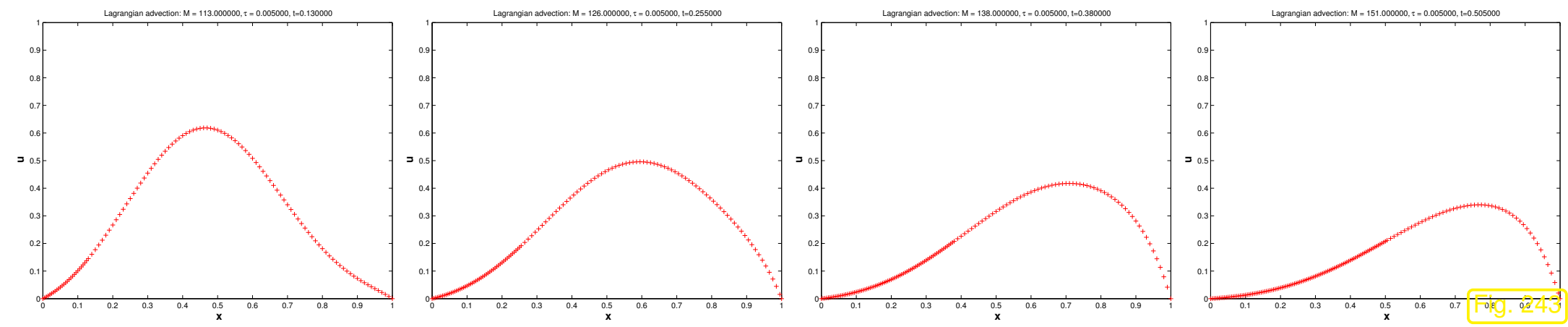
```

```
9
10 T = 0.5; tau = T/M; %
    timestep size
11 h = 1/N; x = 0:h:1; u = max(1-3*abs(x(2:end)-1)-1/3),0)'; % Initial
    value
12
13 [Amat,Mmat] = getdeltamat(x); % Obtain stiffness and mass
    matrix
14 u = (Mmat+0.5*tau*epsilon*Amat) \ (Mmat*u); % Implicit Euler timestep
15
16 for j=1:M+1
17     % Advection step: shift meshpoints, drop those travelling out of  $\Omega = ]0,1[$ ,
    insert
18     % new meshpoints from the left. Solution values are just copied.
19     xm = x(2:end-1)+tau; % Transport of meshpoints (here: explicit
    Euler)
20     idx = find(xm < 1); % Drop meshpoints beyond  $x = 1$ 
21     x = [0,tau,xm(idx),1]; % Insert new meshpoint at left end of  $\Omega$ 
22     u = [0;u(idx)]; % Copy nodal values and feed 0 from left
23
24     % Diffusion partial timestep
25     [Amat,Mmat] = getdeltamat(x); % Obtain stiffness and mass
    matrix on new mesh
26     u = (Mmat+tau*epsilon*Amat) \ (Mmat*u); % Implicit Euler step
27 end
28 end
```

$\epsilon = 10^{-5}$:



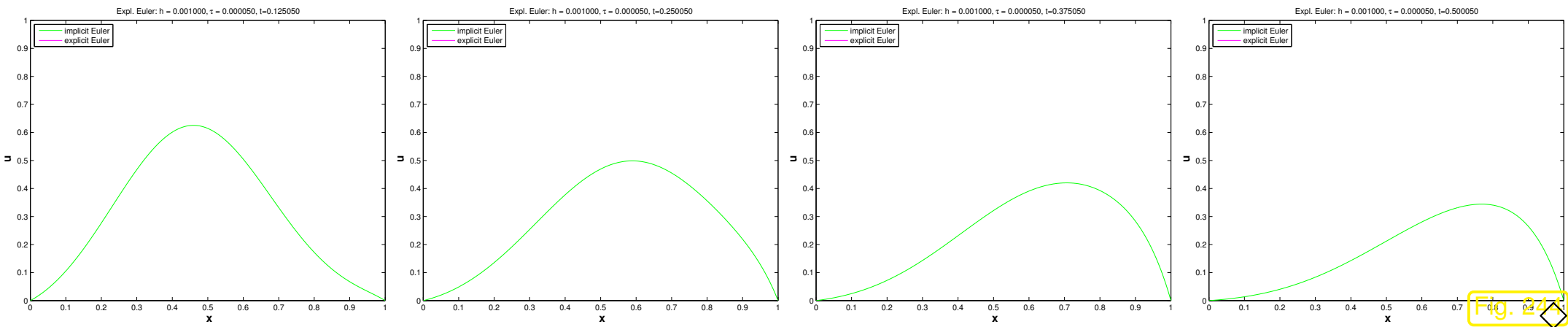
$\epsilon = 0.1$:



“Reference solution” computed by method of lines, see Ex. 7.3.4, with $h = 10^{-3}$, $\tau = 5 \cdot 10^{-5}$:

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Example 7.3.35 (Lagrangian method for convection-diffusion in 2D).

- IBVP (7.3.2) on $\Omega =]0, 1[^2$, $T = 1$,
- Particle mesh method based on Delaunay remeshing, see Ex. 7.3.31, and linear finite element Galerkin discretization for diffusion step.

Code 7.3.36: Particle mesh method in 2D

```

1 function ConvDiffLagr(v, epsilon, u0, n, tau, m)
2 % Point particle method for convection-diffusion problem on the unit square
3 % v: handle to a function returning the velocity field for (an array) of
   points

```

```
4 % u0: handle to a function returning the initial value  $u_0$  for (an array)
5 % of points
6 % n:  $h=1/n$  is the grid spacing of the inintial point distribution
7 % tau: timestep size, m: number of timesteps, that is,  $T = m\tau$ 
8
9 % Initialize points
10 h = 1/n; [Xp,Yp] = meshgrid(0:h:1,0:h:1);
11 P = [reshape(Xp,1,(n+1)^2);reshape(Yp,1,(n+1)^2)];
12 % Initialize points on the boundary
13 BP = [[(0:h:1);zeros(1,n+1)], [ones(1,n+1);(0:h:1)], ...
14        [(0:h:1);ones(1,n+1)], [zeros(1,n+1);(0:h:1)]];
15 % Construct initial mesh by Delaunay algorithm
16 TRI = delaunay(P(1,:),P(2,:));
17
18 U = u0(P); % Initial values
19
20 fp = figure('name','particles','renderer','painters');
21 fs = figure('name','solution','renderer','painters');
22
23 % Visualize mesh, points (interior points in red, boundary points in blue)
24 % the piecewise linear approximate solution
25 figure(fp); plot(P(1,:),P(2),'r+',BP(1,:),BP(2,),'m*'); hold on;
26 triplot(TRI,P(1,:),P(2,),'blue'); hold off;
27 title(sprintf('n = %i, t = %f, \\\tau = %f, %i points',n,0,tau,size(P,2)));
28 drawnow;
29
30 figure(fs); trisurf(TRI,P(1,:),P(2,:),U');
31 axis([0 1 0 1 0 1]); xlabel('{\bf x_1}');
32 ylabel('{\bf x_2}'); zlabel('{\bf u}');
33 title(sprintf('n = %i, t = %f, \\\tau = %f, %i points',n,0,tau,size(P,2)));
```

```
34 pause;
35
36 % Initial diffusion half step (implicit Euler)
37 [Amat,Mmat] = getGalerkinMatrices(TRI,P(1,:),P(2,:)); % Compute Galerkin
   matrices
38 % Isolate indices of interior points
39 j = 1; intidx = [];
40 for p=P
41     if ((p(1) > eps) (p(1) < 1-eps) (p(2) > eps) (p(2) < 1-eps))
42         intidx = [intidx,j];
43     end
44     j = j+1;
45 end
46 Amat = Amat(intidx,intidx); Mmat = Mmat(intidx,intidx);
47 U(intidx) = (Mmat+0.5*epsilon*tau*Amat)\(Mmat*U(intidx));
48
49 % full(Amat), full(Mmat), return;
50
51 t = 0;
52 for l=1:m
53 % Advect points (explicit trapezoidal rule)
54 P1 = P + tau/2*v(P); P = P + tau*v(P1);
55
56 % Remove points on the boundary or outside the domain
57 Pnew = []; Unew = []; l = 1; j = 0;
58 for p=P
59     if ((p(1) > eps) (p(1) < 1-eps) (p(2) > eps) (p(2) < 1-eps))
60         Pnew = [Pnew,p]; Unew = [Unew; U(l)];
61         j = j+1; % Counter for interior points
62     end
63     l = l+1;
```



```
64 end
65
66 % Add points on the boundary (particle injection)
67 P = [Pnew, BP];
68
69 % Delaunay algorithm for building triangulation
70 TRI = delaunay(P(1,:),P(2,:));
71 [Amat,Mmat] = getGalerkinMatrices(TRI,P(1,:),P(2,:)); % Compute Galerkin
    matrices
72 Amat = Amat(1:j,1:j); Mmat = Mmat(1:j,1:j);
73 U = (Mmat+epsilon*tau*Amat)\(Mmat*Unew); % implicit Euler step
74 U = [U; zeros(size(BP,2),1)]; % zero padding for boundary nodes
75
76 % Visualize mesh, points (interior points in red, boundary points in blue)
77 % the piecewise linear approximate solution
78 figure(fp); plot(P(1,:),P(2,:),'r+',BP(1,:),BP(2,),'m*'); hold on;
79 triplot(TRI,P(1,:),P(2,),'blue'); hold off;
80 title(sprintf('n = %i, t = %f, \\\tau = %f, %i points',n,t,tau,size(P,2)));
81 drawnow;
82
83 figure(fs); trisurf(TRI,P(1,:),P(2,:),U');
84 axis([0 1 0 1 0 1]); xlabel('\\bf x_1');
85 ylabel('\\bf x_2'); zlabel('\\bf u');
86 title(sprintf('n = %i, t = %f, \\\tau = %f, %i points',n,t,tau,size(P,2)));
87 t = t+tau;
88 end
```

Invocation: `ConvDiffLagr (@circvel, 0.001, @initvals, 1/40, 0.01, 100)`



Advantage of Lagrangian (particle) methods for convection diffusion:

No artificial diffusion required (no “smearing”)

No stability induced timestep constraint



Drawback of Lagrangian (particle) methods for convection diffusion:

Remeshing (may be) expensive and difficult.

Point advection may produce “voids” in point set.

7.3.4 Semi-Lagrangian method

Now we study a family of methods for transient convection-diffusion that takes into account transport along streamlines, but, in contrast to genuine Lagrangian methods, relies on a *fixed* mesh.

Definition 7.3.37 (Material derivative).

Given a velocity field $\mathbf{v} : \Omega \times]0, T[\mapsto \mathbb{R}^d$, the *material derivative* of a function $f = f(\mathbf{x}, t)$ at (\mathbf{x}, t) is

$$\frac{Df}{D\mathbf{v}}(\mathbf{x}, t_0) = \lim_{\tau \rightarrow 0} \frac{f(\mathbf{x}, t_0) - f(\Phi_{t_0}^{-\tau} \mathbf{x}, t_0 - \tau)}{\tau}, \quad \mathbf{x} \in \Omega, \quad 0 < t_0 < T,$$

with $\Phi_{t_0}^t$ the flow map (at time t_0) associated with \mathbf{v} , that is, cf. (7.1.3), (7.1.4),

$$\frac{d\Phi_{t_0}^t \mathbf{x}}{dt} = \mathbf{v}(\Phi_{t_0}^t \mathbf{x}, t - t_0), \quad \Phi_{t_0}^0 \mathbf{x} = \mathbf{x}.$$

The material derivative $\frac{Df}{D\mathbf{v}}$ is the

rate of change of f experienced by a particle carried along by the flow

because $\Phi_{t_0}^t \mathbf{x}$ describes the trajectory of a particle located at \mathbf{x} at time t_0 ($\leftrightarrow t = 0$).

By a straightforward application of the chain rule for smooth f

$$\frac{Df}{D\mathbf{v}}(\mathbf{x}, t) = \mathbf{grad}_{\mathbf{x}} f(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t) + \frac{\partial f}{\partial t}(\mathbf{x}, t) . \quad (7.3.40)$$

➤ The transient convection-diffusion equation can be rewritten as (7.3.1)

$$\frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[,$$

$$\blacktriangledown \leftarrow (7.3.40)$$

$$\frac{Du}{D\mathbf{v}} - \epsilon \Delta u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[. \quad (7.3.41)$$

Idea: *Backward difference* (“implicit Euler”) discretization of material derivative

$$\frac{Du}{D\mathbf{v}}|_{(\mathbf{x},t)=(\bar{\mathbf{x}},t_0)} \approx \frac{u(\bar{\mathbf{x}}, t_0) - u(\Phi_{t_0}^{-\tau}\bar{\mathbf{x}}, t_0 - \tau)}{\tau},$$

with timestep $\tau > 0$, where $t \mapsto \Phi^t \bar{\mathbf{x}}$ solves the initial value problem

$$\frac{d\Phi_{t_0}^t \bar{\mathbf{x}}}{dt}(t) = \mathbf{v}(\Phi_{t_0}^t \bar{\mathbf{x}}, t_0 + t) \quad , \quad \Phi_{t_0}^0 \bar{\mathbf{x}} = \bar{\mathbf{x}} .$$



► *Semi-discretization* of (7.3.41) *in time* (with fixed timestep $\tau > 0$)

$$\frac{u^{(j)}(\mathbf{x}) - u^{(j-1)}(\Phi_{t_j}^{-\tau} \mathbf{x})}{\tau} - \epsilon \Delta u^{(j)}(\mathbf{x}) = f(\mathbf{x}, t_j) \quad \text{in } \Omega, \quad (7.3.43)$$

+ boundary conditions at $t = t_j$,

where $u^{(j)} : \Omega \mapsto \mathbb{R}$ is an approximation for $u(\cdot, t_j)$, $t_j := j\tau$, $j \in \mathbb{N}$.

Note the difference to the method of lines (\rightarrow Sects. 6.1.3, 6.2.3, 7.3.1): in (7.3.43) semidiscretization in time was carried out first, now followed by discretization in space, which reverses the order adopted in the method of lines.

Cast (7.3.43) into variational form according to the recipe of Sect. 2.8 and apply *Galerkin discretization* (here discussed for linear finite elements, homogeneous Dirichlet boundary conditions $u = 0$ on $\partial\Omega$).

This yields one timestep (size τ) for the *semi-Lagrangian method*: the approximation $u_N^{(j)}$ for $u(j\tau)$ (equidistant timesteps) is computed from the previous timestep according to

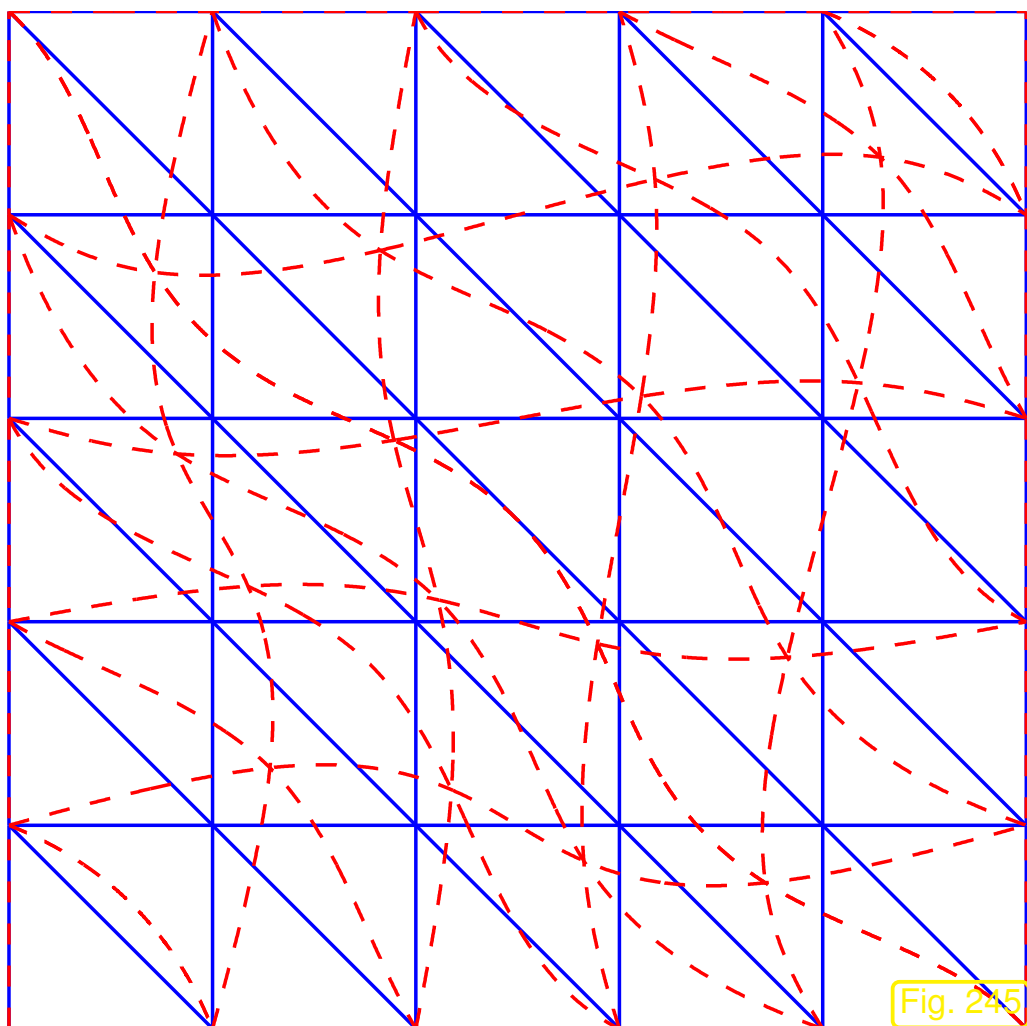
$$u_N^{(j)} \in \mathcal{S}_{1,0}^0(\mathcal{M}): \int_{\Omega} \frac{u_N^{(j)}(\mathbf{x}) - u_N^{(j-1)}(\Phi_{t_j}^{-\tau} \mathbf{x})}{\tau} v_N(\mathbf{x}) \, d\mathbf{x} + \epsilon \int_{\Omega} \mathbf{grad} u_N^{(j)} \cdot \mathbf{grad} v_N \, d\mathbf{x} \\ = \int_{\Omega} f(\mathbf{x}, t_j) v_N(\mathbf{x}) \, d\mathbf{x} \quad \forall v_N \in \mathcal{S}_{1,0}^0(\mathcal{M}) . \quad (7.3.44)$$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Here, \mathcal{M} is supposed to be a *fixed* triangular mesh of Ω .

However, (7.3.44) cannot be implemented: $\mathbf{x} \mapsto u_N^{(j-1)}(\Phi_{t_j}^{-\tau} \mathbf{x})$ is a finite element function that has been “transported with the (reversed) flow” (in the sense of pullback, see Def. 3.6.2)



◁ $-\cdot-\cdot-$ $\hat{=}$ image of \mathcal{M} ($-\cdot-$) under $\Phi_{t_j}^{-\tau}$

The pullback $\mathbf{x} \mapsto v_N(\Phi_{t_j}^{-\tau} \mathbf{x})$ of $v_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$ is piecewise smooth w.r.t. the mapped mesh drawn with $-\cdot-\cdot-$. Hence, it is not smooth inside the cells of \mathcal{M} .

Fig. 245

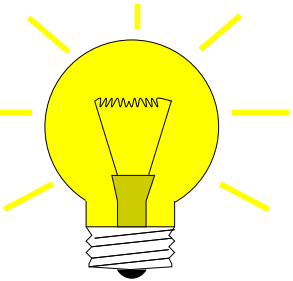
- the transported function may **not** be a finite element function on \mathcal{M} ,
- the transported function may not even be piecewise smooth on \mathcal{M}

- local quadrature for the approximate computation of the integral in (7.3.44) that involves $u_N^{(j-1)}(\Phi_{t_j}^{-\tau} \mathbf{x})$ is not possible, because accurate numerical quadrature requires a (locally) smooth integrand.

Idea: • replace $u_N^{(j-1)}(\mathbf{y}(\mathbf{x}(-\tau)))$ with linear interpolant (\rightarrow Def. 5.3.13)

$$l_1(u_N^{(j-1)} \circ \Phi_{t_j}^{-\tau}) \in \mathcal{S}_{1,0}^0(\mathcal{M}),$$

- approximate $\Phi_{t_j}^{-\tau} \mathbf{x}$ by $\mathbf{x} - \tau \mathbf{v}(\mathbf{x}, t_j)$ (explicit Euler).
("streamline backtracking")



$$u_N^{(j)} \in \mathcal{S}_{1,0}^0(\mathcal{M}): \int_{\Omega} \frac{u_N^{(j)}(\mathbf{x}) - l_1(u_N^{(j-1)}(\cdot - \tau \mathbf{v}(\cdot, t_j)))(\mathbf{x})}{\tau} v_N(\mathbf{x}) \, d\mathbf{x} + \epsilon \int_{\Omega} \mathbf{grad} u_N^{(j)} \cdot \mathbf{grad} v_N \, d\mathbf{x}$$

$$= \int_{\Omega} f(\mathbf{x}, t_j) v_N(\mathbf{x}) \, d\mathbf{x} \quad \forall v_N \in \mathcal{S}_{1,0}^0(\mathcal{M}).$$

Then apply local vertex based numerical quadrature (2D trapezoidal rule (3.2.18) = global trapezoidal rule) to the first integral. This amounts to using **mass lumping**, see Rem. 6.2.34.

► Implementable version of (7.3.44):

$$\begin{aligned}
 u_N^{(j)} \in \mathcal{S}_{1,0}^0(\mathcal{M}): \quad & \frac{1}{3}|U_{\mathbf{p}}|(\mu_{\mathbf{p}}^{(j)} - u_N^{(j-1)}(\mathbf{p} - \tau \mathbf{v}(\mathbf{p}, t_j))) + \tau \int_{\Omega} \mathbf{grad} u_N^{(j)} \cdot \mathbf{grad} b_N^{\mathbf{p}} \, d\mathbf{x} \\
 & = \frac{1}{3}|U_{\mathbf{p}}|f(\mathbf{p}), \quad \mathbf{p} \in \mathcal{N}(\mathcal{M}) \cap \Omega, \quad (7.3.45)
 \end{aligned}$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

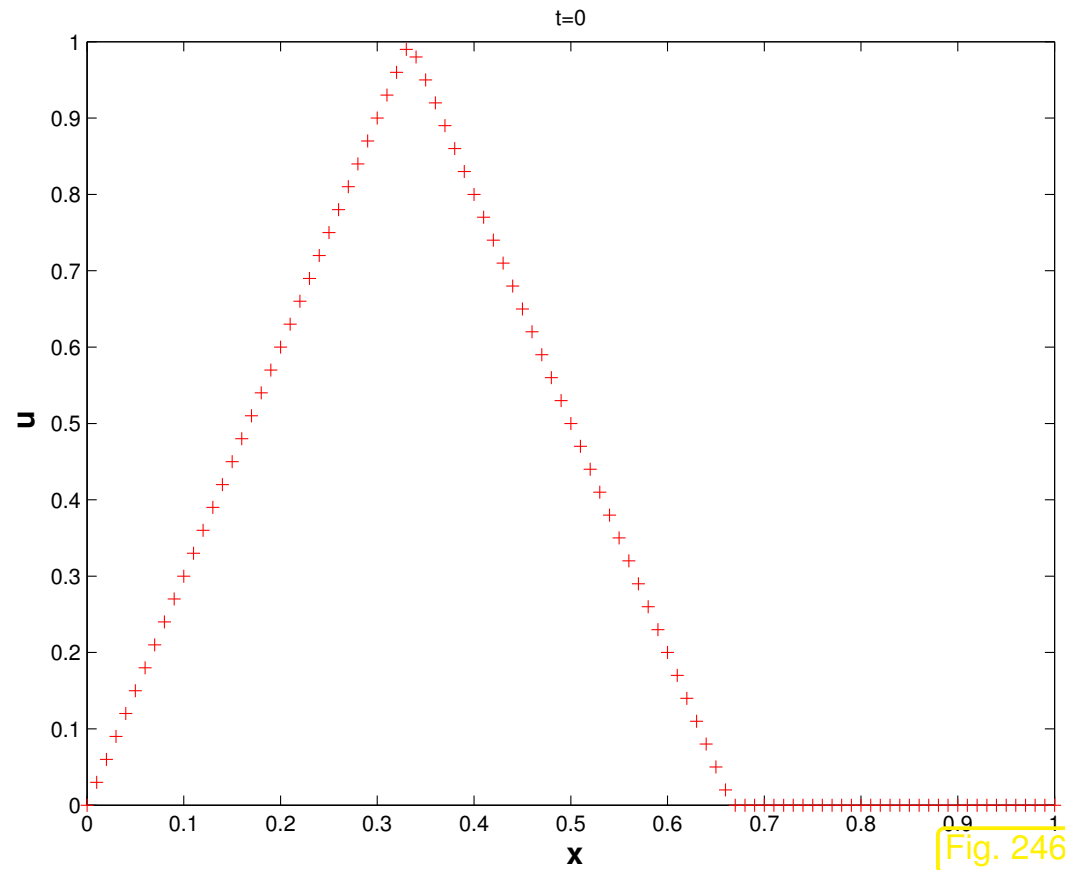
where $\mu_{\mathbf{p}}^{(j)}$ are the nodal values of $u_N^{(j)} \in \mathcal{S}_{1,0}^0(\mathcal{M})$ associated with the interior nodes of the mesh \mathcal{M} , $b_N^{\mathbf{p}}$ is the “tent function” belonging to node \mathbf{p} , $|U_{\mathbf{p}}|$ is the sum of the areas of all triangles adjacent to \mathbf{p} .

SAM, ETHZ

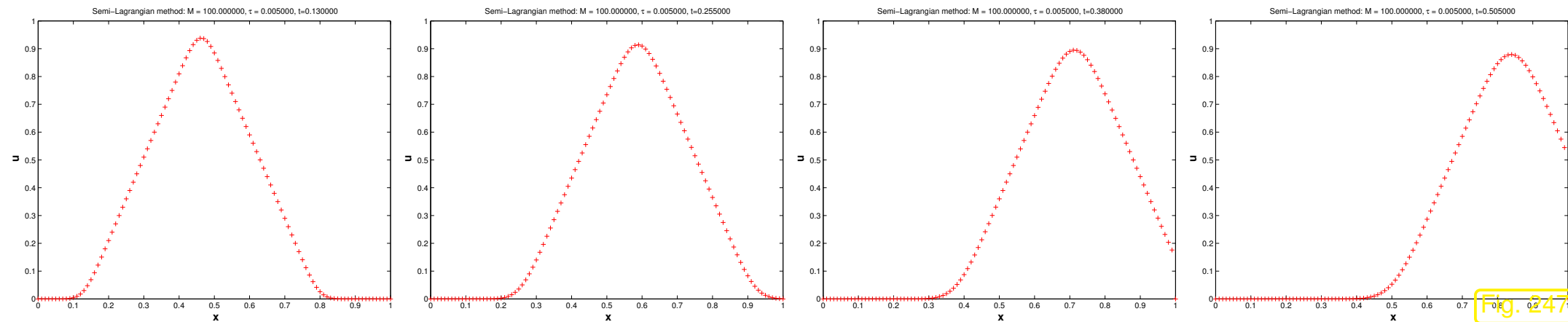
Example 7.3.46 (Semi-Lagrangian method for convection-diffusion in 1D).

Same IBVP as in Ex. 7.3.33

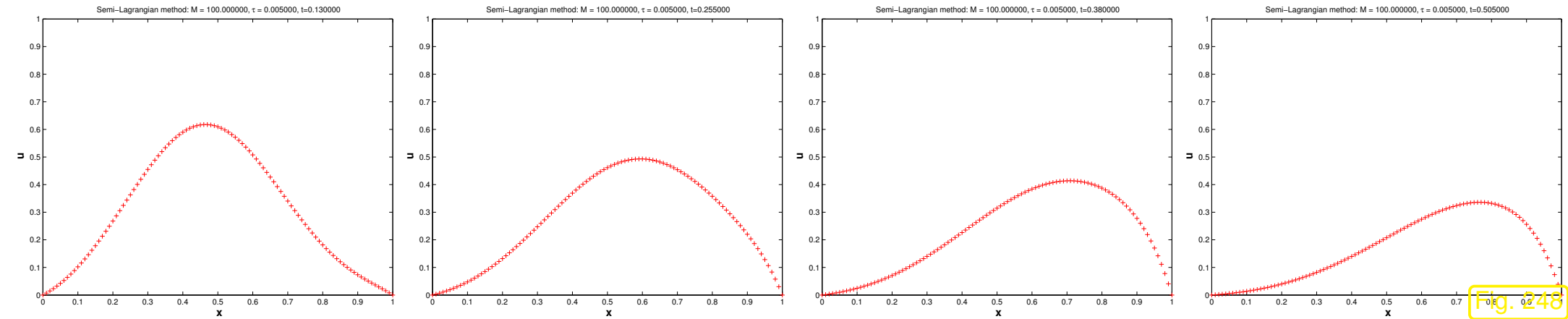
- Linear finite element Galerkin discretization with mass lumping in space
- Semi-Lagrangian method: 1D version of (7.3.44)
- Explicit Euler streamline backtracking



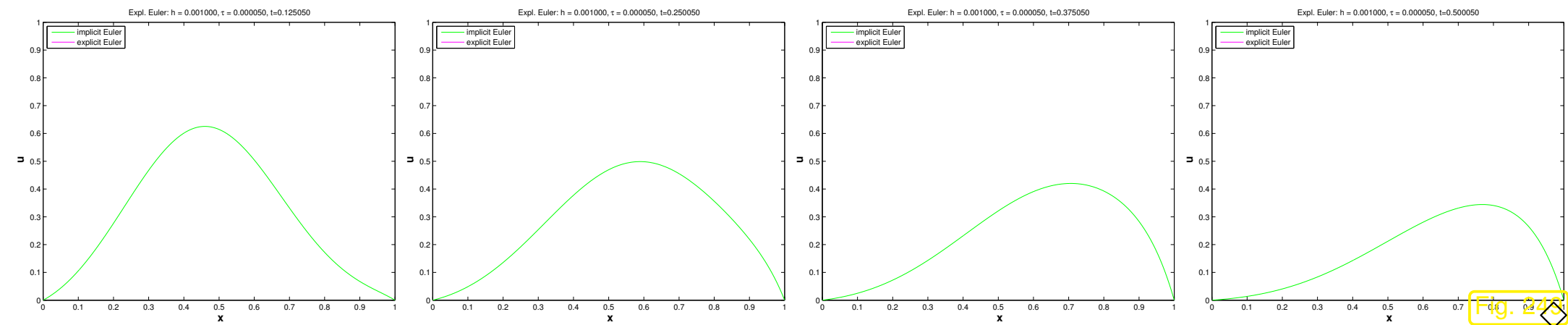
$\epsilon = 10^{-5}$:



$\epsilon = 0.1$:



“Reference solution” computed by method of lines, see Ex. 7.3.4, with $h = 10^{-3}$, $\tau = 5 \cdot 10^{-5}$:



Example 7.3.47 (Semi-Lagrangian method for convection-diffusion in 2D).

- 2nd-order scalar convection diffusion problem (7.3.2), $\Omega :=]0, 1[^2$, $f = 0$, $g = 0$,
- velocity field

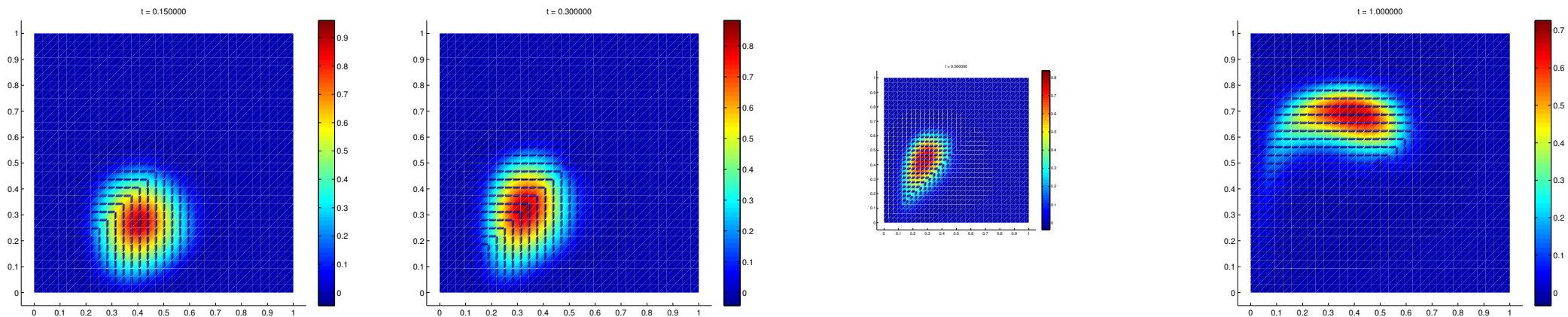
$$\mathbf{v}(\mathbf{x}) := \begin{pmatrix} -\sin(\pi x_1) \cos(\pi x_2) \\ \sin(\pi x_2) \cos(\pi x_1) \end{pmatrix} .$$

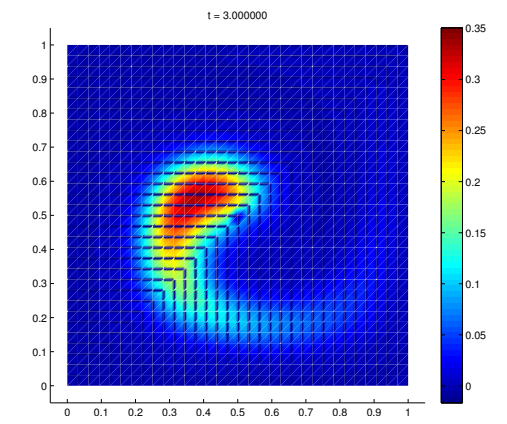
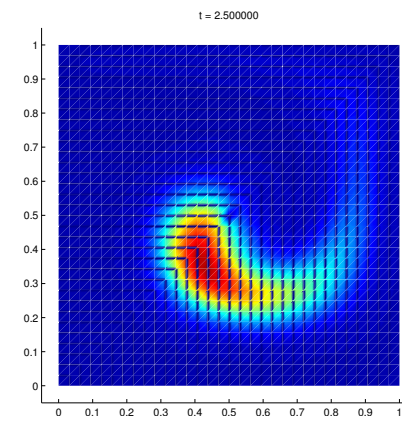
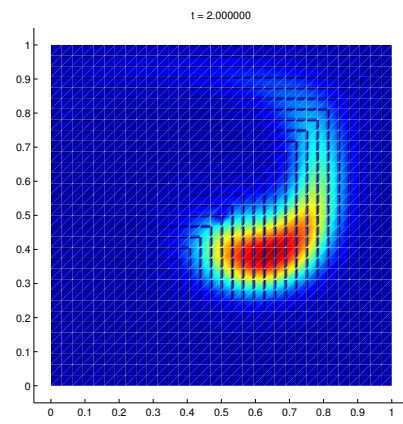
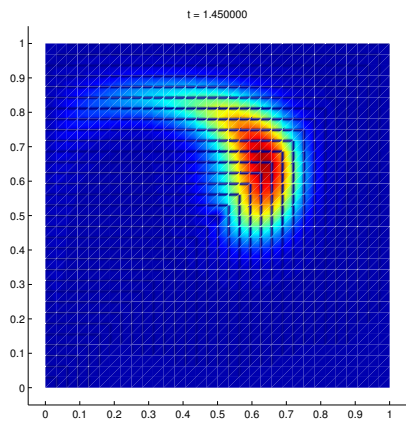
- Initial condition: “compactly supported cone shape”

$$u_0(\mathbf{x}) = \max(0, 1 - 4 * \text{sqrt}((\mathbf{x}(:, 1) - 0.5) .^2 + (\mathbf{x}(:, 2) - 0.25) .^2)) ;$$

- semi-Lagrangian finite element Galerkin discretization according to (7.3.44) on regular triangular meshes of square domain Ω , see Fig. 124.

Example with $\epsilon = 0$:





We observe smearing of initial data due to numerical diffusion inherent in the interpolation step of the semi-Lagrangian method.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



8

Numerical Methods for Conservation Laws

Conservation laws describe physical phenomena governed by

- *conservation* laws for certain physical quantities (e.g., mass momentum, energy, etc.),
- *transport* of conserved physical quantities.

We have already examined problems of this type in connection with transient heat conduction in Sect. 7.1.4. There thermal energy was the conserved quantity and a *prescribed* external velocity field \mathbf{v} determined the transport.

A new aspect emerging for general conservation laws is that the transport velocity itself may depend on the conserved quantities themselves, which gives rise to *non-linear models*.



Supplementary and further reading:

[25]: Comprehensive monograph and textbook about so-called finite volume method providing detailed explanations.

[23]: concisely written textbook adopting a mathematical perspective and delving into technical details.

[12]: mathematical monograph about the theory of initial value problems for conservation laws.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

8.1 Conservation laws: Examples

Focus:

Cauchy problems

Spatial domain $\Omega = \mathbb{R}^d$ (unbounded!)

➤ Cauchy problems are pure initial value problems (no boundary values).

Rationale: ❶ *Finite speed of propagation* typical of conservation laws

(Potential spatial boundaries will not affect the solution for some time in the case of compactly supported initial data, *cf.* situation for wave equation, where we also examined the Cauchy problem, see (6.2.15).)

❷ No spatial boundary ➤ need not worry about (spatial) boundary conditions!

(Issue of spatial boundary conditions can be very intricate for conservation laws)

8.1.1 Linear advection

Cauchy problem for linear **transport equation** (**advection equation**) \rightarrow Sect. 7.1.4, (7.1.16):

$$\frac{\partial}{\partial t}(\rho u) + \operatorname{div}(\mathbf{v}(\mathbf{x}, t)(\rho u)) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega} := \mathbb{R}^d \times]0, T[, \quad (8.1.1)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathbb{R}^d \quad (\text{initial conditions}) . \quad (8.1.2)$$

$u = u(\mathbf{x}, t) \hat{=}$ temperature, $\rho > 0 \hat{=}$ heat capacity, $\mathbf{v} = \mathbf{v}(\mathbf{x}, t) \hat{=}$ prescribed velocity field.

(8.1.1) = **linear scalar conservation law**



Conserved quantity: thermal energy (density) ρu

(Recall the derivation of (7.1.16) through conservation of energy, *cf.* (6.1.1).)

Simplified problem: assume constant heat capacity $\rho \equiv 1$, no sources $f \equiv 0$, stationary velocity field $\mathbf{v} = \mathbf{v}(\mathbf{x}) \quad \triangleright$ rescaled initial value problem *written in conserved variables*

$$\begin{aligned} \frac{\partial u}{\partial t} + \operatorname{div}(\mathbf{v}(\mathbf{x})u) &= 0 \quad \text{in} \quad \tilde{\Omega} := \mathbb{R}^d \times]0, T[, \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) \quad \text{for all} \quad \mathbf{x} \in \mathbb{R}^d \quad (\text{initial conditions}) . \end{aligned} \quad (8.1.4)$$

Convention: differential operator div acts on spatial independent variable only,

$$(\operatorname{div} \mathbf{f})(\mathbf{x}, t) := \frac{\partial f_1}{\partial x_1} + \dots + \frac{\partial f_d}{\partial x_d} , \quad \mathbf{f}(\mathbf{x}, t) = \begin{pmatrix} f_1(\mathbf{x}, t) \\ \vdots \\ f_d(\mathbf{x}, t) \end{pmatrix} .$$

Special case: Constant coefficient linear advection in 1D

- $d = 1 \quad \triangleright \quad \Omega = \mathbb{R}$,
- constant velocity $v = \text{const.}$.

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(vu) = 0 \quad \text{in} \quad \tilde{\Omega} = \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) \quad \forall x \in \mathbb{R} . \quad (8.1.5)$$

This is the 1D version of the transport equation (7.3.7) \triangleright solution given by (7.3.11)

$$(7.3.11) \quad \blacktriangleright \quad u(x, t) = u_0(x - vt), \quad x \in \mathbb{R}, \quad 0 \leq t < T. \quad (8.1.6)$$

Solution $u = u(x, t)$ = initial data “travelling” with velocity v .

Solution formula (8.1.6) makes perfect sense even for *discontinuous* initial data u_0 !

➔ We should not expect $u = u(x, t)$ to be differentiable in space or time.
A “weaker” concept of solution is required, see Sect. 8.2.3 below.

This consideration should be familiar: for second order elliptic boundary value problems, for which classical solutions are to be twice continuously differentiable, the concept of a variational solution

made it possible to give a meaning to solutions $\in H^1(\Omega)$ that are merely continuous and piecewise differentiable, see Rem. 1.3.39.

Related to (8.1.6): d'Alembert solution formula (6.2.16) for 1D wave equation (6.2.15).

Remark 8.1.11 (Boundary conditions for linear advection).

Recall the discussion in Sects. 7.2.1, 7.3.2, *cf.* solution formula (7.3.12):

For the scalar linear advection initial boundary value problem

$$\frac{\partial u}{\partial t} + \operatorname{div}(\mathbf{v}(\mathbf{x}, t)u) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[, \quad (8.1.12)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \Omega , \quad (8.1.13)$$

on a bounded domain $\Omega \subset \mathbb{R}^d$, **boundary conditions** (e.g., prescribed temperature)

$$u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \text{on } \Gamma_{\text{in}}(t) \times]0, T[,$$

can be imposed on the **inflow boundary**

$$\Gamma_{\text{in}}(t) := \{ \boldsymbol{x} \in \partial\Omega : \mathbf{v}(\boldsymbol{x}, t) \cdot \boldsymbol{n}(\boldsymbol{x}) < 0 \} , \quad 0 < t < T . \quad (8.1.14)$$

Note: Γ_{in} can change with time!

Bottom line:

Knowledge of local and current direction of transport
needed to impose meaningful boundary conditions!

8.1.2 Traffic modeling [2]

We design simple mathematical models for non-stationary traffic flow on a *single long highway lane*.
This situation often occurs with bypasses of long highway construction sites.

Simplifying *modeling assumptions* (not quite matching reality):

• Identical cars and behavior of drivers (8.1.18)

• Uniformity of road conditions (8.1.19)

• Speed of a car determined only by (its distance from) the car in front (8.1.20)

8.1.2.1 Particle model

Gist of particle model: tracking of individual cars over times $[0, T]$

$x_i(t) \hat{=}$ position of i -th car at time t , $i = 1, \dots, N$ ($N \hat{=}$ total number of cars)

We will always take for granted ordering: $x_i(t) < x_{i+1}(t)$

In order to describe the dynamics of the moving cars we need a *velocity model*.

► Here: **optimal velocity model**

$$\dot{x}_i(t) = v_{\text{opt}}(\Delta x_i) \quad , \quad \Delta x_i(t) = x_{i+1}(t) - x_i(t) > 0 \quad , \quad i = 1, \dots, N - 1 . \quad (8.1.24)$$

↔ relies on Assumptions (8.1.18)–(8.1.20) above, in particular (8.1.20).

$v_{\text{opt}}(\Delta x_i)$ from the assumption that

each car drives as fast as possible under safety constraints.
(drive more slowly if the you are close to the car in front)

$$\text{►} \quad v_{\text{opt}}(\Delta x) = v_{\text{max}} \left(1 - \frac{\Delta_0}{\Delta x} \right) , \quad (8.1.25)$$

with $\Delta_0 \hat{=}$ length of a car = distance of cars in bumper to bumper traffic jam.

► (8.1.24) + (8.1.25): **ordinary differential equation** (ODE) on state space \mathbb{R}^N

In order to get a well-posed initial value problem, the ODE has to be supplemented with **initial conditions**

$$x_i(0) = x_{i,0} \in \mathbb{R} \quad , \quad x_{i,0} \leq x_{i+1,0} - \Delta_0 . \quad (8.1.26)$$

Obviously (why?): solution of (8.1.24), (8.1.25), (8.1.26) satisfies $x_i(t) \leq x_{i+1}(t) - \Delta_0$.

Remark 8.1.29 (Acceleration based traffic modeling).

The speed of a car is a consequence of drivers accelerating and breaking.

➤ acceleration based modeling of car dynamics under Assumptions (8.1.18)–(8.1.20)

$$\ddot{x}_i(t) = F(\Delta x_i(t), \Delta v_i(t)) \quad , \quad \Delta v_i(t) = \dot{x}_{i+1} - \dot{x}_i . \quad (8.1.30)$$

Models of this type are popular in practice.

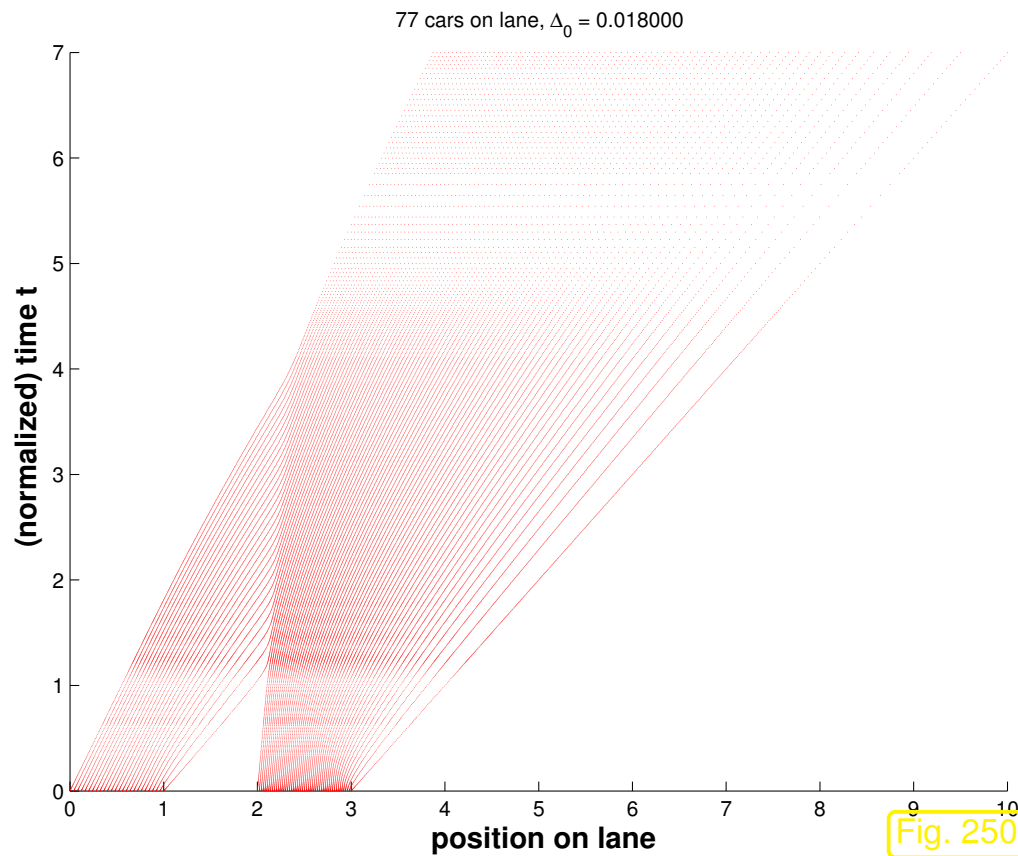


R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Example 8.1.31 (Particle simulation of traffic flow).

Usually: $v_{\max} = 1$ by **rescaling** of spatial/temporal units, *cf.* Rem. 1.2.6.



Initial positions of cars:

```
1 x0 = [0:1/25:1, 2:1/50:3]
```

◁ Simulation based on optimal velocity model
(8.1.25) with (dimensionless) $\Delta_0 = 0.0180$.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Code 8.1.33: Particle simulation of cars based on optimal velocity model

```
1 function [times, Y, fig] = carsim(x0, T, xL, xR, d0)
2 % Particle simulation of single lane traffic flow using the normalization
3 %  $v_{\max} = 1$  and  $d_0 := \frac{1}{N}$ , where  $N$  is the
```

```
4 % total number of cars. x0 passes the initial positions of the cars.
5 % This vector is assumed to be sorted with the last component providing the
6 % position of the rightmost car.
7
8 % Total number N of cars
9 N = length(x0); x0 = reshape(x0,N,1);
10 % Bumper to bumper distance d0 of the cars
11 if ( nargin < 5), d0 = (xR-xL)/(5*N); end
12 u0 = 1/d0; % Maximal number density of cars in a bumper to bumper jam
13
14 % Check validity of initial positions
15 dist0 = diff(x0); % compute  $\Delta x_i$ 
16 if (min(dist0) < 0.99*d0), d0, min(dist0), error('Cars too
    close'); end
17
18 % right hand side of the numerical integrator according to (8.1.24) and
19 % (8.1.25) with  $v_{\max} = 1$ . Note that x has to be a row
20 % vector. The rightmost car travels at speed  $v_{\max}$ .
21 rhs = @(t,x) [1-d0*1./diff(x);1];
22
23 % perform numerical integration using MATLAB's standard integrator
24 options = odeset('abstol',1E-8,'reltol',1E-7);
25 [times,X] = ode45(rhs,[0 T],x0,options);
```

```
26
27 % Compute density of cars normalized with the maximal density  $(\Delta_0)^{-1}$ ,
28 % based on averages over  $\frac{N}{5}$  equally long sections of the lane, that
29 % is  $\delta = \frac{5|x_R-x_L|}{N}$  in (8.1.35).
30 Y = []; M = floor(N/6);
31 for k=1:length(times)
32     Y = [Y; cardensity(X(k,:), xL, xR, M) / u0];
33 end
34
35 % Plot positions of cars as a function of time ("fan plot")
36 fig = figure('name', 'positions of cars');
37 axis([xL xR 0 T]); hold on;
38 k = 1;
39 for t=times'
40     plot(X(k,:), t*ones(N,1), 'r.', 'markersize', 1);
41     k = k+1;
42 end
43 xlabel('\bf position on lane', 'fontsize', 14);
44 ylabel('\bf (normalized) time t', 'fontsize', 14);
45 title(sprintf('%d cars on lane, \Delta_{0} = %f', N, d0));
46 hold off;
47
48 % (Animated) plot of normalized density of cars
```

```
49 figure ;  
50 for k=1:length (times)  
51     stairs ((xL: (xR-xL)/M:xR), Y(k, :), 'm') ;  
52     axis ([xL xR 0 1]) ;  
53     xlabel ('{\bf position on lane}', 'fontsize', 14) ;  
54     ylabel ('{\bf density of cars}', 'fontsize', 14) ;  
55     title (sprintf ('%d cars on lane, time = %f', N, times(k))) ;  
56     drawnow ;  
57 end
```

Initial conditions: congestion on two sections of the road.

Evolution: traffic jams merge and dissolve as cars “escape” to the right. *Fan-shaped* patterns emerge.



Extraction of macroscopic quantities.

terminology: “macroscopic quantities” $\hat{=}$ quantities describing the traffic flow detached from existence of individual cars.

macroscopic quantities can be obtained by *averaging* from the microscopic particle description.

Key macroscopic quantity:

(normalized) **density** of cars

$$u_\delta(x, t) := \frac{\Delta_0}{\delta} \#\{i \in \{1, \dots, N\} : x - \delta \leq x_i(t) < x + \delta\}, \quad (8.1.35)$$

where $\delta > 0$ is the **spatial averaging length**.

(The density defined in (8.1.35) is “normalized” because it is the ratio of the number density of cars and the maximal density Δ_0^{-1} . Hence, invariably, $0 \leq u_\delta(x, t) \leq 1$.)

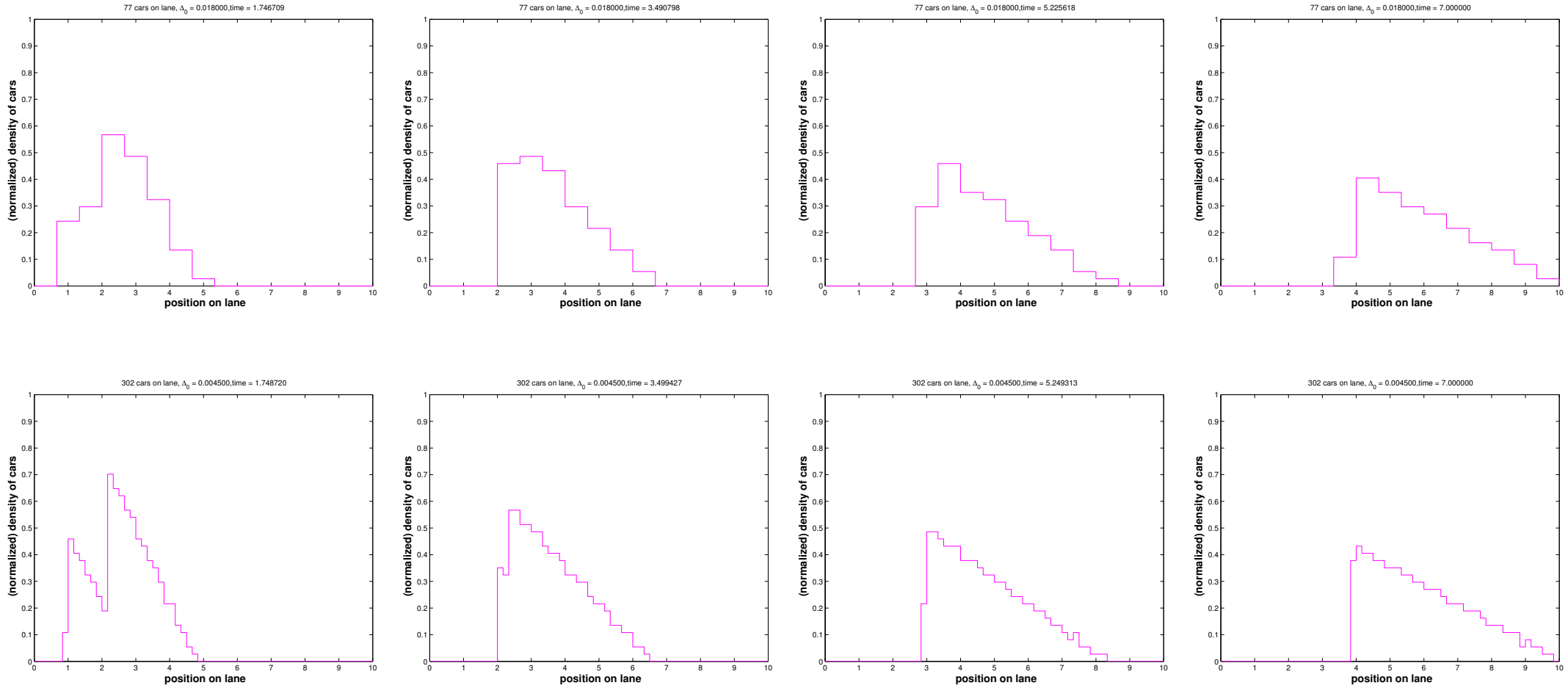
Note: u_δ will crucially depend on δ

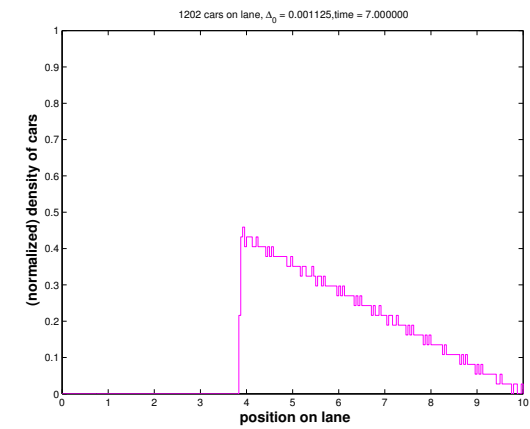
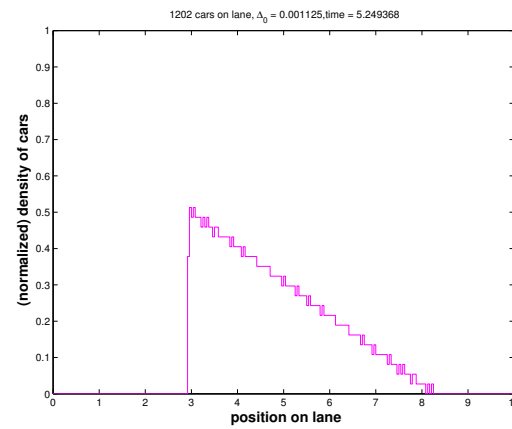
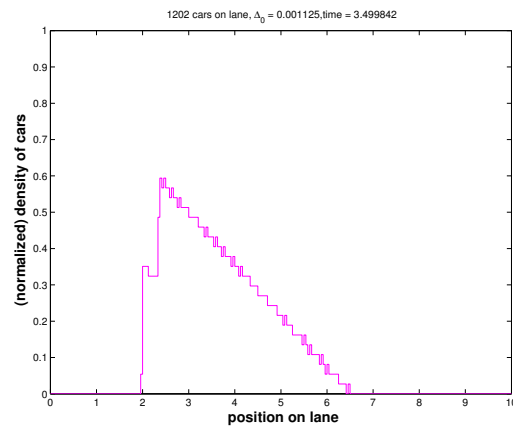
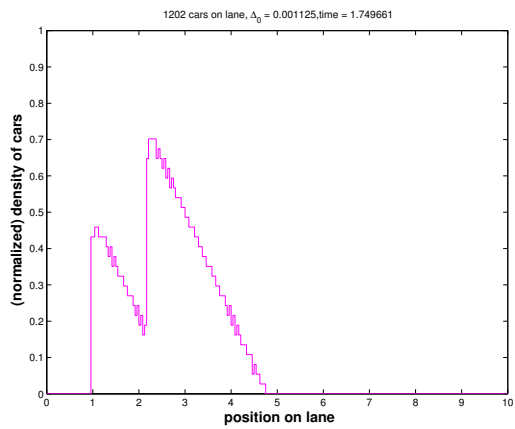
Example 8.1.36 (Particle simulation of traffic flow, cnt'd). \rightarrow Ex. 8.1.31

Now $x_0 = [0:2/k:1, 2:1/k:3]$ with $k=50, 200, 800, \Delta_0 = 0.9/k$, see (8.1.25), $\delta = 3.33k$ in (8.1.35). Simulation based on Code 8.1.32.

R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ





Striking observation:

For $N \rightarrow \infty$, $\Delta_0 \sim N^{-1}$, $\delta \sim N^{-1}$ the normalized car densities $u_\delta(x, t)$ seem to approach a *limit density*. What is it? Can it be obtained as a solution of a “limit model”. These issues will be addressed next.

Notice: Similar observation with the mass-spring model of Sect. 1.2.2 in the limit $n \rightarrow \infty$.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



8.1.2.2 Continuum traffic model

In Ex. 8.1.36 we observed the emergence of a stable limit density in the microscopic particle model of traffic flow according to (8.1.24) and (8.1.25), when the number of cars and their maximum density tended to ∞ in tandem, while the spatial averaging length tends to zero.

Now we derive a **macroscopic model** describing this limit.

(Macroscopic model \leftrightarrow stated in terms of macroscopic quantities)

Notice: Parallels with derivation of continuum elastic string model in Sect. 1.2.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 8.1.38 (Suitability of macroscopic models for traffic flow).

The limit $N \rightarrow \infty$ in traffic modeling is commonly denounced as dubious, because the number of cars on a road is way too small to render the limit a good approximation of actual traffic flow, see [2, Sect. 2.3].

Nevertheless, here we introduce a limit model, because

- it yields a least a qualitatively correct representation of patterns observed in real traffic flow,
- it provides an important **model problem** for scalar non-linear conservation laws.



Ingredients of macroscopic (continuum) traffic model:

- spatial domain $\Omega = \mathbb{R} \hat{=}$ infinitely long single highway lane (\rightarrow Cauchy problem),
- traffic flow described by the macroscopic quantity

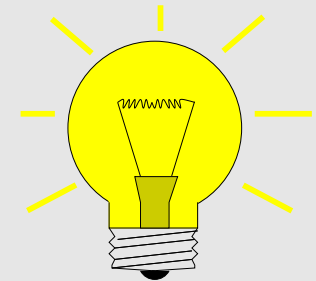
normalized density of cars $u : \Omega \times [0, T] \mapsto [0, 1]$,

- optimal velocity speed model (8.1.25).

However, (8.1.25) does not fit the spirit of macroscopic modeling: δx_i is not a macroscopic quantity!

Required: concept of a **macroscopic velocity**

Idea: spatial averaging of velocities of cars



$$v_\delta(x, t) = \frac{\sum_{i \in \mathcal{U}_\delta(x)} \dot{x}_i(t)}{\#\mathcal{U}_\delta} (x), \quad (8.1.43)$$

$$\mathcal{U}_\delta(x) := \{i \in \{1, \dots, N\} : x - \delta \leq x_i(t) < x + \delta\} .$$

▶ derived macroscopic quantity:

(normalized) **flux** of cars: $q_\delta(x, t) = u_\delta(x, t)v_\delta(x, t) . \quad (8.1.44)$

Interpretation: $q(x, t) \approx$ no. of cars passing site x in unit time around instance t in time.

▶ approximate **balance law** (“conservation of cars”)

$$\underbrace{\int_{x_0}^{x_1} u_\delta(x, t_1) dt - \int_{x_0}^{x_1} u_\delta(x, t_0) dt}_{\text{change of no. of cars on } [x_0, x_1] \text{ in } [t_0, t_1]} \approx \underbrace{\int_{t_0}^{t_1} q_\delta(x_0, t) dx - \int_{t_0}^{t_1} q_\delta(x_1, t) dx}_{\text{no. of cars entering/leaving } [x_0, x_1] \text{ in } [t_0, t_1]} . \quad (8.1.45)$$

Now we consider $N \rightarrow \infty$ (many cars) and $\delta \sim N^{-1} \rightarrow 0$ and drop the subscript δ , which hints at the averaging.

The balance law (8.1.46) will remain valid in the limit and will even become exact !

$$\underbrace{\int_{x_0}^{x_1} u(x, t_1) dt - \int_{x_0}^{x_1} u(x, t_0) dt}_{\text{change of no. of cars on } [x_0, x_1] \text{ in } [t_0, t_1]} = \underbrace{\int_{t_0}^{t_1} q(x_0, t) dx - \int_{t_0}^{t_1} q(x_1, t) dx}_{\text{no. of cars entering/leaving } [x_0, x_1] \text{ in } [t_0, t_1]} . \quad (8.1.46)$$

 R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

In the “infinitely many cars” limit $u(x, t)$, $v(x, t)$, and $q(x, t)$ can be expected to become (piecewise) smooth functions. This justifies the transition to a **differential macroscopic model**:

Temporarily assume that $u = u(x, t)$ is smooth in both x and t and set $x_1 = x_0 + h$, $t_1 = t_0 + \tau$. First approximate the integrals in (8.1.46).

$$\int_{x_0}^{x_1} u(x, t_1) - u(x, t_0) dx = h(u(x_0, t_1) - u(x_0, t_0)) + O(h^2) \quad \text{for } h \rightarrow 0 ,$$

$$\int_{t_0}^{t_1} q(x_1, t) - q(x_0, t) dt = \tau (q(x_1, t_0) - q(x_0, t_0)) + O(\tau^2) \quad \text{for } \tau \rightarrow 0 .$$

Then employ Taylor expansion for the differences:

$$u(x_0, t_1) - u(x_0, t_0) = \frac{\partial u}{\partial t}(x_0, t_0)\tau + O(\tau^2) \quad \text{for } \tau \rightarrow 0 ,$$

$$q(x_1, t_0) - q(x_0, t_0) = \frac{\partial q}{\partial x}(x_0, t_0)h + O(h^2) \quad \text{for } h \rightarrow 0 .$$

Finally, divide by h and τ and take the limit $\tau \rightarrow 0, h \rightarrow 0$:

$$\blacktriangleright \quad \frac{\partial u}{\partial t}(x, t) + \frac{\partial q}{\partial x}(x, t) = 0 \quad \text{in } \Omega \times]0, T[. \quad (8.1.48)$$

We still need to link u and q :

From (8.1.25) (with $v_{\max} = 1$ after rescaling) we deduce the macroscopic **constitutive relationship** between the (averaged and normalized) density (\rightarrow (8.1.35)) of cars and their averaged speed (\rightarrow

(8.1.43)):

$$v(x, t) = 1 - u(x, t) . \quad (8.1.52)$$

$$(8.1.48) \ \& \ (8.1.52) \ \& \ (8.1.44) \ \blacktriangleright \quad \boxed{\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} (u(1 - u)) = 0 \quad \text{in } \Omega \times]0, T[} . \quad (8.1.53)$$

+ macroscopic counterpart of initial conditions (8.1.26):

$$u(x, 0) = u_0(x) , \quad x \in \mathbb{R} . \quad (8.1.54)$$

8.1.3 Inviscid gas flow

Introduction. In this section we study modeling in **fluid mechanics**, a special field of continuum mechanics. In spirit this is close to the modeling of traffic flow in Sect. 8.1.2, because the macroscopic

behavior of fluids also results from the interaction of many small particles (molecules). However, in fluid mechanics the limit model for infinitely many particles enjoys a much more solid foundation than that for traffic, because the number of particles involved is tremendous ($\approx 10^{20} - 10^{30}$).



Fig. 251

Terminology: frictionless $\hat{=}$ inviscid

Assumption: variation of gas density negligible (“near incompressibility”)
 motion of fluid driven by inertia \leftrightarrow *conservation of linear momentum*

We derive a **continuum model** for inviscid, nearly incompressible fluid in a straight infinitely long pipe $\leftrightarrow \Omega = \mathbb{R}$ (Cauchy problem).

This simple model will be based on *conservation of linear momentum*, whereas conservation of mass and energy will be neglected (and violated). Hence, the crucial conserved quantity will be the momentum.

by near incompressibility

Unknown: $u = u(x, t)$ = momentum density \sim local velocity $v = v(x, t)$ of fluid

Conserved quantity:

(linear) **momentum** of fluid $u = u(x, t)$

- flux of linear momentum $f \sim v \cdot u$ (after scaling: $f(u) = \frac{1}{2}u \cdot u$)
 (“momentum u advected by velocity u ”)

Conservation of linear momentum ($\sim u$): for all control volumes $V :=]x_0, x_1[\subset \Omega$:

$$\underbrace{\int_{x_0}^{x_1} u(x, t_1) - u(x, t_0) dx}_{\text{change of momentum in } V} + \underbrace{\int_{t_0}^{t_1} \frac{1}{2}u^2(x_1, t) - \frac{1}{2}u^2(x_0, t) dt}_{\text{outflow of momentum}} = 0 \quad \forall 0 < t_0 < t_1 < T. \quad (8.1.59)$$

Temporarily assume that $u = u(x, t)$ is smooth in both x and t and set $x_1 = x_0 + h$, $t_1 = t_0 + \tau$.
 First approximate the integrals in (8.1.59).

$$\int_{x_0}^{x_1} u(x, t_1) - u(x, t_0) dx = h(u(x_0, t_1) - u(x_0, t_0)) + O(h^2) \quad \text{for } h \rightarrow 0,$$

$$\int_{t_0}^{t_1} \frac{1}{2}u^2(x_1, t) - \frac{1}{2}u^2(x_0, t) dt = \tau(\frac{1}{2}u^2(x_1, t_0) - \frac{1}{2}u^2(x_0, t_0)) + O(\tau^2) \quad \text{for } \tau \rightarrow 0.$$

Then employ Taylor expansion for the differences:

$$u(x_0, t_1) - u(x_0, t_0) = \frac{\partial u}{\partial t}(x_0, t_0)\tau + O(\tau^2) \quad \text{for } \tau \rightarrow 0 ,$$

$$\frac{1}{2}u^2(x_1, t_0) - \frac{1}{2}u^2(x_0, t_0) = \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right)(x_0, t_0)h + O(h^2) \quad \text{for } h \rightarrow 0 .$$

Finally, divide by h and τ and take the limit $\tau \rightarrow 0, h \rightarrow 0$:

$$\blacktriangleright \quad \frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0 \quad \text{in } \Omega \times]0, T[. \quad (8.1.60)$$

(8.1.60) = **Burgers equation**: a one-dimensional scalar conservation law (without sources)

Remark 8.1.61 (Euler equations).

The above gas model blatantly ignores the fundamental laws of conservation of mass and of energy. These are taken into account in a famous more elaborate model of inviscid fluid flow:

Euler equations [9], a more refined model for inviscid gas flow in an infinite pipe

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{pmatrix} = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad (8.1.62)$$

$$u(x, 0) = u_0(x) \quad , \quad \rho(x, 0) = \rho_0(x) \quad , \quad E(x, 0) = E_0(x) \quad \text{for } x \in \mathbb{R} ,$$

- where
- $\rho = \rho(x, t) \hat{=}$ fluid density, $[\rho] = \text{kg m}^{-1}$,
 - $u = u(x, t) \hat{=}$ fluid velocity, $[u] = \text{m s}^{-1}$,
 - $p = p(x, t) \hat{=}$ fluid pressure, $[p] = \text{N}$,
 - $E = E(x, t) \hat{=}$ total energy density, $[E] = \text{J m}^{-1}$.

+ **state equation** (material specific constitutive equations), e.g., for ideal gas

$$p = (\gamma - 1)(E - \frac{1}{2}\rho u^2), \quad \text{with adiabatic index } 0 < \gamma < 1.$$

Conserved quantities (**densities**):

$$\rho \leftrightarrow \text{mass density} \quad , \quad \rho u \leftrightarrow \text{momentum density} \quad , \quad E \leftrightarrow \text{energy density}.$$

Underlying physical conservation principles for individual densities:

- First equation $\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0 \leftrightarrow$ *conservation of mass*,
- Second equation $\frac{\partial(\rho u)}{\partial t} + \frac{\partial}{\partial x}(\rho u^2 + p) = 0 \leftrightarrow$ *conservation of momentum*,
- Third equation $\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}((E + p)u) = 0 \leftrightarrow$ *conservation of energy*.

Euler equations (8.1.62) = non-linear **system of conservation laws** (in 1D)

As is typical of non-linear systems of conservations laws, the analysis of the Euler equations is intrinsically difficult: hitherto not even existence and uniqueness of solutions for general initial values could be established. Moreover, solutions display a wealth of complicated structures. Therefore, this course is confined to scalar conservation laws, for which there is only one unknown real-valued function of space and time.

8.2 Scalar conservation laws in 1D

8.2.1 Integral and differential form

What we have seen so far (except for Euler's equations in Rem. 8.1.61)

$$\text{Burgers equation: } \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{1}{2} u^2 \right) = 0 \quad \text{in } \Omega \times]0, T[, \quad (8.1.60)$$

$$\text{traffic flow equation: } \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} (u(1 - u)) = 0 \quad \text{in } \Omega \times]0, T[, \quad (8.1.53)$$

$$\text{linear advection: } \frac{\partial}{\partial t} (\rho u) + \text{div}(\mathbf{v}(\mathbf{x}, t)(\rho u)) = f(\mathbf{x}, t) \quad \text{in } \mathbb{R}^d \times]0, T[. \quad (8.1.1)$$

Now, we learn about a class of Cauchy problems to which these three belong. First some notations and terminology:

- $\Omega \subset \mathbb{R}^d \hat{=}$ fixed (bounded/unbounded) spatial domain ($\Omega = \mathbb{R}^d =$ Cauchy problem)
- computational domain: space-time cylinder $\tilde{\Omega} := \Omega \times]0, T[$, $T > 0$ final time
- $U \subset \mathbb{R}^m$ ($m \in \mathbb{N}$) $\hat{=}$ **phase space** (state space) for **conserved quantities** u_i (usually $U = \mathbb{R}^m$)

Our focus below:

scalar case $m = 1$

Conservation law for transient state distribution $u : \tilde{\Omega} \mapsto U : u = u(\mathbf{x}, t)$, for $0 \leq t \leq T$

$$\frac{d}{dt} \int_V u \, d\mathbf{x} + \int_{\partial V} \mathbf{f}(u, \mathbf{x}) \cdot \mathbf{n} \, dS(\mathbf{x}) = \int_V s(u, \mathbf{x}, t) \, d\mathbf{x} \quad \forall \text{ "control volumes" } V \subset \Omega. \quad (8.2.1)$$

change of amount

inflow/outflow

production term

Terminology: \triangleright **flux function** $\mathbf{f} : U \times \Omega \mapsto \mathbb{R}^d$
 \triangleright **source function** $s : U \times \Omega \times]0, T[\mapsto \mathbb{R}$ (here usually $s = 0$)

- For Burgers equation (8.1.60): $f(u, x) = f(u) = \frac{1}{2}u^2$, $s = 0$,
- For traffic flow equation (8.1.53): $f(u, x) = f(u) = u(1 - u)$, $s = 0$,
- For linear advection (8.1.1): $\mathbf{f}(u, \mathbf{x}) = \mathbf{v}(\mathbf{x}, t)u$, $s = f(\mathbf{x}, t)$
 (Note: in this case the conserved quantity is actually ρu , which was again denoted by u)

☞ (8.2.1) has the same structure as the “conservation of energy law” (6.1.1) for heat conduction.

Conservation of energy:

$$\frac{d}{dt} \int_V \rho u \, d\mathbf{x} + \int_{\partial V} \mathbf{j} \cdot \mathbf{n} \, dS = \int_V f \, d\mathbf{x} \quad \text{for all “control volumes” } V \quad (6.1.1)$$

energy stored in V

power flux through ∂V

heat generation in V

In this case the heat flux was given by

$$\text{Fourier's law} \quad \mathbf{j}(\mathbf{x}) = -\kappa(\mathbf{x}) \mathbf{grad} u(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (2.5.3)$$

or its extended version (7.1.5). In Fourier's law the flux is a *linear* function of *derivatives* of u .

Conversely, for the **flux function** $\mathbf{f} : U \times \Omega \mapsto \mathbb{R}^d$ in (8.2.1) we assume

\mathbf{f} only depends on local state u , not on derivatives of u !

On the other hand we go far beyond Fourier's law, since

\mathbf{f} will in general be a *non-linear* function of u !

Remark 8.2.3 (Diffusive flux).

Taking into account the relationship with heat “diffusion”, a flux function of the form of Fourier’s law (2.5.3)

$$\mathbf{f}(u) = -\kappa(\mathbf{x}) \operatorname{grad} u ,$$

is called a **diffusive flux**.



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Now, integrate (8.2.1) over time period $[t_0, t_1] \subset [0, T]$ and use fundamental theorem of calculus:

► Space-time **integral form** of (8.2.1), *cf.* (8.1.59),

$$\int_V u(\mathbf{x}, t_1) \, d\mathbf{x} - \int_V u(\mathbf{x}, t_0) \, d\mathbf{x} + \int_{t_0}^{t_1} \int_{\partial V} \mathbf{f}(u, \mathbf{x}) \cdot \mathbf{n} \, dS(\mathbf{x}) \, dt = \int_{t_0}^{t_1} \int_V s(u, \mathbf{x}, t) \, d\mathbf{x} \, dt \quad (8.2.4)$$

for all $V \subset \Omega, 0 < t_0 < t_1 < T, \mathbf{n} \hat{=} \text{exterior unit normal at } \partial V$

► [Gauss theorem Thm. 2.4.9] (local) **differential form** of (8.2.1):

$$\frac{\partial}{\partial t} u + \operatorname{div}_{\mathbf{x}} \mathbf{f}(u, \mathbf{x}) = s(u, \mathbf{x}, t) \quad \text{in } \tilde{\Omega} . \quad (8.2.5)$$

div acting on spatial variable \mathbf{x} only

+ initial condition

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega$$

Special case $d = 1 \iff (8.2.5) = \text{one-dimensional scalar conservation law for "density" } u : \tilde{\Omega} \mapsto \mathbb{R}$

$$\frac{\partial u}{\partial t}(x, t) + \frac{\partial}{\partial x} (f(u(x, t), x)) = s(u(x, t), x, t) \quad \text{in }]\alpha, \beta[\times]0, T[, \alpha, \beta \in \mathbb{R} \cup \{\pm\infty\} . \quad (8.2.6)$$

Remark 8.2.7 (Boundary values for conservation laws).

Suitable boundary values on $\partial\Omega \times]0, T[$? \rightarrow usually tricky question (highly f -dependent)

Reason: remember discussion in Rem. 8.1.11, meaningful boundary conditions hinge on knowledge of local (in space and time) transport direction, which, in a *non-linear* conservation law, will usually depend on the unknown solution $u = u(x, t)$.



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

8.2.2 Characteristics

We consider Cauchy problem ($\Omega = \mathbb{R}$) for one-dimensional scalar conservation law (8.2.6):

$$\blacktriangleright \begin{cases} \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 & \text{in } \mathbb{R} \times]0, T[, \\ u(x, 0) = u_0(x) & \text{in } \mathbb{R} . \end{cases} \quad (8.2.9)$$

Assumption:

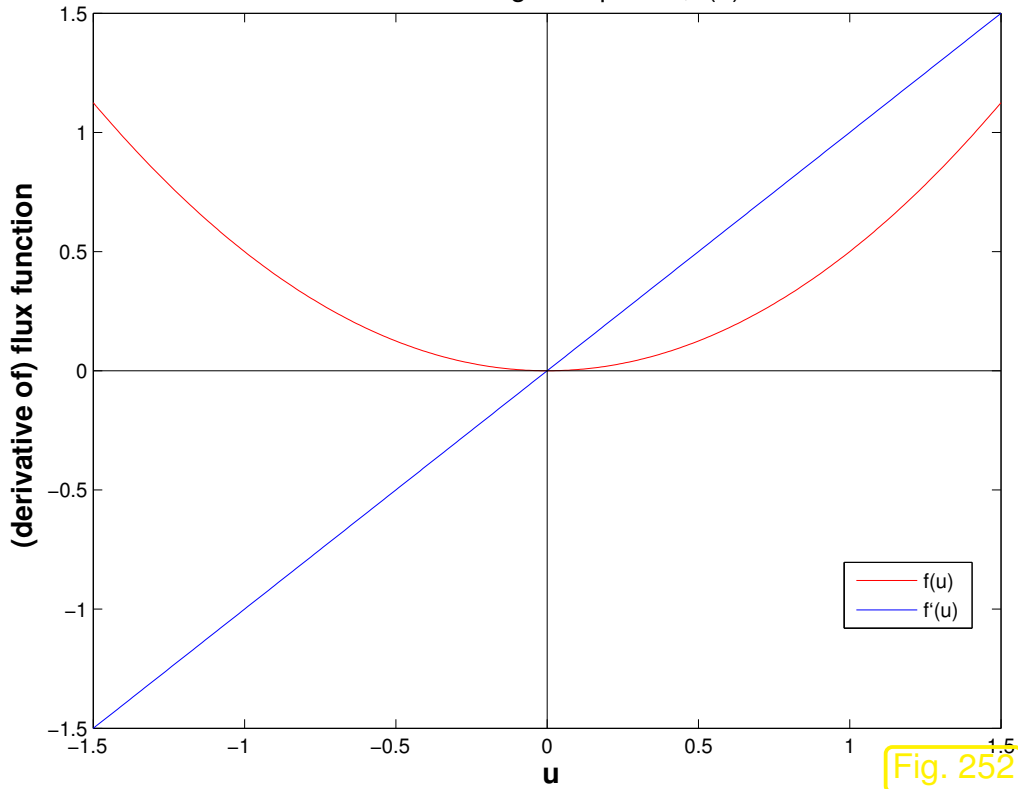
flux function $f : \mathbb{R} \mapsto \mathbb{R}$ smooth ($f \in C^2$),
and **convex** or **concave** [32, Def. 5.5.2]

Recall [32, Thm. 5.5.2]:

f convex \Rightarrow derivative f' increasing
 f concave \Rightarrow derivative f' decreasing

flux function for Burgers' equation (8.1.60)

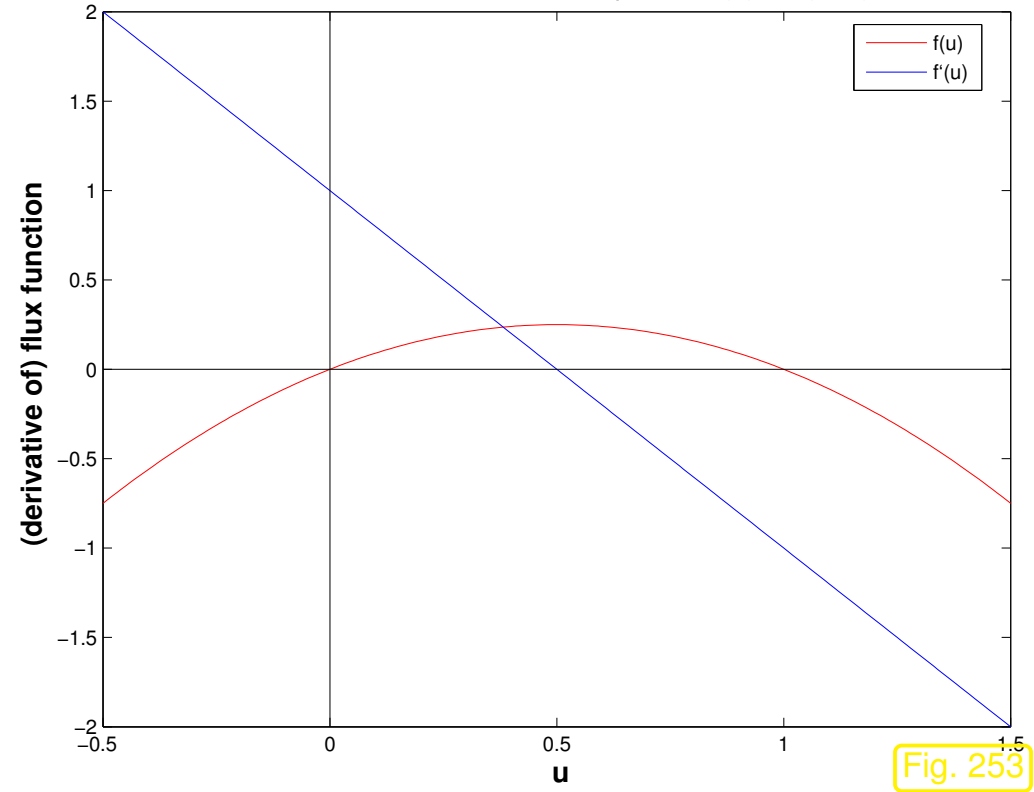
Flux function for Burgers equation, $f(u) = 1/2u^2$



f convex

flux function for traffic flow equation (8.1.53)

Flux function for traffic flow equation, $f(u) = u(1-u)$



f concave

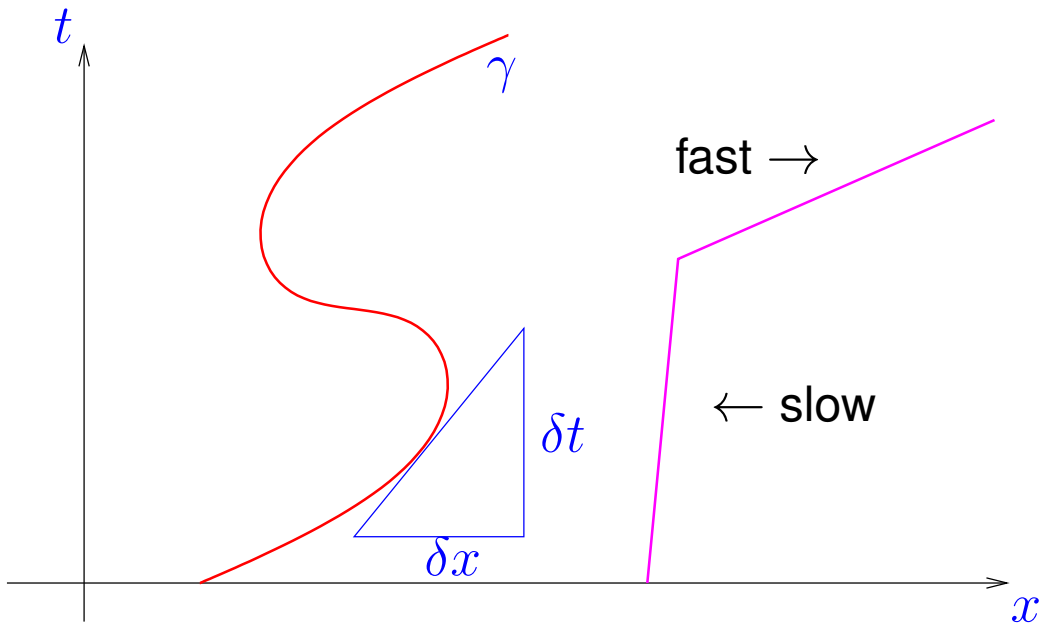
Burgers' equation (8.1.60) and the traffic flow equation (8.1.53) will serve as main examples for scalar conservation laws in one spatial dimension. The opposite curvatures of their flux functions will be reflected by a “mirror symmetric” behavior of their solutions in many cases. Below most examples will be discussed for both model problems in order to elucidate these differences, but the reader may focus on only one model problem.

Definition 8.2.10 (Characteristic curve for one-dimensional scalar conservation law).

A curve $\Gamma := (\gamma(\tau), \tau) : [0, T] \mapsto \mathbb{R} \times]0, T[$ in the (x, t) -plane is a **characteristic curve** of (8.2.9), if

$$\frac{d}{d\tau}\gamma(\tau) = f'(u(\gamma(\tau), \tau)) , \quad 0 \leq \tau \leq T , \quad (8.2.11)$$

where u is a continuously differentiable solution of (8.2.9).



◁ $x - t$ -diagram

$$\frac{d}{d\tau}\gamma(\tau) = \text{speed of interface } \gamma.$$

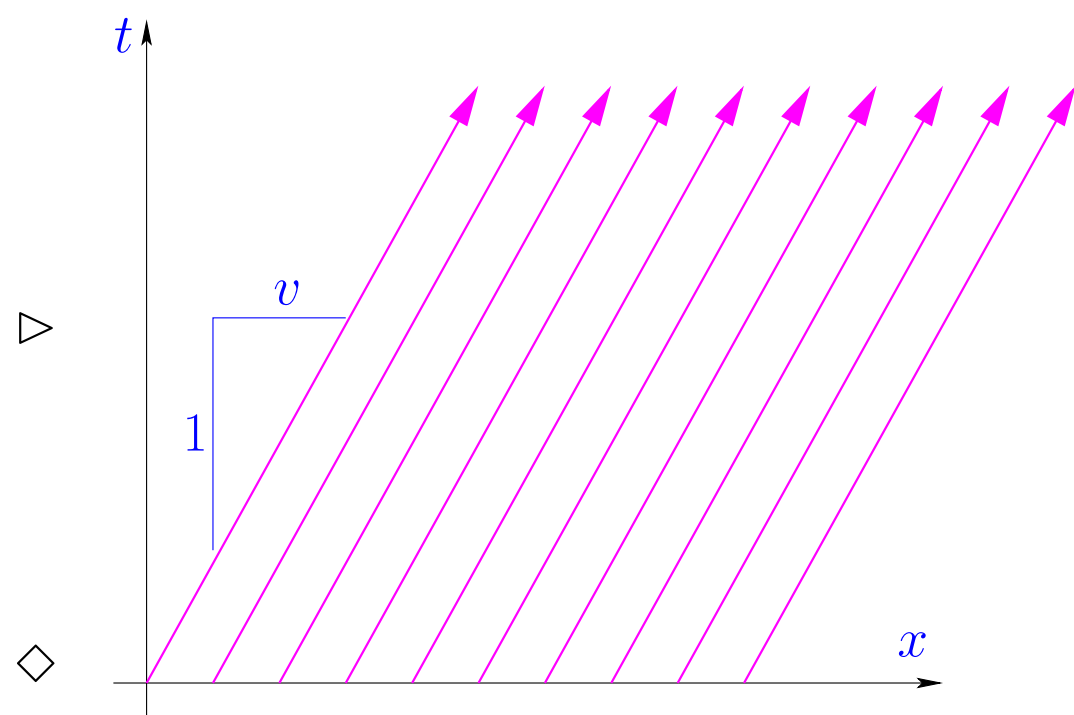
Example 8.2.12 (Characteristics for advection).

Constant linear advection (8.1.5): $f(u) = vu$

➔ characteristics $\gamma(\tau) = v\tau + c, c \in \mathbb{R}$.

solution (8.1.6) $u(x, t) = u_0(x - vt)$

meaningful for *any* u_0 ! (cf. Sect. 7.3.2)




This example reveals a close relationship between streamlines (\rightarrow Sect. 7.1.1) and characteristic curves. That the latter are a true generalization of the former is also reflected by the following simple observation, which generalizes the considerations in Sect. 7.3.2, (7.3.9).

Lemma 8.2.13 (Classical solutions and characteristic curves).

Smooth solutions of (8.2.9) are constant along characteristic curves.

Proof. Apply chain rule twice, cf. (7.3.9), and use the defining equation (8.2.11) for a characteristic curve:

$$\begin{aligned} \frac{d}{d\tau}u(\gamma(\tau), \tau) &\stackrel{\text{chain rule}}{=} \frac{\partial u}{\partial x}(\gamma(\tau), \tau) \frac{d}{d\tau}\gamma(\tau) + \frac{\partial u}{\partial t}(\gamma(\tau), \tau) \\ &\stackrel{(8.2.11)}{=} \frac{\partial u}{\partial x}(\gamma(\tau), \tau) \cdot f'(u(\gamma(\tau), \tau)) + \frac{\partial u}{\partial t}(\gamma(\tau), \tau) \\ &\stackrel{\text{chain rule}}{=} \left(\frac{\partial}{\partial x} f(u) \right) (\gamma(\tau), \tau) + \frac{\partial u}{\partial t}(\gamma(\tau), \tau) = 0 . \end{aligned}$$

 notation: $f' \hat{=}$ derivative of flux function $f : U \subset \mathbb{R} \mapsto \mathbb{R}$

So, u is constant on a characteristic curve.

➤ $f'(u)$ is constant on a characteristic curve.

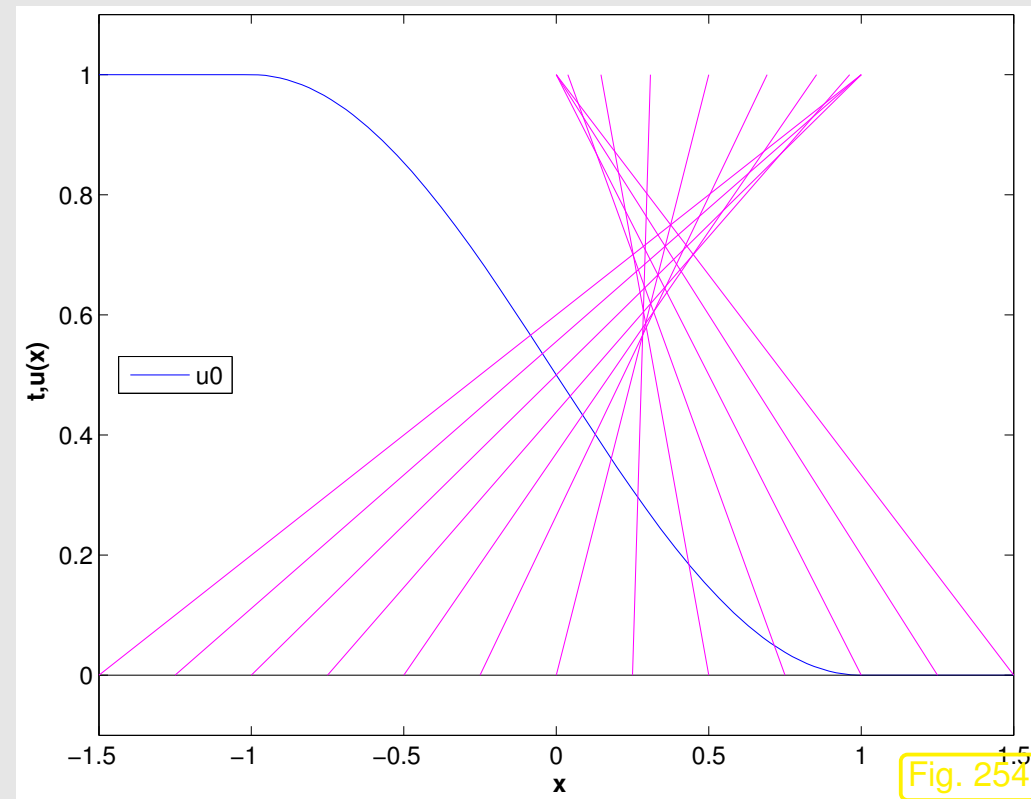
(8.2.11) \Rightarrow slope of characteristic curve is constant!

▶ Characteristic curve through $(x_0, 0) = \text{straight line } (x_0 + f'(u_0(x_0))\tau, \tau), 0 \leq \tau \leq T !$

!?! implicit solution formula for (8.2.9) (f' monotone !):

$$u(x, t) = u_0(x - f'(u(x, t))t) . \quad (8.2.14)$$

Example 8.2.15 (Breakdown of characteristic solution formula).



for Burger's equation (8.1.60):

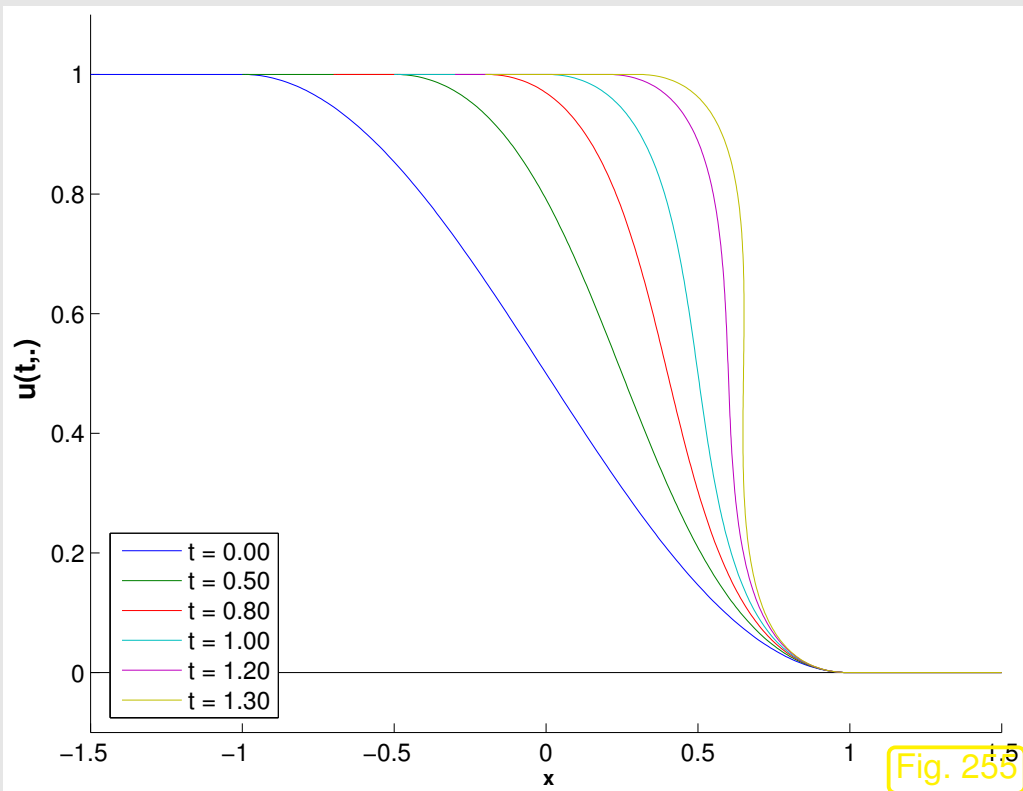
$(f(u) = \frac{1}{2}u^2$ smooth and strictly convex)

▷ $f'(u) = u$ (increasing)

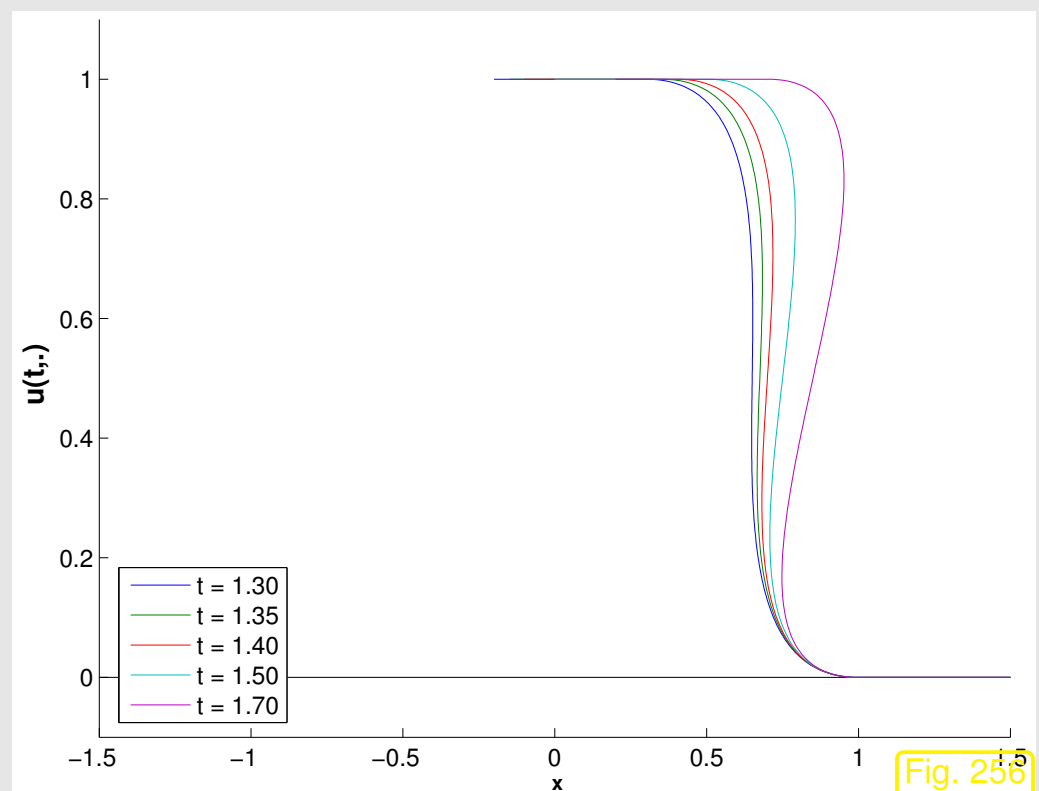
◁ if u_0 smooth and decreasing

➤ characteristic curves intersect !

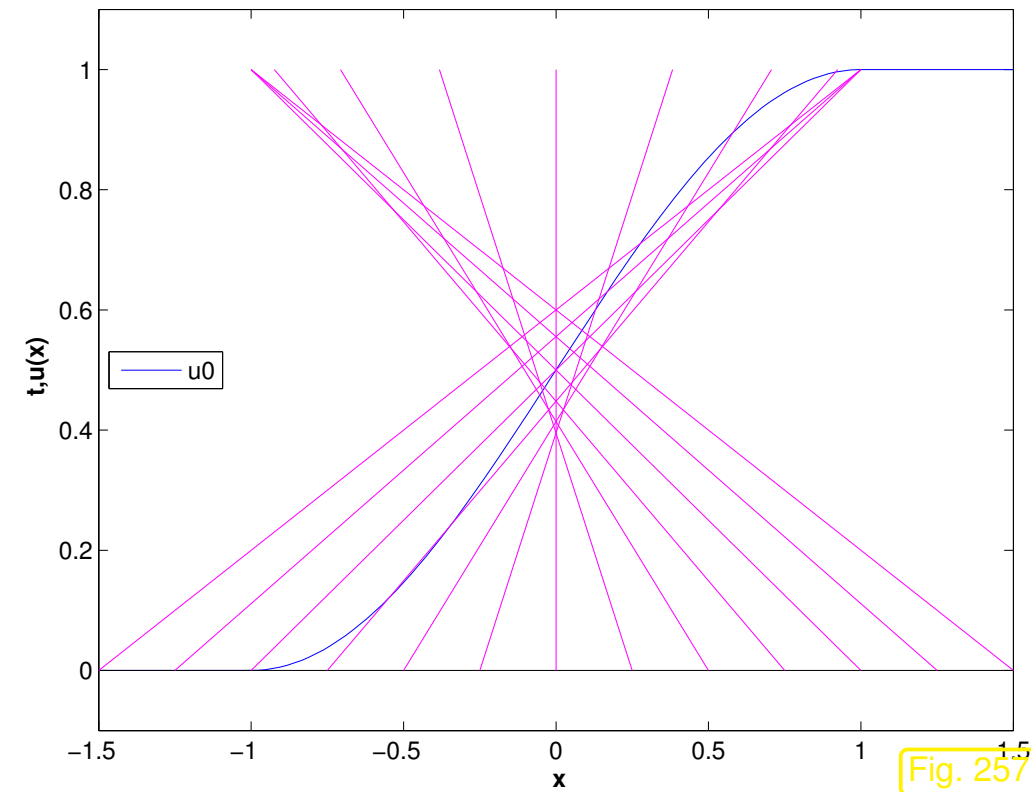
➤ solution formula (8.2.14) becomes invalid



$t < 1.3$: solution by (8.2.14)



the wave breaks: “multivalued solution”



for traffic flow equation (8.1.53):

($f(u) = u(1 - u)$ smooth and strictly concave)

▷ $f'(u) = 1 - 2u$ (decreasing)

◁ if u_0 smooth and increasing

➤ characteristic curves intersect !

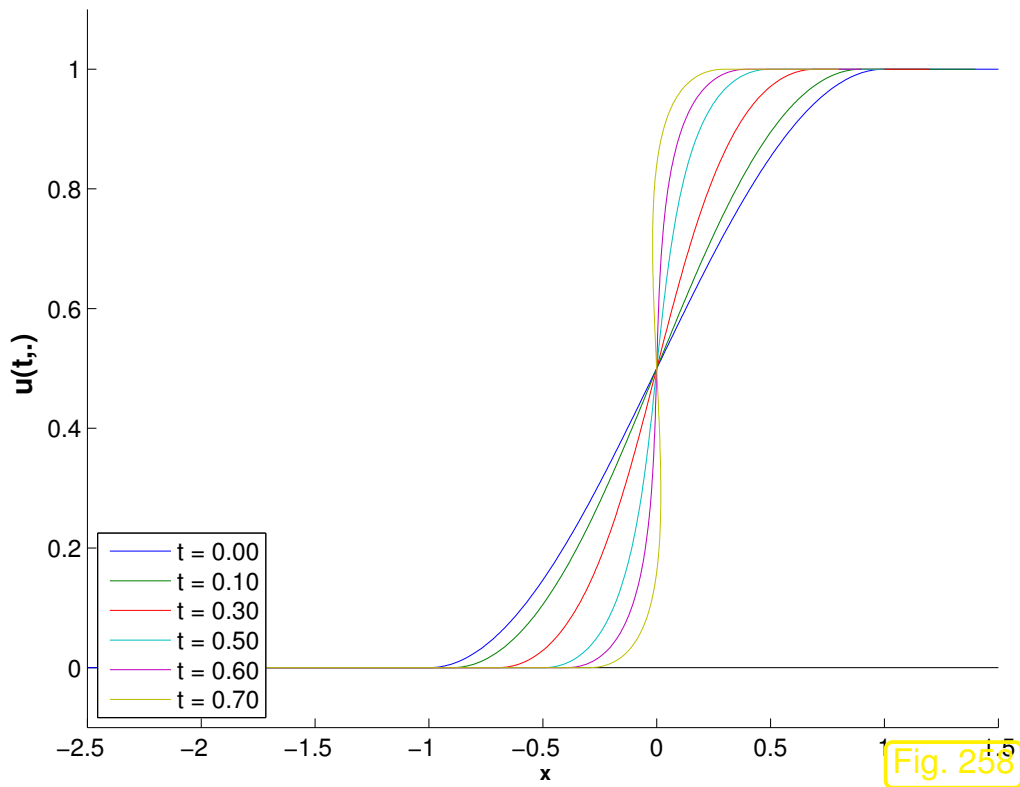


Fig. 258

$t < 0.7$: solution by (8.2.14)

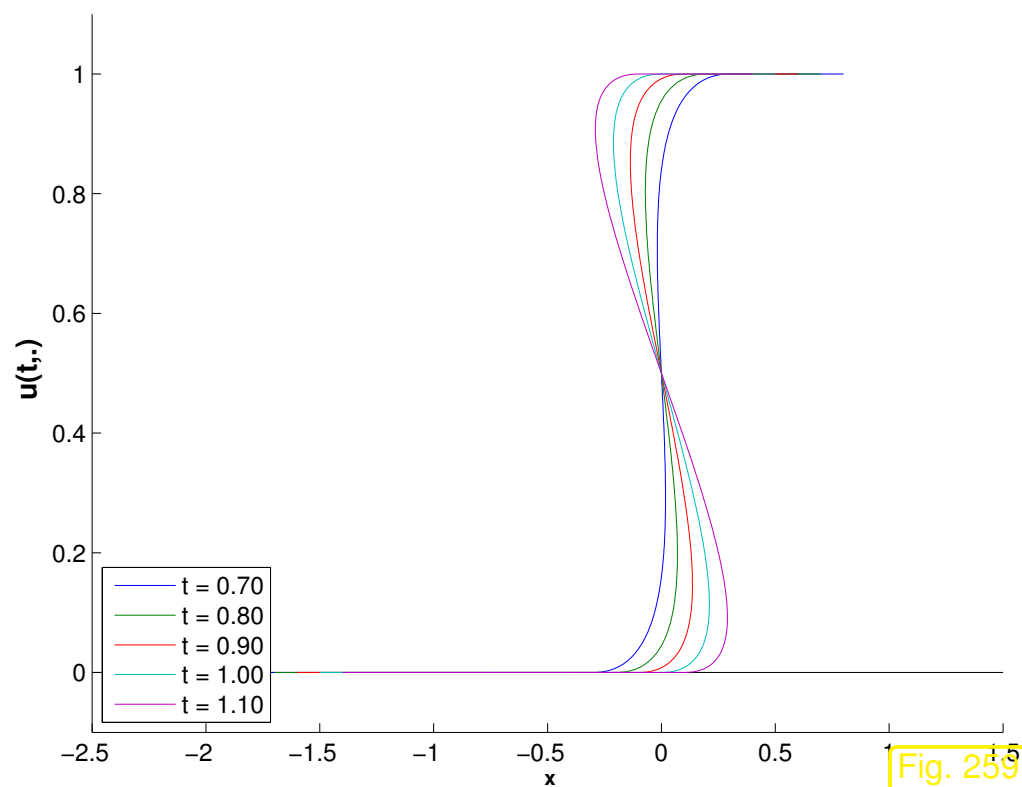


Fig. 259

the wave breaks: “multivalued solution”



breakdown of classical solutions & Ex. 8.2.12 → new concept of solution of (8.2.9)

8.2.3 Weak solutions

“Space-time Gaussian theorem”

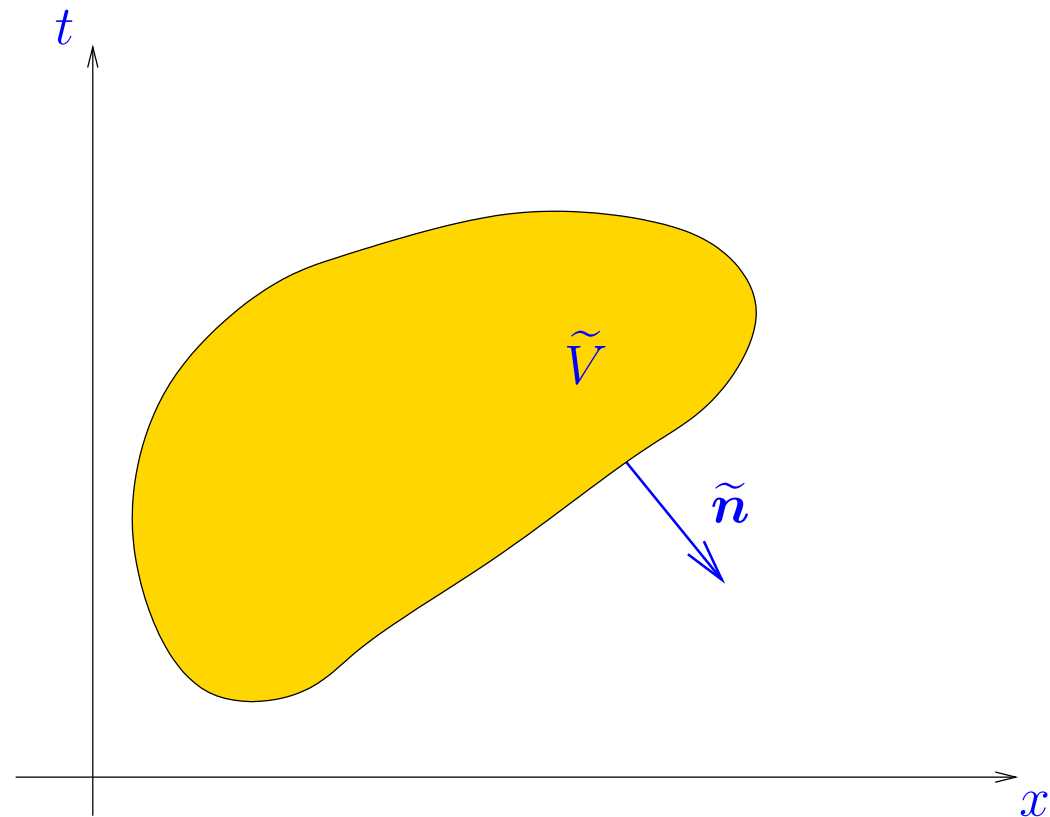
$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad (8.2.16)$$

$$\begin{array}{c} \Downarrow \\ \text{div}_{(x,t)} \begin{pmatrix} f(u) \\ u \end{pmatrix} = 0 \quad \text{in } \tilde{\Omega}. \end{array} \quad (8.2.17)$$

► \forall “space-time control volumes” $\tilde{V} \subset \tilde{\Omega}$:

$$\int_{\partial \tilde{V}} \begin{pmatrix} f(u(\tilde{\mathbf{x}})) \\ u(\tilde{\mathbf{x}}) \end{pmatrix} \cdot \begin{pmatrix} n_x(\tilde{\mathbf{x}}) \\ n_t(\tilde{\mathbf{x}}) \end{pmatrix} dS(\tilde{\mathbf{x}}) = 0,$$

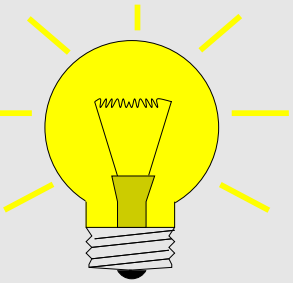
$\tilde{\mathbf{n}} := (n_x, n_t)^T \hat{=}$ space-time unit normal



(8.2.17) for space-time rectangle $\tilde{V} =]x_0, x_1[\times]t_0, t_1[$ ► **integral form** of (8.2.16), cf. (8.2.4):

$$\int_{x_0}^{x_1} u(x, t_1) dx - \int_{x_0}^{x_1} u(x, t_0) dx = \int_{t_0}^{t_1} f(u(x_0, t)) dt - \int_{t_0}^{t_1} f(u(x_1, t)) dt. \quad (8.2.18)$$

Still, (8.2.18) encounters problems, if a discontinuity of u coincides with an edge of the space-time rectangle.



Idea: Obtain weak form of (8.2.16) from (8.2.17) by integration by parts, that is, application of Green's first formula Thm. 2.4.11 in space-time!

STEP I: Test (8.2.17) with **compactly supported smooth** function $\Phi : \tilde{\Omega} \mapsto \mathbb{R}$, $\Phi(\cdot, T) = 0$, and integrate over space-time cylinder $\tilde{\Omega} = \mathbb{R} \times [0, T]$:

$$(8.2.17) \quad \blacktriangleright \quad \int_{\tilde{\Omega}} \operatorname{div}_{(x,t)} \begin{pmatrix} f(u) \\ u \end{pmatrix} \Phi(\mathbf{x}, t) \, d\mathbf{x} \, dt = 0 .$$

STEP II: Perform integration by parts using Green's first formula Thm. 2.4.11 on $\tilde{\Omega}$:

$$\int_{\tilde{\Omega}} \operatorname{div}_{(x,t)} \begin{pmatrix} f(u) \\ u \end{pmatrix} \Phi(\mathbf{x}, t) \, d\mathbf{x} \, dt = 0$$

$$\stackrel{\text{Thm. 2.4.11}}{\Rightarrow} \int_{\tilde{\Omega}} \begin{pmatrix} f(u) \\ u \end{pmatrix} \cdot \mathbf{grad}_{(x,t)} \Phi \, d\mathbf{x} \, dt + \int_{-\infty}^{\infty} u(x, 0) \Phi(x, 0) \, dx = 0 ,$$

because $\partial\tilde{\Omega} = \mathbb{R} \times \{0\} \cup \mathbb{R} \times \{T\}$ with “normals” $\mathbf{n} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ ($t = 0$ boundary) and $\mathbf{n} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ ($t = T$ boundary), which has to be taken into account in the boundary term in Green’s formula. The “ $t = T$ boundary part” does not enter as $\Phi(\cdot, T) = 0$.

Note that $u(x, 0)$ is fixed by the initial condition: $u(x, 0) = u_0(x)$.

Definition 8.2.19 (Weak solution of Cauchy problem for scalar conservation law).

For $u_0 \in L^\infty(\mathbb{R})$, $u : \mathbb{R} \times]0, T[\mapsto \mathbb{R}$ is a **weak solution** of the Cauchy problem (8.2.9), if

$$u \in L^\infty(\mathbb{R} \times]0, T[) \quad \wedge \quad \int_{-\infty}^{\infty} \int_0^T \left\{ u \frac{\partial \Phi}{\partial t} + f(u) \frac{\partial \Phi}{\partial x} \right\} dt dx + \int_{-\infty}^{\infty} u_0(x) \Phi(x, 0) dx = 0,$$

for all $\Phi \in C_0^\infty(\mathbb{R} \times [0, T[)$, $\Phi(\cdot, T) = 0$.

Remark 8.2.21 (Properties of weak solutions).

By reversing integration by parts, it is easy to see that

$$u \text{ weak solution of (8.2.9) \& } u \in C^1 \iff u \text{ classical solution of (8.2.9).}$$

Arguments from mathematical integration theory confirm

$$u \in L_{\text{loc}}^{\infty}(\mathbb{R} \times]0, T[) \text{ weak solution of (8.2.9)} \implies \begin{array}{l} u \text{ satisfies integral form (8.2.18)} \\ \text{for "almost all" } x_0 < x_1, 0 < t_0 < t_1 < T. \end{array}$$



8.2.4 Jump conditions

For piecewise smooth vectorfield $\mathbf{j} : \Omega \subset \mathbb{R}^2$:

$$\text{“div } \mathbf{j} = 0\text{”}$$

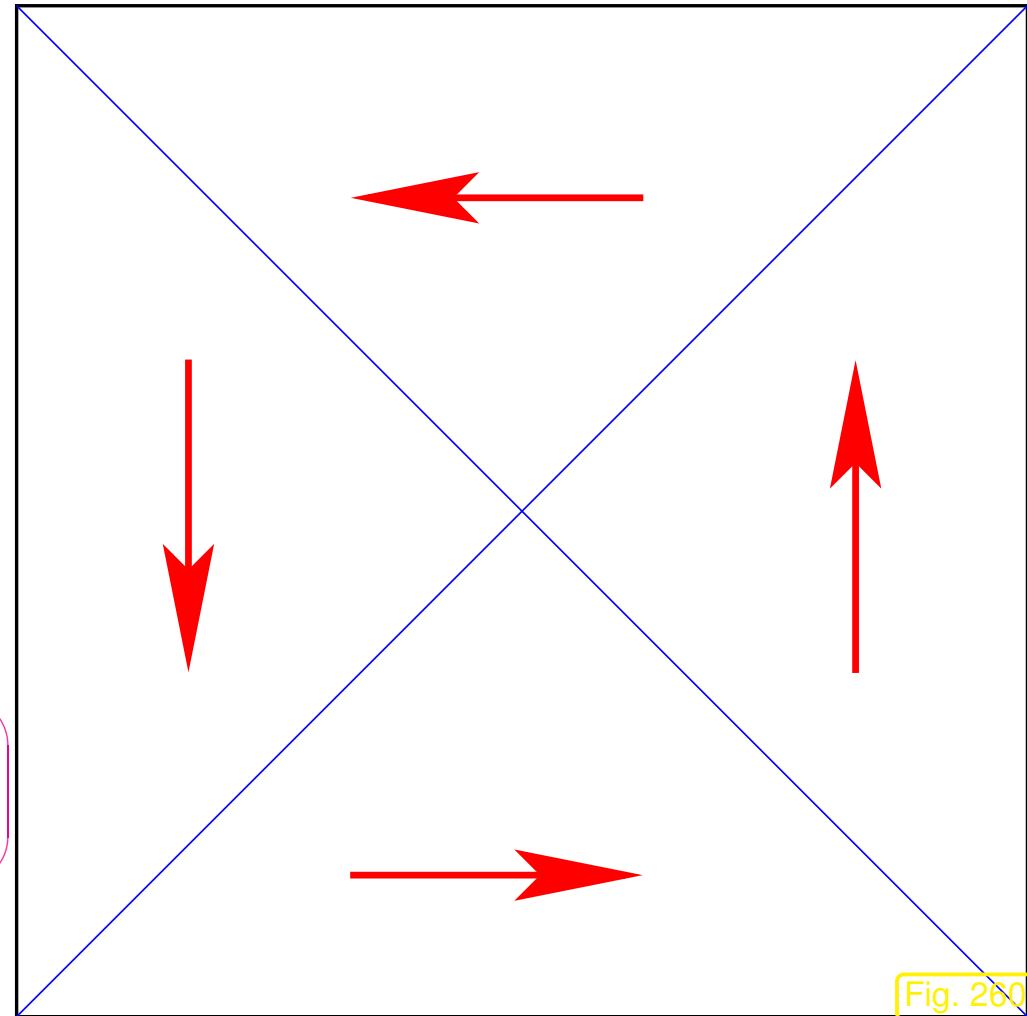


$$\int_{\partial V} \mathbf{j} \cdot \mathbf{n} \, dS = 0 \quad \forall \text{ control volumes } V \subset \Omega$$

Necessary condition:

Continuity of **normal components**
across discontinuities

discontinuous divergence-free vectorfield \triangleright



R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

To see this, consider a slender tiny rectangle aligned with a line of discontinuity of \mathbf{j} . In the absence of normal continuity a net flux through its boundary will result, provided that the rectangle is small enough (“pillbox argument”).

Apply this insight to vectorfield on space-time domain $\tilde{\Omega} = \mathbb{R} \times]0, T[$:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \Leftrightarrow \quad \operatorname{div}_{(x,t)} \underbrace{\begin{pmatrix} f(u) \\ u \end{pmatrix}}_{=: \mathbf{j}} = 0 \quad \text{in } \tilde{\Omega}. \quad (8.2.17)$$

Normal at C^1 -curve $\Gamma := \tau \mapsto (\gamma(\tau), \tau)$ in $(\gamma(\tau), \tau)$

$$\tilde{\mathbf{n}} = \frac{1}{\sqrt{1 + |\dot{s}|^2}} \begin{pmatrix} 1 \\ -\dot{s} \end{pmatrix}, \quad \dot{s} := \frac{d\gamma}{d\tau}(\tau) \quad \text{“speed of curve”}.$$

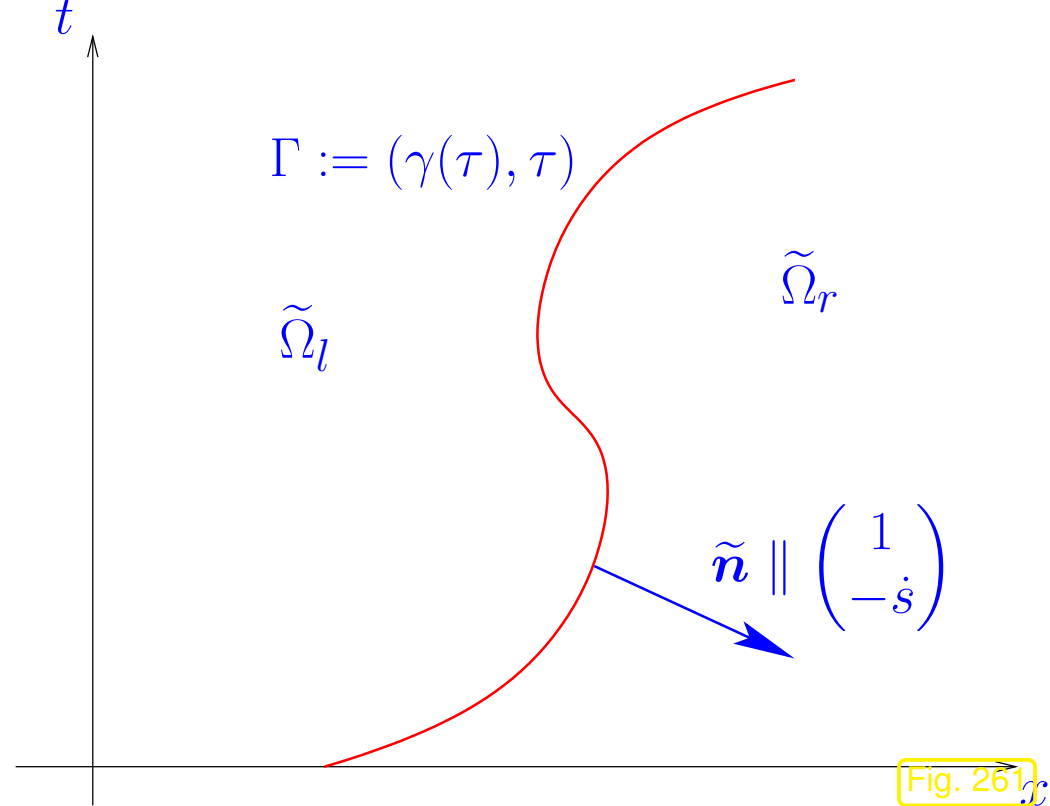
To see this, recall that the normal is orthogonal to the tangent vector $\begin{pmatrix} \dot{s} \\ 1 \end{pmatrix}$ and that in 2D the direction orthogonal to $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ is given by $\begin{pmatrix} -x_2 \\ x_1 \end{pmatrix}$.

“normal continuity” of piecewise smooth vectorfield $(f(u), u)^T$

\Leftrightarrow

$$\begin{pmatrix} 1 \\ -\frac{d\gamma}{d\tau} \end{pmatrix} \cdot \begin{pmatrix} [f(u)] \\ [u] \end{pmatrix} = 0, \quad (8.2.22)$$

where $[\cdot] \hat{=}$ jump across Γ .



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

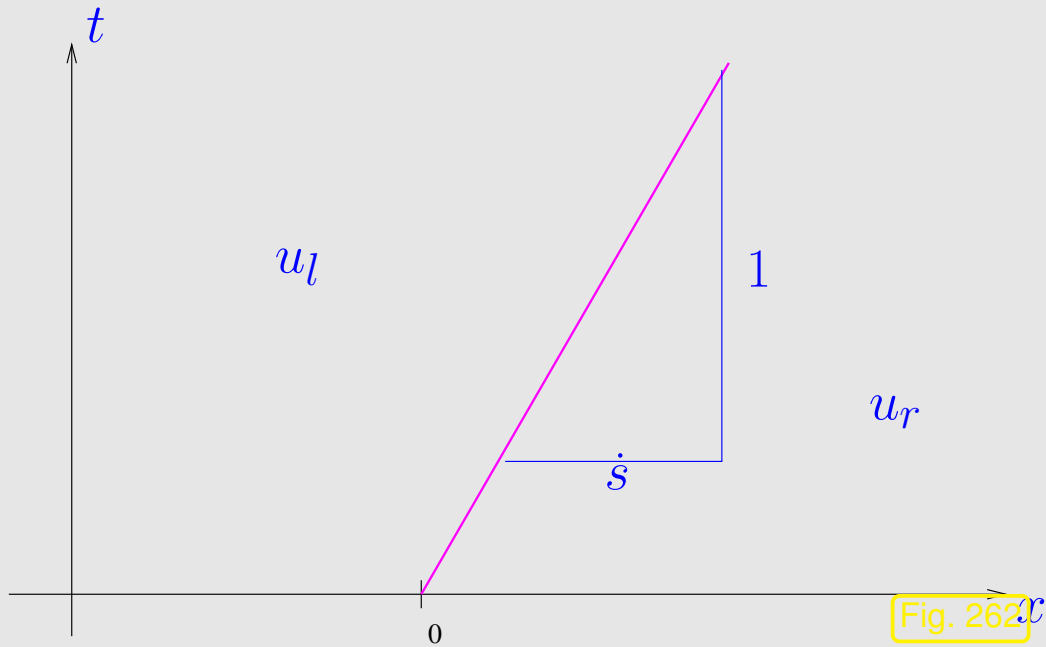
SAM, ETHZ

Terminology: (8.2.22) = Rankine-Hugoniot (jump) condition, shorthand notation:

$$\dot{s}(u_l - u_r) = f_l - f_r, \quad \dot{s} := \frac{d\gamma}{d\tau} \quad \text{“propagation speed of discontinuity”} \quad (8.2.23)$$

Remark 8.2.26 (Discontinuity connecting constant states).

The simplest situation compliant with Rankine-Hugoniot jump condition: *constant states* to the left and right of the curve of discontinuity (8.2.22):



$$u(x, t) = \begin{cases} u_l \in \mathbb{R} & , \text{ for } x < \dot{s}t , \\ u_r \in \mathbb{R} & , \text{ for } x > \dot{s}t , \end{cases} \quad (8.2.27)$$

with **constant** speed \dot{s} of discontinuity, according to (8.2.23) given by (for $u_l \neq u_r$)

$$\dot{s} = \frac{f(u_l) - f(u_r)}{u_l - u_r} .$$



8.2.5 Riemann problem

Rem. 8.2.26: situation of locally constant states is particularly easy.

► Consider: Cauchy-problem (8.2.9) for piecewise constant initial data u_0 .

Definition 8.2.28 (Riemann problem).

$$u_0(x) = \begin{cases} u_l \in \mathbb{R} & , \text{ if } x < 0 , \\ u_r \in \mathbb{R} & , \text{ if } x > 0 . \end{cases} \hat{=} \text{Riemann problem for (8.2.9)}$$

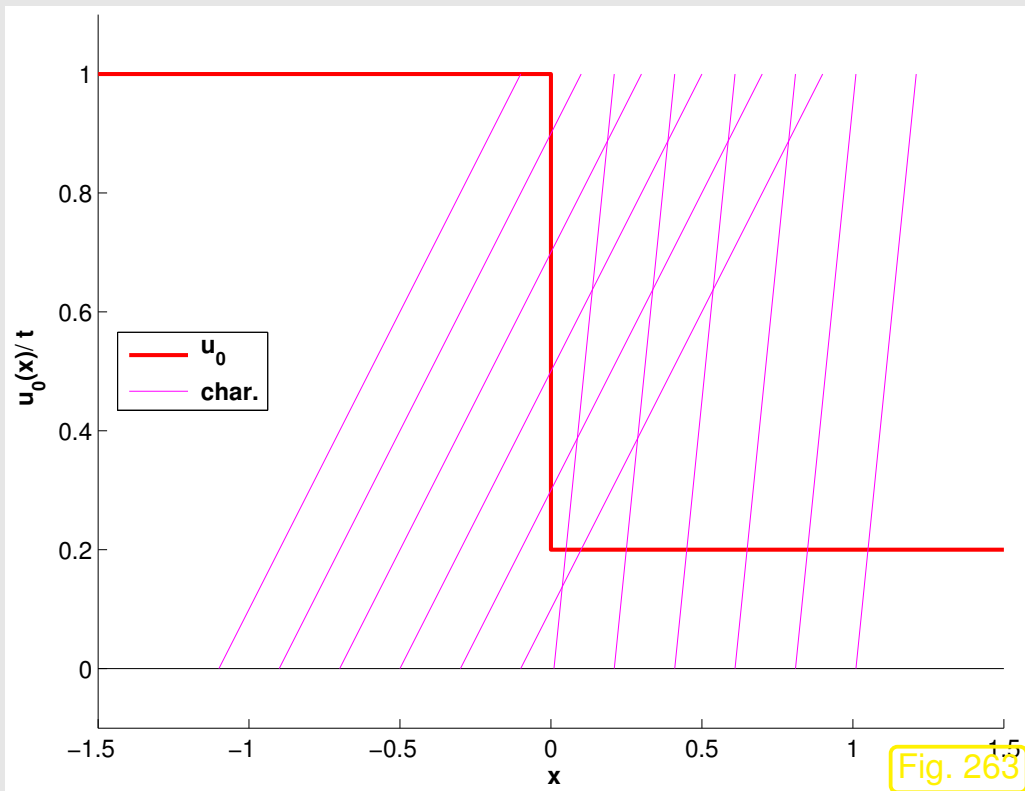
R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

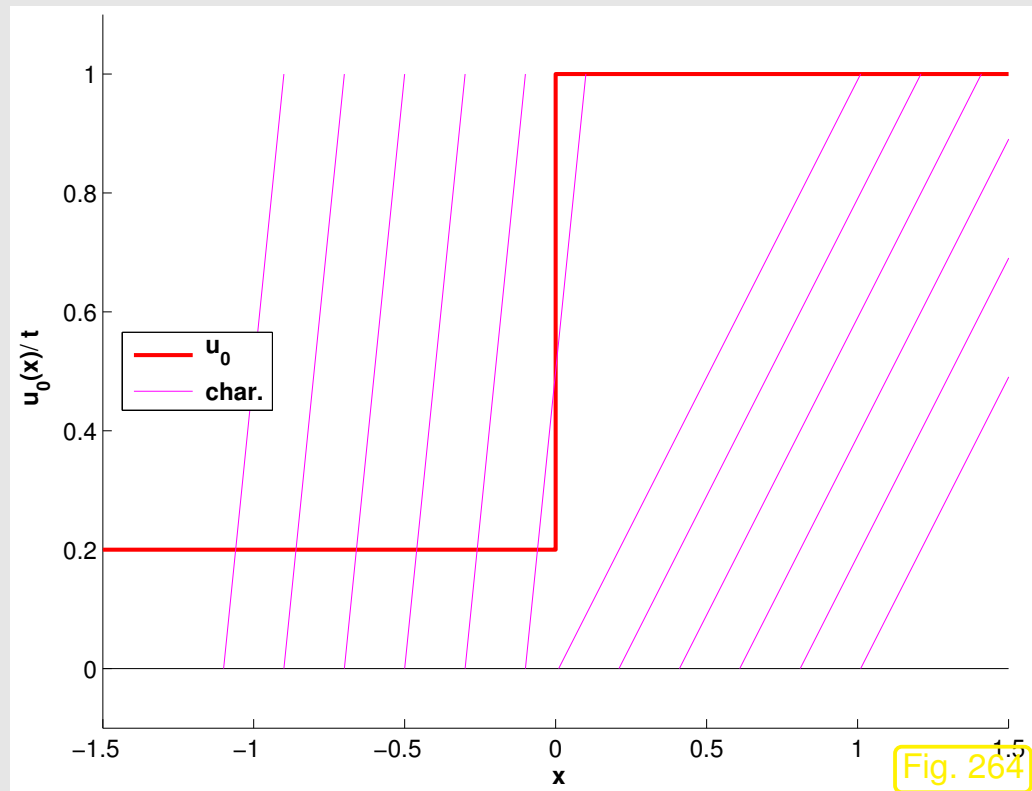
Assumption, cf. Sect. 8.2.2:

flux function $f : \mathbb{R} \mapsto \mathbb{R}$ smooth & convex

► f' non-decreasing ► pattern of characteristic curves for Riemann problem:



intersecting characteristics



diverging characteristics

Assumption, cf. Sect. 8.2.2:

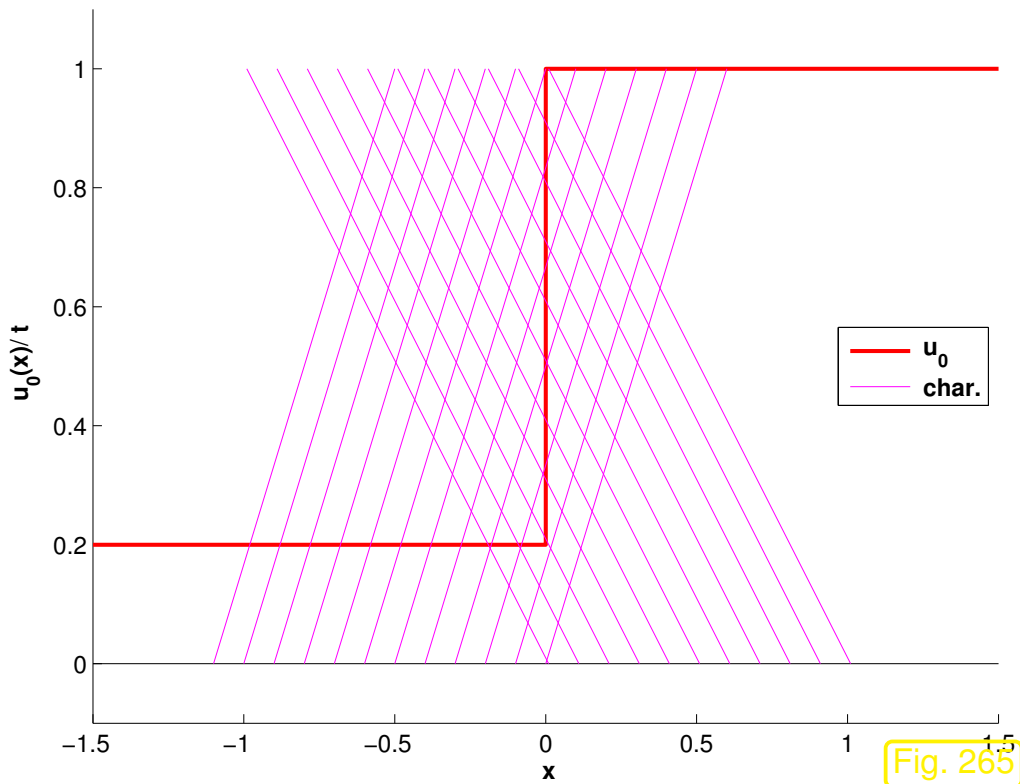
flux function $f : \mathbb{R} \mapsto \mathbb{R}$ smooth & concave



f' non-increasing

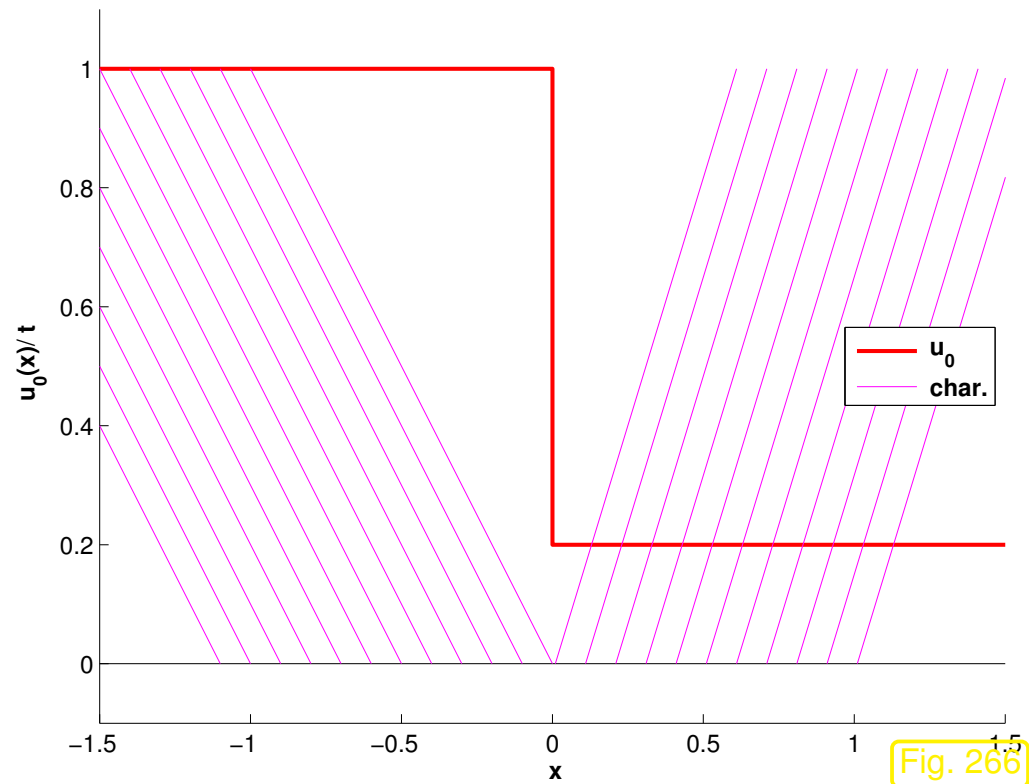


pattern of characteristic curves for Riemann problem:



intersecting characteristics

Fig. 265



diverging characteristics

Fig. 266

Definition 8.2.29 (Shock).

If Γ is a smooth curve in the (x, t) -plane and u a weak solution of (8.2.9), a discontinuity of u across Γ is called a **shock**.

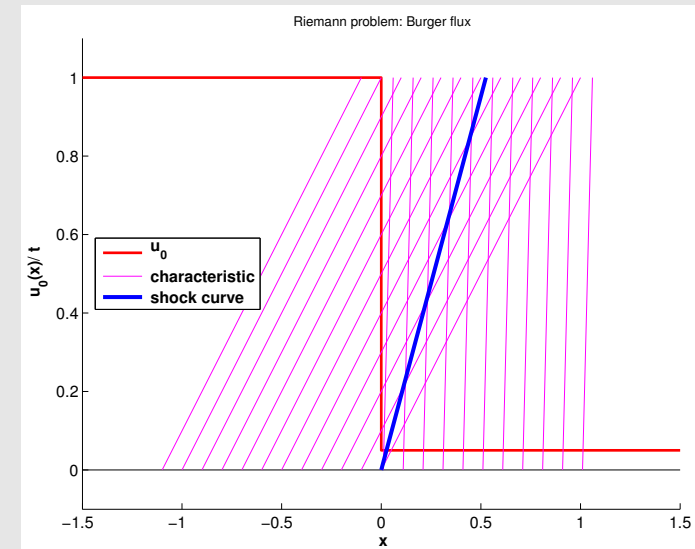
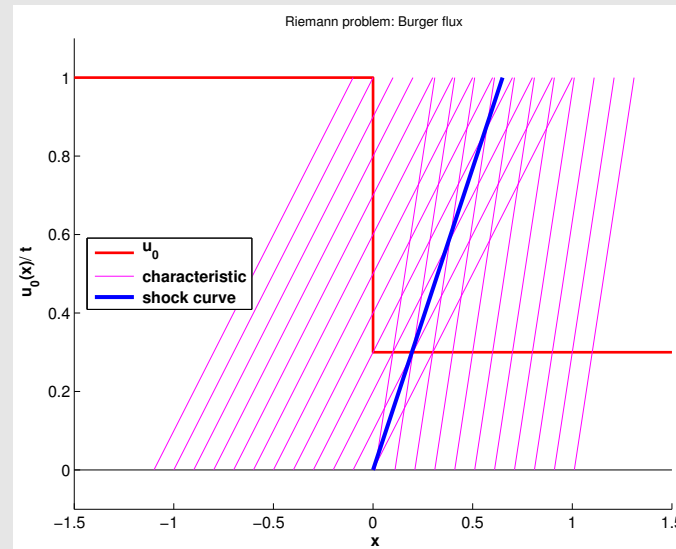
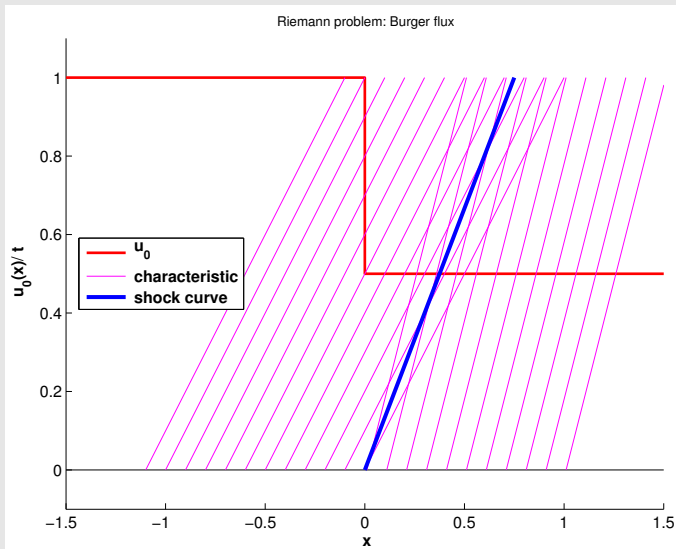
Rem. 8.2.26 ➤ **shock speed** s \leftrightarrow Rankine-Hugoniot jump conditions:

$$(x_0, t_0) \in \Gamma: \quad \dot{s} = \frac{f(u_l) - f(u_r)}{u_l - u_r}, \quad \begin{aligned} u_l &:= \lim_{\epsilon \rightarrow 0} u(x_0 - \epsilon, t_0), \\ u_r &:= \lim_{\epsilon \rightarrow 0} u(x_0 + \epsilon, t_0). \end{aligned} \quad (8.2.30)$$

Lemma 8.2.31 (Shock solution of Riemann problem).

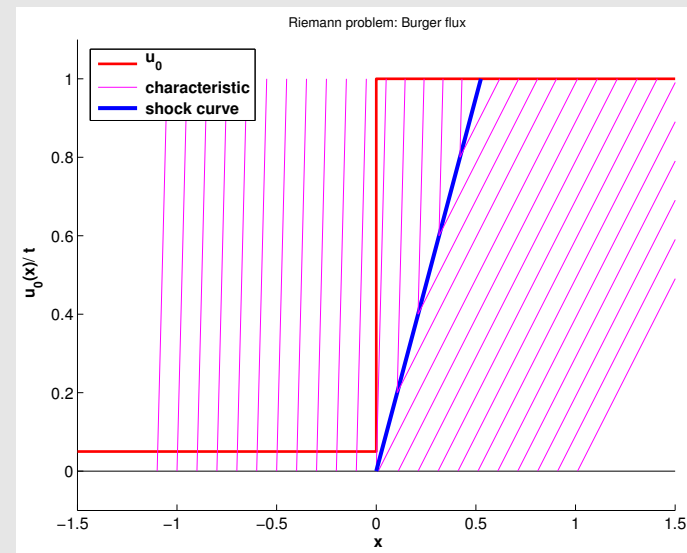
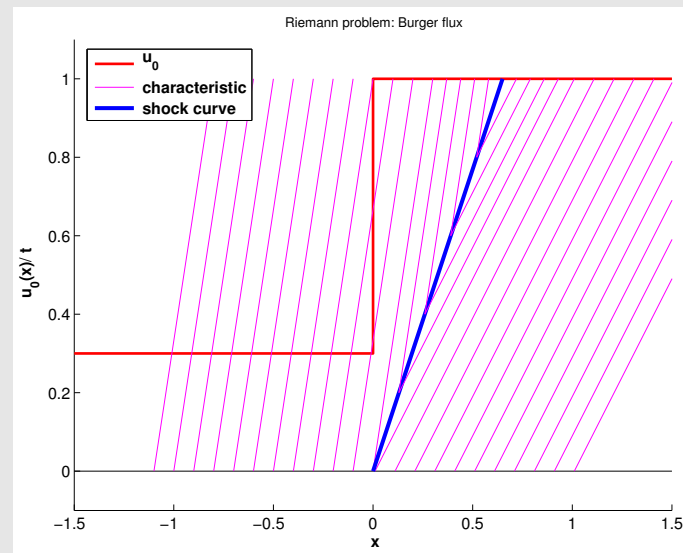
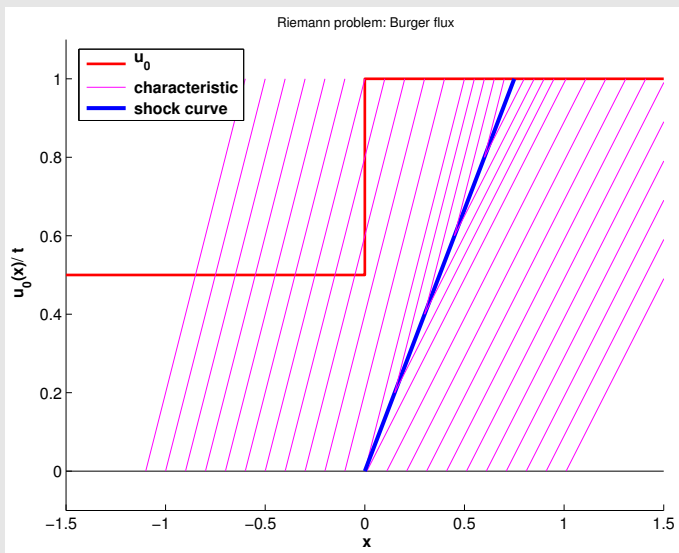
$$u(x, t) = \begin{cases} u_l & \text{for } x < \dot{s}t, \\ u_r & \text{for } x > \dot{s}t, \end{cases} \quad \dot{s} := \frac{f(u_l) - f(u_r)}{u_l - u_r}, \quad x \in \mathbb{R}, 0 < t < T,$$

is weak solution of Riemann problem (\rightarrow Def. 8.2.28) for (8.2.9).



Burgers flux $f(u) = \frac{1}{2}u^2$, $u_l > u_r$: characteristic curves impinge on shock

Fig. 267

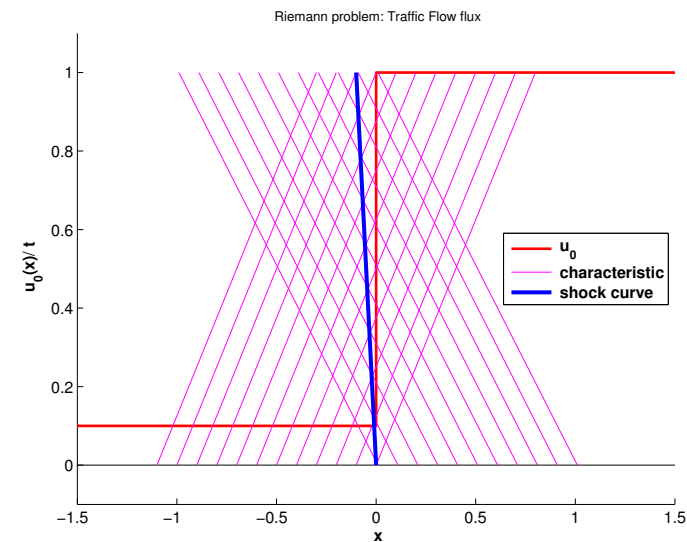
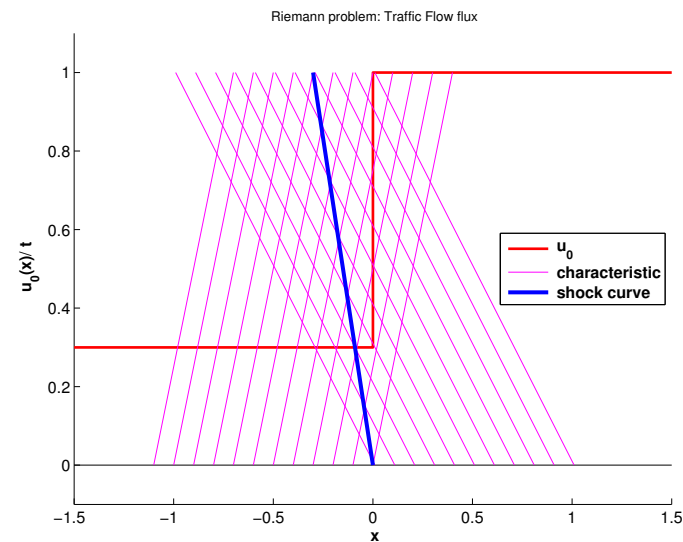
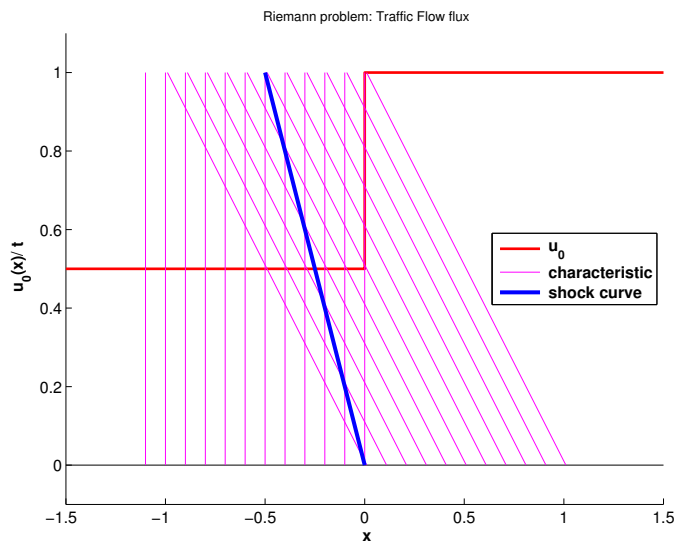


Burgers flux $f(u) = \frac{1}{2}u^2$, $u_l < u_r$: characteristic curves emanate from shock (expansion shock)

Fig. 268

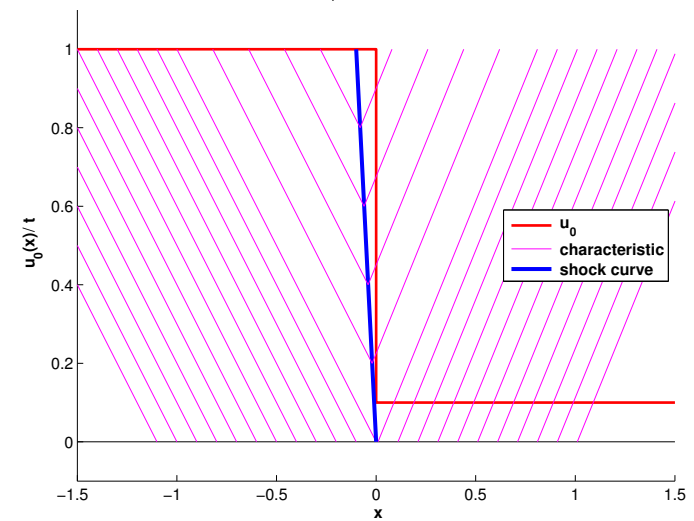
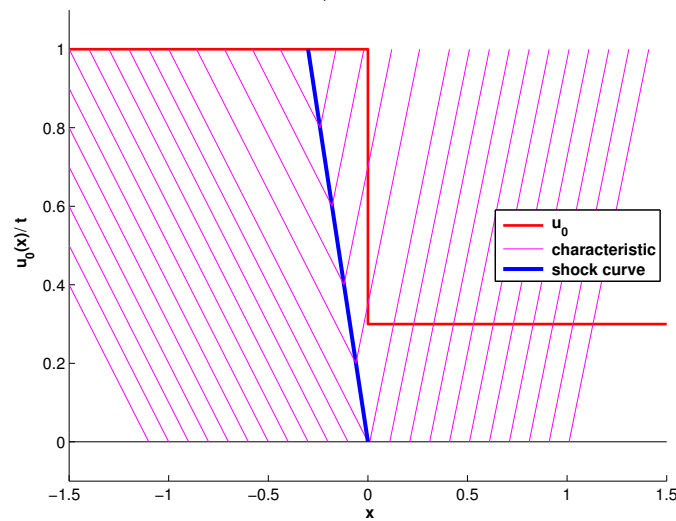
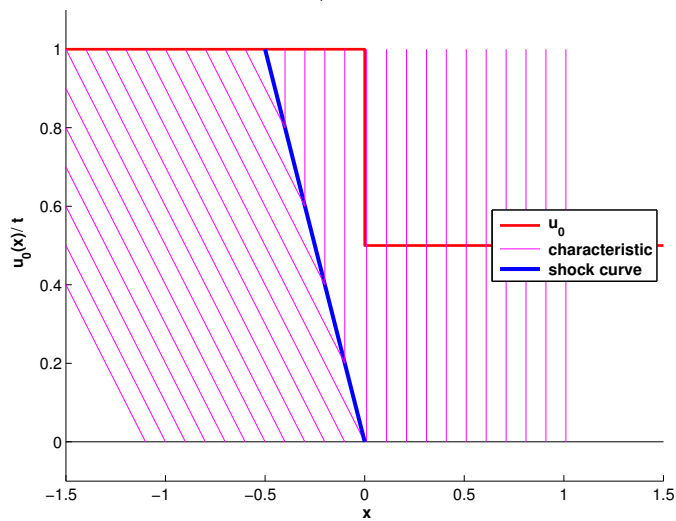
R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Traffic Flow flux $f(u) = u(1 - u)$, $u_l < u_r$: characteristic curves impinge on shock

Fig. 269



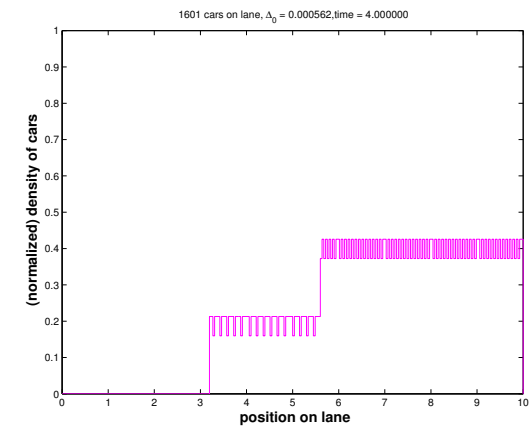
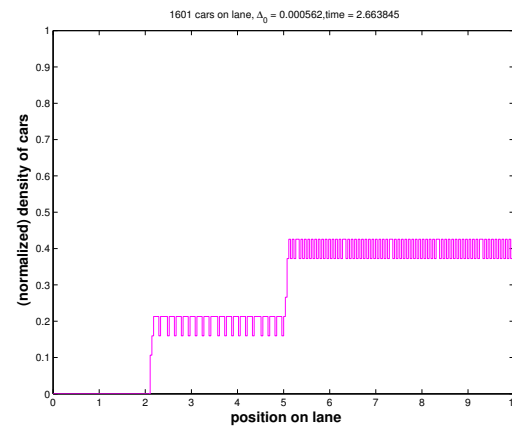
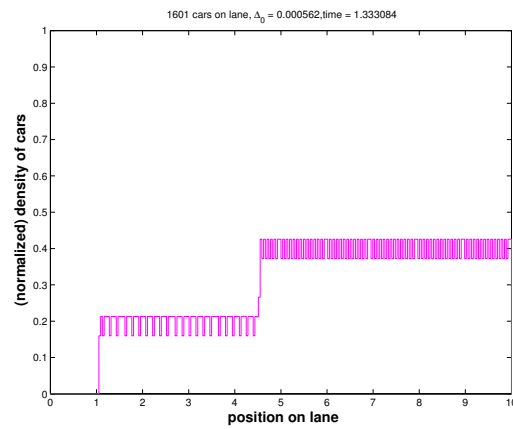
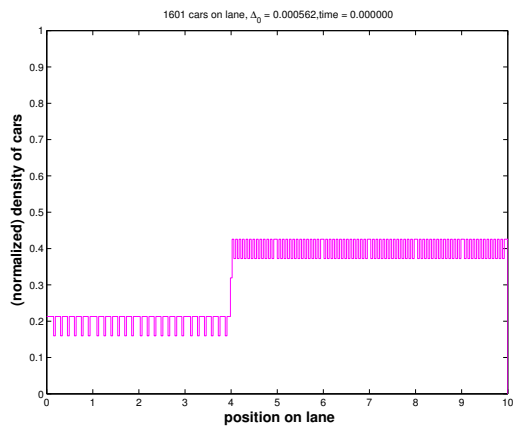
Traffic Flow flux $f(u) = u(1 - u)$, $u_l > u_r$: characteristic curves emanate from shock (expansion shock)

Fig

Example 8.2.32 (Shock patterns in traffic flow).

Simulation of microscopic particle model of traffic flow as in Ex. 8.1.36, $x_0 = [(0:0.01:4), (4.005:0.005:10)]$, $\Delta_0 = 0.002$, normalized car density by averaging.

Situation: column of fast going cars approach a zone of dense traffic.



Observation: abrupt changes of car density (= shocks) present in initial conditions persist throughout the evolution. Sites of discontinuity travel with constant speed close to the speed predicted by the jump conditions (8.2.23).

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

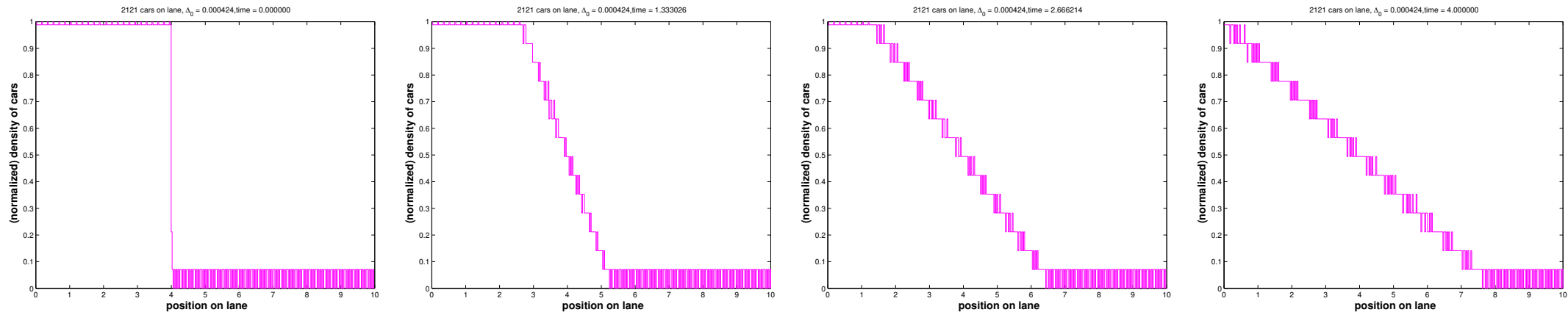
SAM, ETHZ



Example 8.2.33 (Fan patterns in traffic flow).

Simulation of microscopic particle model of traffic flow as in Ex. 8.1.36, $x_0 = [(0:0.002:4) , (4.05:0.05:10)]$, $\Delta_0 = 0.002$, normalized car density by averaging.

Situation: front end of a traffic jam



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Observation: abrupt changes of car density present in initial conditions disappear and are replaced with a zone of *linearly decreasing* car density, whose edges move with constant speed in opposite direction.

No shock solution!



Example 8.2.36 (Vanishing viscosity for Burgers equation).

Recall modeling explained in Sect. 8.1.3. There is no such material as an “inviscid” fluid in nature, because in any physical system there will be a tiny amount of friction. This leads us to the very general understanding that conservation laws can usually be regarded as limit problems $\epsilon = 0$ for singularly perturbed transport-diffusion problems with an “ ϵ -amount” of diffusion.

In 1D, for any $\epsilon > 0$ these transport-diffusion problems will possess a unique smooth solution. Studying its behavior for $\epsilon \rightarrow 0$ will tell us, what are “physically meaningful” solutions for the conservation law. This consideration is called the **vanishing viscosity** method to define solutions for conservation laws.

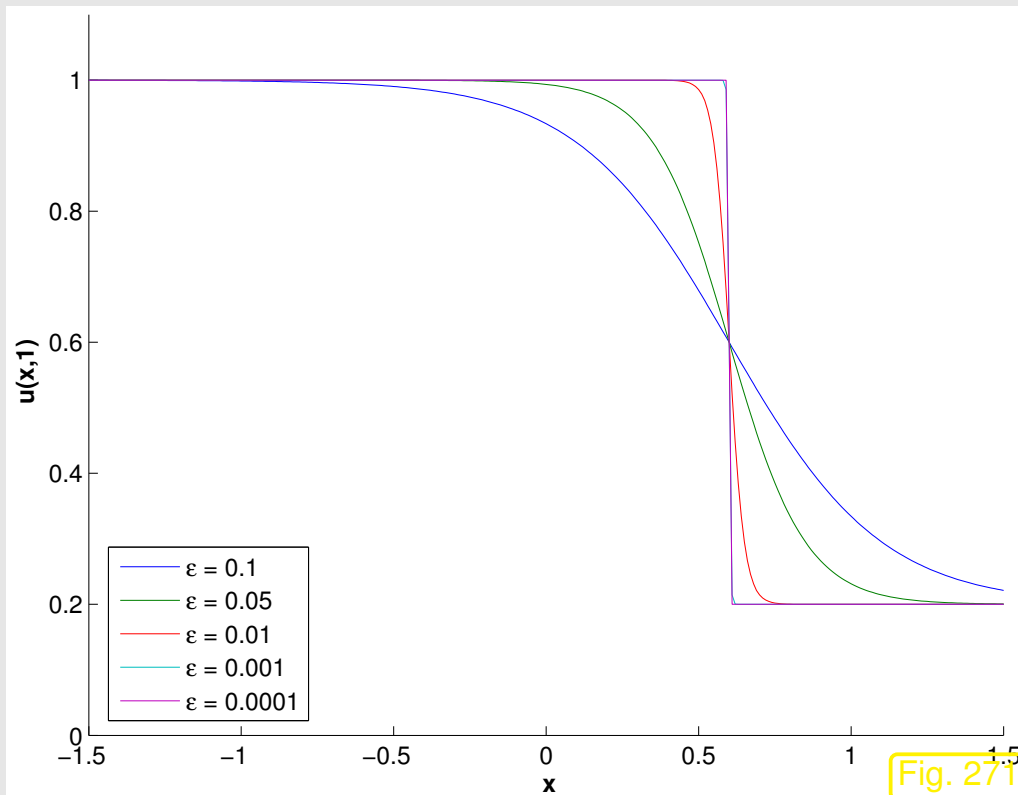
Here we pursue this idea for Burgers equation, see Sect. 8.1.3.

Viscous Burgers equation:
$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{1}{2} u^2 \right) = \epsilon \frac{\partial^2 u}{\partial x^2} . \quad (8.2.37)$$

dissipative (viscous) term

Travelling wave solution of Riemann problem for (8.2.37) via Cole-Hopf transform \rightarrow [15, Sect. 4.4.1]

$$u_\epsilon(x, t) = w(x - \dot{s}t) \quad , \quad w(\xi) = u_r + \frac{1}{2}(u_l - u_r) \left(1 - \tanh \left(\frac{\xi(u_l - u_r)}{4\epsilon} \right) \right) \quad , \quad \dot{s} = \frac{1}{2}(u_l + u_r) .$$



$u_\epsilon(x, t)$ = classical solution of (8.2.37) for all $t > 0$,
 $x \in \mathbb{R}$ (only for $u_l > u_r$!).

◁

$$u_l > u_r, \quad t = 0.5$$

emerging shock for $\epsilon \rightarrow 0$

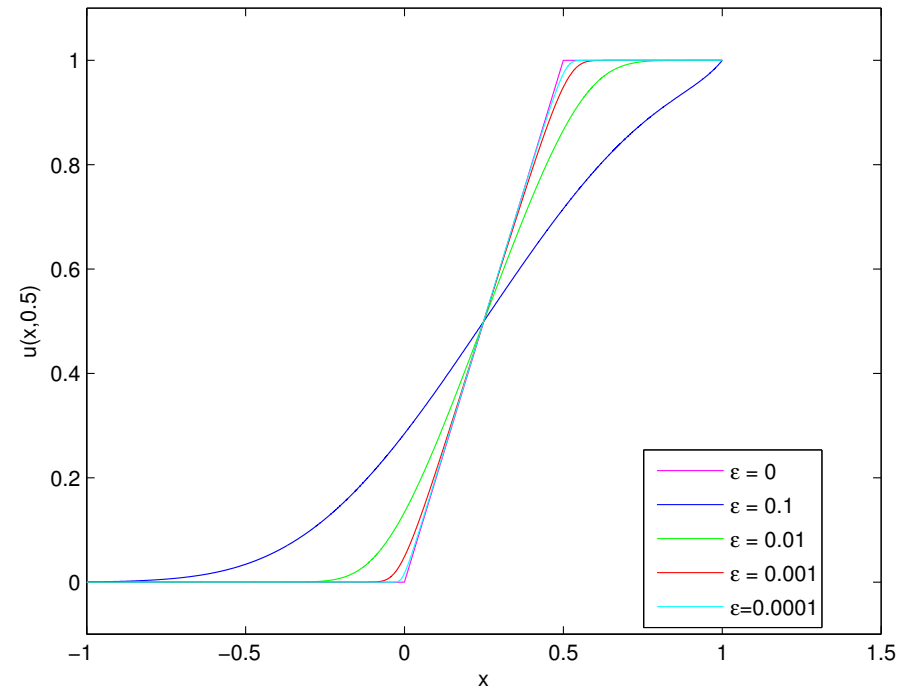
$u_\epsilon \rightarrow u$ from Lemma 8.2.31 in $L^\infty(\mathbb{R})$.

Highly accurate numerical solution of
Riemann problem for (8.2.37)

$$u_l < u_r \quad u_\epsilon(x, 0.5) \triangleright$$

no shock as $\epsilon \rightarrow 0$!

$u_\epsilon \rightarrow$ a piecewise linear function!



Let us try to derive a (weak) solution of the homogeneous scalar conservation law (8.2.16) with the structure observed in Ex. 8.2.36.

Idea: conservation law (8.2.16) homogeneous in spatial/temporal derivatives:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times \mathbb{R}^+ \quad \Rightarrow \quad \frac{\partial u_\lambda}{\partial t} + \frac{\partial}{\partial x} f(u_\lambda) = 0 \quad \text{in } \mathbb{R} \times \mathbb{R}^+,$$

$u_\lambda(x, t) := u(\lambda x, \lambda t)$, $\lambda > 0$. This suggests that we look for solutions of the Riemann problem that are constant on all straight lines in the $x - t$ -plane that cross $(0, 0)^T$.

▶ try **similarity solution**:

$$u(x, t) = \psi(x/t)$$

▼ ← insert in $\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0$

$$f'(\psi(x/t))\psi'(x/t) = (x/t)\psi'(x/t) \quad \forall x \in \mathbb{R}, 0 < t < T.$$

$$\psi' \equiv 0 \quad \vee \quad f'(\psi(w)) = w \iff \psi(w) = (f')^{-1}(w).$$

f' strictly monotone !

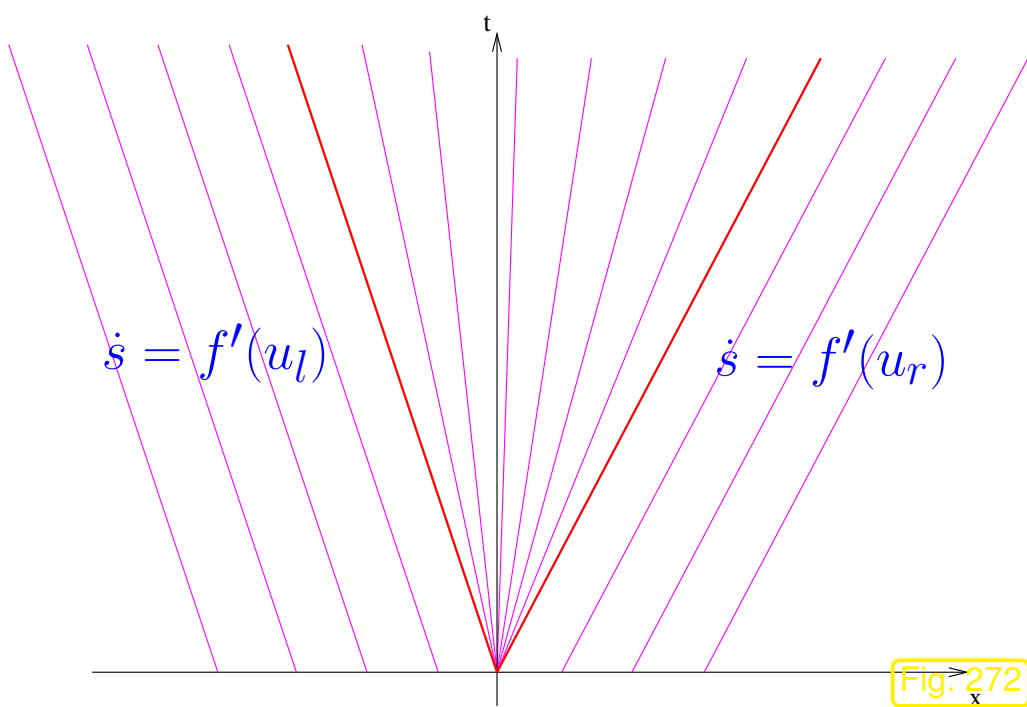


Fig. 272

$f(u)$ strictly **convex**, $u_l < u_r$

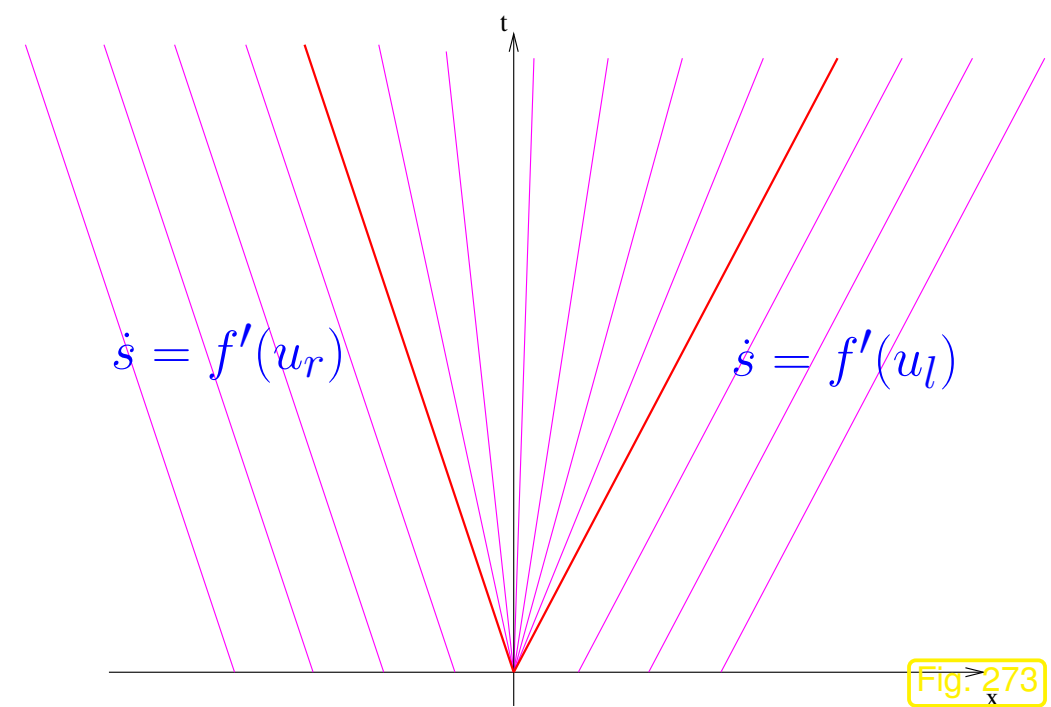


Fig. 273

$f(u)$ strictly **concave**, $u_r < u_l$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Lemma 8.2.38 (Rarefaction solution of Riemann problem).

If $f \in C^2(\mathbb{R})$ is strictly $\begin{cases} \text{convex and } u_l < u_r, \\ \text{concave and } u_r < u_l, \end{cases}$ then

$$u(x, t) := \begin{cases} u_l & \text{for } x < \min\{f'(u_l), f'(u_r)\} \cdot t, \\ g\left(\frac{x}{t}\right) & \text{for } \min\{f'(u_l), f'(u_r)\} < \frac{x}{t} < \max\{f'(u_l), f'(u_r)\}, \\ u_r & \text{for } x > \max\{f'(u_l), f'(u_r)\} \cdot t, \end{cases}$$

$g := (f')^{-1}$, is a weak solution of the Riemann problem (\rightarrow Def. 8.2.28).

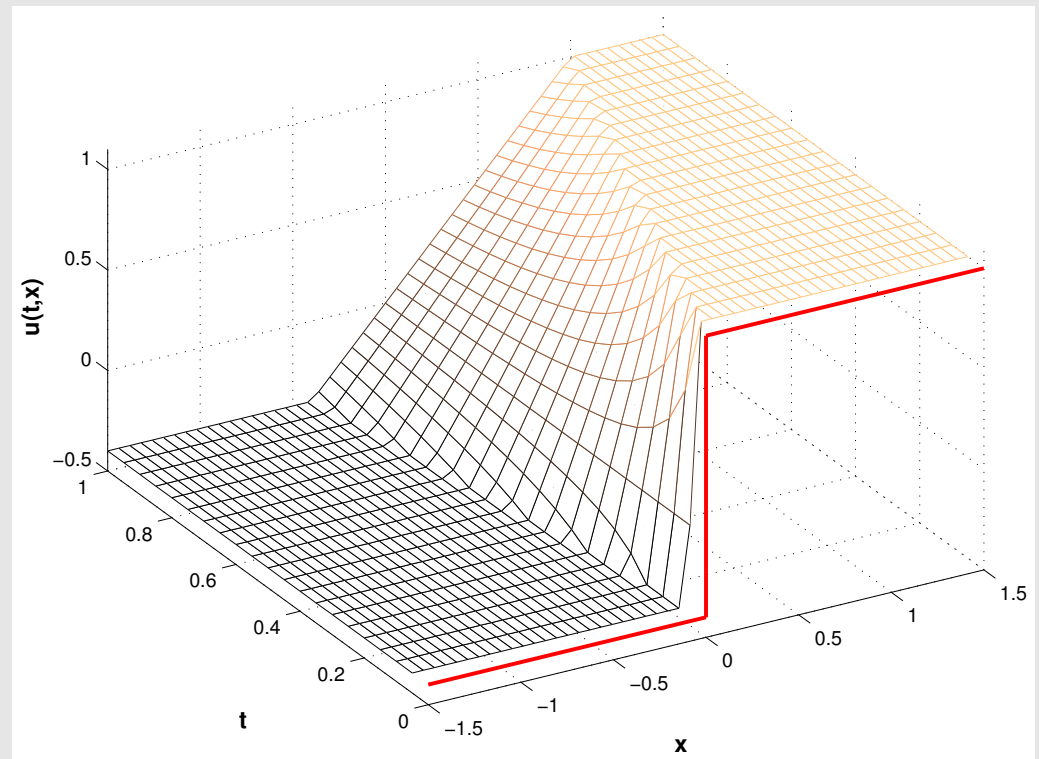
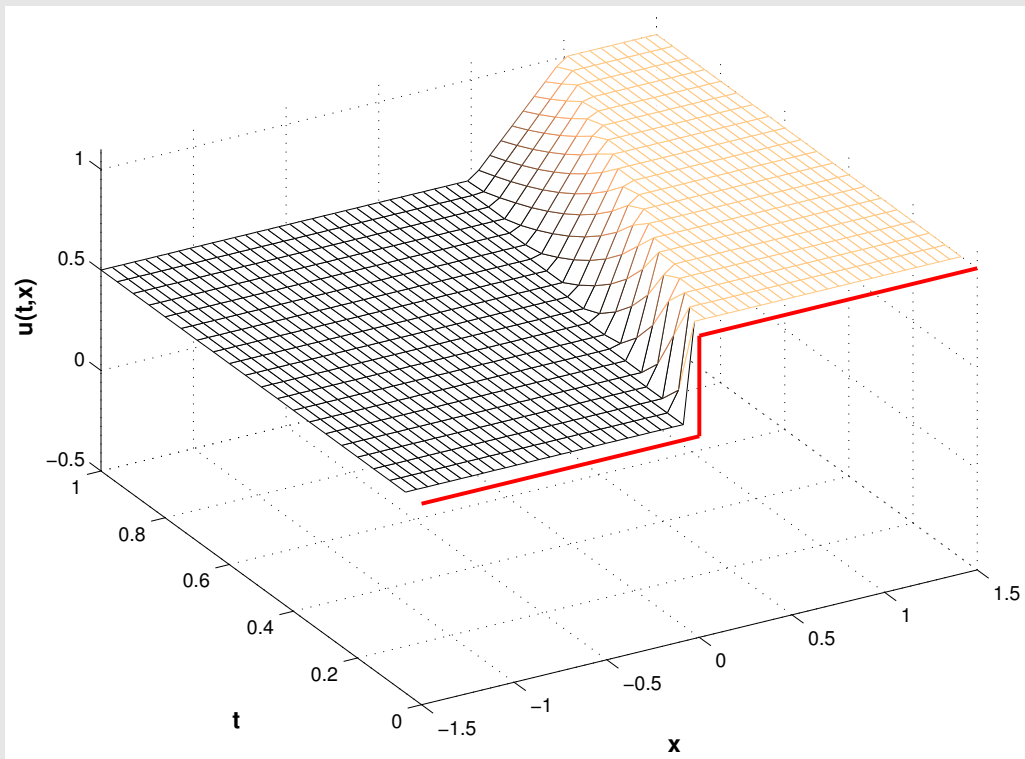
Proof. We show that the rarefaction solution is a weak solution according to Def. 8.2.19 \triangleright for $\Phi \in C_0^\infty(\mathbb{R} \times]0, T[)$

$$\int_0^T \left\{ \int_{-\infty}^{f'(u_l)t} u_l \frac{\partial \Phi}{\partial t} + f(u_l) \frac{\partial \Phi}{\partial x} dx + \int_{f'(u_l)t}^{f'(u_r)t} g\left(\frac{x}{t}\right) \frac{\partial \Phi}{\partial t} + f\left(g\left(\frac{x}{t}\right)\right) \frac{\partial \Phi}{\partial x} dx + \int_{f'(u_r)t}^{\infty} u_r \frac{\partial \Phi}{\partial t} + F(u_r) \frac{\partial \Phi}{\partial x} dx \right\} dt$$

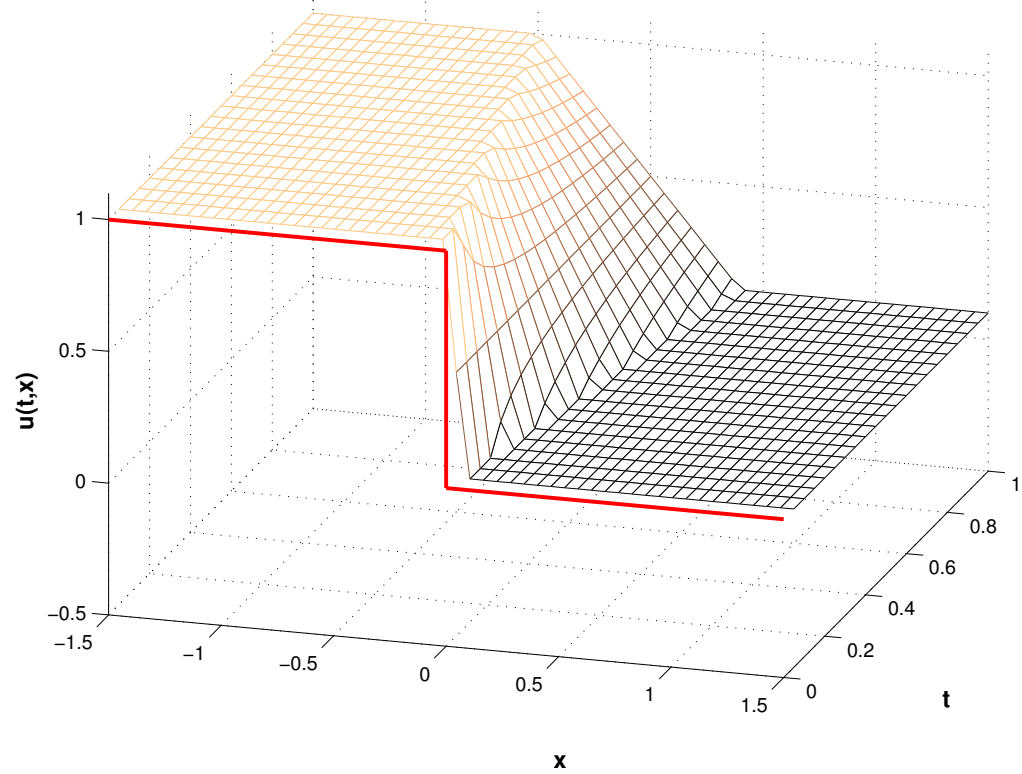
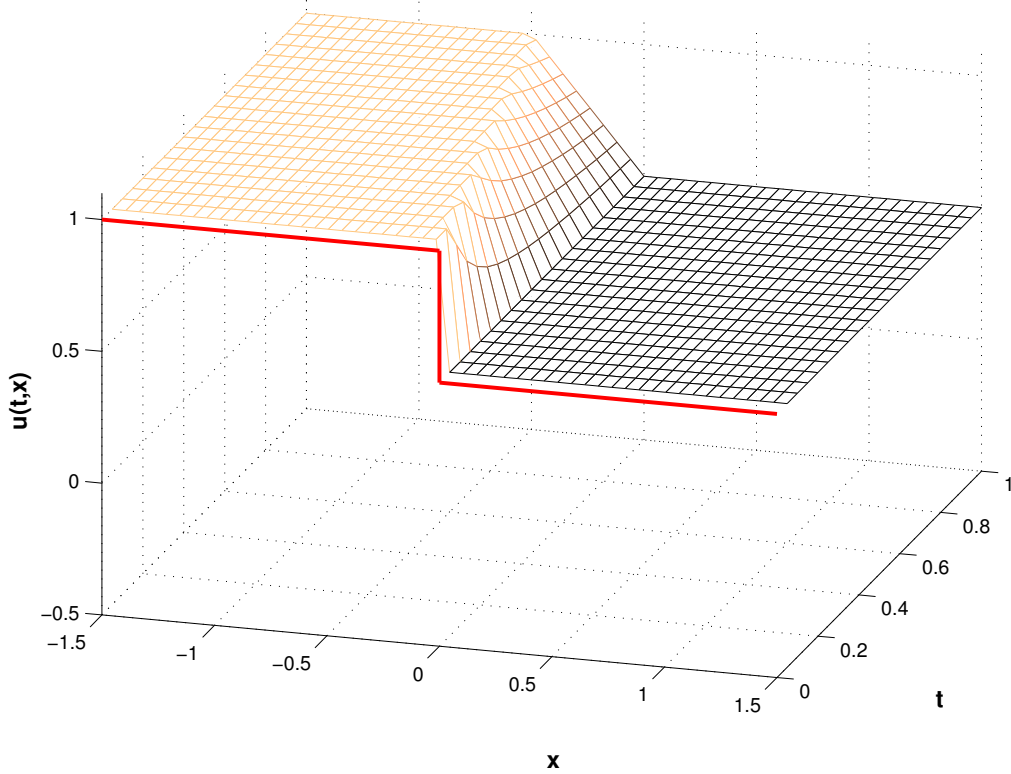
$$= \int_0^T \int_{f'(u_l)t}^{f'(u_r)t} g'\left(\frac{x}{t}\right) \frac{x}{t^2} \Phi - f'\left(g\left(\frac{x}{t}\right)\right) \frac{1}{t} g'\left(\frac{x}{t}\right) \Phi dx dt = 0,$$

because $(f' \circ g)(x/t) = x/t$ and by fundamental theorem of calculus. \square

Terminology: solution of Lemma 8.2.38 = **rarefaction wave**: *continuous solution* !



Burger flux function $f(u) = \frac{1}{2}u^2$, $u_l < u_r$: rarefaction wave solutions

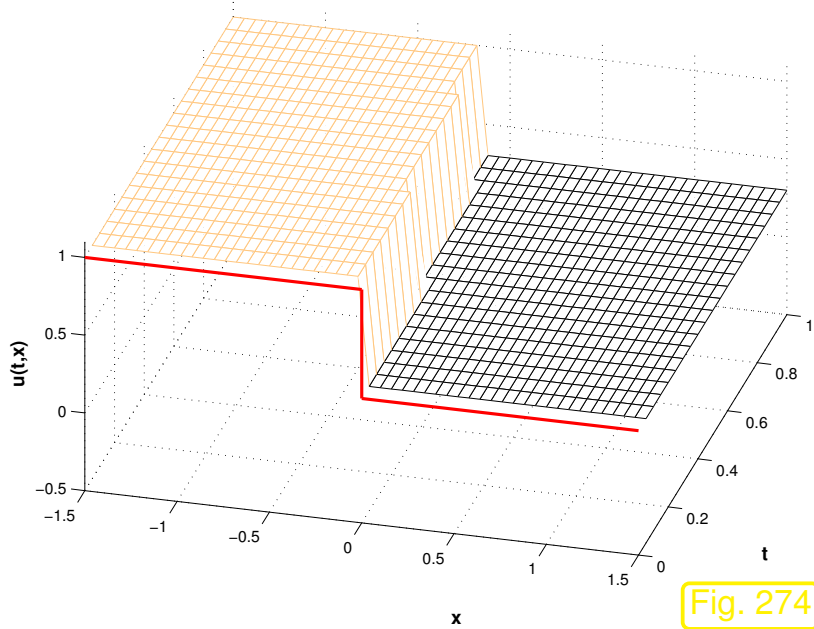


Traffic flow flux function $f(u) = \frac{1}{2}u(1 - u)$, $u_l > u_r$: rarefaction wave solutions

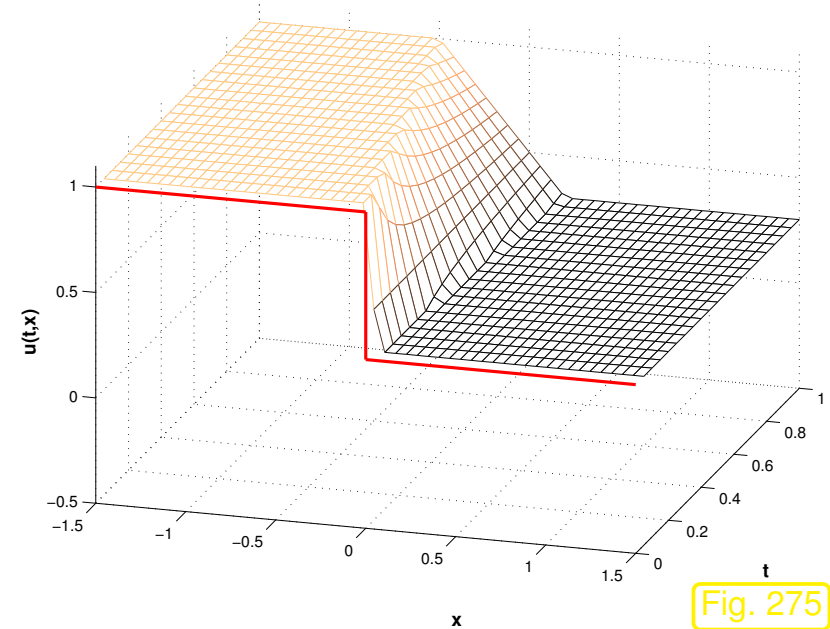
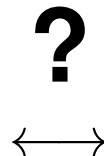
8.2.6 Entropy condition

Sect 8.2.5 ➤ *Non-uniqueness* of weak solutions:

if f' is decreasing as in the traffic flow equation (8.1.53) and $u_l > u_r$ both a shock and a rarefaction wave provide valid weak solutions.



Riemann solution: shock



Riemann solution: rarefaction wave

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

How to select “physically meaningful” = admissible solution ?

❶ Comparison with results from microscopic models, see Ex. 8.2.33 for the case of traffic flow.

② **Vanishing viscosity technique** (→ Ex. 8.2.36 for Burgers' equation): add an “ ϵ -amount” of diffusion (“friction”) and study solution for $\epsilon \rightarrow 0$.

However, desirable: simple selection criteria (**entropy conditions**)

Definition 8.2.39 (Lax entropy condition).

$u \hat{=}$ weak solution of (8.2.9), piecewise classical solution in a neighborhood of C^2 -curve $\Gamma := (\gamma(\tau), \tau), 0 \leq \tau \leq T$, discontinuous across Γ .

u satisfies the **Lax entropy condition** in $(x_0, t_0) \in \Gamma \iff f'(u_l) > \dot{s} := \frac{f(u_l) - f(u_r)}{u_l - u_r} > f'(u_r)$.



Characteristic curves must not emanate from shock \leftrightarrow no “generation of information”

Parlance: shock satisfying Lax entropy condition = **physical shock**

Note: f' increasing \blacktriangleright by Def 8.2.39 necessary for physical shock

$u_l > u_r$
$u_l < u_r$

Physically meaningful weak solution of conservation law = **entropy solution**

For *scalar* conservation laws with locally Lipschitz-continuous flux function f :

Existence & uniqueness of entropy solutions

Remark 8.2.40 (General entropy solution for 1D scalar Riemann problem). \rightarrow [26]

Entropy solution of Riemann problem (\rightarrow Def. 8.2.28) for (8.2.9) with arbitrary $f \in C^1(\mathbb{R})$:

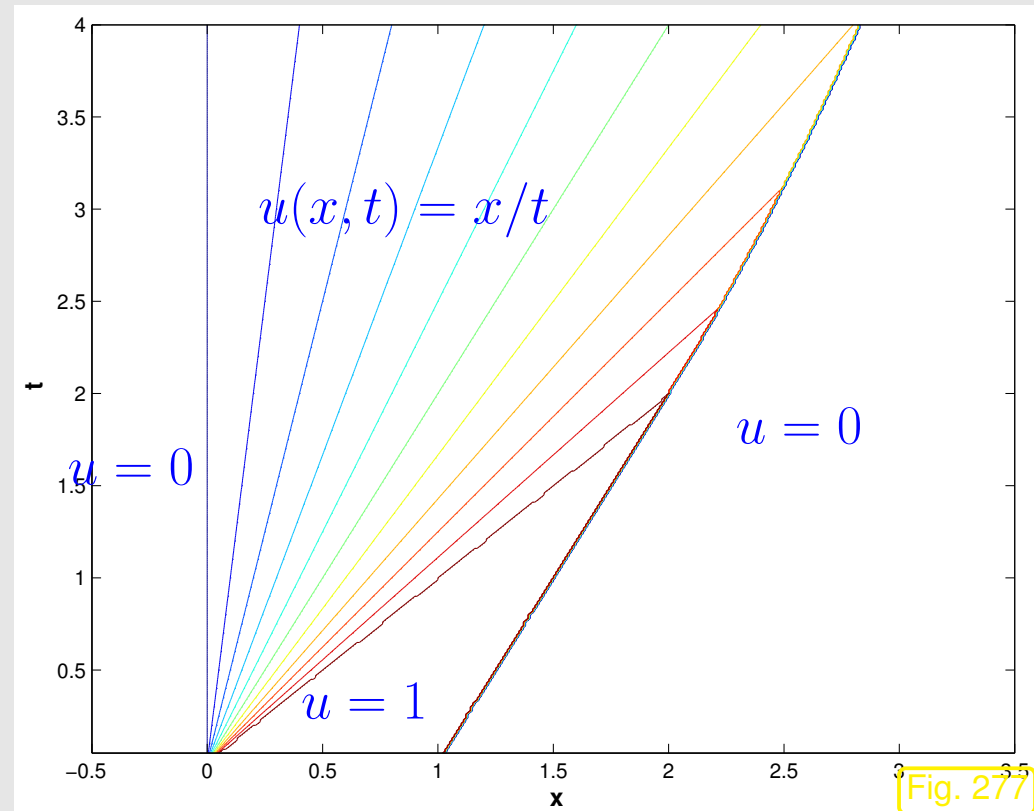
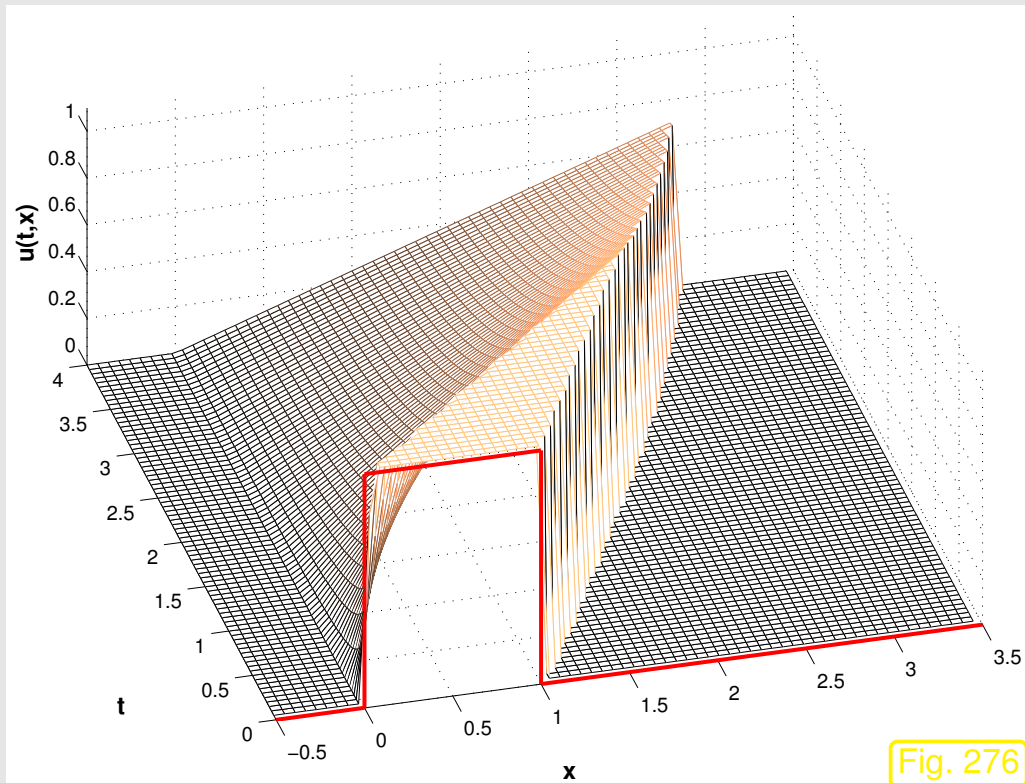
$$u(x, t) = \psi(x/t) \quad , \quad \psi(\xi) := \begin{cases} \operatorname{argmin}_{u_l \leq u \leq u_r} (f(u) - \xi u) & , \text{ if } u_l < u_r \text{ ,} \\ \operatorname{argmax}_{u_r \leq u \leq u_l} (f(u) - \xi u) & , \text{ if } u_l \geq u_r \text{ .} \end{cases} \quad (8.2.41)$$



Example 8.2.43 (Entropy solution of Burgers equation).

Analytical solution available for Burgers equation (8.1.60) with initial data, see [15, Sect. 3.4, Ex. 3]

$$u_0(x) = \begin{cases} 0 & , \text{if } x < 0 \text{ or } x > 1 , \\ 1 & , \text{if } 0 \leq x \leq 1 . \end{cases}$$



Vector field in $x - t$ -plane

$$\begin{pmatrix} f(u(x, t)) \\ u(x, t) \end{pmatrix}$$

for entropy solution $u = u(x, t)$ \triangleright . Observe
the normal continuity across the shock: the vector
field is tangential to the shock curve.

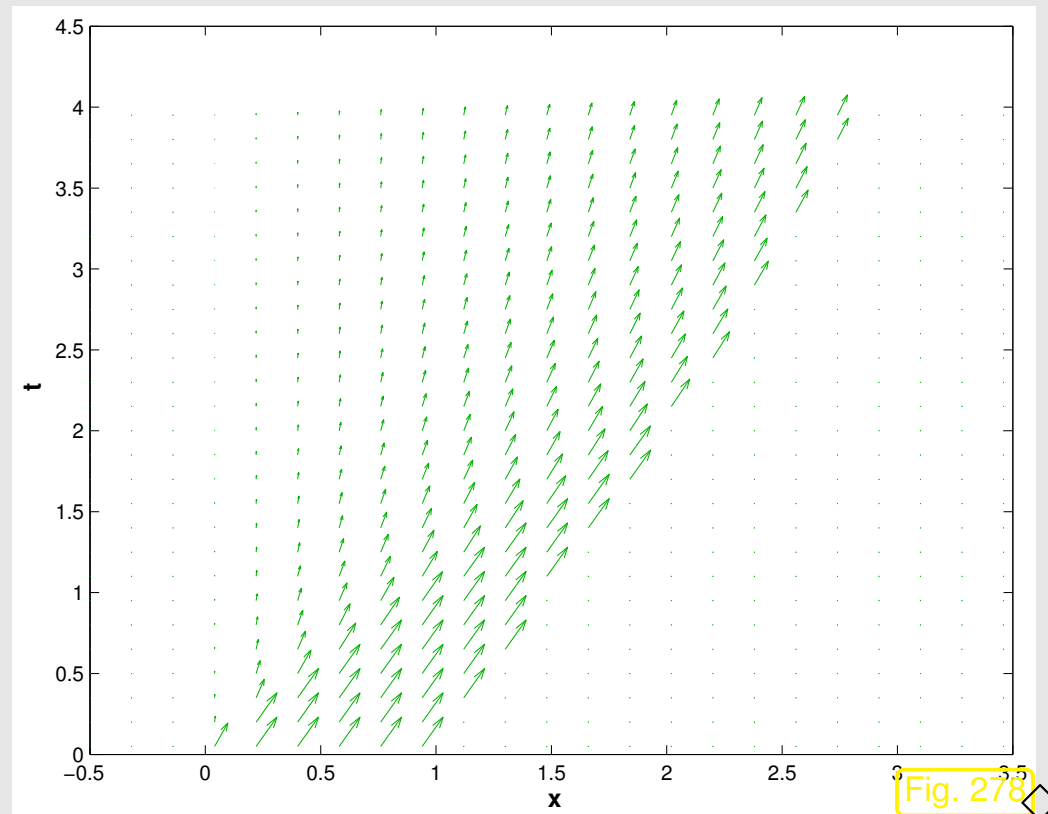


Fig. 278

Example 8.2.44 (Entropy solution of Traffic Flow equation).

Analytical solution available for Traffic Flow equation (8.1.53) with initial data, see [15, Sect. 3.4, Ex. 3]

$$u_0(x) = \begin{cases} 0.5 & , \text{ if } x < 0 \text{ or } x > 1 , \\ 1 & , \text{ if } 0 \leq x \leq 1 . \end{cases}$$

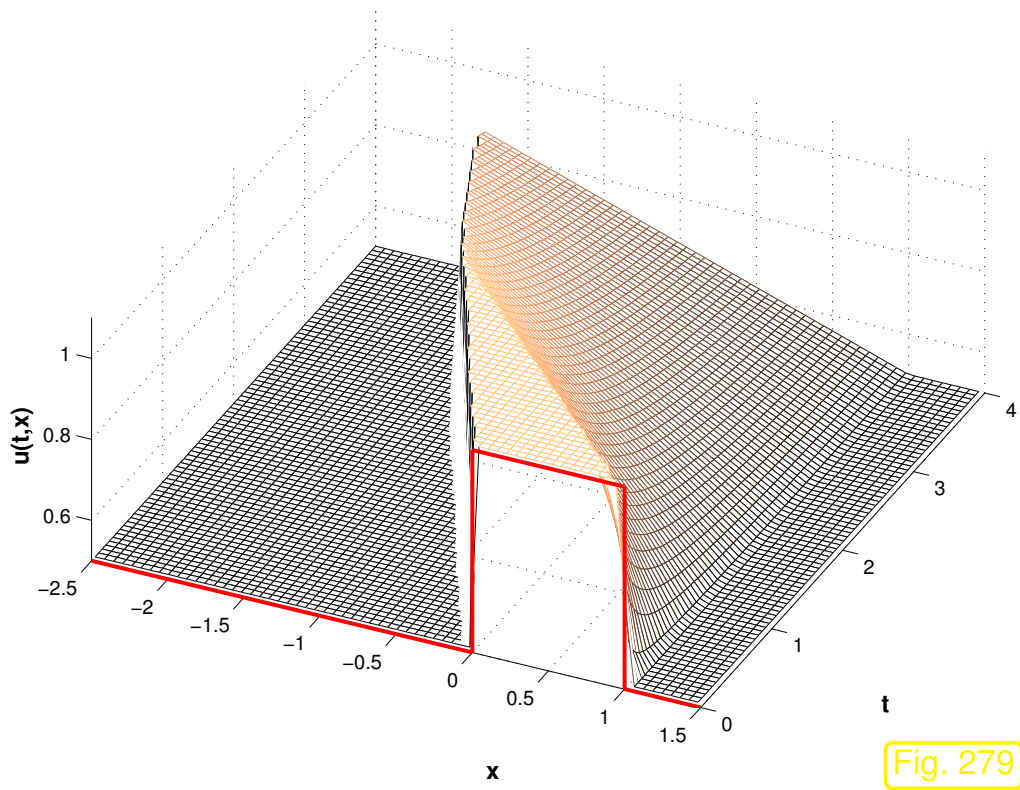


Fig. 279

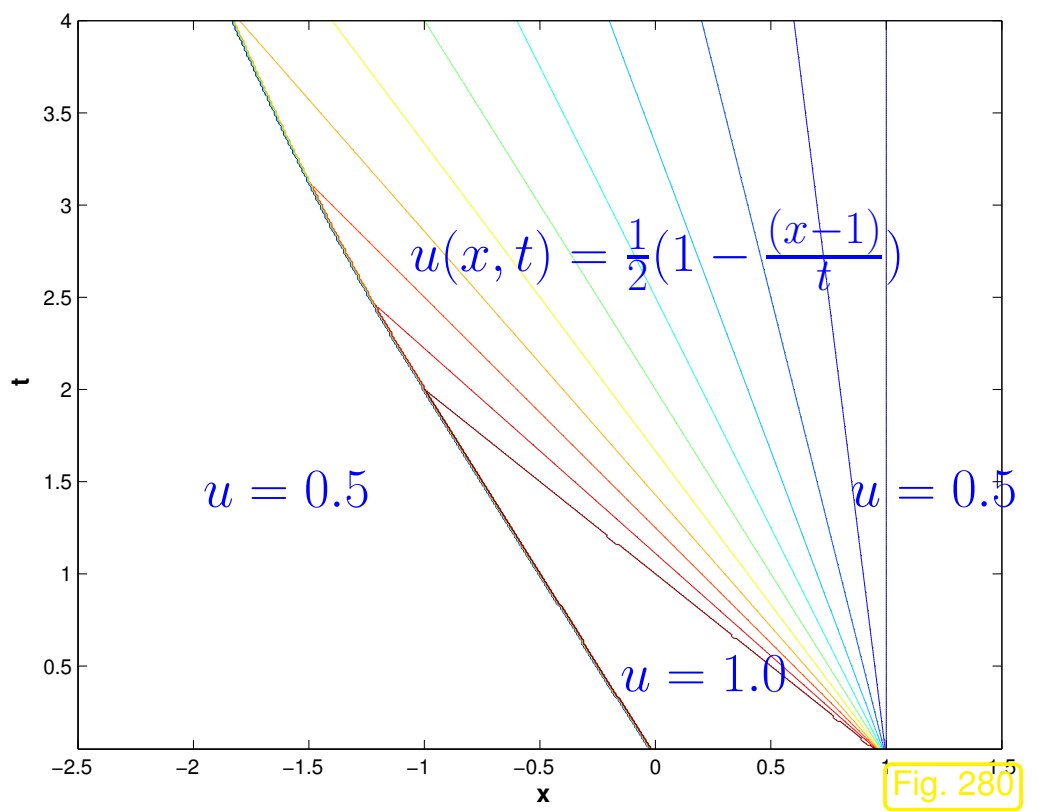


Fig. 280

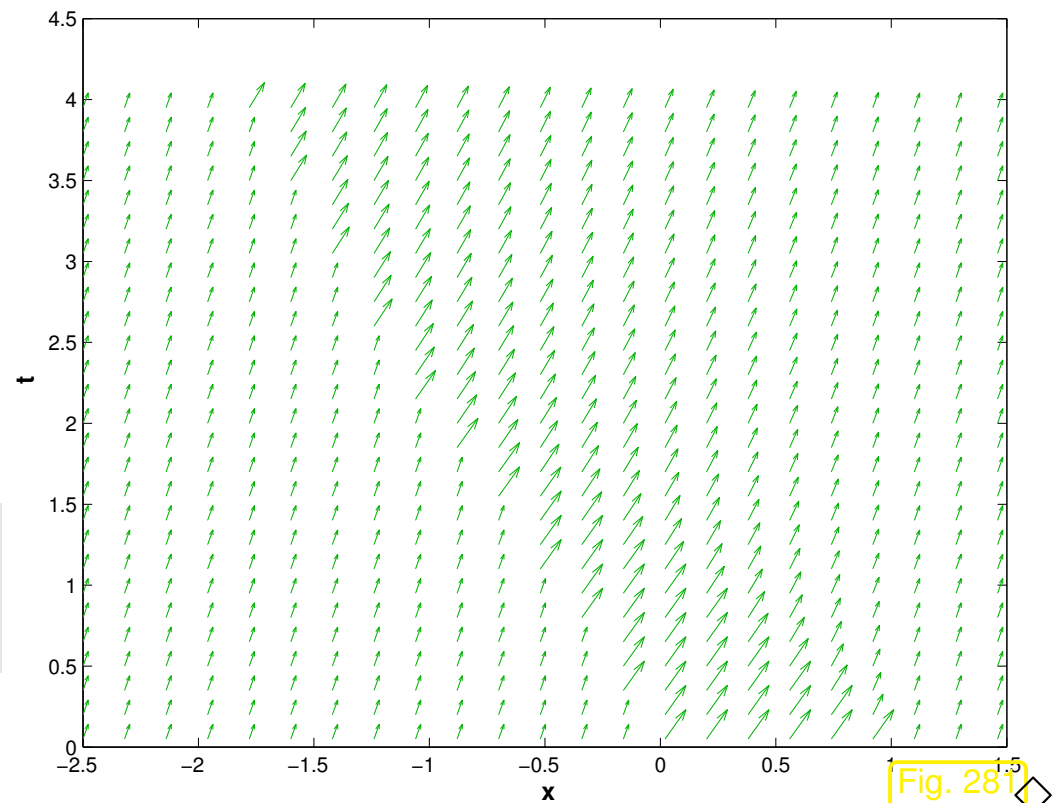
Vector field in $x - t$ -plane

$$\begin{pmatrix} f(u(x, t)) \\ u(x, t) \end{pmatrix}$$

for entropy solution $u = u(x, t)$

▷.

Observe the normal continuity across the shock:
the vector field is tangential to the shock curve.



8.2.7 Properties of entropy solutions

Setting: $u \in L^\infty(\mathbb{R} \times]0, T[)$ weak (\rightarrow Def. 8.2.19) entropy solution of Cauchy problem

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times]0, T[\quad , \quad u(x, 0) = u_0(x) \quad , \quad x \in \mathbb{R} . \quad (8.2.9)$$

with flux function $f \in C^1(\mathbb{R})$ (not necessarily convex/concave).

Notation: $\bar{u} \in L^\infty(\mathbb{R} \times]0, T[) \hat{=}$ entropy solution w.r.t. initial data $\bar{u}_0 \in L^\infty(\mathbb{R})$.

Theorem 8.2.45 (**Comparison principle** for scalar conservation laws).

$$\text{If } u_0 \leq \bar{u}_0 \text{ a.e. on } \mathbb{R} \quad \Rightarrow \quad u \leq \bar{u} \text{ a.e. on } \mathbb{R} \times]0, T[$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

With obvious consequences:

$$\blacktriangleright \quad u_0(x) \in [\alpha, \beta] \text{ on } \mathbb{R} \quad \Rightarrow \quad u(x, t) \in [\alpha, \beta] \text{ on } \mathbb{R} \times]0, T[$$

Note: this guarantees the normalization condition $0 \leq u(x, t) \leq 1$ for the traffic flow model, if it is satisfied for the initial data u_0 .

► L^∞ -stability (► no blow-up can occur!)

$$\forall 0 \leq t \leq T: \|u(\cdot, t)\|_{L^\infty(\mathbb{R})} \leq \|u_0\|_{L^\infty(\mathbb{R})} . \quad (8.2.46)$$

Theorem 8.2.47 (L^1 -contractivity of evolution for scalar conservation law).

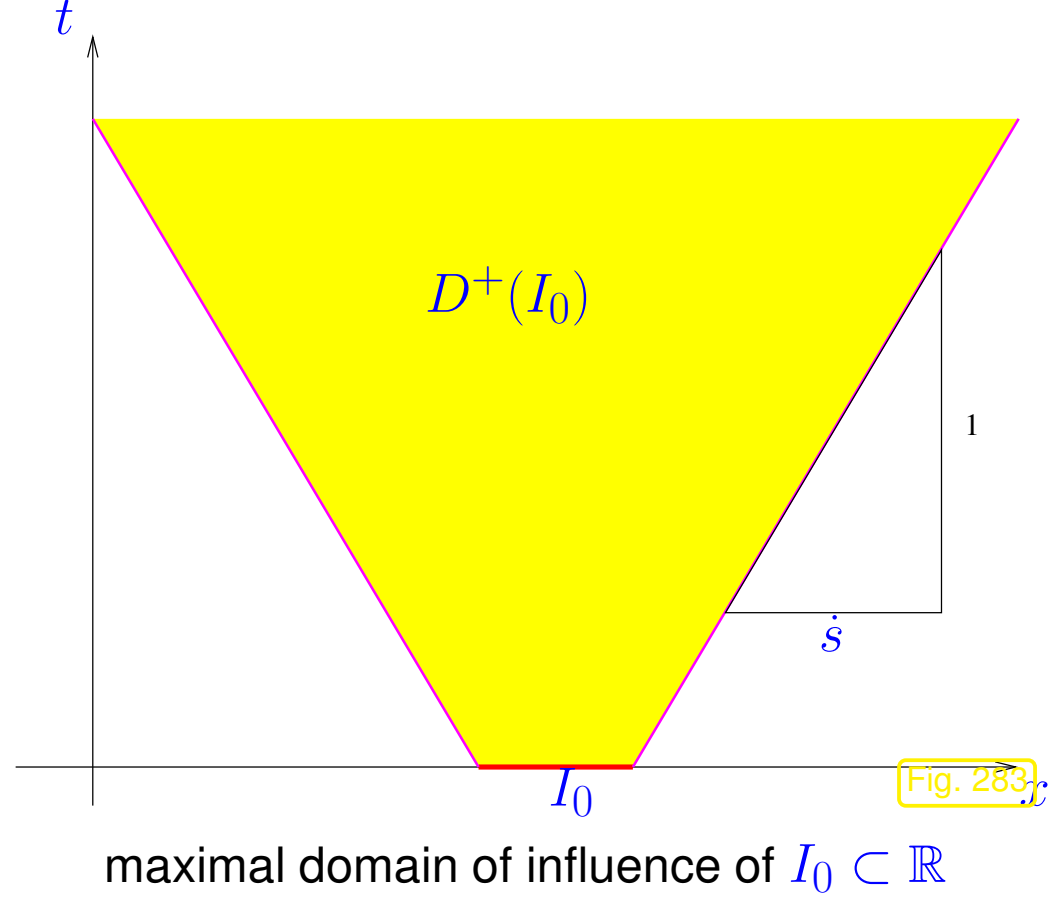
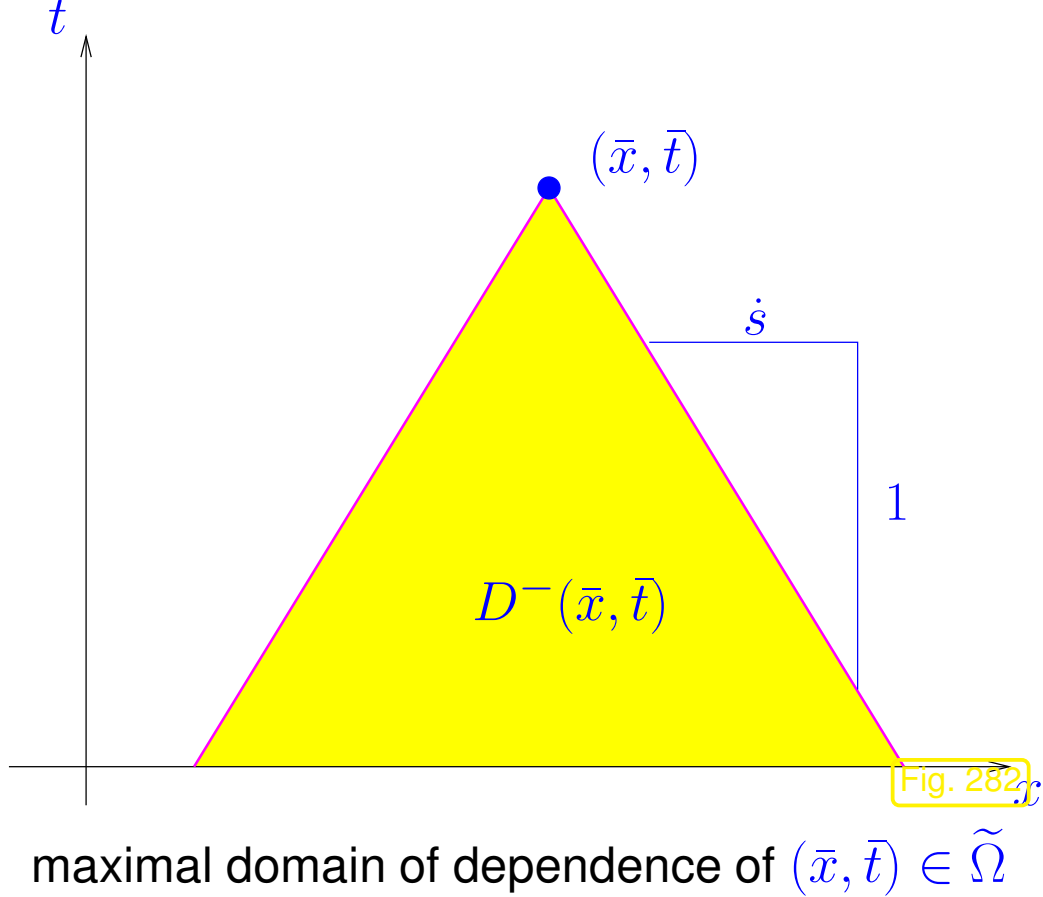
$$\forall t \in]0, T[, R > 0: \int_{|x| < R} |u(x, t)| dx \leq \int_{|x| < R+st} |u_0(x)| dx ,$$

with *maximal speed of propagation*

$$\dot{s} := \max\{|f'(\xi)|: \inf_{x \in \mathbb{R}} u_0(x) \leq \xi \leq \sup_{x \in \mathbb{R}} u_0(x)\} . \quad (8.2.48)$$

Thm. 8.2.47 ► *finite speed of propagation* in conservation law, bounded by \dot{s} from (8.2.48):

► As in the case of the wave equation → Sect. 6.2.2:



Analoguous to Thm. 6.2.18:

Corollary 8.2.49 (Domain of dependence for scalar conservation law). $\rightarrow [12, \text{Cor. 6.2.2}]$

The value of the entropy solution at $(\bar{x}, \bar{t}) \in \tilde{\Omega}$ depends only on the restriction of the initial data to $\{x \in \mathbb{R}: |x - \bar{x}| < s\bar{t}\}$.

Another strand of theoretical results asserts that the solution of a 1D scalar conservation law cannot develop oscillations:

u solves (8.2.9) ➤ No. of local extrema (in space) of $u(\cdot, t)$ decreasing with time

8.3 Conservative finite volume discretization

Example 8.3.1 (Naive finite difference scheme).

Cauchy problem for Burgers equation (8.1.60) rewritten using product rule:

$$\frac{\partial u}{\partial t}(x, t) + u(x, t) \frac{\partial u}{\partial x}(x, t) = 0 \quad \text{in } \mathbb{R} \times]0, T[.$$

↔ related to advection with velocity $v(x, t) = u(x, t)$:

$$\frac{\partial u}{\partial t}(x, t) + u(x, t) \frac{\partial u}{\partial x}(x, t) = 0 \quad \text{in } \mathbb{R} \times]0, T[.$$

$$\updownarrow$$

$$\updownarrow$$

$$\frac{\partial u}{\partial t}(x, t) + v(x, t) \frac{\partial u}{\partial x}(x, t) = 0 \quad \text{in } \mathbb{R} \times]0, T[.$$

If $u_0(x) \geq 0$, then, by Thm. 8.2.45, $u(x, t) \geq 0$ for all $0 < t < T$, that is, positive direction of transport throughout.

Heeding guideline from Sect. 7.3.1: use **upwind discretization** (backward differences) in space!

► on (infinite) equidistant grid, meshwidth $h > 0$, $x_j = hj$, $j \in \mathbb{Z}$, obtain semi-discrete problem for nodal values $\mu_j = \mu_j(t) \approx u(x_j, t)$

$$\frac{\partial u}{\partial t}(x, t) + u(x, t) \frac{\partial u}{\partial x}(x, t) = 0 \quad \text{in } \mathbb{R} \times]0, T[.$$

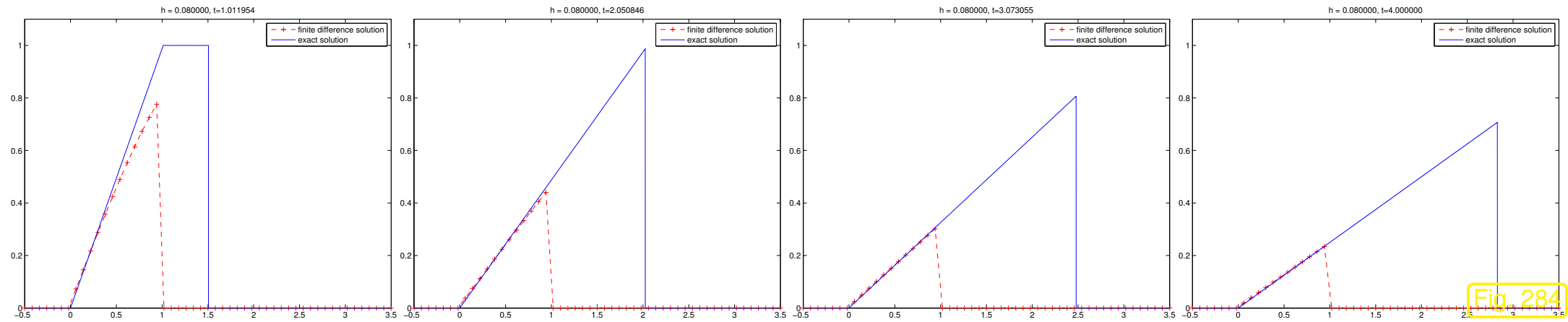
$$\updownarrow$$

$$\updownarrow$$

(8.3.3)

$$\dot{\mu}_j(t) + \mu_j \frac{\mu_j - \mu_{j-1}}{h} = 0, \quad j \in \mathbb{Z}, \quad 0 < t < T .$$

Numerical experiment with Cauchy problem from Ex. 8.2.43, $h = 0.08$, integration of (8.3.3) with MATLAB `ode45`.



Observation from numerical experiment: OK for rarefaction wave, but *scheme cannot capture speed of shock correctly!*

Analysis: consider $\mu_j(0) = \begin{cases} 1 & , \text{ if } j < 0 , \\ 0 & , \text{ if } j \geq 0 . \end{cases}$

\Leftrightarrow Riemann problem with $u_0(x) = 1$ for $x < 0 - \epsilon$, $u_0(x) = 0$ for $x > 0 - \epsilon$, $\epsilon \ll 1$.

Entropy solution (for this u_0) = travelling shock (\rightarrow Lemma 8.2.31), speed $\dot{s} = \frac{1}{2} > 0$

$\triangleleft \triangleright$

Numerical solution:
 $\vec{\mu}(t) = \vec{\mu}_0$ for all $t > 0$!

\blacktriangleright 3-point FDM (8.3.3) “converges” to wrong solution !

8.3.1 Semi-discrete conservation form

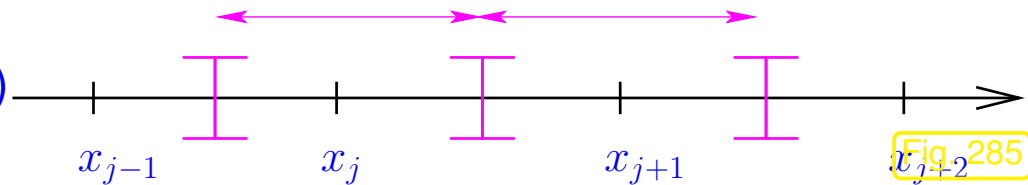
Objective: spatial semi-discretization of Cauchy problem

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times]0, T[\quad , \quad u(x, 0) = u_0(x) \quad , \quad x \in \mathbb{R} . \quad (8.2.9)$$

on (infinite) equidistant spatial mesh with mesh width $h > 0$

$$\mathcal{M} := \{]x_{j-1}, x_j[: x_j := jh, j \in \mathbb{Z} \} . \quad (8.3.4)$$

mesh cells and dual cells ▷




Finite volume interpretation of nodal unknowns μ_j ($\leftrightarrow x_j, j \in \mathbb{Z}$):


= conserved quantities in **dual cells** $]x_{j-1/2}, x_{j+1/2}[$, midpoints $x_{j-1/2} := \frac{1}{2}(x_j + x_{j-1})$:

$$\mu_j(t) \approx \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t) dx . \quad (8.3.5)$$

$$\vec{\mu}(t) := (\mu_j(t))_{j \in \mathbb{Z}} \in \mathbb{R}^{\mathbb{Z}} \quad \longleftrightarrow \quad u_N(x, t) = \sum_{j \in \mathbb{Z}} \mu_j(t) \chi_{]x_{j-1/2}, x_{j+1/2}[}(x) . \quad (8.3.6)$$



a function !


 notation: **characteristic function** $\chi_{]x_{j-1/2}, x_{j+1/2}[}(x) = \begin{cases} 1 & , \text{ if } x_{j-1/2} < x \leq x_{j+1/2} , \\ 0 & \text{ elsewhere.} \end{cases}$

 $(\mu_j(t))_{j \in \mathbb{Z}} \longleftrightarrow$ **piecewise constant** approximation $u_N(t) \approx u(\cdot, t)$

Note: $u_N(t)$ discontinuous at dual cell boundaries $x_{j+1/2}$!

By spatial integration over dual cells, which now play the role of the control volumes in (8.2.1):

$$\frac{d}{dt} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t) dx + f(u(x_{j+1/2}, t)) - f(u(x_{j-1/2}, t)) = 0, \quad j \in \mathbb{Z}, \quad (8.3.9)$$

(8.3.5) 
$$\frac{d\mu_j}{dt}(t) + \frac{1}{h} \left(\underbrace{f(u_N(x_{j+1/2}, t))}_{?} - \underbrace{f(u_N(x_{j-1/2}, t))}_{?} \right) = 0, \quad j \in \mathbb{Z}. \quad (8.3.10)$$

Problem: jump of $u_N(t)$ \triangleright ambiguity of values $u_N(x_{j+1/2}, t)$, $u_N(x_{j-1/2}, t)$, as we encountered it in the context of upwind quadrature in Sect. 7.2.2.1.

 R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Abstract “solution”:

Approximation $f(u_N(x_{j+1/2}, t)) \approx f_{j+1/2}(t) := F(\mu_{j-m_l+1}(t), \dots, \mu_{j+m_r}(t)), \quad j \in \mathbb{Z},$

with **numerical flux function** $F : \mathbb{R}^{m_l+m_r} \mapsto \mathbb{R}, \quad m_l, m_r \in \mathbb{N}_0.$

Note: the **same** numerical flux function is used for all dual cells!

► Finite volume semi-discrete evolution for (8.2.9) in **conservation form**:

$$\frac{d\mu_j}{dt}(t) = -\frac{1}{h} \left(F(\mu_{j-m_l+1}(t), \dots, \mu_{j+m_r}(t)) - F(\mu_{j-m_l}(t), \dots, \mu_{j+m_r-1}(t)) \right), \quad j \in \mathbb{Z}. \quad (8.3.11)$$

numerical flux (function) $F : \mathbb{R}^{m_l+m_r} \mapsto \mathbb{R}$

Special case: **2-point numerical flux** ($m_l = m_r = 1$): $F = F(v, w)$
($v \hat{=}$ left state, $w \hat{=}$ right state)

$$(8.3.11) \quad \blacktriangleright \quad \frac{d\mu_j}{dt}(t) = -\frac{1}{h} (F(\mu_j(t), \mu_{j+1}(t)) - F(\mu_{j-1}(t), \mu_j(t))), \quad j \in \mathbb{Z}. \quad (8.3.12)$$

Assumption on numerical flux functions: F Lipschitz-continuous in each argument.

Code 8.3.13: Wrapper code for finite volume evolution with 2-point flux

```

1 function ufinal = consformevl(a,b,N,u0,T,F)
2 % finite volume discrete evolution in conservation form with 2-point flux, see
   (8.3.12)
3 % Cauchy problem over time [0,T] restricted to finite interval [a,b],
4 % equidistant mesh with meshwidth N cells, meshwidth h := (b-a)/N.
5 % 2-point numerical flux function F = F(v,w) passed in handle F
6 h = (b-a)/N; x = a+0.5*h:h:b-0.5*h; % centers of dual cells
7 mu0 = h*u0(x)'; % vector  $\vec{\mu}_0$  of initial cell averages (column vector)
8 % right hand side function for MATLAB ode solvers
    
```

```
9 odefun = @(t,mu) (-1/h*fluxdiff(mu,F));
10 % timestepping by explicit Runge-Kutta method of order 5
11 options = odeset('abstol',1E-8,'reltol',1E-6,'stats','on');
12 [t,MU] = ode45(odefun,[0 T],mu0,options);
13 % 3D graphical output of u(x,t) over space-time plane
14 [X,T] = meshgrid(x,t);
15 figure; surf(X,T,MU/h); colormap(copper);
16 xlabel('\bf x','fontsize',14);
17 ylabel('\bf t','fontsize',14);
18 zlabel('\bf u','fontsize',14);
19 ufinal = MU(:,end);
20 end
21
22 function fd = fluxdiff(mu,F)
23 n = length(mu); fd = zeros(n,1);
24 % constant continuation of data outside [a,b]
25 fd(1) = F(mu(1),mu(2)) - F(mu(1),mu(1));
26 for j=2:n-1
27     fd(j) = F(mu(j),mu(j+1)) - F(mu(j-1),mu(j)); % see (8.3.12)
28 end
29 fd(n) = F(mu(n),mu(n)) - F(mu(n-1),mu(n));
30 end
```

8.3.2 Discrete conservation property

An evident first property of finite volume methods in conservation form:

$$\mu_j(0) = \mu_0 \in \mathbb{R} \quad \forall j \in \mathbb{Z} \quad \Rightarrow \quad \mu_j(t) = \mu_0 \quad \forall j \in \mathbb{Z}, \quad \forall t > 0. \quad (8.3.15)$$

that is, constant solutions are preserved by the method.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

A “telescopic sum argument” combined with the interpretation (8.3.6) shows that the conservation form (8.3.11) of the semi-discrete conservation law involves

$$\frac{d}{dt} \int_{x_{k-1/2}}^{x_{m+1/2}} u_N(x, t) \, dx = h \sum_{l=k}^m \frac{d\mu_l}{dt}(t) = -(f_{m+1/2}(t) - f_{k-1/2}(t)) \quad \forall k, m \in \mathbb{Z}.$$



$$\frac{d}{dt} \int_{x_{k-1/2}}^{x_{m+1/2}} u(x, t) dx = - (f(u(x_{j+1/2}, t)) - f(u(x_{k-1/2}, t))) ,$$

► With respect to unions of dual cells and numerical fluxes, the semidiscrete solution $u_N(t)$ satisfies a balance law of the same structure as a (weak) solution of (8.2.9).

Of course, the numerical flux function F has to fit the flux function f of the conservation law:

Definition 8.3.16 (Consistent numerical flux function).

A numerical flux function $F : \mathbb{R}^{m_l+m_r} \mapsto \mathbb{R}$ is **consistent** with the flux function $f : \mathbb{R} \mapsto \mathbb{R}$, if

$$F(u, \dots, u) = f(u) \quad \forall u \in \mathbb{R} .$$

Focus: solution of Riemann problem (\rightarrow Def. 8.2.28) by finite volume method in conservation form (8.3.11):

Initial data “constant at $\pm\infty$ ”: $\mu_{-j}(0) = u_l$, $\mu_j(0) = u_r$ for large j .

Consistency of the numerical flux function implies for large $m \gg 1$

$$\frac{d}{dt} \int_{-x_{-m-1/2}}^{x_{m+1/2}} u_N(x, t) dt = - (F(u_r, \dots, u_r) - F(u_l, \dots, u_l)) = -(f(u_r) - f(u_l)) . \quad (8.3.18)$$

Exactly the same balance law holds for any weak solutions of the Riemann problem!

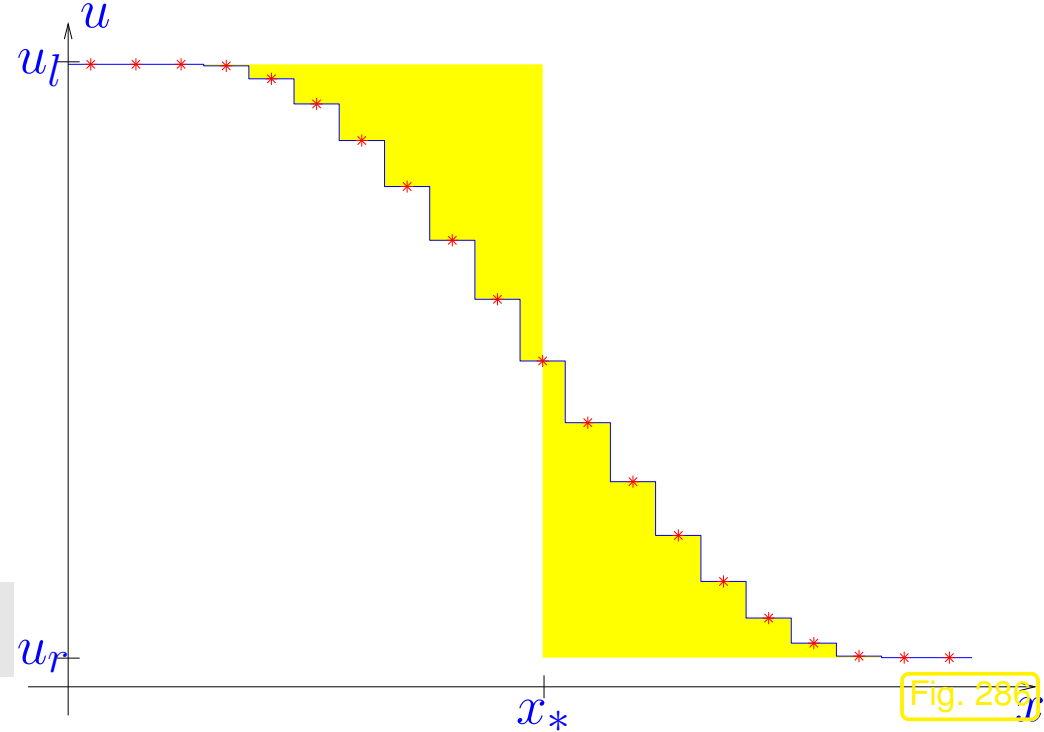
Situation: discrete solution $u_N(t)$ decreasing & supposed to approximate a shock

approximate location of shock at time t):

$$x_*(t) \in \mathbb{R}:$$

$$\int_{-\infty}^{x_*(t)} u_l - u_N(x, t) dx = \int_{x_*(t)}^{\infty} u_N(x, t) - u_r dx$$

equality of yellow areas \triangleright



$$\blacktriangleright \int_{x_{-m-1/2}}^{x_{m+1/2}} u_N(x, t) dx = (x_*(t) + x_{-m-1/2})u_l + (x_{m+1/2} - x_*(t))u_r .$$

$$(8.3.18) \quad \implies \frac{dx_*}{dt}(t) = \frac{1}{u_l - u_r} \sum_{j \in \mathbb{Z}} \frac{d\mu_j}{dt}(t) = \frac{f(u_l) - f(u_r)}{u_l - u_r} \stackrel{(8.2.23)}{=} \dot{s} .$$

Conservation form with consistent numerical flux yields correct “discrete shock speed”
(not liable to effect of Ex. 8.3.1)

8.3.3 Numerical flux functions

8.3.3.1 Central flux

Example 8.3.21 (Central flux for Burgers equation).

- Cauchy problem for Burgers equation (8.1.60) (flux function $f(u) = \frac{1}{2}u^2$) from Ex. 8.2.43 (“box” initial data)
- Spatial finite volume discretization in conservation form (8.3.11) with **central numerical fluxes**

$$F_1(v, w) := \frac{1}{2}(f(v) + f(w)) \quad , \quad F_2(v, w) := f\left(\frac{1}{2}(v + w)\right) . \quad (8.3.22)$$

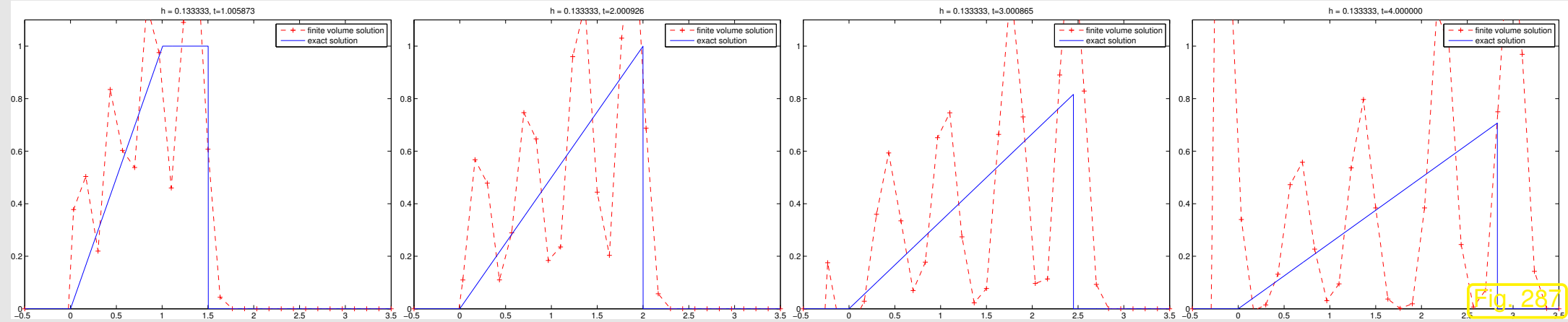
Obviously the 2-point numerical fluxes F_1 and F_2 are consistent according to Def. 8.3.16. The resulting spatially semi-discrete scheme is given by, see (8.3.12)

$$F_1: \quad \frac{d\mu_j}{dt}(t) = -\frac{1}{2h}(f(\mu_{j+1}(t)) - f(\mu_{j-1}(t))) ,$$

$$F_2: \quad \frac{d\mu_j}{dt}(t) = -\frac{1}{h}(f(\frac{1}{2}(\mu_j(t) + \mu_{j+1}(t))) - f(\frac{1}{2}(\mu_j(t) + \mu_{j-1}(t)))) .$$

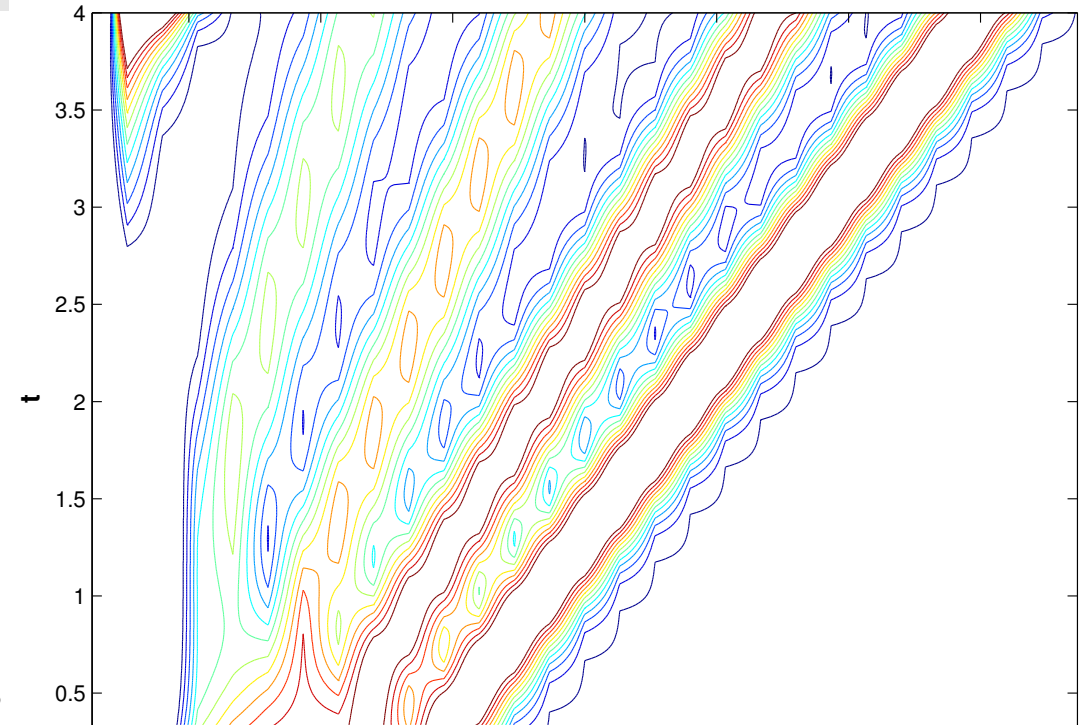
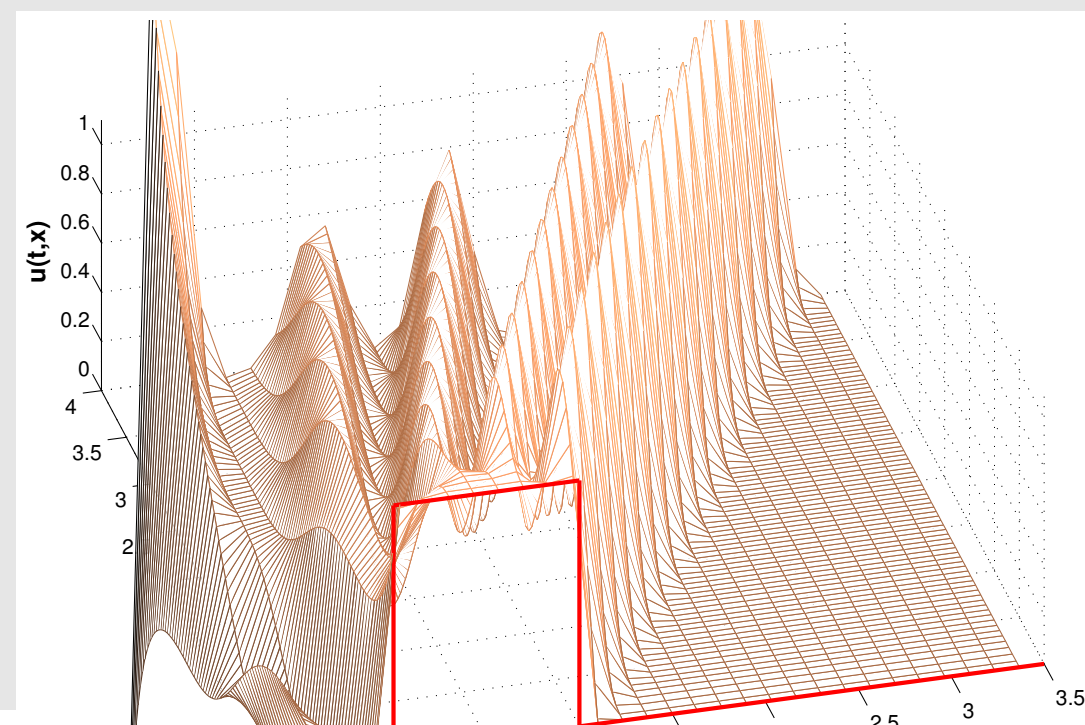
- timestepping based on adaptive Runge-Kutta method `ode45` of MATLAB
 (`opts = odeset('abstol', 1E-7, 'reltol', 1E-6);`).

Fully discrete evolution for central numerical flux F_1 : $h = 0.03$



R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ



Fully discrete evolution for central numerical flux F_2 : $h = 0.017$

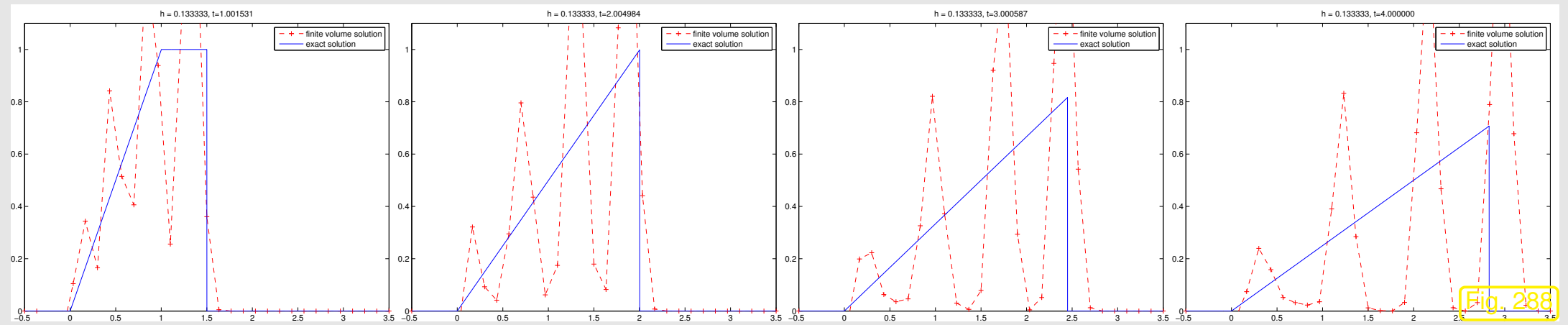
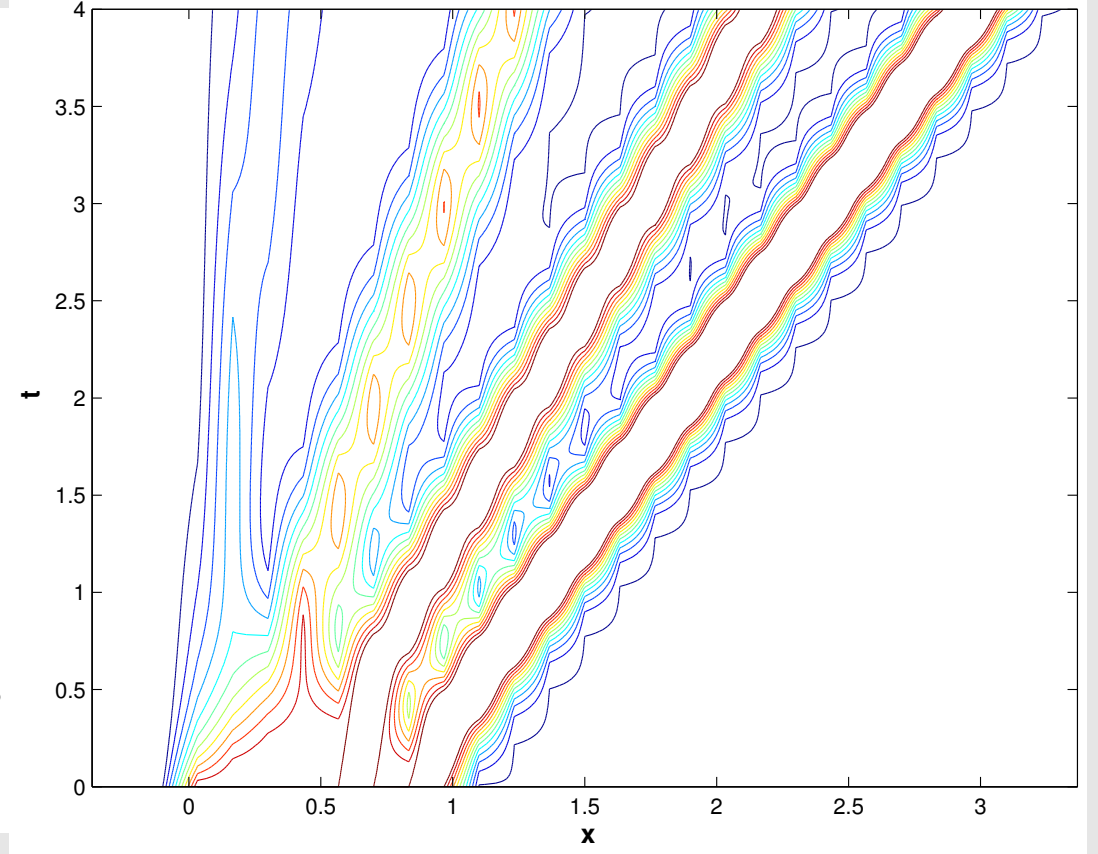
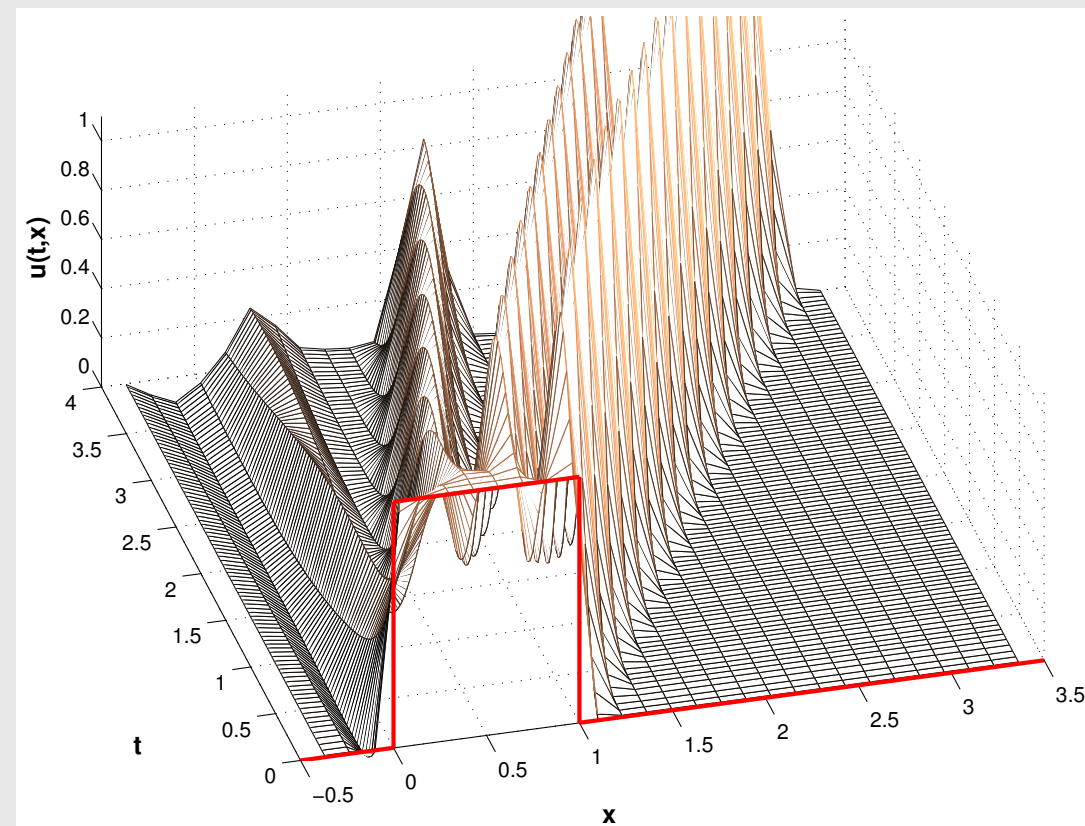


Fig. 288



R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

Observation: massive spurious oscillations utterly pollute numerical solution

Example 8.3.23 (Central flux for Traffic Flow equation).

- Cauchy problem for Traffic Flow equation (8.1.53) (flux function $f(u) = u(1-u)$) from Ex. 8.2.44 (“box” initial data)

Fully discrete evolution for central numerical flux F_1 : $h = 0.03$

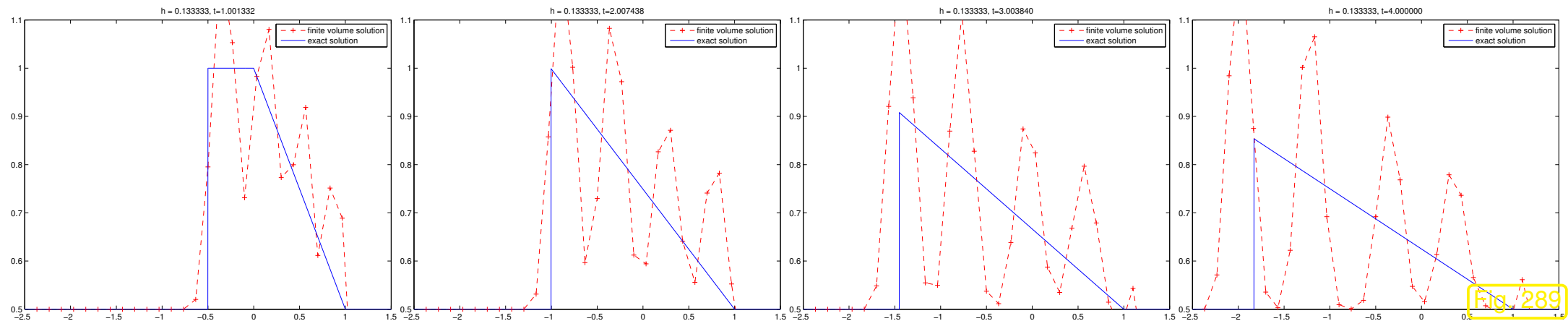
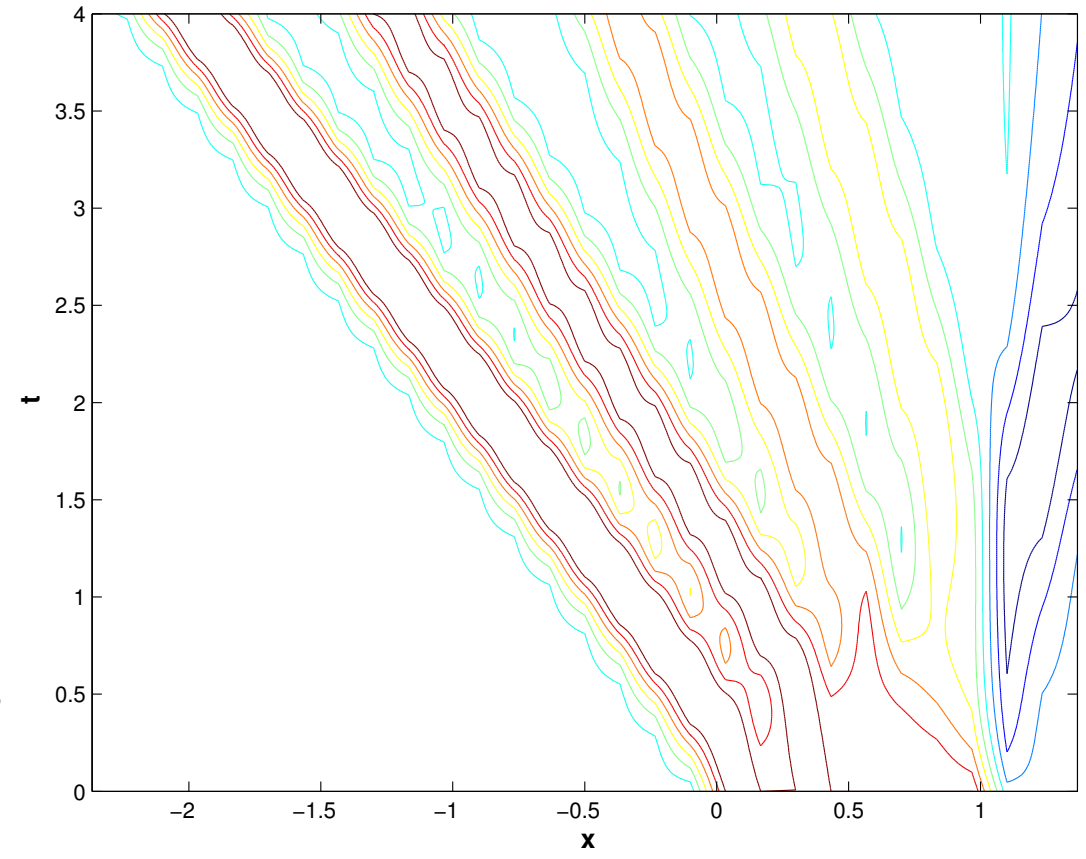
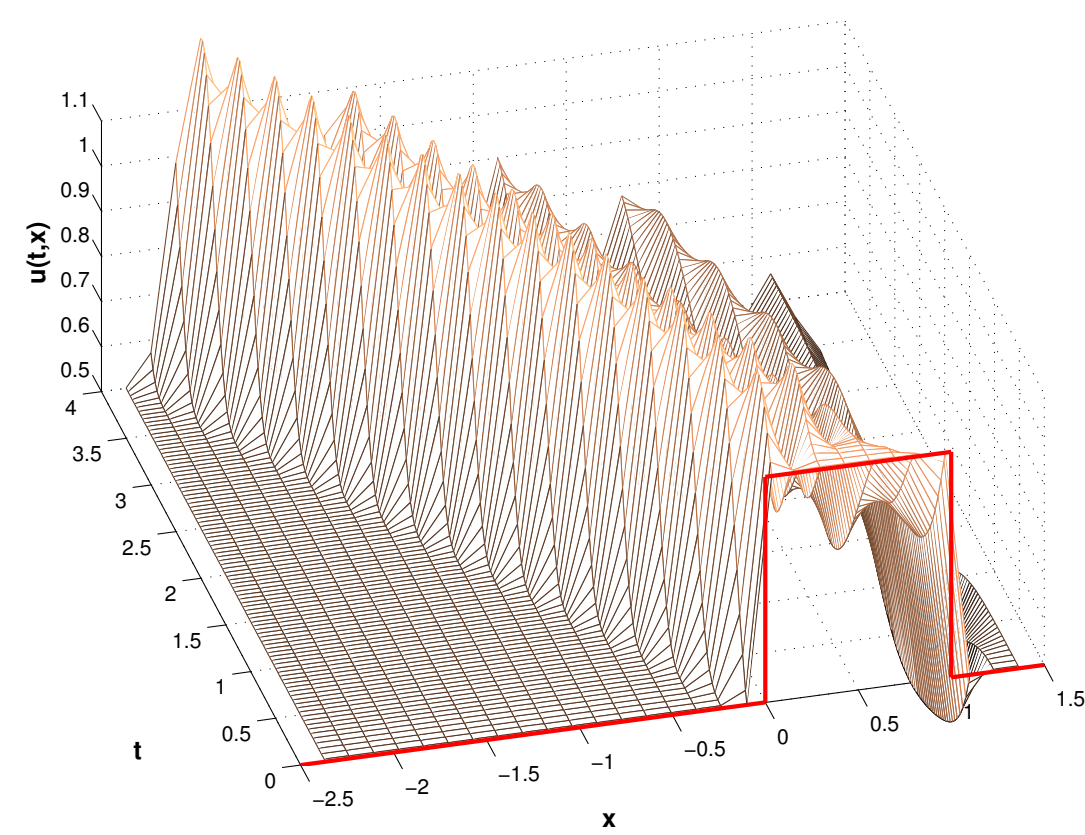


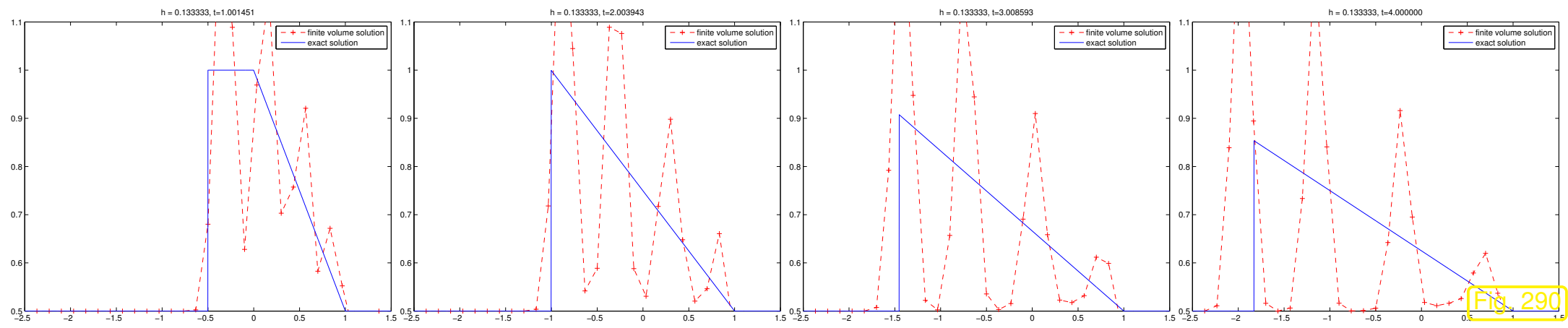
Fig. 289

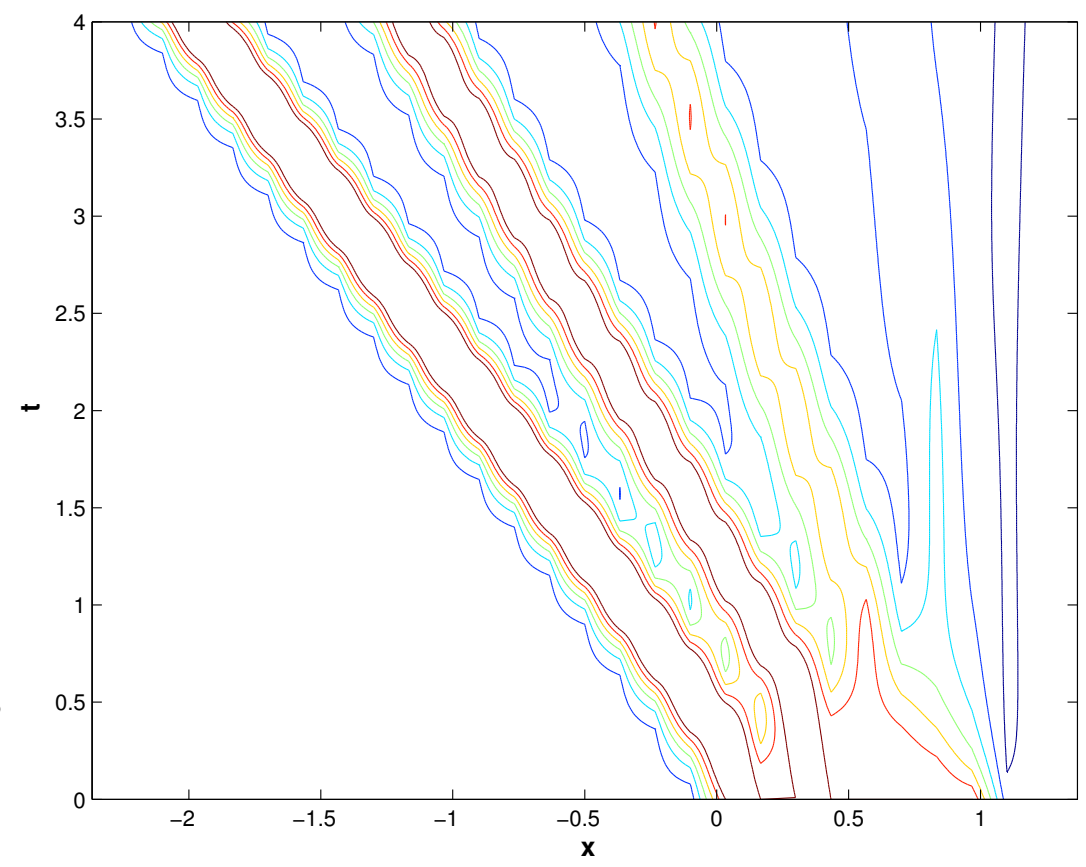
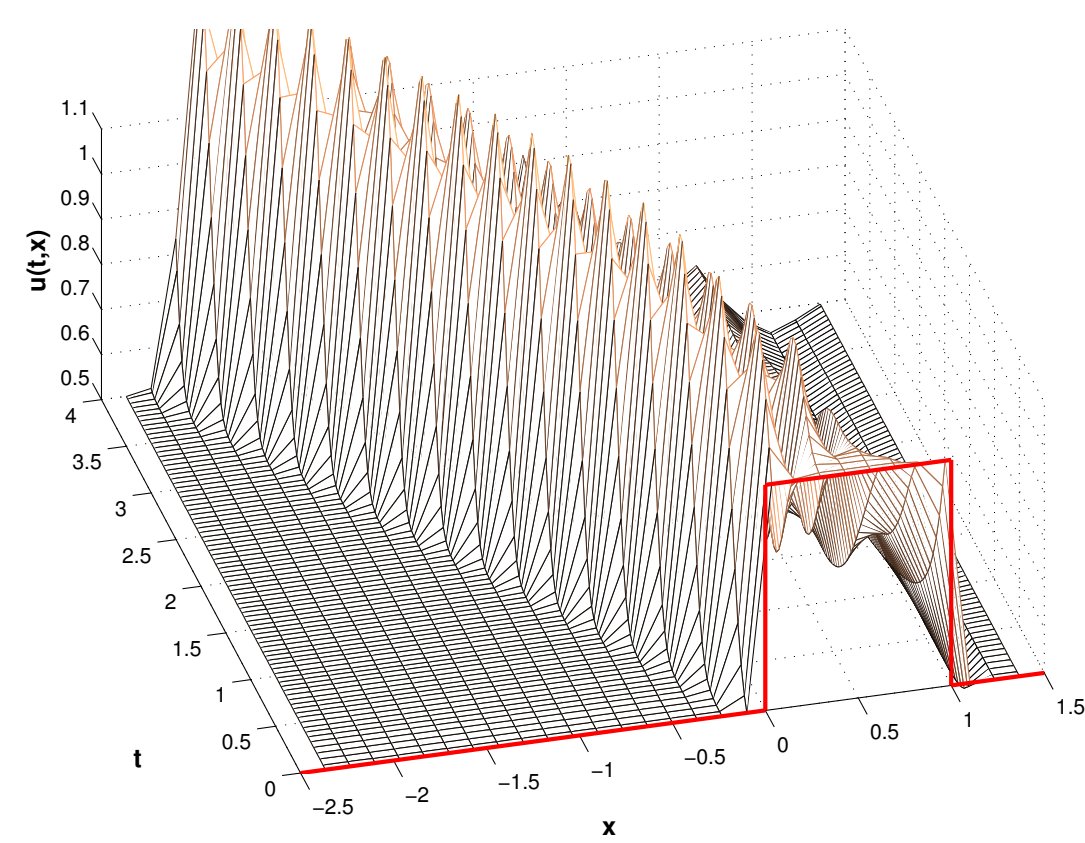
R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Fully discrete evolution for central numerical flux F_2 : $h = 0.017$





Observation: massive spurious oscillations utterly pollute numerical solution



Example 8.3.24 (Central flux for linear advection).

Cauchy problem (8.1.5): constant velocity scalar linear advection, $v = 1$, flux function $f(u) = vu$

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0 \quad \text{in } \tilde{\Omega} = \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) \quad \forall x \in \mathbb{R} . \quad (8.1.5)$$

= Cauchy problem for 1D transport equation (7.3.7)!

Finite volume spatial discretization in conservation form (8.3.11) with central numerical fluxes from (8.3.22):

$$\begin{aligned} F_1(v, w) &:= \frac{1}{2}(f(v) + f(w)) \\ F_2(v, w) &:= f\left(\frac{1}{2}(v + w)\right) \end{aligned} \quad \Rightarrow \quad \frac{d\mu_j}{dt}(t) = -\frac{v}{2h}(\mu_{j+1}(t) - \mu_{j-1}(t)) , \quad j \in \mathbb{Z} .$$

= spatial semi-discretization using linear finite element Galerkin discretization of convective term, see (7.2.15).

Sect. 7.3.1: this method is *prone to spurious oscillations*, see Ex. 7.3.4.

This offers an explanation also for its failure for Burgers equation, see Ex. 8.3.21



Sect. 7.2.2.2: **artificial diffusion** cures instability of central difference quotient

$$\begin{aligned} \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} &= 0 \quad \text{in } \mathbb{R} \times]0, T[, \\ \updownarrow & \\ \frac{\partial u}{\partial t} + (ch/2) \underbrace{\frac{-\mu_{j-1} + 2\mu_j - \mu_{j+1}}{h^2}}_{\hat{=} \text{ difference quotient for } \frac{d^2u}{dx^2}} + \underbrace{c \frac{\mu_{j+1} - \mu_{j-1}}{2h}}_{\hat{=} \text{ difference quotient for } c \frac{du}{dx}} &= 0, \quad j \in \mathbb{Z} . \end{aligned}$$

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Can this be rewritten in conservation form (8.3.11)? YES!

$$(ch/2) \frac{-\mu_{j-1} + 2\mu_j - \mu_{j+1}}{h^2} + c \frac{\mu_{j+1} - \mu_{j-1}}{2h} = \frac{1}{h} (F(\mu_j, \mu_{j+1}) - F(\mu_{j-1}, \mu_j)) ,$$

with $F(v, w) := \frac{c}{2}(v + w) - \frac{c}{2}(w - v)$. (8.3.27)

central numerical flux

diffusive/viscous numerical flux

Recall from Rem. 8.2.3: the flux function $f(u) = -\frac{\partial u}{\partial x}$ models diffusion. Hence, the diffusive numerical flux amounts to a central finite difference discretization of the partial derivative in space:

$$-\frac{\partial u}{\partial x}(x, t) \Big|_{x=x_{j+1/2}} \approx -\frac{1}{h}(u(x_{j+1}, t) - u(x_j, t)) .$$

How to adapt this to general scalar conservation laws?

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = \frac{\partial u}{\partial t} + \underbrace{f'(u)}_{\text{local speed of transport} \leftrightarrow c} \frac{\partial u}{\partial x} = 0 \quad (8.3.28)$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

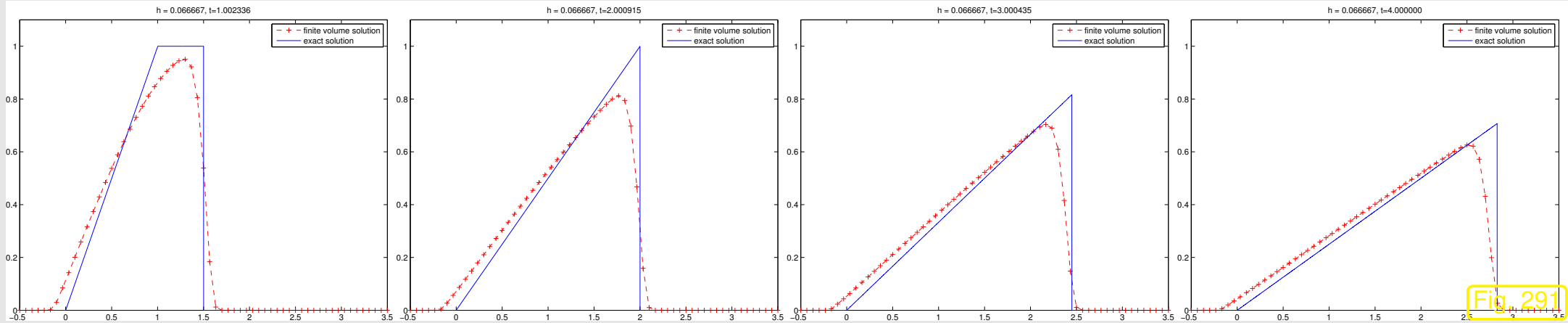
► (local) **Lax-Friedrichs flux**

$$F_{\text{LF}}(v, w) = \frac{1}{2}(f(v) + f(w)) - \frac{1}{2} \max_{\min\{v, w\} \leq u \leq \max\{v, w\}} |f'(u)|(w - v) . \quad (8.3.29)$$

Example 8.3.31 (Lax-Friedrichs flux for Burgers equation).

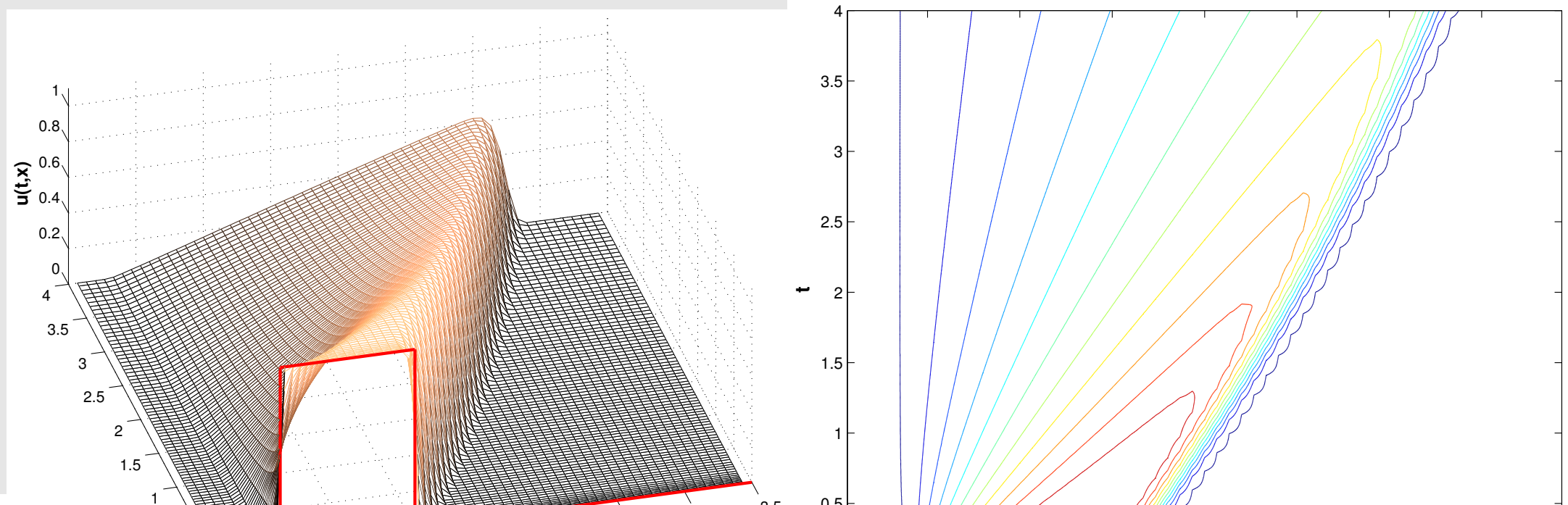
☞ same setting and conservative discretization as in Ex. 8.3.21

☞ Numerical flux function: Lax-Friedrichs flux (8.3.29)



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Observation: spurious completely suppressed, qualitatively good resolution of both shock and rarefaction.

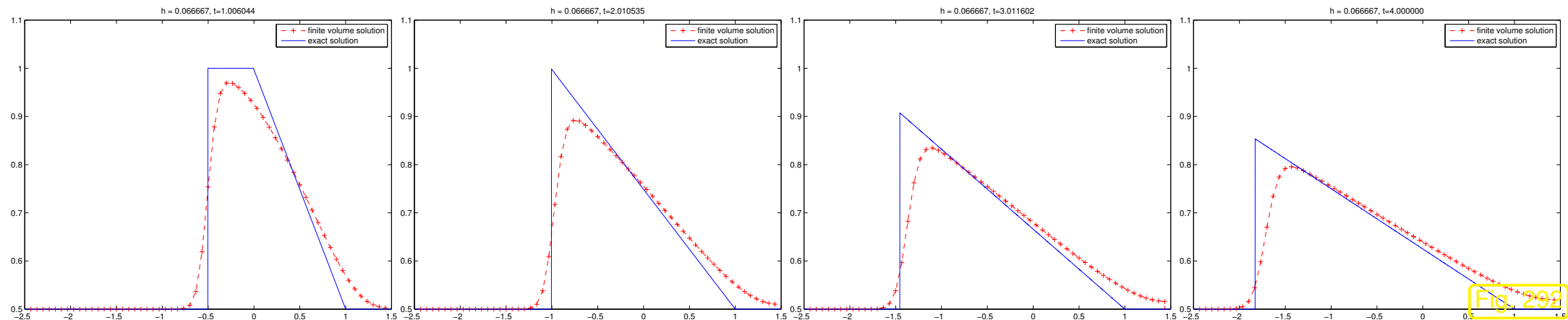
Effect of artificial diffusion: smearing of shock, cf. discussion in Ex. 7.2.27.



Example 8.3.32 (Lax-Friedrichs flux for traffic flow equation).

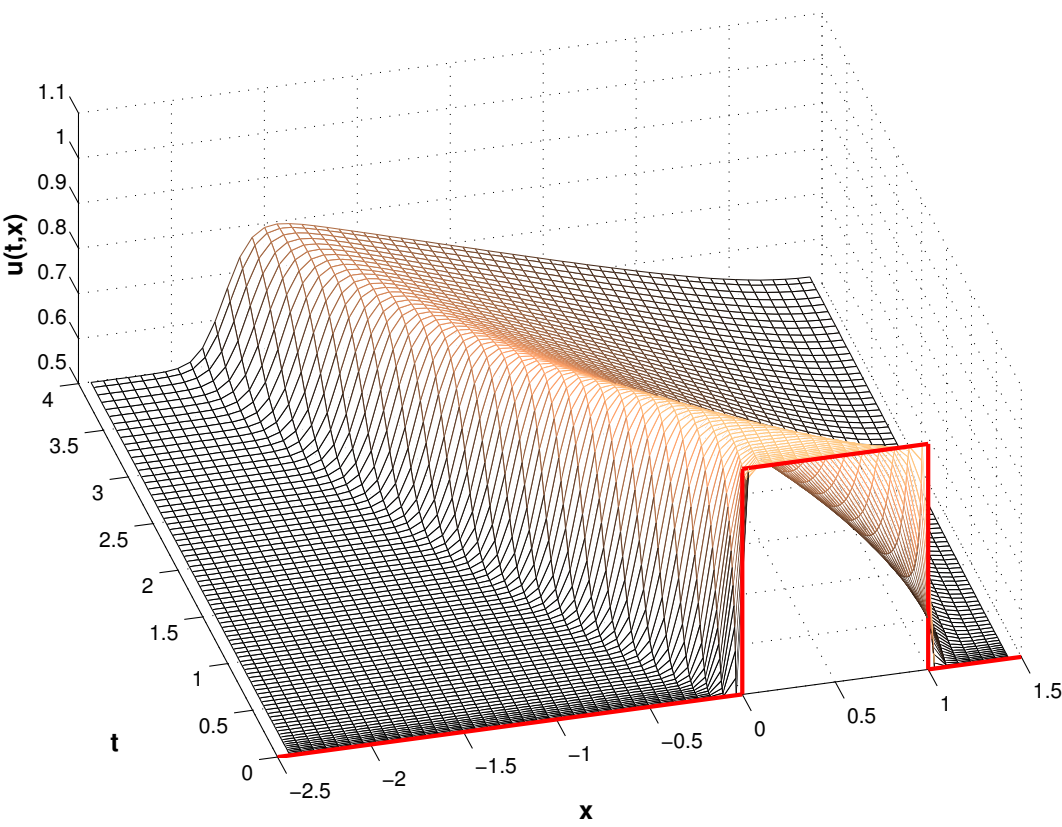
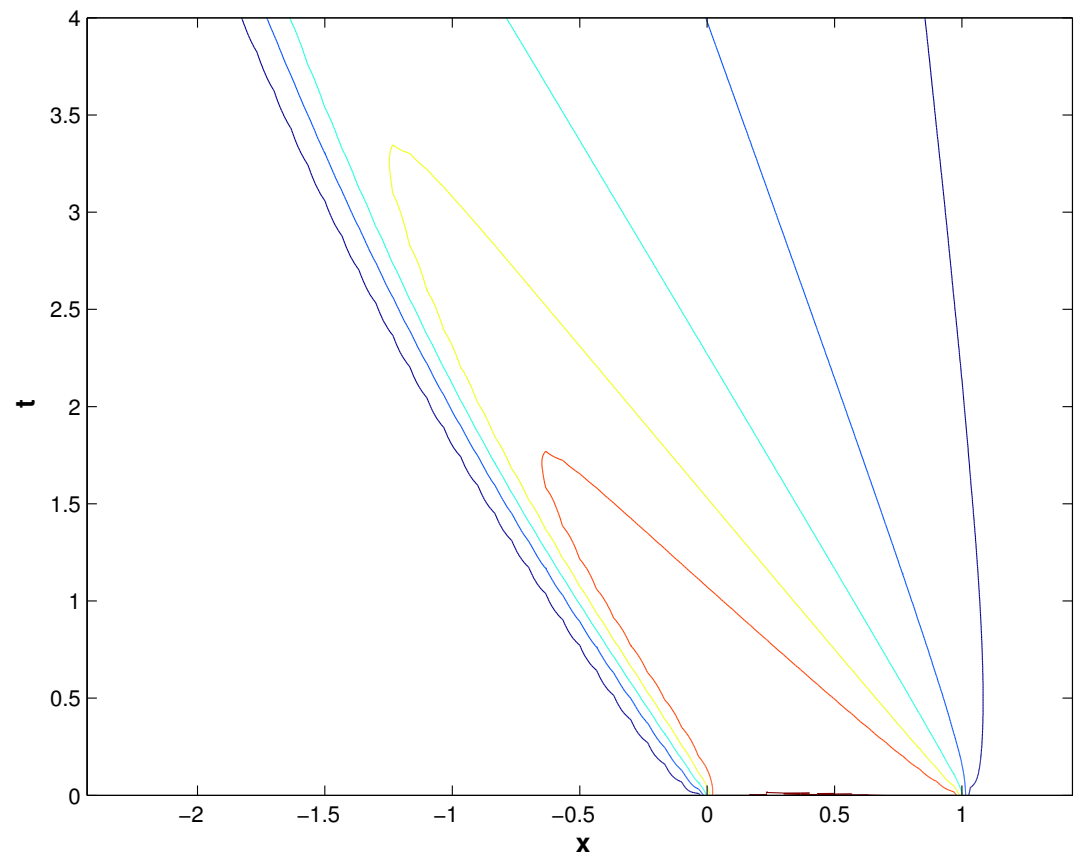
☞ same setting and conservative discretization as in Ex. 8.3.21

☞ Numerical flux function: Lax-Friedrichs flux (8.3.29)



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



8.3.3.3 Upwind flux

Another idea for stable spatial discretization of stationary transport in Sect. 7.2.2.1:

“upwinding” = obtain information from where transport brings it

☞ remedy for ambiguity of evaluation of discontinuous gradient in upwind quadrature

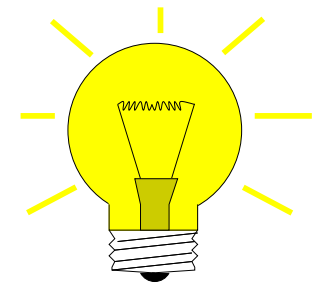
Ambiguity also faced in the evaluation of $f(u_N(x_{j+1/2}), t), f(u_N(x_{j-1/2}), t)$, see (8.3.10), which forced us to introduce numerical flux functions in (8.3.11).

(8.3.28): local velocity of transport at $(x, t) \in \tilde{\Omega}$ is given by $f'(u(x, t))$

➤ ambiguous local velocity of transport at discontinuity of u_N !

Idea: deduce local velocity of transport from Rankine-Hugoniot jump condition

(8.2.23)



▶ local velocity of transport $= \begin{cases} f'(u) \\ \frac{f(u_r) - f(u_l)}{u_r - u_l} \end{cases}$ for unique state, $u = u_l = u_r$
at discontinuity.

$(u_l, u_r \hat{=}$ states to left and right of discontinuity)

$$F_{\text{uw}}(v, w) = \begin{cases} f(v) & , \text{ if } \dot{s} \geq 0 , \\ f(w) & , \text{ if } \dot{s} < 0 , \end{cases} \quad \dot{s} := \begin{cases} \frac{f(w) - f(v)}{w - v} & \text{ for } v \neq w , \\ f'(v) & \text{ for } v = w . \end{cases} \quad (8.3.33)$$

Example 8.3.35 (Upwind flux for Burgers equation).

same setting and conservative discretization as in Ex. 8.3.21

Numerical flux function: upwind flux (8.3.33)

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

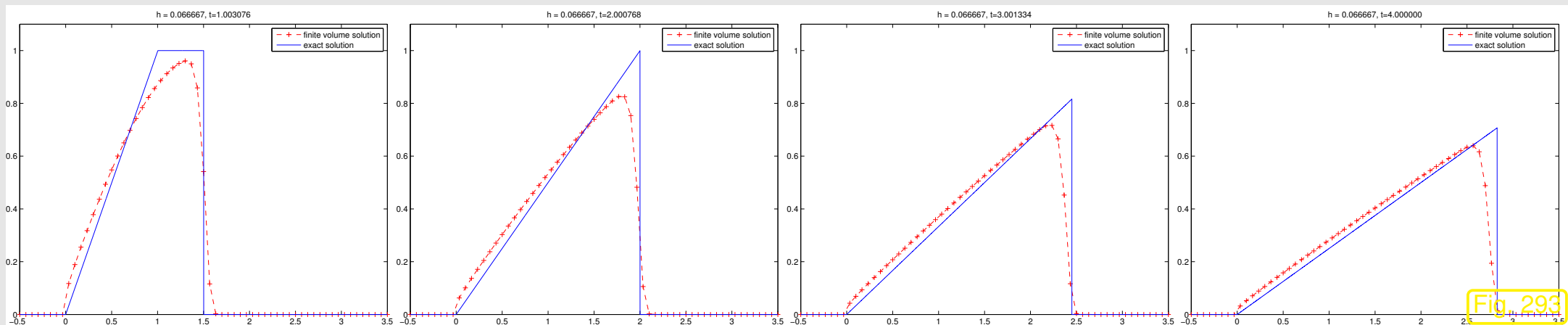
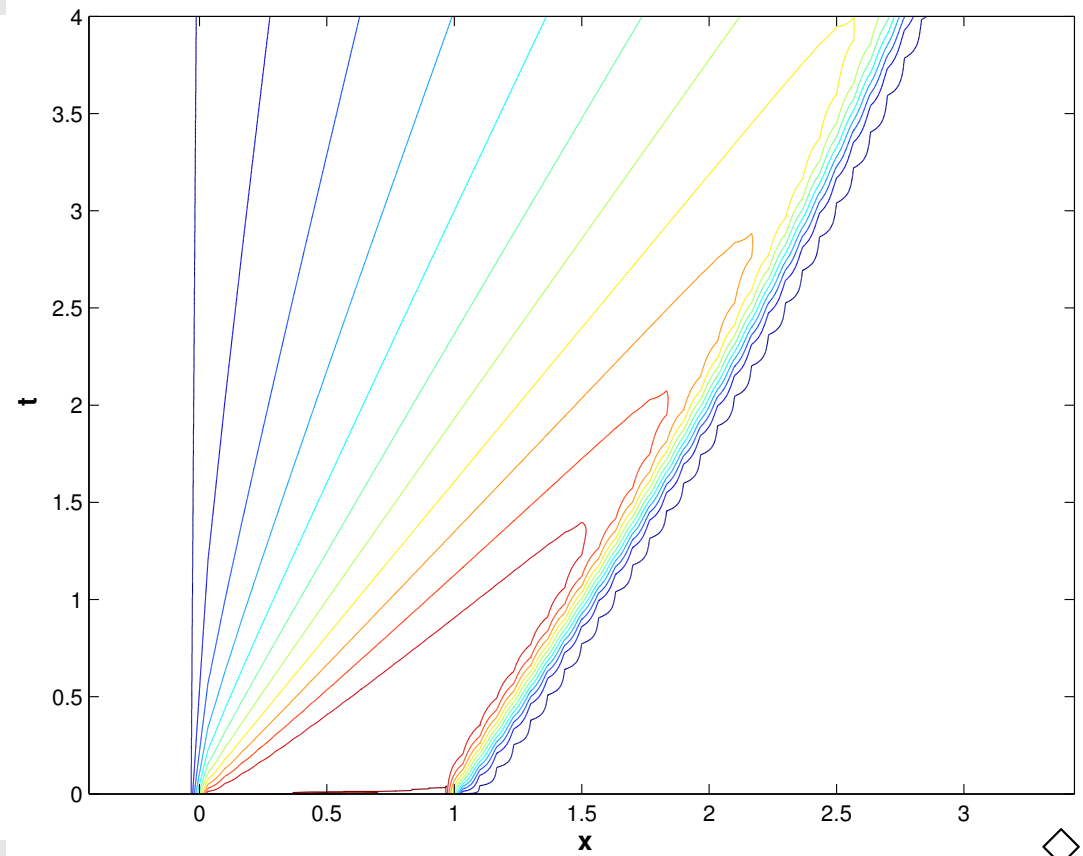
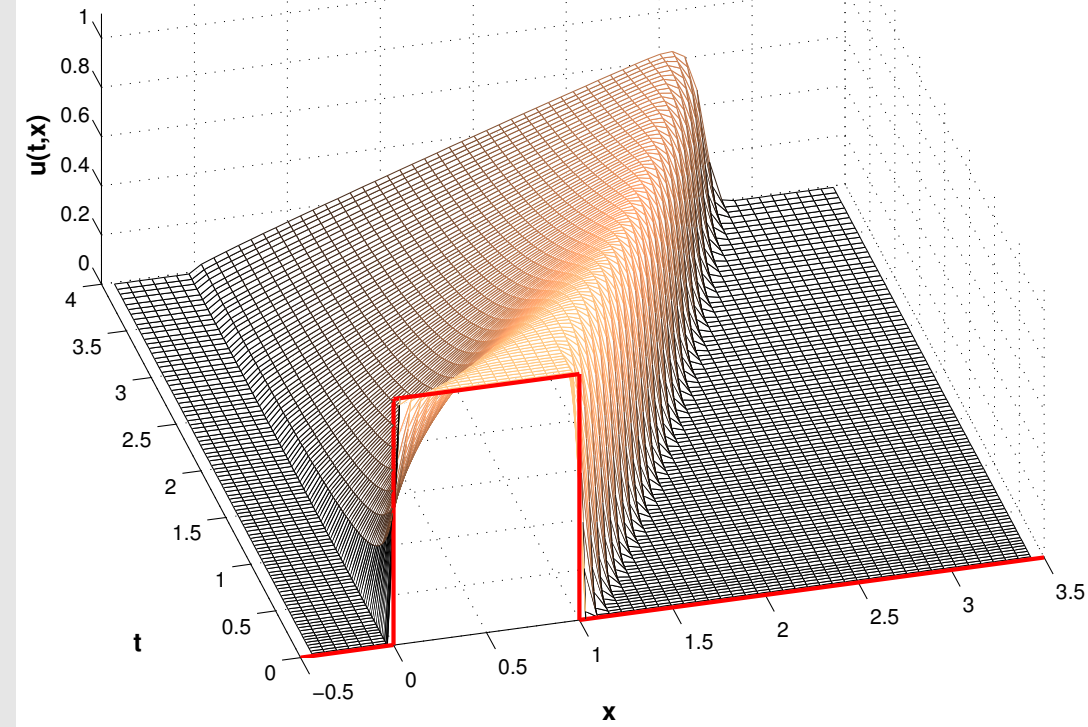


Fig. 293



Example 8.3.36 (Upwind flux for traffic flow equation).

- ☞ same setting and conservative discretization as in Ex. 8.3.21
- ☞ Numerical flux function: upwind flux (8.3.33)

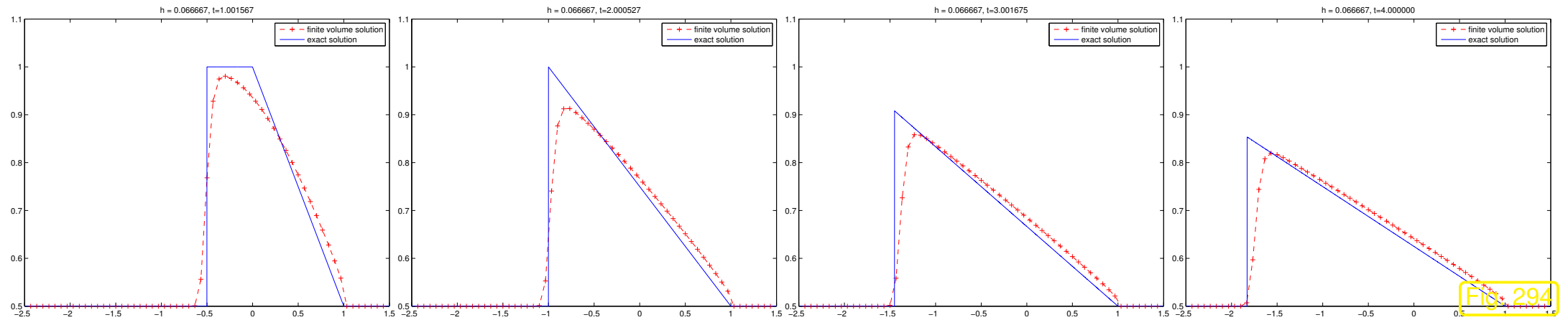
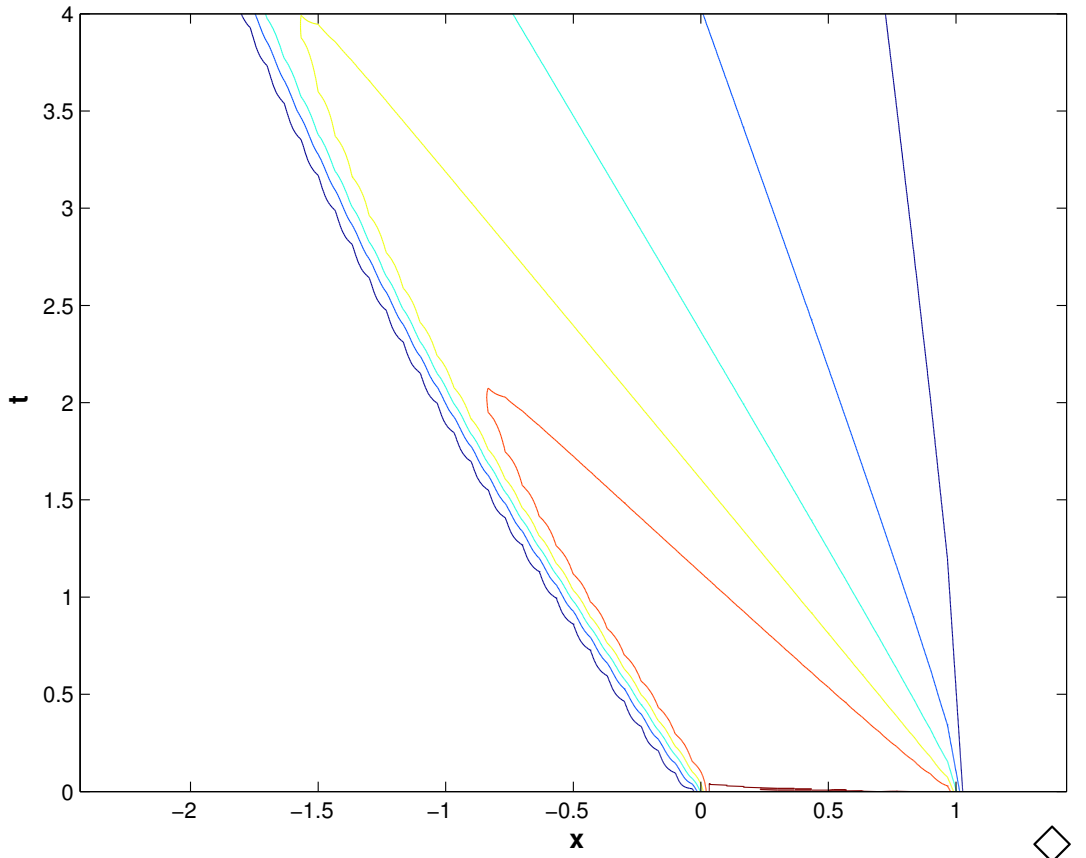
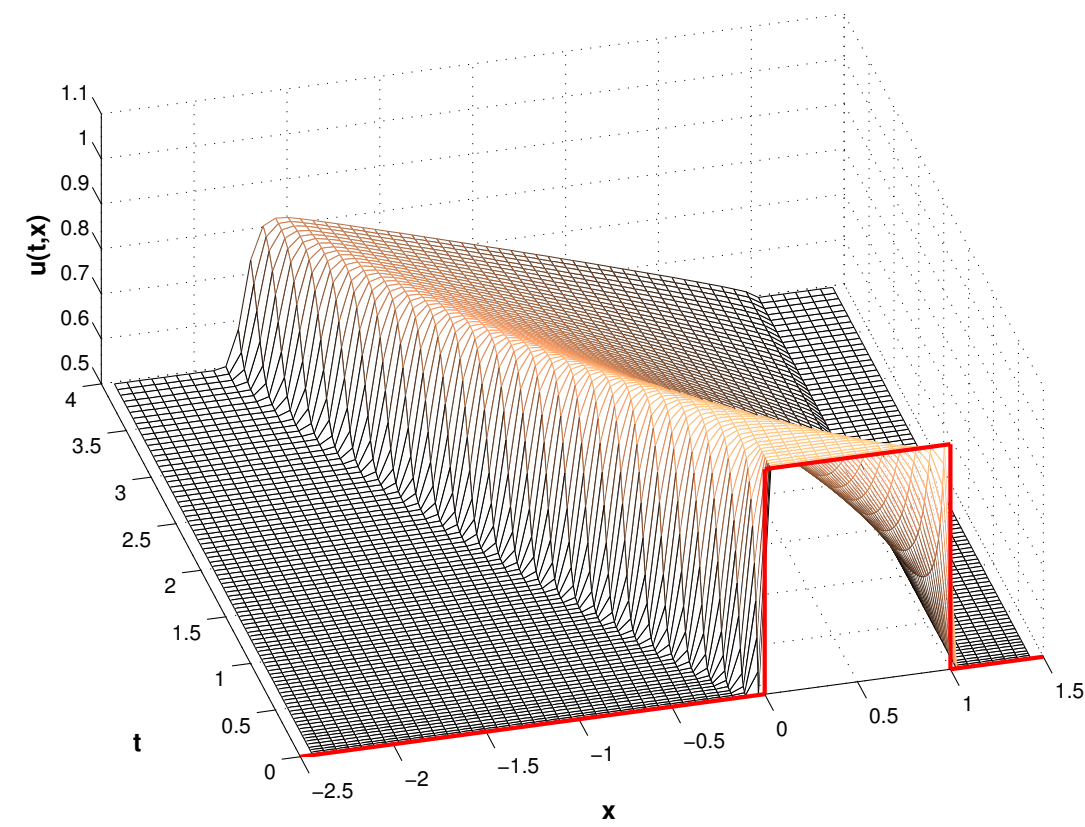


Fig. 294



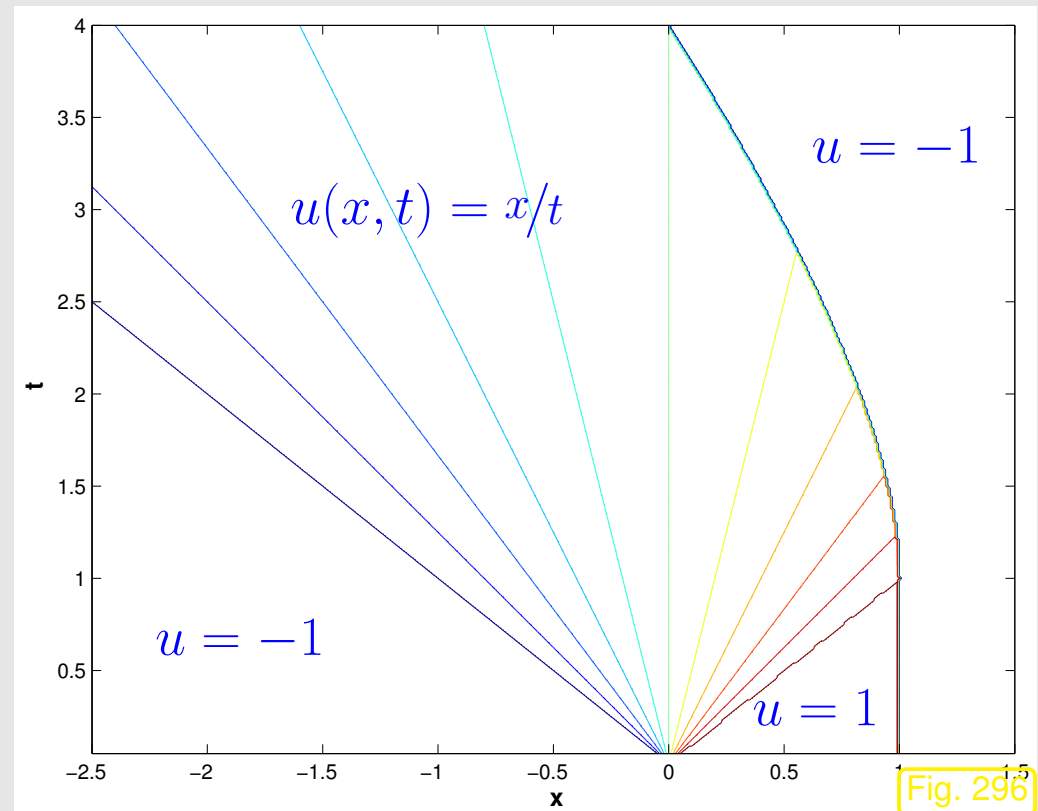
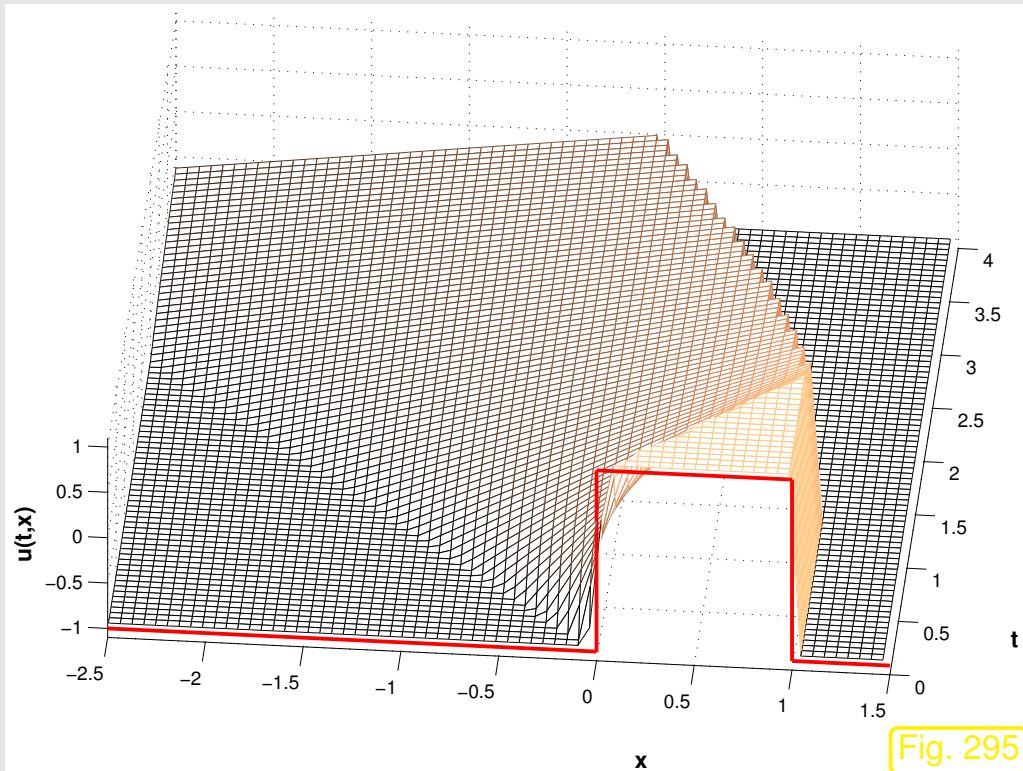
R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs
 SAM, ETHZ



Example 8.3.37 (Upwind flux and transsonic rarefaction).

Cauchy problem (8.2.9) for Burgers equation (8.1.60), i.e., $f(u) = \frac{1}{2}u^2$ and initial data

$$u_0(x) = \begin{cases} -1 & \text{for } x < 0 \text{ or } x > 1, \\ 1 & \text{for } 0 < x < 1. \end{cases} \quad (8.3.39)$$

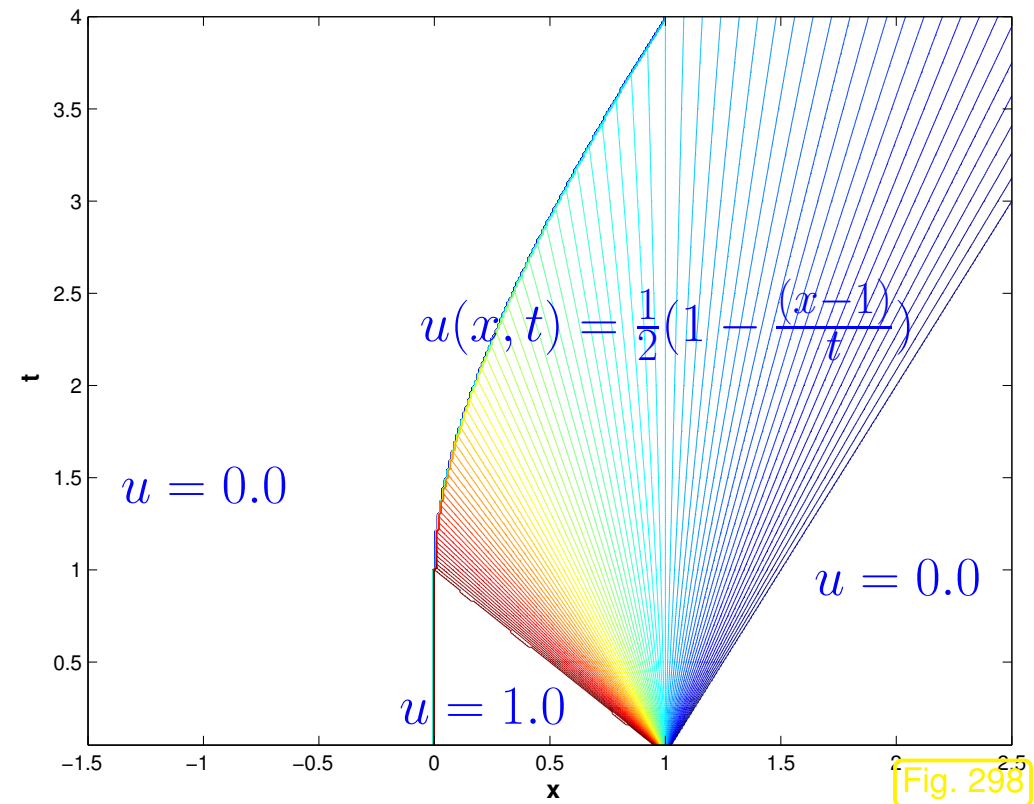
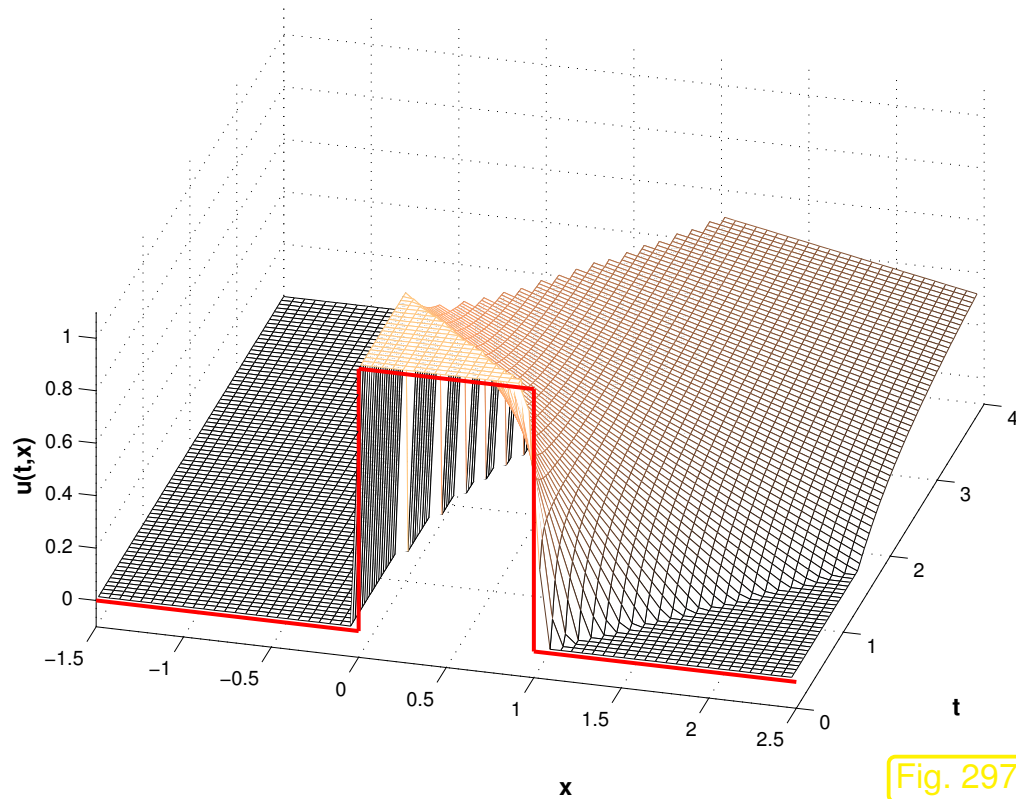


R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Cauchy problem (8.2.9) for traffic flow equation (8.1.53), i.e., $f(u) = u(1 - u)$ and initial data

$$u_0(x) = \begin{cases} 0 & \text{for } x < 0 \text{ or } x > 1, \\ 1 & \text{for } 0 < x < 1. \end{cases} \quad (8.3.40)$$

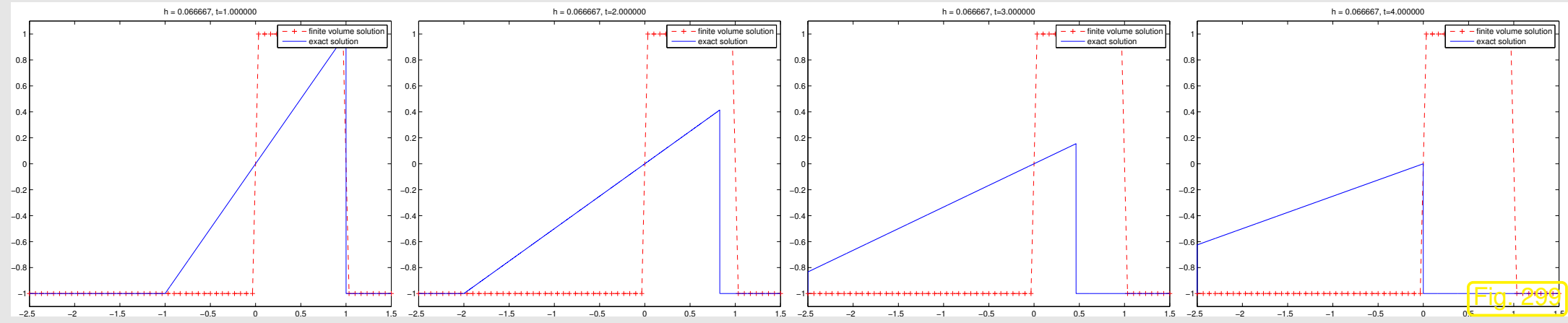


R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

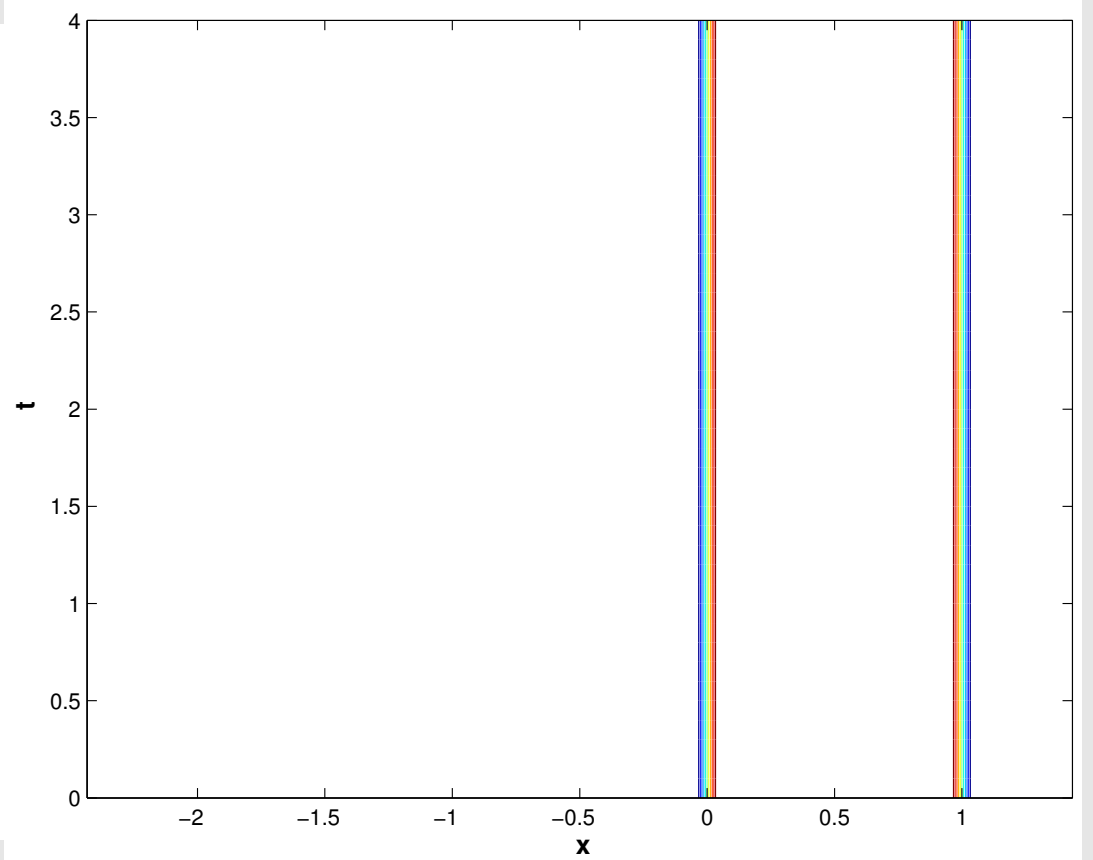
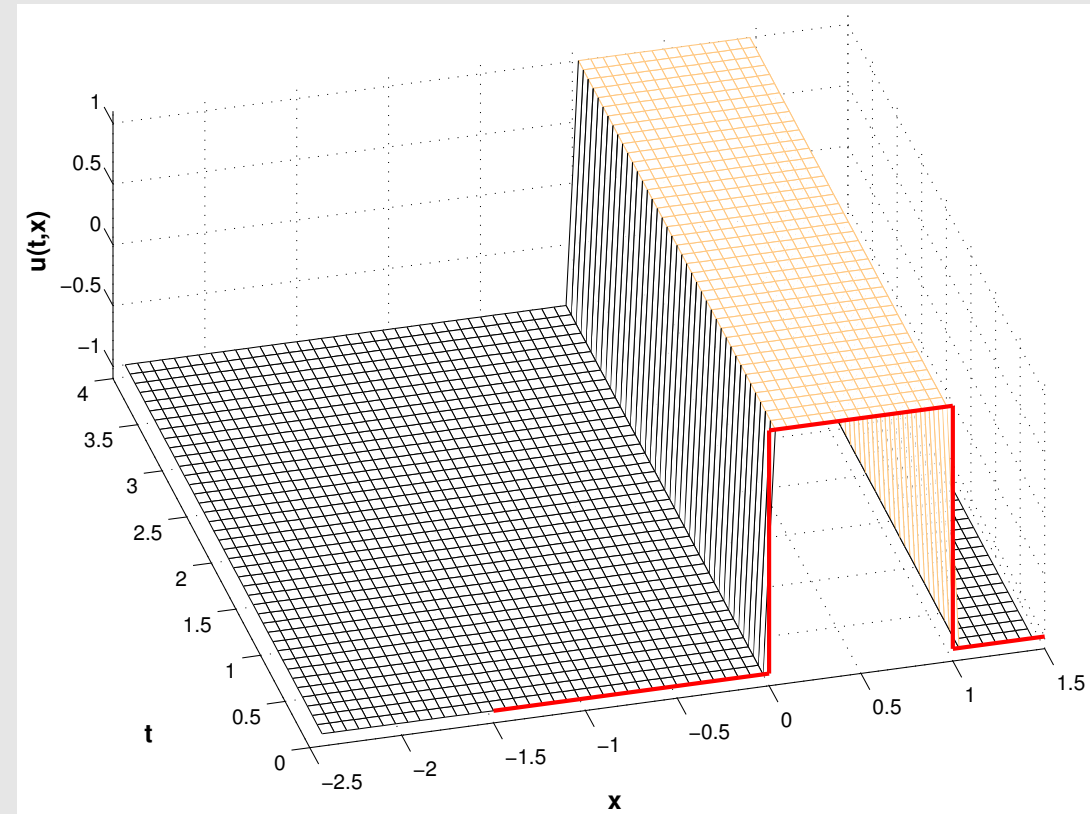
The *entropy solution* (\rightarrow Sect. 8.2.6) of these Cauchy problem features a **transsonic rarefaction fan** at $x = 1$: this is a rarefaction solution (\rightarrow Lemma 8.2.38) whose “edges” move in opposite directions.

Burgers' equation, initial density (8.3.39): numerical solution with finite volume method with upwind flux (8.3.33).



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Traffic flow equation, initial data (8.3.40): numerical solution with finite volume method with upwind flux (8.3.33).

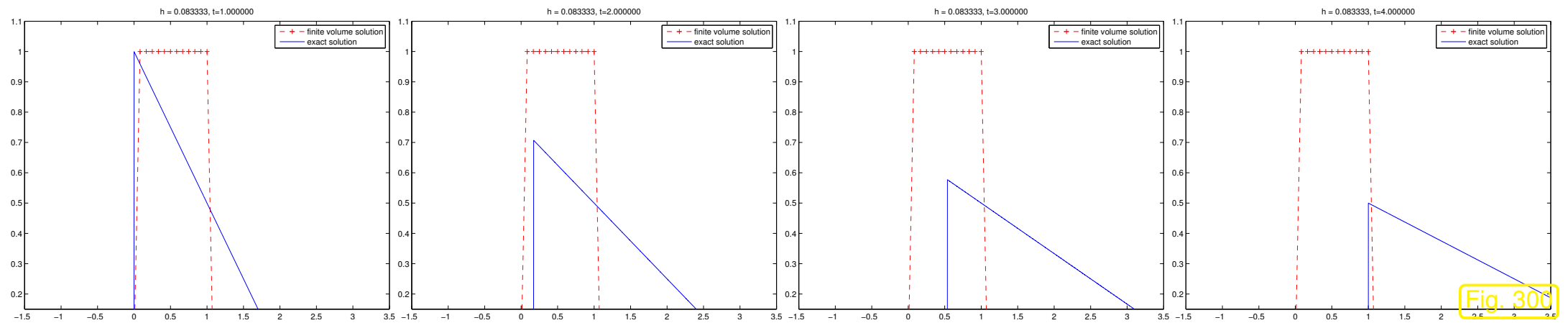
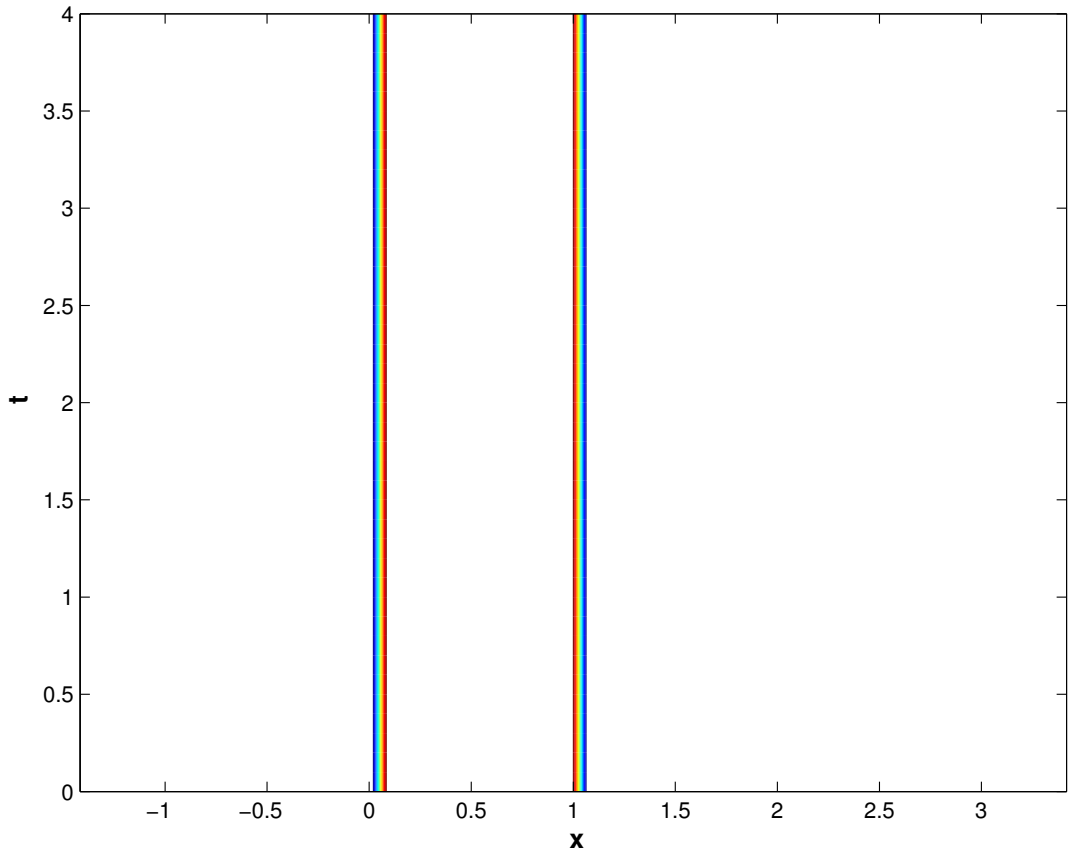
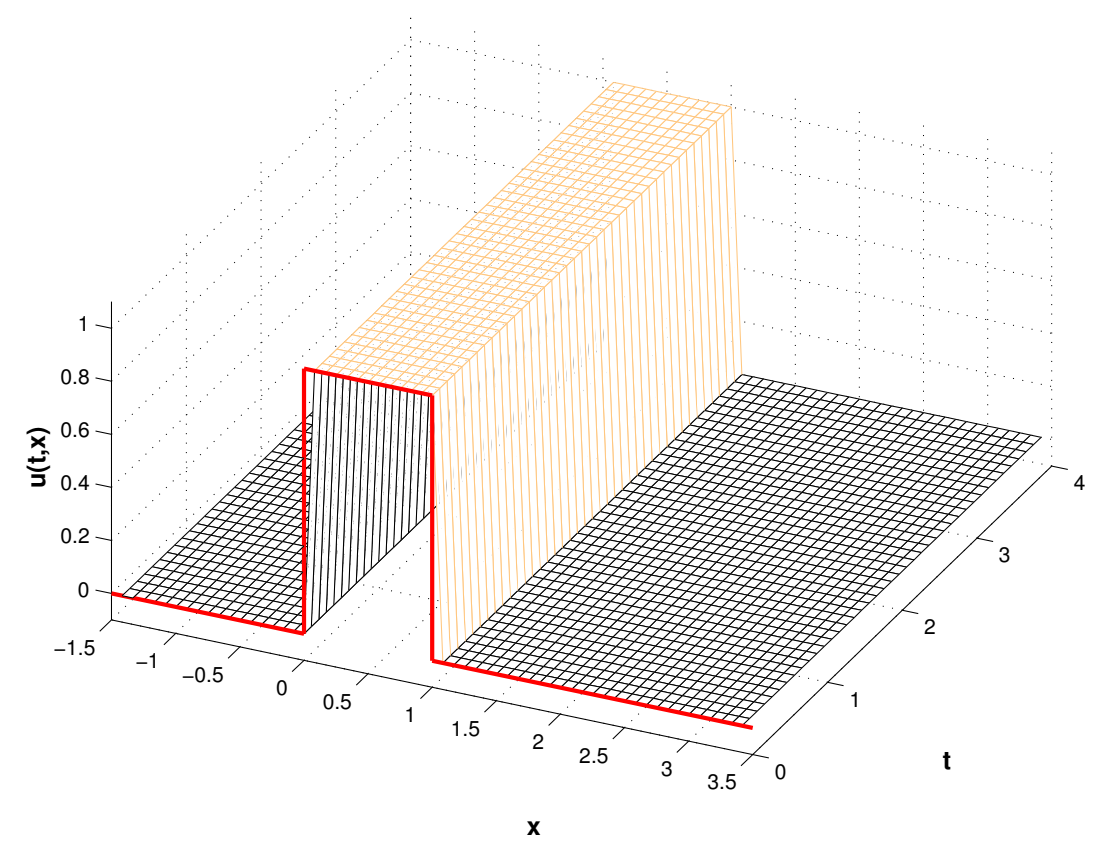


Fig. 306



R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs
 SAM, ETHZ

Conservative finite volume discretization with upwind flux produces (stationary) *expansion shock* instead of transonic rarefaction!

Sect. 8.2.6: this is a weak solution, but it violates the entropy condition, “non-physical shock”.



Example 8.3.41 (Upwind flux: Convergence to expansion shock).

- Cauchy problem (8.2.9) for Burgers equation (8.1.60), i.e., $f(u) = \frac{1}{2}u^2$
- $u_0(x) = 1$ for $x > 0$, $u_0(x) = -1$ for $x < 0$
 - entropy solution = rarefaction wave (\rightarrow Lemma 8.2.38)
- FV in conservation form, upwind flux (8.3.33), on equidistant grid, $x_j = (j + \frac{1}{2})h$, meshwidth $h > 0$

► initial nodal values $\mu_j(0) = \begin{cases} -1 & \text{for } j < 0, \\ 1 & \text{for } j \geq 0. \end{cases}$

► Semi-discrete evolution equation:

$$\frac{d\mu_j}{dt}(t) = -\frac{1}{2h} \cdot \begin{cases} \mu_{j+1}^2(t) - \mu_j^2(t) & \text{for } j \geq 0, \\ \mu_j^2(t) - \mu_{j-1}^2(t) & \text{for } j < 0. \end{cases}$$

► $\mu_j(t) = \mu_j(0)$ for all t ► for $h \rightarrow 0$, convergence to entropy violating expansion shock !

► finite volume method may converge to non-physical weak solutions !



8.3.3.4 Godunov flux

(The following discussion is for convex flux functions only. The reader is encouraged to figure out the modifications necessary if the flux function is concave.)

The upwind flux (8.3.33) is a numerical flux of the form

$$F(v, w) = f(u^\downarrow(v, w)) \quad \text{with an intermediate state } u^\downarrow(v, w) \in \mathbb{R} .$$

For the upwind flux the intermediate state is not really “intermediate”, but coincides with one of the states v, w depending on the sign of the “local shock speed” $\dot{s} := \frac{f(w) - f(v)}{w - v}$.

Idea: obtain suitable intermediate state as

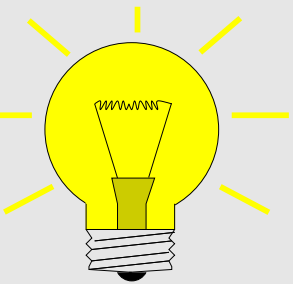
$$u^\downarrow(v, w) = \psi(0) , \quad (8.3.46)$$

where $u(x, t) = \psi(x/t)$ solves the Riemann problem (\rightarrow Def. 8.2.28)

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad , \quad u(x, 0) = \begin{cases} v & , \text{ for } x < 0 , \\ w & , \text{ for } x \geq 0 . \end{cases} \quad (8.3.47)$$

We focus on $f : \mathbb{R} \mapsto \mathbb{R}$ strictly convex & smooth (e.g. Burgers equations (8.1.60))

► Riemann problem (8.3.47) (\rightarrow Def. 8.2.28) has the *entropy solution* (\rightarrow Sect. 8.2.6):



① If $v > w$ ➤ discontinuous solution, **shock** (\rightarrow Lemma 8.2.31)

$$u(t, x) = \begin{cases} v & \text{if } x < \dot{s}t, \\ w & \text{if } x > \dot{s}t, \end{cases} \quad \dot{s} = \frac{f(v) - f(w)}{v - w}. \quad (8.3.48)$$

② If $v \leq w$ ➤ continuous solution, **rarefaction wave** (\rightarrow Lemma 8.2.38)

$$u(t, x) = \begin{cases} v & \text{if } x < f'(v)t, \\ g(x/t) & \text{if } f'(v) \leq x/t \leq f'(w), \\ w & \text{if } x > f'(w)t, \end{cases} \quad g := (f')^{-1}. \quad (8.3.49)$$

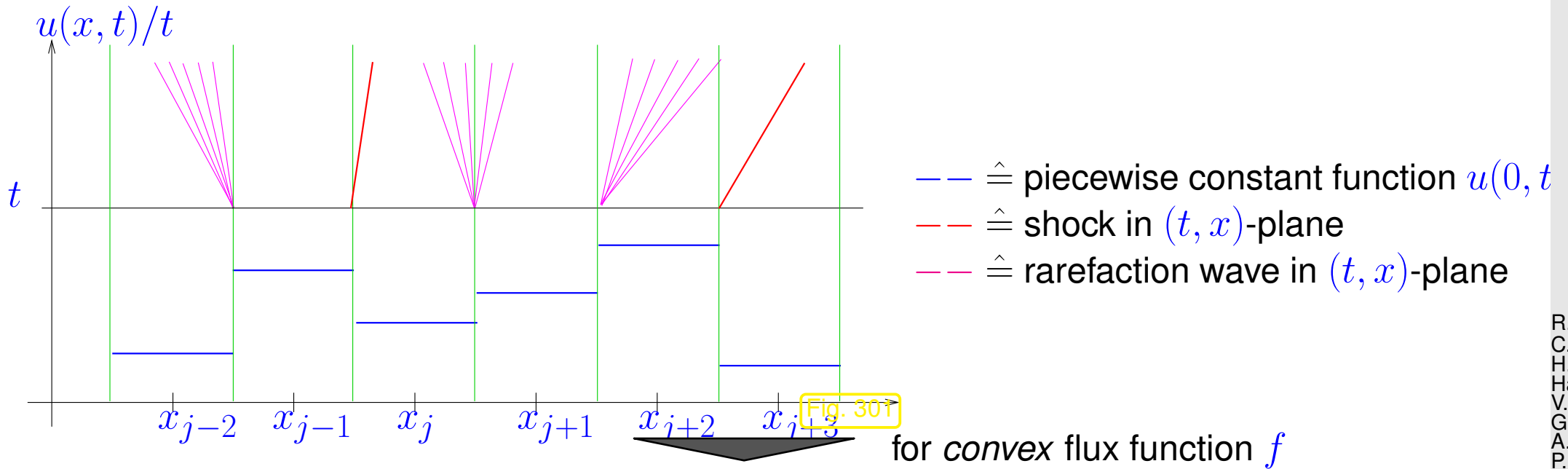
➤ All weak solutions of a Riemann problem are of the form $u(x, t) = \psi(x/t)$ with a suitable function ψ , which is

- piecewise constant with a jump at $\dot{s} := \frac{f(w) - f(v)}{w - v}$ for a shock solution (8.3.48),

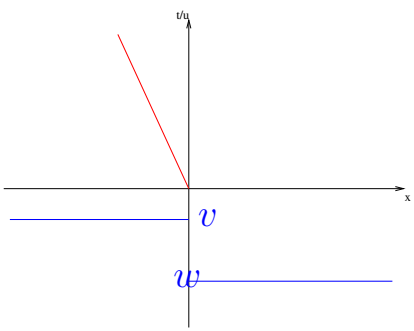
- the continuous function (in the case of strictly convex flux function f)

$$\psi(\xi) := \begin{cases} v & , \text{ if } \xi < f'(v), \\ (f')^{-1}(\xi) & , \text{ if } f'(v) < \xi < f'(w), \\ w & , \text{ if } \xi > f'(w), \end{cases}$$

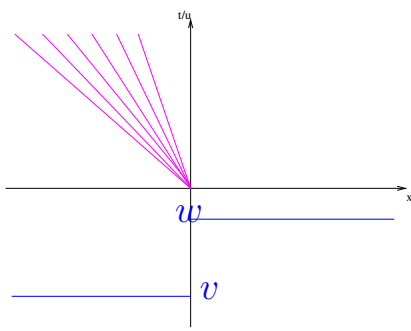
provided that $w > v$ = situation of a rarefaction solution (8.3.49), see Lemma 8.2.38.



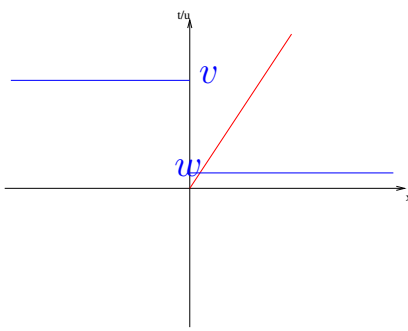
$$u^\downarrow(v, w) = \begin{cases} w & , \text{ if } v > w \wedge \dot{s} < 0 \text{ (shock 1) ,} \\ & v < w \wedge f'(w) < 0 \text{ (rarefaction 2) ,} \\ v & , \text{ if } v > w \wedge \dot{s} > 0 \text{ (shock 3) ,} \\ & v < w \wedge f'(v) > 0 \text{ (rarefaction 4) ,} \\ (f')^{-1}(0) & , \text{ if } v < w \wedge f'(v) \leq 0 \leq f'(w) \text{ (rarefaction 5).} \end{cases} \quad (8.3.50)$$



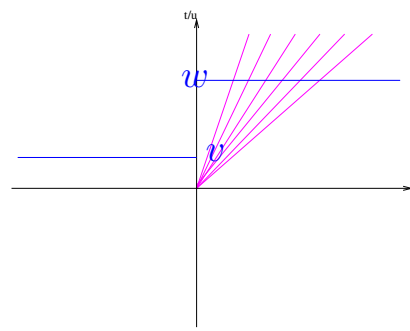
①: subsonic shock



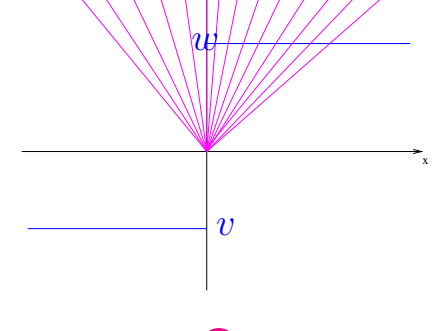
②: subsonic rarefaction



③: supersonic shock



④: supersonic rarefaction



⑤: transonic rarefaction

Detailed analysis of (8.3.50):

$$v > w \quad (\text{shock case}): \quad f(u^\downarrow(v, w)) = \begin{cases} f(v) & , \text{ if } \frac{f(w) - f(v)}{w - v} > 0 \Leftrightarrow f(w) < f(v) , \\ f(w) & , \text{ if } \frac{f(w) - f(v)}{w - v} \leq 0 \Leftrightarrow f(w) \geq f(v) . \end{cases}$$

$$\blacktriangleright \quad f(u^\downarrow(v, w)) = \max\{f(v), f(w)\} .$$

For a convex flux function f :

$$v < w \Rightarrow f'(v) \leq \frac{f(w) - f(v)}{w - v} \leq f'(w).$$

► For $v < w$ (rarefaction case)

$$f(u^\downarrow(v, w)) = \begin{cases} f(v) & , \text{ if } f'(v) > 0, \\ f(z) & , \text{ if } f'(v) < 0 < f'(w), \\ f(w) & , \text{ if } f'(w) < 0, \end{cases}$$

where $f'(z) = 0 \Leftrightarrow f$ has a global minimum in z .

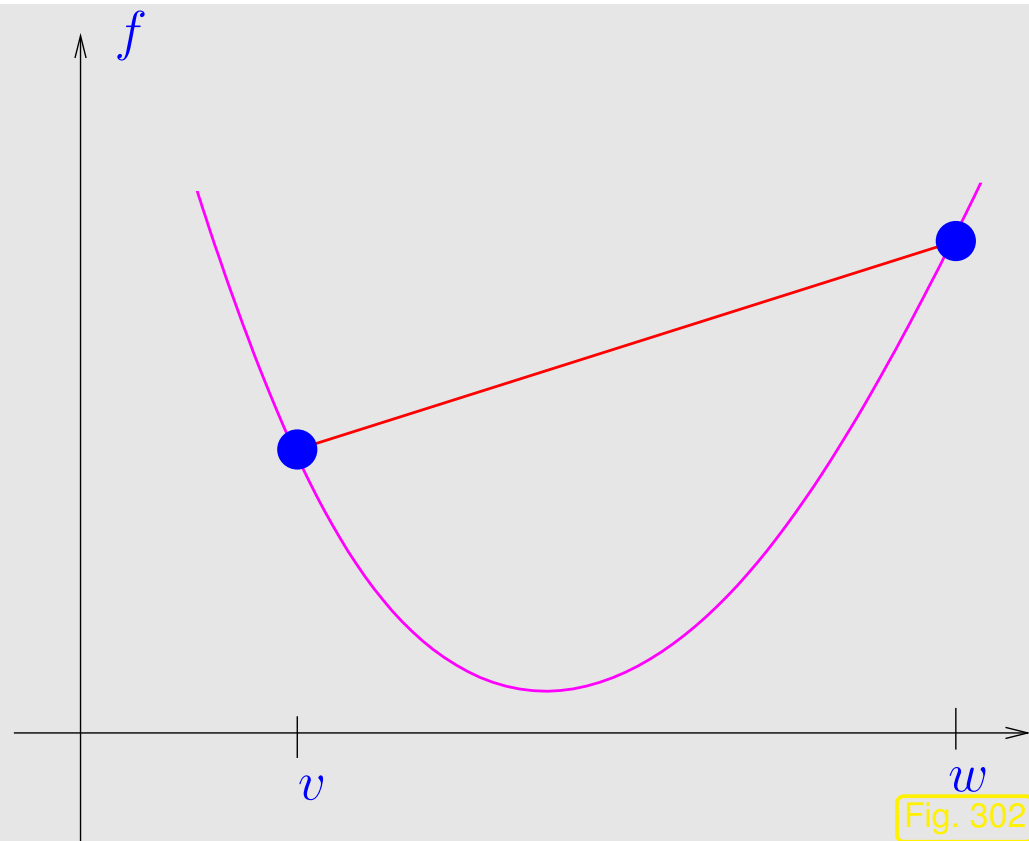


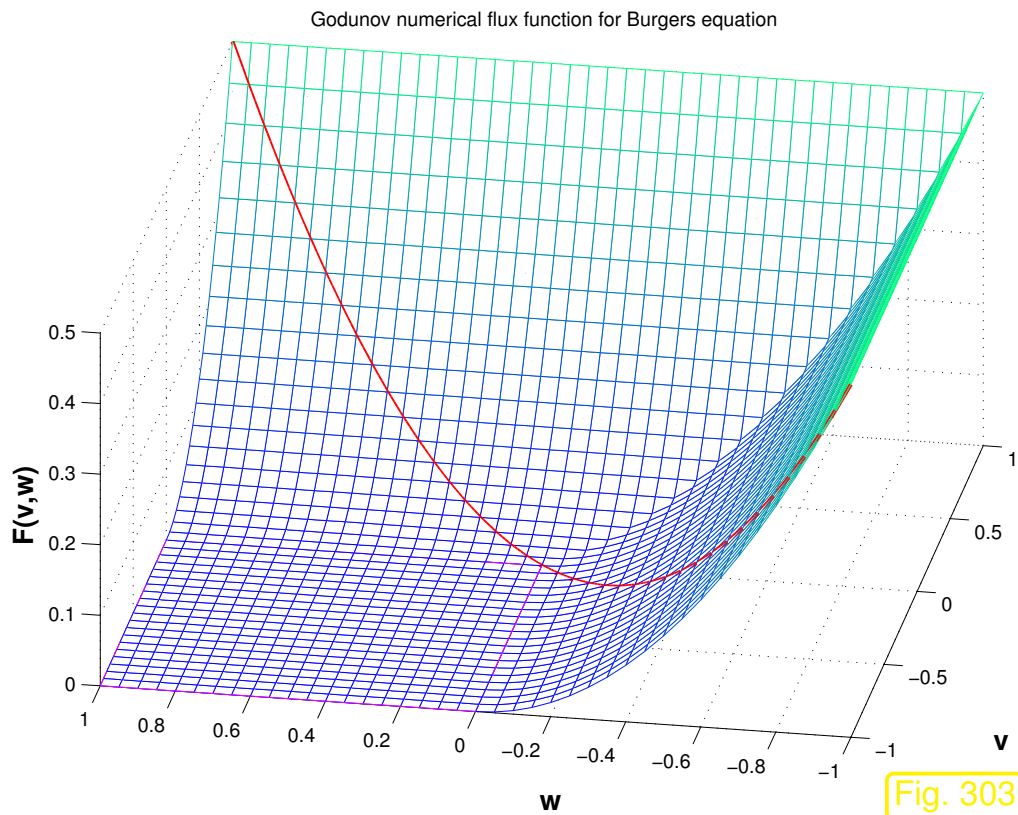
Fig. 302

2-point numerical flux function according to (8.3.46) and (8.3.47): **Godunov numerical flux**

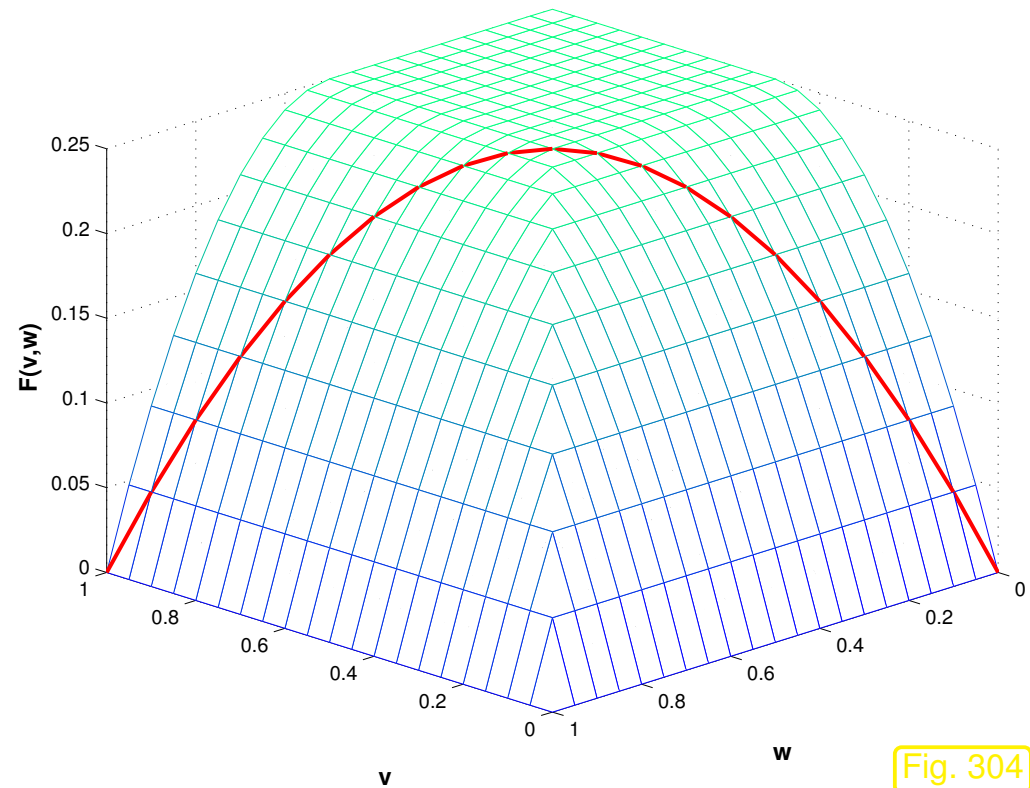
Using general Riemann solution (8.2.41) we get for **any** flux function:

► Godunov numerical flux function

$$F_{\text{GD}}(v, w) = \begin{cases} \min_{v \leq u \leq w} f(u) & , \text{ if } v < w, \\ \max_{w \leq u \leq v} f(u) & , \text{ if } w \leq v. \end{cases} \tag{8.3.51}$$



for Burgers' equation (8.1.60)



for traffic flow equation (8.1.53)

Remark 8.3.52 (Upwind flux and expansion shocks).

$$F_{\text{UW}}(v, w) = F_{\text{GD}}(v, w), \text{ except for the case of } \textit{transsonic rarefaction}!$$

(transsonic rarefaction = rarefaction fan with edges moving in opposite direction, see Ex. 8.3.37)

What does the upwind flux $F_{\text{uw}}(v, w)$ from (8.3.33) yield in the case of transsonic rarefaction?

If f convex, $v < w$, $f'(v) < 0 < f'(w)$,

$$\blacktriangleright F_{\text{uw}}(v, w) = f(\psi(0)) ,$$

where $u(x, t) = \psi(x/t)$ is a non-physical *entropy-condition violating* (\rightarrow Def. 8.2.39) expansion shock weak solution of (8.3.47).

Upwind flux treats transsonic rarefaction as expansion shock!

➤ Explanation for observation made in Ex. 8.3.37.



Example 8.3.54 (Godunov flux for Burgers equation).

- ☞ same setting and conservative discretization as in Ex. 8.3.37
- ☞ Numerical flux function: Godunov numerical flux (8.3.51)

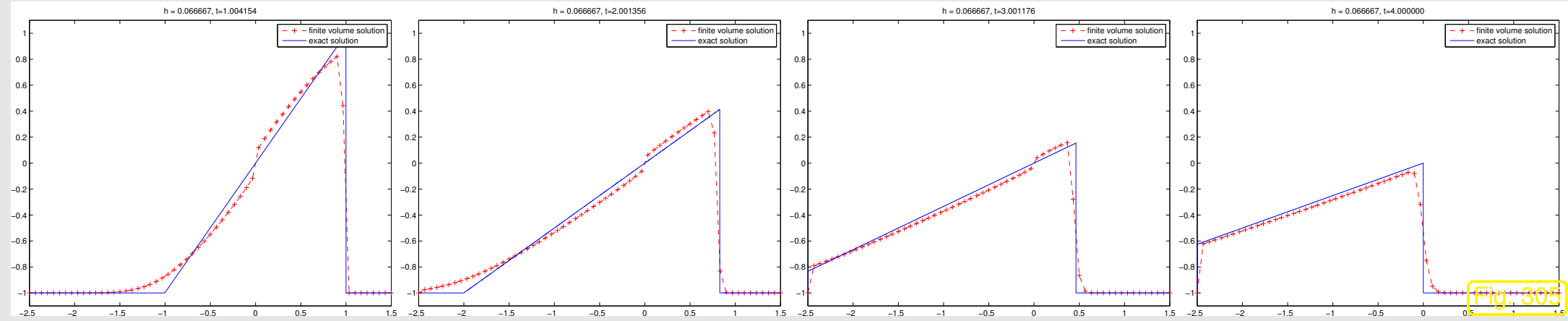
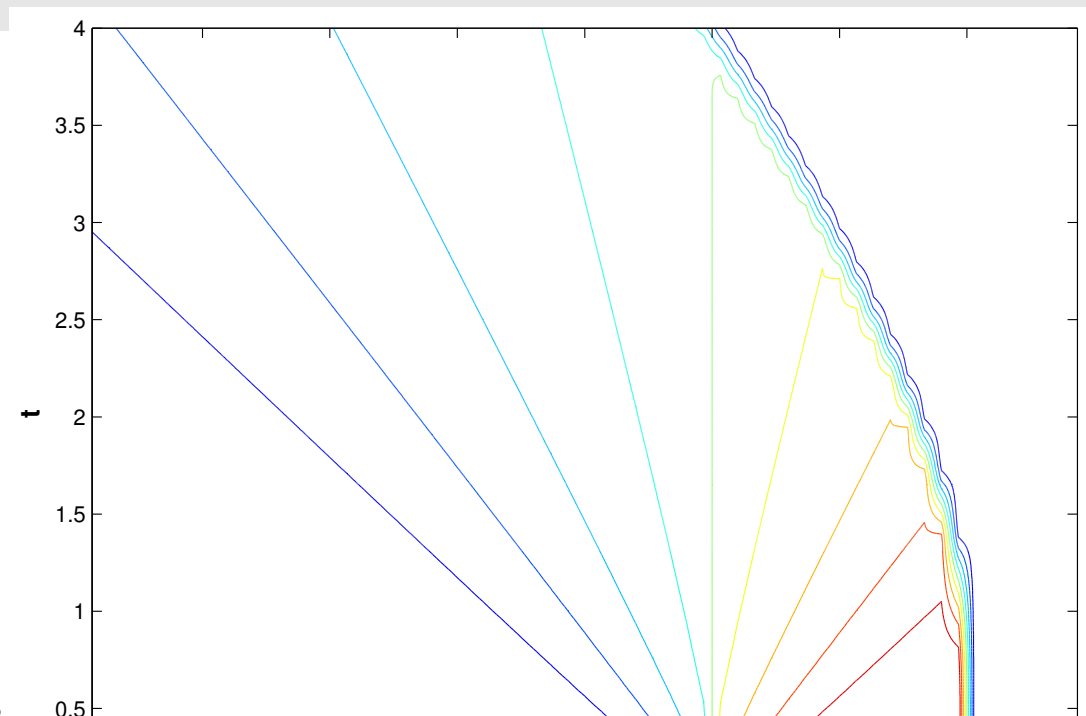
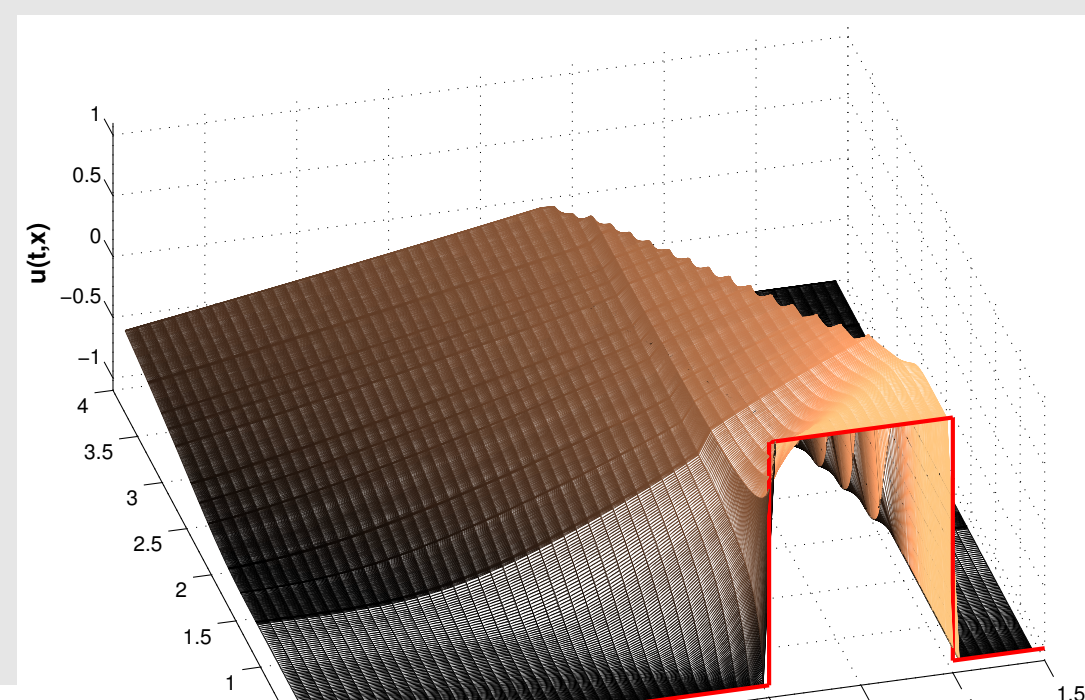


Fig. 305



R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ



Observation: Transonic rarefaction captured by discretization, but small remnants of an expansion shock still observed.

Example 8.3.55 (Godunov flux for traffic flow equation).

☞ same setting and conservative discretization as in Ex. 8.3.37

☞ Numerical flux function: Godunov numerical flux (8.3.51)

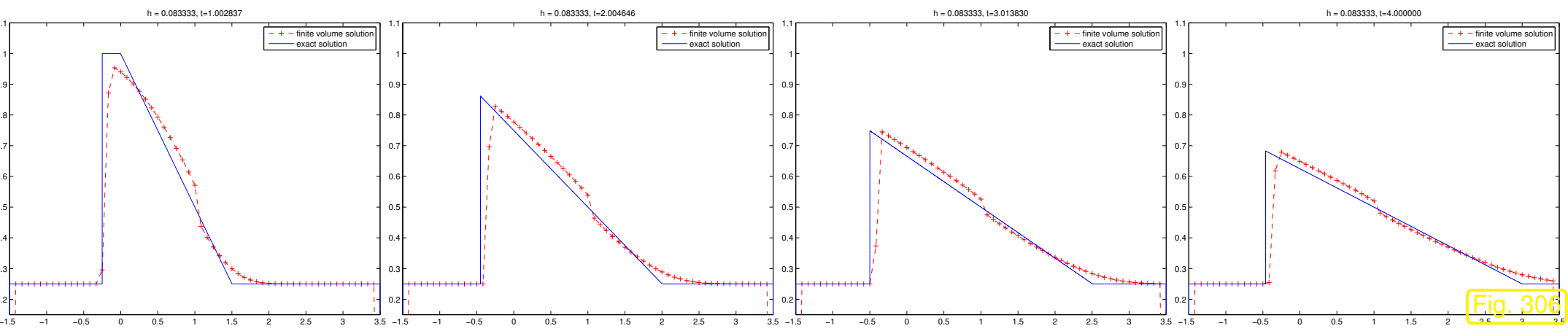
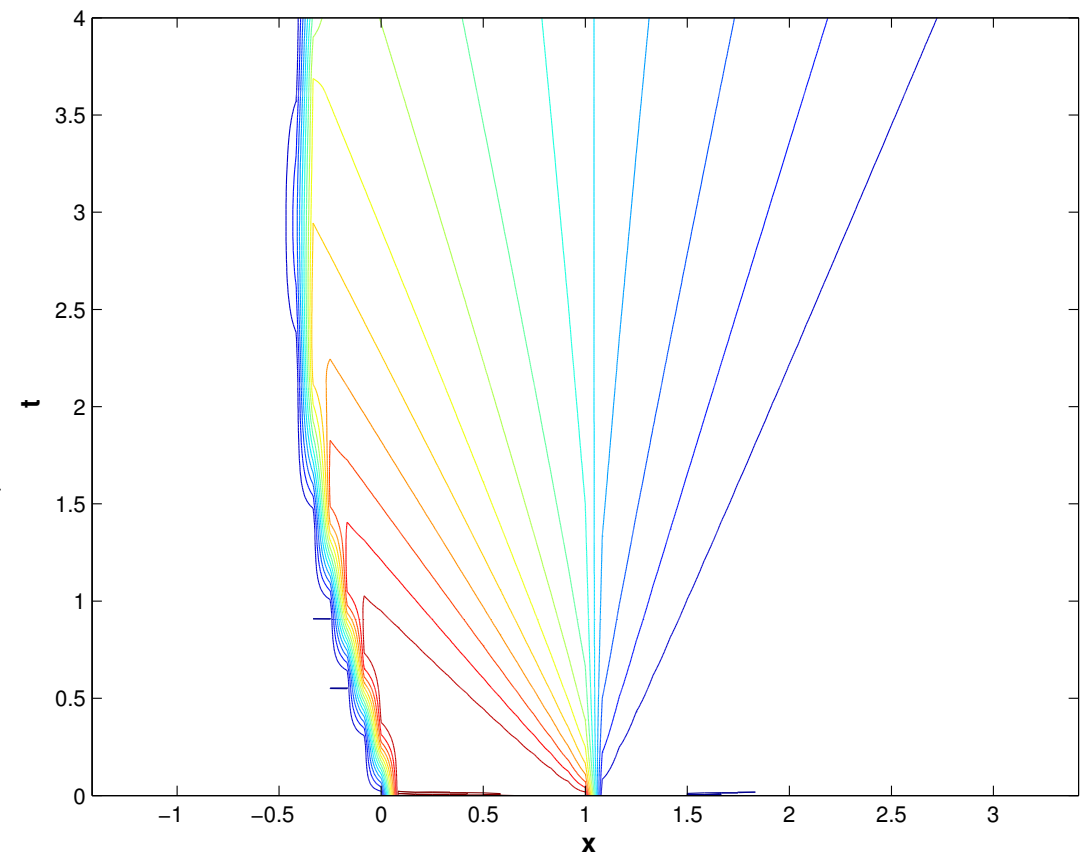
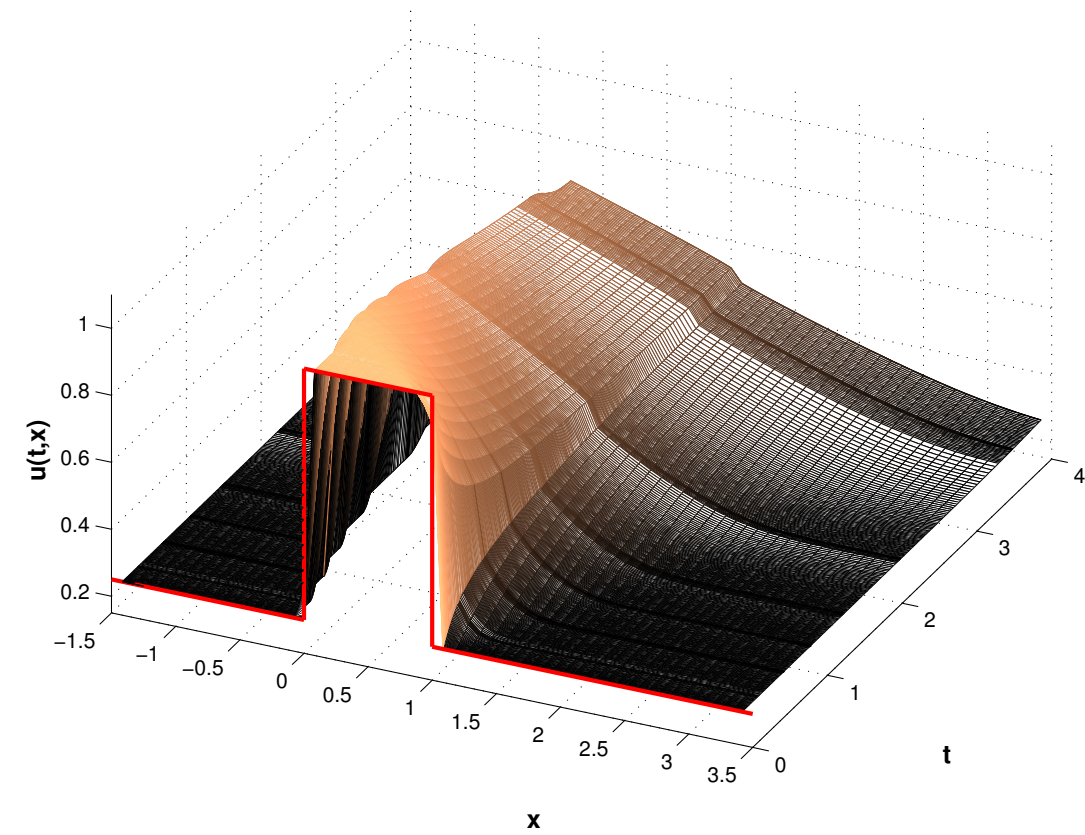


Fig. 306



Observation: Transonic rarefaction captured by discretization, but small remnants of an expansion shock still observed.



8.3.4 Montone schemes

Observations made for some piecewise constant solutions $u_N(t)$ of semi-discrete evolutions arising from spatial finite volume discretization in conservation form (8.3.12):

- Ex. 8.3.31 (Lax-Friedrichs numerical flux (8.3.29)) • $\min_{x \in \mathbb{R}} u_0(x) \leq u_N(x, t) \leq \max_{x \in \mathbb{R}} u_0(x)$
 Ex. 8.3.54 (Godunov numerical flux (8.3.51)) • *no new* local extrema in numerical solution

In these respects the conservative finite volume discretizations based on either the Lax-Friedrichs numerical flux or the Godunov numerical flux inherit crucial structural properties of the exact solution, see Sect. 8.2.7, in particular, Thm. 8.2.45 and the final remark: they display **structure preservation**, *cf.* (5.7).

Is this coincidence for the special settings examined in Ex. 8.3.31 and Ex. 8.3.54?

Focus: semi-discrete evolution (8.3.12) resulting from finite volume discretization in conservation form on an equidistant infinite mesh

$$(8.3.11) \quad \blacktriangleright \quad \frac{d\mu_j}{dt}(t) = -\frac{1}{h} \left(F(\mu_j(t), \mu_{j+1}(t)) - F(\mu_{j-1}(t), \mu_j(t)) \right), \quad j \in \mathbb{Z}, \quad (8.3.12)$$

for Cauchy problem

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times]0, T[\quad , \quad u(x, 0) = u_0(x), \quad x \in \mathbb{R}, \quad (8.2.9)$$

induced by Lax-Friedrichs numerical flux (8.3.29)

$$F_{\text{LF}}(v, w) = \frac{1}{2}(f(v) + f(w)) - \frac{1}{2} \max_{\min\{v, w\} \leq u \leq \max\{v, w\}} |f'(u)|(w - v). \quad (8.3.29)$$

$$\blacktriangleright \quad \frac{d\mu_j}{dt} = -\frac{1}{2h} \left(f(\mu_{j+1}) - f(\mu_{j-1}) - \max_{u \in [\mu_j, \mu_{j+1}]} |f'(u)|(\mu_{j+1} - \mu_j) + \max_{u \in [\mu_{j-1}, \mu_j]} |f'(u)|(\mu_j - \mu_{j-1}) \right). \quad (8.3.56)$$

Goal: show that $u_N(t)$ linked to $\vec{\mu}(t)$ from (8.3.56) through piecewise constant reconstruction (8.3.6) satisfies

$$\min_{x \in \mathbb{R}} u_N(x, 0) \leq u_N(x, t) \leq \max_{x \in \mathbb{R}} u_N(x, 0) \quad \forall x \in \mathbb{R}, \quad \forall t \in [0, T]. \quad (8.3.57)$$

Recall from Sect. 8.2.7: estimate (8.3.57) for the exact solution $u(x, t)$ of (8.2.9) is a consequence of the comparison principle of Thm. 8.2.45 and the fact that constant initial data are preserved during the evolution. The latter property is straightforward for conservative finite volume spatial semi-discretization, see (8.3.15).

➤ Goal: Establish comparison principle for finite volume semi-discrete solutions based on Lax-Friedrichs numerical flux:

$$\left\{ \begin{array}{l} \vec{\mu}(t), \vec{\eta}(t) \text{ solve (8.3.56) ,} \\ \eta_j(0) \leq \mu_j(0) \quad \forall j \in \mathbb{Z} \end{array} \right\} \Rightarrow \eta_j(t) \leq \mu_j(t) \quad \forall j \in \mathbb{Z}, \quad \forall 0 \leq t \leq T.$$

Assumption: $\vec{\mu} = \vec{\mu}(t)$ and $\vec{\eta} = \vec{\eta}(t)$ solve (8.3.56) and satisfy for some $t \in [0, T]$

$$\eta_k(t) \leq \mu_k(t) \quad \forall k \in \mathbb{Z}, \quad \xi := \eta_j(t) = \mu_j(t) \quad \text{for some } j \in \mathbb{Z}.$$

Can η_j raise above μ_j ?

$$\frac{d}{dt}(\mu_j - \eta_j) = -\frac{1}{h} \left(F_{\text{LF}}(\xi, \mu_{j+1}) - F_{\text{LF}}(\xi, \eta_{j+1}) + F_{\text{LF}}(\eta_{j-1}, \xi) - F_{\text{LF}}(\mu_{j-1}, \xi) \right).$$

To show: $\frac{d}{dt}(\mu_j - \eta_j) \geq 0 \quad \Rightarrow \quad \mu_j(t)$ will stay above $\eta_j(t)$.

This can be concluded, if

$$F_{\text{LF}}(\xi, \mu_{j+1}) - F_{\text{LF}}(\xi, \eta_{j+1}) \leq 0 \quad \text{and} \quad F_{\text{LF}}(\eta_{j-1}, \xi) - F_{\text{LF}}(\mu_{j-1}, \xi) \leq 0. \quad (8.3.58)$$

The only piece of information we are allowed to use is

$$\mu_{j+1} \geq \eta_{j+1} \quad \text{and} \quad \mu_{j-1} \geq \eta_{j-1}.$$

This would imply (8.3.58), if F_{LF} was increasing in the first argument and decreasing in the second argument.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Definition 8.3.59 (Monotone numerical flux function).

A 2-point numerical flux function $F = F(v, w)$ is called *monotone*, if

F is an *increasing* function of its *first* argument

and

F is a *decreasing* function of its *second* argument.

Simple criterion: A continuously differentiable 2-point numerical flux function $F = F(v, w)$ is monotone, if and only if

$$\frac{\partial F}{\partial v}(v, w) \geq 0 \quad \text{and} \quad \frac{\partial F}{\partial w}(v, w) \leq 0 \quad \forall (v, w). \quad (8.3.60)$$

Lemma 8.3.61 (Monotonicity of Lax-Friedrichs numerical flux and Godunov flux).

For any continuously differentiable flux function f the associated Lax-Friedrichs flux (8.3.29) and Godunov flux (8.3.51) are monotone.

Proof.

① Lax-Friedrichs numerical flux:

$$F_{\text{LF}}(v, w) = \frac{1}{2}(f(v) + f(w)) - \frac{1}{2} \max_{\min\{v, w\} \leq u \leq \max\{v, w\}} |f'(u)|(w - v). \quad (8.3.29)$$

Application of the criterion (8.3.60) is straightforward:

$$\frac{\partial F_{\text{LF}}}{\partial v}(v, w) = f'(v) + \max_{\min\{v, w\} \leq u \leq \max\{v, w\}} |f'(u)| \geq 0,$$

$$\frac{\partial F_{\text{LF}}}{\partial w}(v, w) = f'(w) - \max_{\min\{v, w\} \leq u \leq \max\{v, w\}} |f'(u)| \leq 0 .$$

② Godunov numerical flux

$$F_{\text{GD}}(v, w) = \begin{cases} \min_{v \leq u \leq w} f(u) & , \text{ if } v < w , \\ \max_{w \leq u \leq v} f(u) & , \text{ if } w \leq v . \end{cases} \quad (8.3.51)$$

$v < w$: If v increases, then the range of values over which the minimum is taken will shrink, which makes $F_{\text{GD}}(v, w)$ increase.

If w is raised, then the minimum is taken over a larger interval, which causes $F_{\text{GD}}(v, w)$ to become smaller.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

$v \geq w$: If v increases, then the range of values over which the maximum is taken will grow, which makes $F_{\text{GD}}(v, w)$ increase.

If w is raised, then the maximum is taken over a smaller interval, which causes $F_{\text{GD}}(v, w)$ to decrease. \square

Lemma 8.3.62 (Comparison principle for monotone semi-discrete conservative evolutions).

Let the 2-point numerical flux function $F = F(v, w)$ be monotone (\rightarrow Def. 8.3.59) and $\vec{\mu} = \vec{\mu}(t)$, $\vec{\eta} = \vec{\eta}(t)$ solve (8.3.12). Then

$$\eta_k(0) \leq \mu_k(0) \quad \forall k \in \mathbb{Z} \quad \Rightarrow \quad \eta_k(t) \leq \mu_k(t) \quad \forall k \in \mathbb{Z}, \quad \forall 0 \leq t \leq T.$$

The assertion of Lemma 8.3.62 means that for monotone numerical flux, the semi-discrete evolution satisfies the **comparison principle** of Thm. 8.2.45.

Proof (of Lemma 8.3.62, following the above considerations for the Lax-Friedrichs flux).

The two sequences of nodal values satisfy (8.3.12)

$$\frac{d\mu_j}{dt}(t) = -\frac{1}{h} (F(\mu_j(t), \mu_{j+1}(t)) - F(\mu_{j-1}(t), \mu_j(t))) , \quad j \in \mathbb{Z} , \quad (8.3.63)$$

$$\frac{d\eta_j}{dt}(t) = -\frac{1}{h} (F(\eta_j(t), \eta_{j+1}(t)) - F(\eta_{j-1}(t), \eta_j(t))) , \quad j \in \mathbb{Z} . \quad (8.3.64)$$

Let t_0 be the *earliest* time, at which $\vec{\eta}$ “catches up” with $\vec{\mu}$ in at least one node x_j , $j \in \mathbb{Z}$, of the mesh, that is

$$\eta_k(t_0) \leq \mu_k(t_0) \quad \forall k \in \mathbb{Z} \quad , \quad \xi := \eta_j(t_0) = \mu_j(t_0) .$$

By subtracting (8.3.63) and (8.3.64) we get

$$\frac{d}{dt}(\mu_j - \eta_j)(t_0) = -\frac{1}{h} \left(F(\xi, \mu_{j+1}(t_0)) - F(\xi, \eta_{j+1}(t_0)) + F(\eta_{j-1}(t_0), \xi) - F(\mu_{j-1}(t_0), \xi) \right) \geq 0 ,$$

because for a *monotone* numerical flux function (\rightarrow Def. 8.3.59)

$$\begin{array}{ll} \eta_{j-1}(t_0) \leq \mu_{j-1}(t_0) & \begin{array}{l} \text{increasing in first argument} \\ \Rightarrow \end{array} & F(\eta_{j-1}(t_0), \xi) - F(\mu_{j-1}(t_0), \xi) \leq 0 , \\ \eta_{j+1}(t_0) \leq \mu_{j+1}(t_0) & \begin{array}{l} \text{decreasing in second argument} \\ \Rightarrow \end{array} & F(\xi, \mu_{j+1}(t_0)) - F(\xi, \eta_{j+1}(t_0)) \leq 0 . \end{array}$$

This means that “ η_j cannot overtake μ_j ”: no value η_j can ever raise above μ_j . □

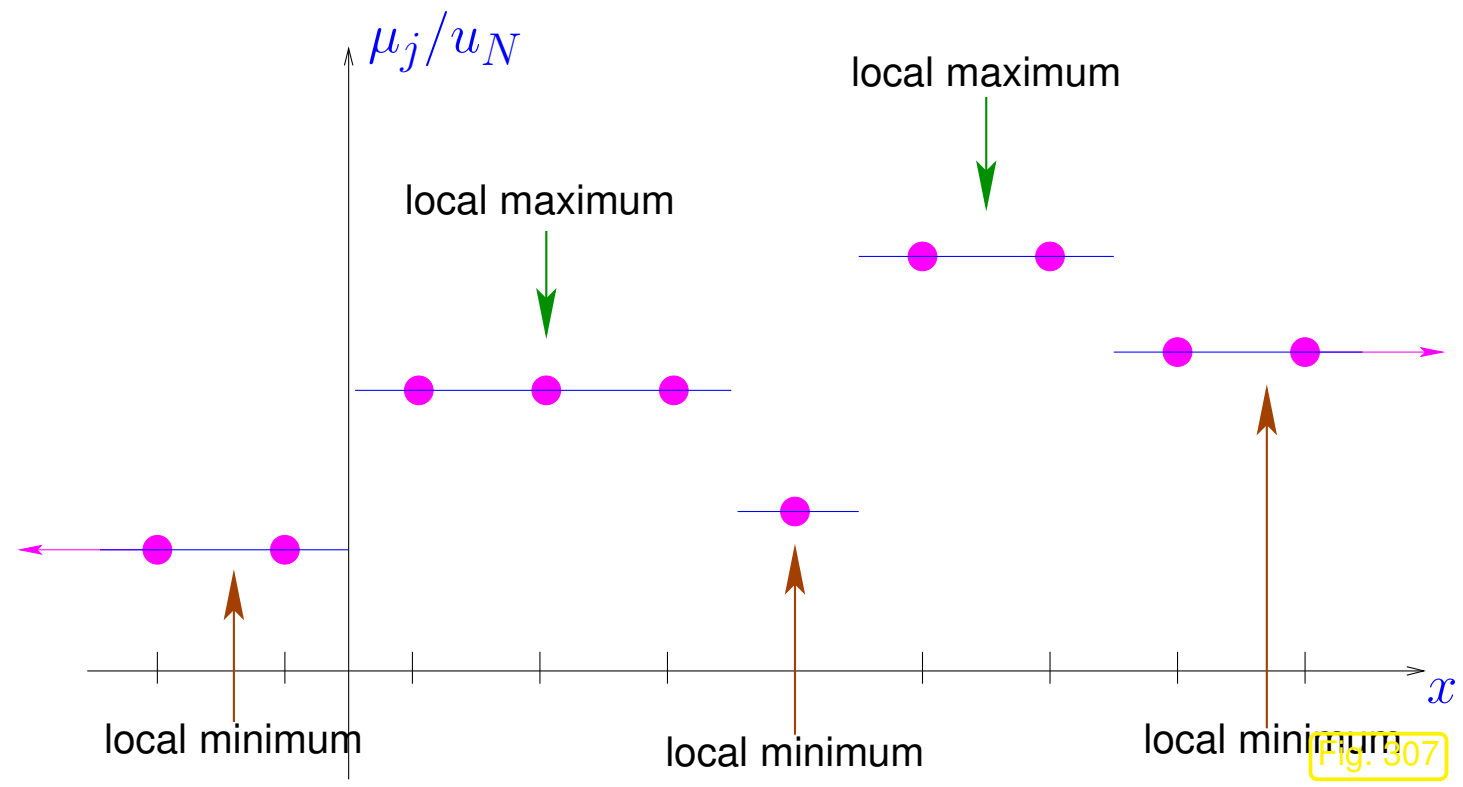
Now we want to study the “preservation of the number of local extrema” during a semi-discrete evolution, another *structural property* of exact solutions of conservations laws, see Sect. 8.2.7.

Intuitive terminology: $\vec{\mu}$ has a **local maximum** $u_m \in \mathbb{R}$, if

$$\exists j \in \mathbb{Z}: \mu_j = u_m \quad \text{and} \quad \exists k_l < j < k_r \in \mathbb{N}: \max_{k_l < l < k_r} \mu_l = u_m \quad \text{and} \quad \mu_{k_l} < u_m, \mu_{k_r} < u_m.$$

In analogous fashion, we define a local minimum. If $\vec{\mu}$ is constant for large indices, these values are also regarded as local extrema.

Counting local extrema of $\vec{\mu}$ and the associated piecewise constant reconstruction.



Lemma 8.3.65 (Non-oscillatory monotone semi-discrete evolutions).

If $\vec{\mu} = \vec{\mu}(t)$ solves (8.3.12) with a **monotone** numerical flux function $F = F(v, w)$ and $\vec{\mu}(0)$ has finitely many local extrema, then the number of local extrema of $\vec{\mu}(t)$ cannot be larger than that of $\vec{\mu}(0)$.

Proof. $i \hat{=}$ index of local maximum of $\vec{\mu}(t)$, t fixed

$$\begin{array}{l} \mu_{i-1}(t) \leq \mu_i(t) \\ \mu_{i+1}(t) \leq \mu_i(t) \end{array} \xrightarrow{\text{monotone flux}} F(\mu_i, \mu_{i+1}) \geq F(\mu_i, \mu_i) \geq F(\mu_{i-1}, \mu_i) ,$$

$$\Rightarrow \frac{d}{dt} \mu_i(t) = -\frac{1}{h} (F(\mu_i, \mu_{i+1}) - F(\mu_{i-1}, \mu_i)) \leq 0 .$$

➤ maxima of $\vec{\mu}$ subside, (minima of $\vec{\mu}$ rise !)

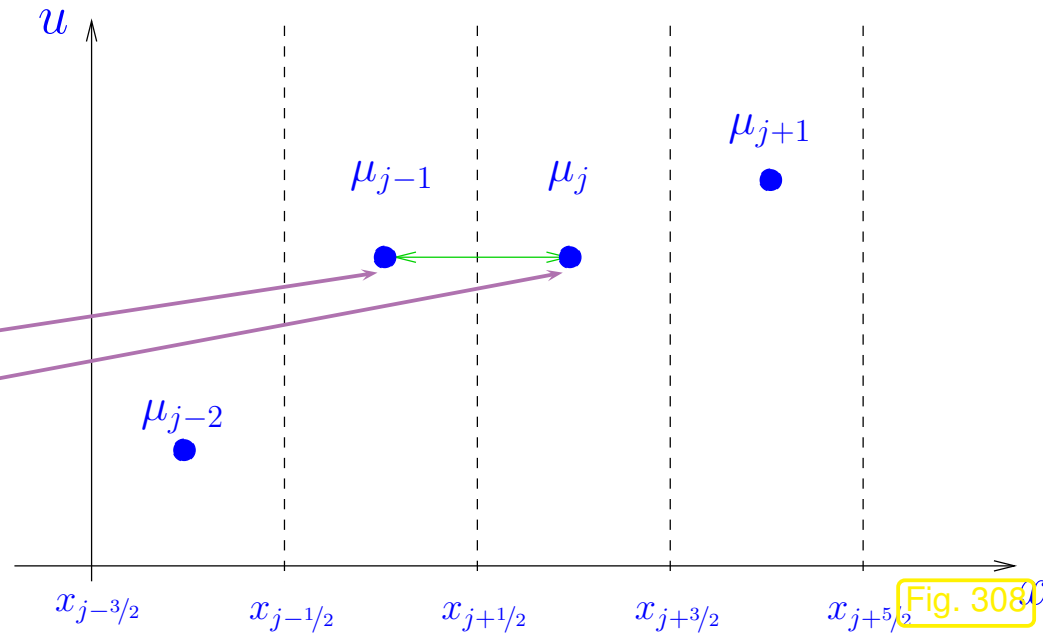
Idea of proof:

No new (local) extrema can arise !

Adjacent values cannot “overtake”:

local maximum: cannot move up

local minimum: cannot move down



8.4 Timestepping

Focus: *Explicit* Runge-Kutta timestepping methods (→ Def. 6.1.33)

Recall [21, Def. 12.4.8]: for explicit s -stage Runge-Kutta single step methods the coefficients a_{ij} vanish for $j \geq i, 1 \leq i, j \leq s$ ➤ the increments \mathbf{K}_i can be computed in turns (without solving a non-linear system of equations).

Initial value problem for abstract semi-discrete evolution in $\mathbb{R}^{\mathbb{Z}}$:

$$\frac{d\vec{\mu}}{dt}(t) = \mathcal{L}_h(\vec{\mu}(t)) , \quad 0 \leq t \leq T \quad , \quad \vec{\mu}(0) = \vec{\mu}_0 \in \mathbb{R}^{\mathbb{Z}} . \quad (8.4.1)$$

Here: $\mathcal{L}_h : \mathbb{R}^{\mathbb{Z}} \mapsto \mathbb{R}^{\mathbb{Z}} \hat{=}$ (non-linear) **finite difference operator**, e.g. for finite volume semi-discretization in conservation form with 2-point numerical flux:

$$(8.3.12) \quad \triangleright \quad (\mathcal{L}_h \vec{\mu})_j := -\frac{1}{h} (F(\mu_j, \mu_{j+1}) - F(\mu_{j-1}, \mu_j)) . \quad (8.4.2)$$

\mathcal{L}_h is local: $(\mathcal{L}_h(\vec{\mu}))_j$ depends only on “neighboring values” $\mu_{j-n_l}, \dots, \mu_{j+n_r}$.

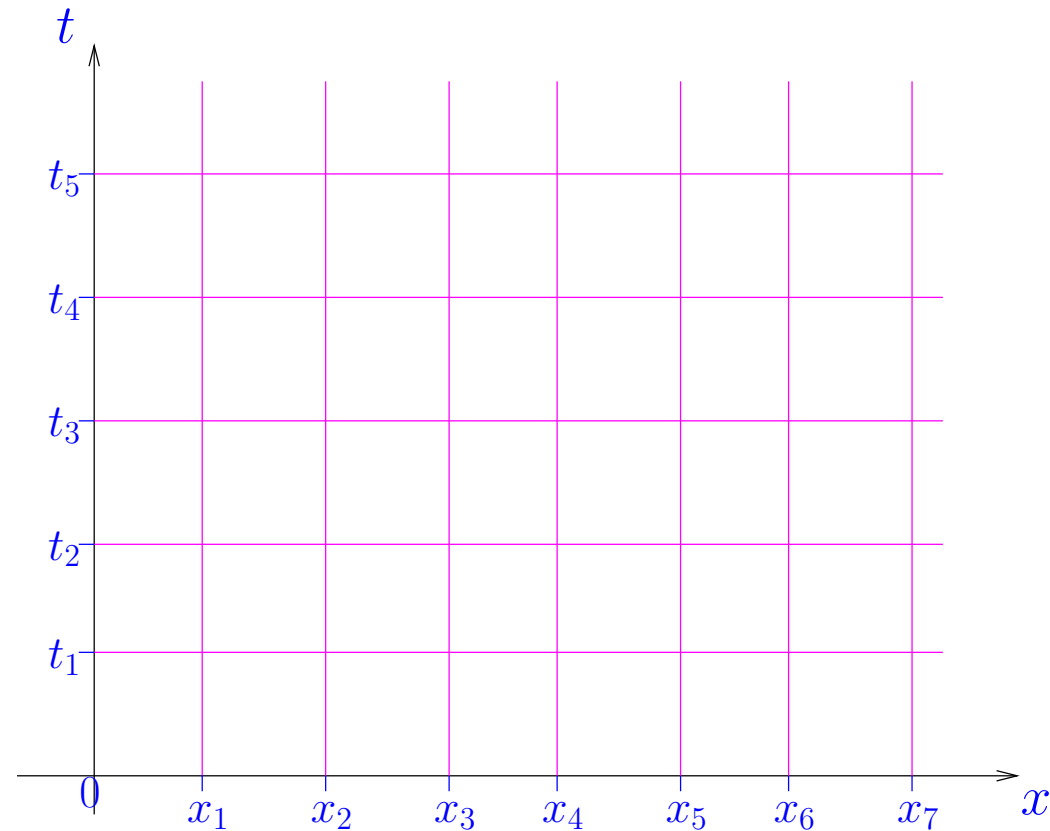
 R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

► Explicit s -stage Runge-Kutta single step method for (8.4.1), timestep $\tau > 0$:

$$\begin{aligned} \vec{\kappa}_1 &= \mathcal{L}_h(\vec{\mu}^{(k)}) , \\ \vec{\kappa}_2 &= \mathcal{L}_h(\vec{\mu}^{(k)} + \tau a_{21} \vec{\kappa}_1) , \\ \vec{\kappa}_3 &= \mathcal{L}_h(\vec{\mu}^{(k)} + \tau a_{31} \vec{\kappa}_1 + \tau a_{32} \vec{\kappa}_2) , \\ &\vdots \\ \vec{\kappa}_s &= \mathcal{L}_h(\vec{\mu}^{(k)} + \tau \sum_{j=1}^{s-1} a_{sj} \vec{\kappa}_j) , \end{aligned} \quad \vec{\mu}^{(k+1)} = \vec{\mu}^{(k)} + \tau \sum_{l=1}^s b_l \vec{\kappa}_l . \quad (8.4.3)$$

Here, $a_{ij} \in \mathbb{R}$ and $b_l \in \mathbb{R}$ are the coefficients from the Butcher scheme (6.1.34). For explicit RK-methods the coefficient matrix \mathcal{A} is strictly lower triangular.



Setting: equidistant spatial mesh \mathcal{M} , meshwidth $h > 0$, nodes $x_j := hj, j \in \mathbb{Z}$,

uniform timestep $\tau > 0, t_k := \tau k, k \in \mathbb{N}_0$.



Single step timestepping for (8.4.1) produces a sequence $(\vec{\mu}^{(k)})_{k \in \mathbb{N}_0}$

$$\mu_j^{(k)} \approx u(x_j, t_k), \quad j \in \mathbb{Z}, k \in \mathbb{N}_0.$$

► Fully discrete evolution

$$\vec{\mu}^{(k+1)} = \mathcal{H}_h(\vec{\mu}^{(k)}), \quad k \in \mathbb{N}_0.$$

$\mathcal{H}_h : \mathbb{R}^Z \mapsto \mathbb{R}^Z$: fully discrete evolution operator, arising from applying single step timestepping (8.4.3) to (8.4.1).

Example 8.4.4 (Fully discrete evolutions).

Fully discrete evolution arising from finite volume semi-discretization in conservation form with 2-point numerical flux $F = F(v, w)$

$$(8.3.12) \quad \triangleright \quad (\mathcal{L}_h \vec{\mu})_j := -\frac{1}{h} (F(\mu_j, \mu_{j+1}) - F(\mu_{j-1}, \mu_j)) . \quad (8.4.2)$$

in combination with *explicit Euler* timestepping ($\hat{=}$ 1-stage explicit RK-method)

$$\vec{\mu}^{(k+1)} = \vec{\mu}^{(k)} + \tau \mathcal{L}_h(\vec{\mu}^{(k)}) .$$

$$\triangleright \quad (\mathcal{H}_h(\vec{\mu}))_j = \mu_j^{(k)} - \frac{\tau}{h} (F(\mu_j^{(k)}, \mu_{j+1}^{(k)}) - F(\mu_{j-1}^{(k)}, \mu_j^{(k)})) . \quad (8.4.5)$$

In the case of *explicit trapezoidal rule* timestepping [21, Eq. 12.4.5] (method of Heun)

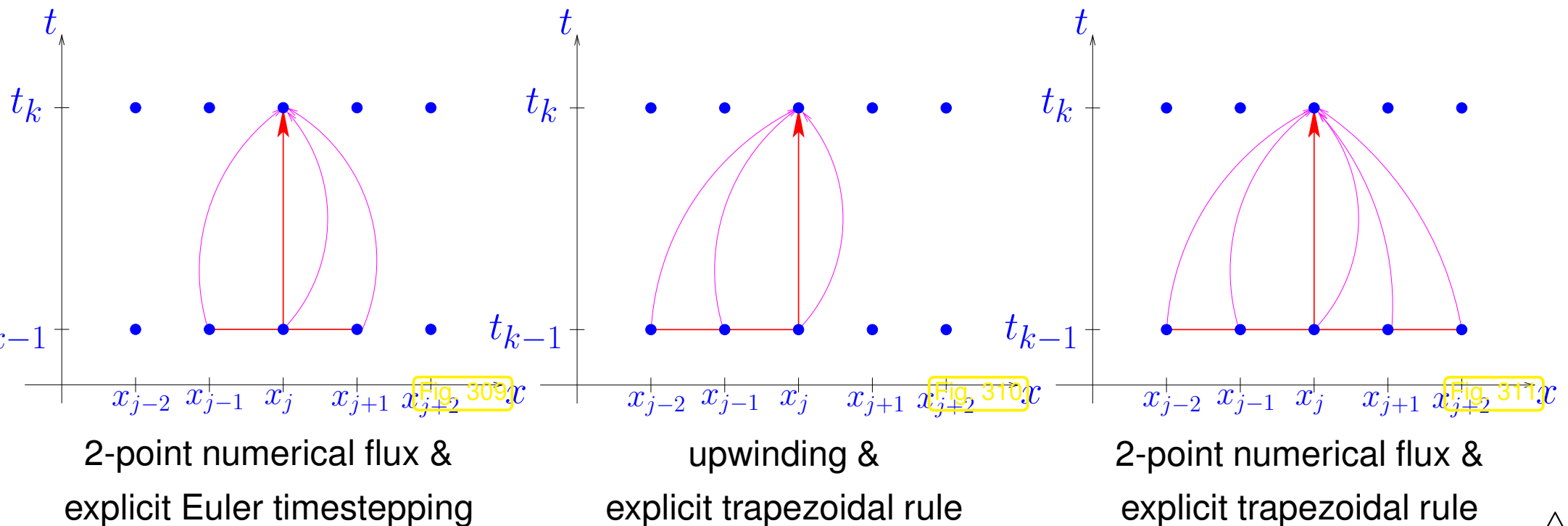
$$\vec{\kappa} = \vec{\mu}^{(k)} + \frac{\tau}{2} \mathcal{L}_h(\vec{\mu}^{(k)}) \quad , \quad \vec{\mu}^{(k+1)} = \vec{\mu}^{(k)} + \tau \mathcal{L}_h(\vec{\kappa}) .$$

$$\triangleright \quad \begin{aligned} \kappa_j &:= (\vec{\kappa})_j = \mu_j^{(k)} - \frac{\tau}{h} (F(\mu_j^{(k)}, \mu_{j+1}^{(k)}) - F(\mu_{j-1}^{(k)}, \mu_j^{(k)})) , \\ (\mathcal{H}_h(\vec{\mu}))_j &= \mu_j^{(k)} - \frac{\tau}{h} (F(\kappa_j, \kappa_{j+1}) - F(\kappa_{j-1}, \kappa_j)) . \end{aligned} \quad (8.4.6)$$

8.4.1 CFL-condition

Remark 8.4.7 (Difference stencils).

Stencil notation: Visualization of flow of information in fully discrete *explicit* evolution (action of \mathcal{H}_h), cf. Fig. 204.



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



A consequence of *explicit* timestepping: **locality** of fully discrete evolution operator:

$$\exists m_l, m_r \in \mathbb{N}_0: \quad (\mathcal{H}(\vec{\mu}))_j = \mathcal{H}_j(\mu_{j-m_l}, \dots, \mu_{j+m_r}) . \quad (8.4.8)$$

If flux function f does not depend on x , $f = f(u)$ as in (8.2.9), we can expect

$$\mathcal{H}_h \text{ is translation-invariant:} \quad \mathcal{H}_j = \mathcal{H} \quad \forall j \in \mathbb{Z} .$$

This is the case for (8.4.5) and (8.4.6).

By inspection of (8.4.3): if \mathcal{L}_h is translation-invariant

$$(\mathcal{L}_h(\vec{\mu}))_j = \mathcal{L}(\mu_{j-n_l}, \dots, \mu_{j+n_r}) , \quad j \in \mathbb{Z} ,$$

and timestepping relies on an s -stage explicit Runge-Kutta method, then we conclude for m_l, m_r in (8.4.8)

$$m_l \leq s \cdot n_l \quad , \quad m_r \leq s \cdot n_r .$$

Now we revisit a concept from Sect. 6.2.5, see, in particular, Rem. 6.2.41:

Definition 8.4.9 (Numerical domain of dependence).

Consider explicit translation-invariant fully discrete evolution $\vec{\mu}^{(k+1)} := \mathcal{H}(\vec{\mu}^{(k)})$ on uniform spatio-temporal mesh $(x_j = hj, j \in \mathbb{Z}, t_k = k\tau, k \in \mathbb{N}_0)$ with

$$\exists m \in \mathbb{N}_0: \quad (\mathcal{H}(\vec{\mu}))_j = \mathcal{H}(\mu_{j-m}, \dots, \mu_{j+m}), \quad j \in \mathbb{Z}. \quad (8.4.10)$$

Then the **numerical domain of dependence** is given by

$$D_h^-(x_j, t_k) := \{(x_m, t_l) \in \mathbb{R} \times [0, t_k]: j - m(k-l) \leq m \leq j + m(k-l)\}.$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

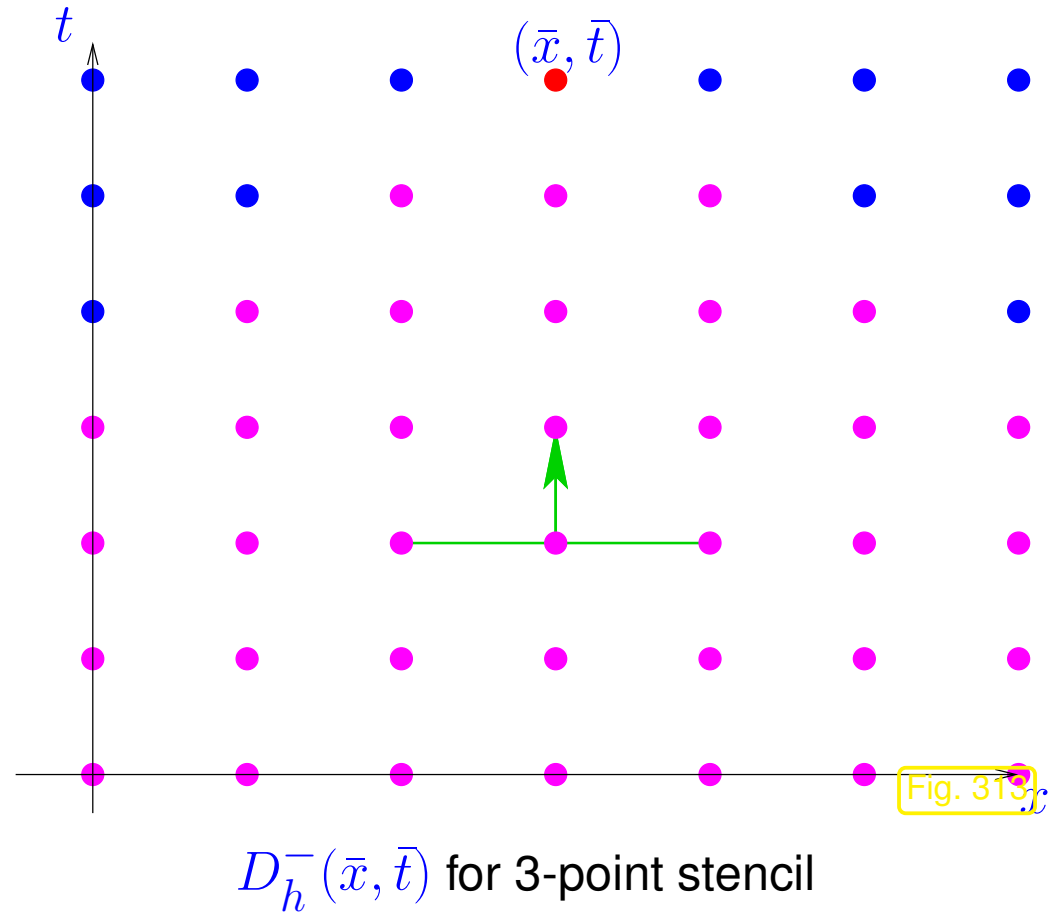
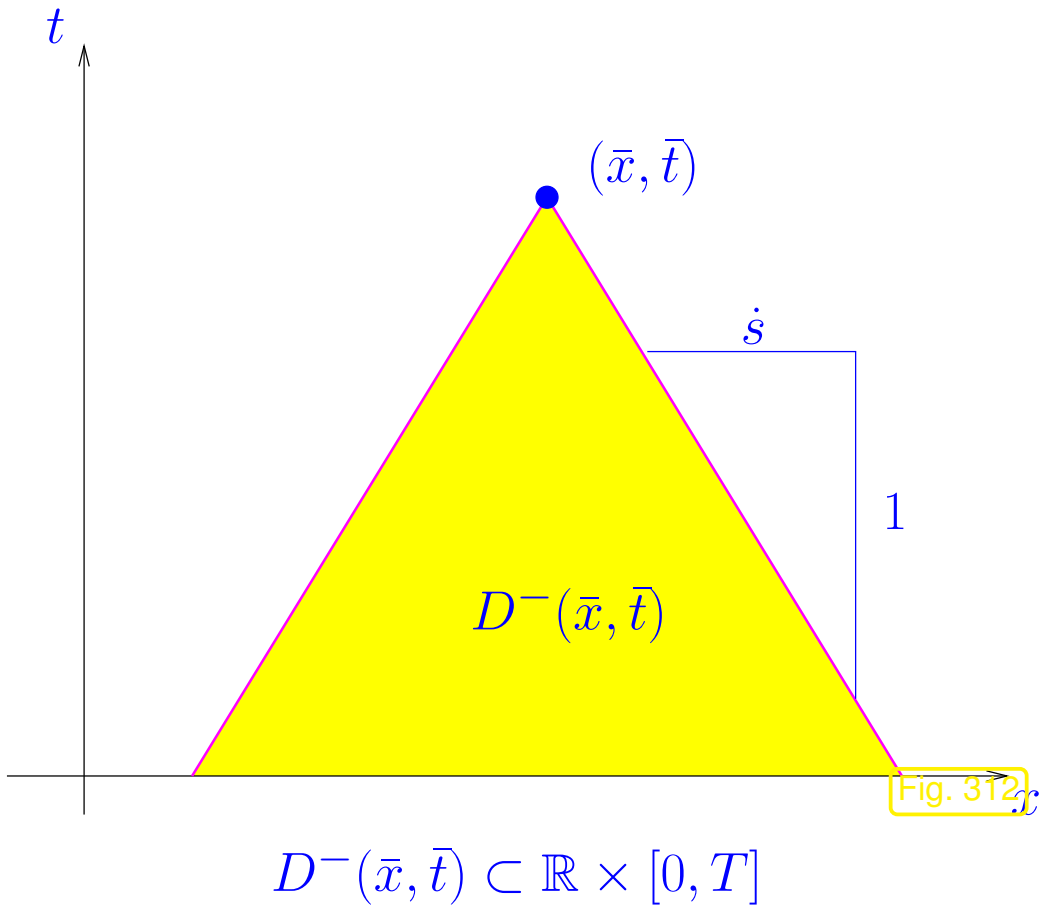
From Thm. 8.2.47 recall the **maximal analytical domain of dependence** for a solution of (8.2.9)

$$D^-(\bar{x}, \bar{t}) := \{(x, t) \in \mathbb{R} \times [0, \bar{t}]: \dot{s}_{\min}(\bar{t} - t) \leq x - \bar{x} \leq \dot{s}_{\max}(\bar{t} - t)\}.$$

with **maximals speeds of propagation**

$$\dot{s}_{\min} := \min\{f'(\xi) : \inf_{x \in \mathbb{R}} u_0(x) \leq \xi \leq \sup_{x \in \mathbb{R}} u_0(x)\}, \quad (8.4.11)$$

$$\dot{s}_{\max} := \max\{f'(\xi) : \inf_{x \in \mathbb{R}} u_0(x) \leq \xi \leq \sup_{x \in \mathbb{R}} u_0(x)\}. \quad (8.4.12)$$



Definition 8.4.13 (Courant-Friedrichs-Lewy (CFL-)condition). \rightarrow *Rem. 6.2.41*

An explicit translation-invariant local fully discrete evolution $\vec{\mu}^{(k+1)} := \mathcal{H}(\vec{\mu}^{(k)})$ on uniform spatio-temporal mesh $(x_j = hj, j \in \mathbb{Z}, t_k = k\tau, k \in \mathbb{N}_0)$ as in Def. 8.4.9 satisfies the **Courant-Friedrichs-Lewy (CFL-)condition**, if the convex hull of its numerical domain of dependence contains the maximal analytical domain of dependence:

$$D^-(x_j, t_k) \subset \text{convex}(D_h^-(x_j, t_k))$$

By definition of $D^-(\bar{x}, \bar{t})$ and $D_h^-(x_j, t_k)$ sufficient for the CFL-condition is

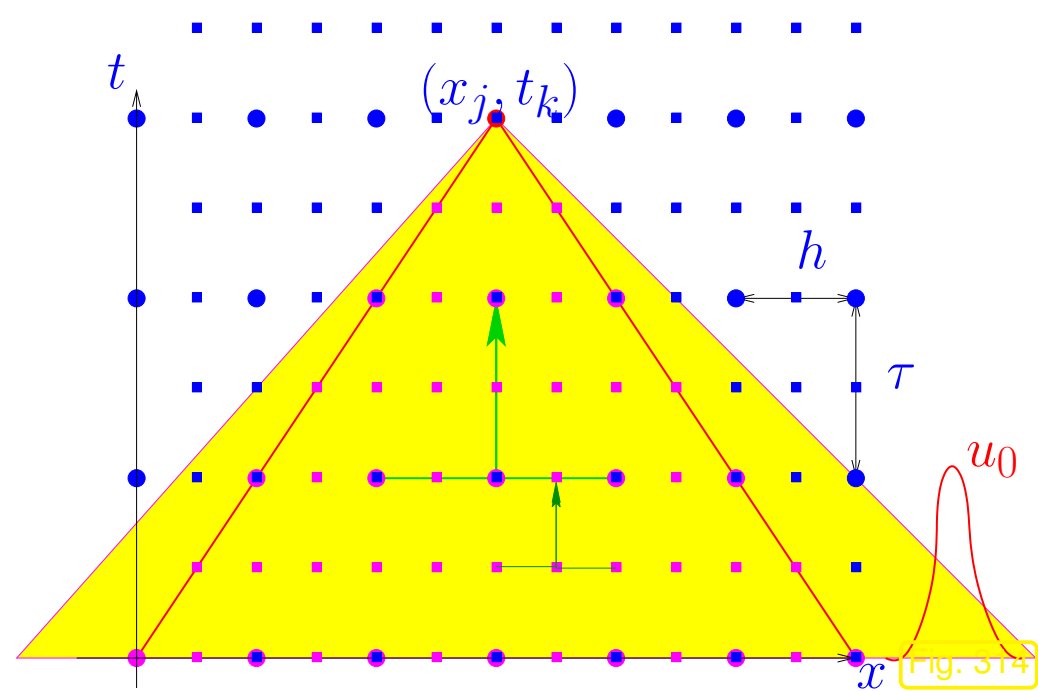
$$\boxed{\frac{\tau}{h} \leq \frac{m}{s}} \iff \text{timestep constraint!} . \quad (8.4.14)$$

This is a timestep constraint similar to the one encountered in Sect. 6.2.5 in the context of leapfrog timestepping for the semi-discrete wave equation.

As discussed in Rem. 6.2.41,

We cannot expect convergence for *fixed ratio* $\tau : h$, for $h \rightarrow 0$ in case the CFL-condition is violated.

Refer to Fig. 207 for a “graphical argument”:



(● $\hat{=}$ coarse grid, ■ $\hat{=}$ fine grid, ■ $\hat{=}$ d.o.d)

◁ Sequence of equidistant space-time grids of $\mathbb{R} \times [0, T]$ with $\tau = \gamma h$ ($\tau/h =$ meshwidth in time/space)

If $\gamma >$ CFL-constraint (8.4.14) then

analytical domain
of dependence

$\not\subset$ numerical domain
of dependence

In Sect. 6.1.4.2 and Sect. 6.2.5 we found that for explicit timestepping

timestep constraints $\tau \leq O(h^r)$, $r \in \{1, 2\}$, *necessary* to avoid exponential blow-up
(*instability*)

Is the timestep constraint (8.4.14) suggested by the CFL-condition also stipulated by stability requirements?

We are going to investigate the question only for the Cauchy problem for scalar *linear* advection in 1D with constant velocity $v > 0$:

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) \quad \forall x \in \mathbb{R}. \quad (8.1.5)$$

Semi-discretization in space on equidistant mesh with meshwidth $h > 0$

➤ *linear, local, and translation-invariant* semi-discrete evolution

$$\frac{d\vec{\mu}}{dt}(t) = \mathcal{L}_h(\vec{\mu}(t)), \quad \text{with} \quad (\mathcal{L}_h(\vec{\mu}))_j = \sum_{l=-m}^m c_l \mu_{j+l}, \quad j \in \mathbb{Z}, \quad (8.4.15)$$

for suitable weights $c_l \in \mathbb{R}$.

Example 8.4.16 (Upwind difference operator for linear advection).

Finite volume semi-discretization of (8.1.5) in conservation form with Godunov numerical flux (8.3.51)
(= upwind flux (8.3.33))

$$(\mathcal{L}_h(\vec{\mu}))_j = -\frac{v}{h}(\mu_j - \mu_{j-1}). \quad (8.4.17)$$

► In (8.4.15): $c_0 = -\frac{v}{h}$, $c_{-1} = \frac{v}{h}$.

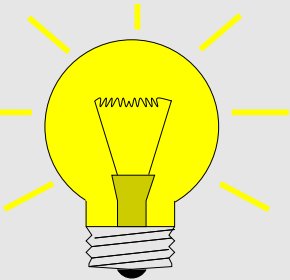
Note: Lax-Friedrichs numerical flux (8.3.29) yields the same \mathcal{L}_h .



As in Sect. 6.1.4.2 and Sect. 6.2.5: **diagonalization technique** (with a new twist)

The new twist is that \mathcal{L}_h acts on the **sequence space** $\mathbb{R}^{\mathbb{Z}}$!

Idea: trial expression for “eigenvectors”



$$\left(\vec{\zeta}^\xi \right)_j := \exp(i\xi j), \quad j \in \mathbb{Z}, \quad -\pi < \xi \leq \pi. \quad (8.4.18)$$

By straightforward computations:

$$(\mathcal{L}_h(\vec{\mu}))_j = \sum_{l=-m}^m c_l \mu_{j+l} \Rightarrow \mathcal{L}_h \zeta^\xi = \underbrace{\left(\sum_{l=-m}^m c_l \exp(i\xi l) \right)}_{\text{“eigenvalue” } \hat{c}_h(\xi)} \zeta^\xi.$$

► spectrum of \mathcal{L}_h : $\sigma(\mathcal{L}_h) = \{ \hat{c}_h(\xi) := \sum_{l=-m}^m c_l \exp(i\xi l) : -\pi < \xi \leq \pi \}.$ (8.4.19)

Terminology: The function $\hat{c}_h(\xi)$ is known as the **symbol** of the difference operator \mathcal{L}_h , cf. the concept of symbol of a differential operator.

Example 8.4.20 (Spectrum of upwind difference operator).

Apply formula (8.4.19) with $c_0 = -\frac{v}{h}$, $c_{-1} = \frac{v}{h}$ (from (8.4.17)):

For \mathcal{L}_h from (8.4.17):
$$\sigma(\mathcal{L}_h) = \left\{ \frac{v}{h}(\exp(-i\xi) - 1) : -\pi < \xi \leq \pi \right\}$$

Spectrum of upwind finite difference operator for linear advection with velocity $v > 0$ (meshwidth $h > 0$) \triangleright

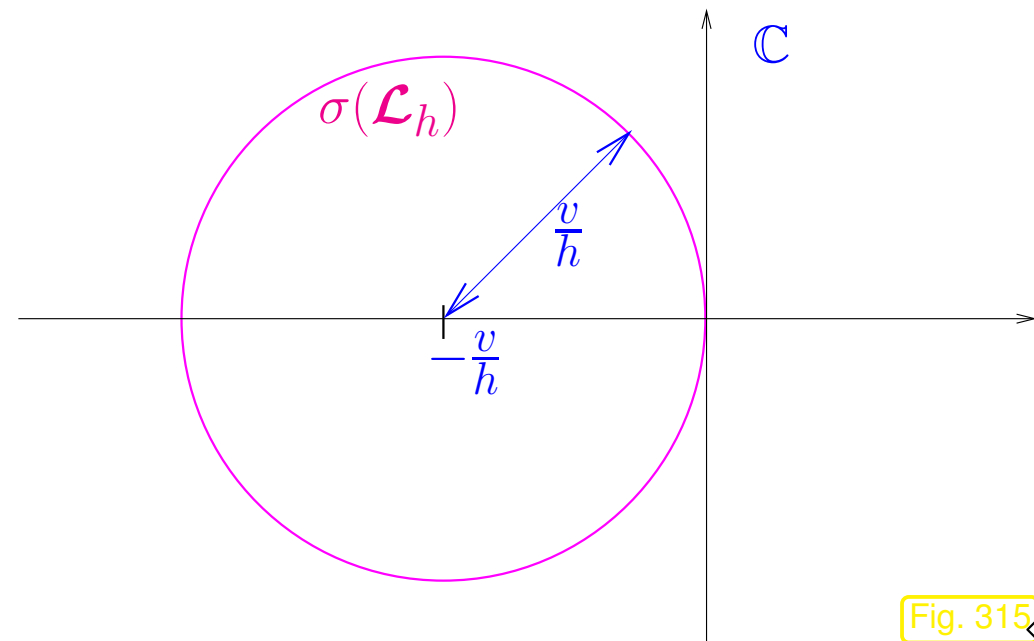


Fig. 315 \diamond

Also here: diagonalization of semi-discrete evolution leads to decoupled scalar linear ODEs. However, now we have uncountably many “eigenvectors” $\vec{\zeta}^\xi$, so that linear combination becomes integration:

$$\vec{\mu}(t) = \int_{-\pi}^{\pi} \hat{\mu}(t, \xi) \vec{\zeta}^\xi d\xi \Leftrightarrow \mu_j(t) = \int_{-\pi}^{\pi} \hat{\mu}(t, \xi) \exp(i\xi j) d\xi . \quad (8.4.21)$$

$$\blacktriangleright \frac{d\vec{\mu}}{dt}(t) = \mathcal{L}_h(\vec{\mu}(t)) \Rightarrow \frac{\partial \hat{\mu}}{\partial t}(t, \xi) = \hat{c}_h(\xi) \hat{\mu}(t, \xi) . \quad (8.4.22)$$

This is a family of scalar, linear ODEs parameterized by $\xi \in] - \pi, \pi]$.

Remark 8.4.23 (Fourier series).

Up to normalization the relationship

$$\vec{\mu}^{(0)} \in \mathbb{R}^{\mathbb{Z}} \quad \leftrightarrow \quad \hat{\mu}^{(0)} :] - \pi, \pi] \mapsto \mathbb{C}$$

from (8.4.21) is the **Fourier series transform**, which maps a sequence to a 2π -periodic function. It has the important isometry property

$$\sum_{j=-\infty}^{\infty} |\mu_j|^2 = 2\pi \int_{-\pi}^{\pi} |\hat{\mu}(\xi)|^2 d\xi .$$

➤ The symbol \hat{c}_h can be viewed as the *representation of a difference operator in Fourier domain*. △

The decoupling manifest in (8.4.22) carries over to Runge-Kutta timestepping in the sense of the commuting diagram (6.1.61).

We introduce the Fourier transforms of the members of the sequence $\left(\vec{\mu}^{(k)}\right)_k$ created by timestepping

$$\vec{\mu}^{(k)} = \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) \vec{\zeta}^{\xi} d\xi \iff \mu_j^{(k)} = \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) \exp(i\xi j) d\xi . \quad (8.4.24)$$

Example 8.4.25 (Explicit Euler in Fourier domain).

Explicit Euler timestepping [21, Eq. 12.2.4] for semi-discrete evolution (8.4.15), see also (8.4.5),

$$\vec{\mu}^{(k+1)} = \vec{\mu}^{(k)} + \tau \mathcal{L}_h \vec{\mu}^{(k)} .$$

$$\blacktriangleright \int_{-\pi}^{\pi} \hat{\mu}^{(k+1)}(\xi) \vec{\zeta}^{\xi} d\xi = (\text{Id} + \tau \mathcal{L}_h) \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) \vec{\zeta}^{\xi} d\xi = \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) (1 + \tau \hat{c}_h(\xi)) d\xi .$$

$$\blacktriangleright \hat{\mu}^{(k+1)}(\xi) = \hat{\mu}^{(k)}(\xi) (1 + \tau \hat{c}_h(\xi)) .$$

In Fourier domain a single explicit Euler timestep corresponds to a multiplication of $\hat{\mu} :] - \pi, \pi] \mapsto \mathbb{C}$ with the function $(1 + \tau \hat{c}_h) :] - \pi, \pi] \mapsto \mathbb{C}$.

Relate this to an explicit Euler step for the ODE $\frac{\partial \hat{\mu}}{\partial t}(t, \xi) = \hat{c}_h(\xi) \hat{\mu}(t, \xi)$ from (8.4.22) with parameter ξ :

$$\hat{\mu}^{(k+1)}(\xi) = (1 + \tau \hat{c}_h(\xi)) \hat{\mu}^{(k)}(\xi) .$$

Generalize the observation made in the previous example:

$$\vec{\mu}^{(k)} = \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) \vec{\zeta}^{\xi} d\xi ,$$

where $(\eta^{(k)}(\xi))_{k \in \mathbb{N}_0}$ is the sequence of approximations created by the Runge-Kutta method when applied to the scalar linear initial value problem

$$\dot{y} = \hat{c}(\xi) y \quad , \quad y(0) = \hat{\mu}^{(0)}(\xi) .$$

Clearly, timestepping can only be stable, if blowup $|\hat{\mu}^{(k)}(\xi)| \rightarrow \infty$ for $k \rightarrow \infty$ can be avoided **for all** $-\pi < \xi \leq \pi$.

From [21, Thm. 13.1.11] we know:

Theorem 8.4.26 (Stability function of explicit Runge-Kutta methods).

The execution of one step of size $\tau > 0$ of an explicit s -stage Runge-Kutta single step method (\rightarrow Def. 6.1.33) with Butcher scheme $\begin{array}{c|c} \mathbf{c} & \mathfrak{A} \\ \hline & \mathbf{b}^T \end{array}$ (see (6.1.34)) for the scalar linear ODE $\dot{y} = \lambda y$, $\lambda \in \mathbb{C}$, amounts to a multiplication with the number

$$\Psi_\lambda^\tau = \underbrace{1 + z\mathbf{b}^T (\mathbf{I} - z\mathfrak{A})^{-1} \mathbf{1}}_{\text{stability function } S(z)} = \det(\mathbf{I} - z\mathfrak{A} + z\mathbf{1}\mathbf{b}^T), \quad z := \lambda\tau, \quad \mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^s.$$

Example 8.4.27 (Stability functions of explicit RK-methods).

- Explicit Euler method (8.4.5) : $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \quad \rightarrow \quad S(z) = 1 + z.$
- Explicit trapezoidal rule (8.4.6) : $\begin{array}{c|cc} 0 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad \rightarrow \quad S(z) = 1 + z + \frac{1}{2}z^2.$

- Classical
Ex. 12.4.10]

RK4-method

[21, :

$$\begin{array}{c|cccc}
 0 & 0 & 0 & 0 & 0 \\
 \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
 \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\
 1 & 0 & 0 & 1 & 0 \\
 \hline
 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6}
 \end{array}$$

$$\triangleright S(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4 .$$


 R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Thm 8.4.26 together with the combinatorial formula for the determinant means that $\Psi_\lambda^\tau(z)$ is a polynomial of degree $\leq s$ in $z \in \mathbb{C}$.

So we conclude for the evolution of “Fourier transforms” $\hat{\mu}^{(k)}(\xi)$:

$$\hat{\mu}^{(k+1)}(\xi) = S(\tau \hat{c}(\xi)) \cdot \hat{\mu}^{(k)}(\xi) , \quad k \in \mathbb{N}_0 , \quad -\pi < \xi \leq \pi ,$$

where $z \mapsto S(z)$ is the **stability function** of the Runge-Kutta timestepping method, see Thm. 8.4.26. For the explicit Euler method we recover the formula of Ex. 8.4.25.

Stability of RK-timestepping of linear semi-discrete evolution $\iff \max_{-\pi < \xi \leq \pi} |S(\tau \hat{c}(\xi))| \leq 1$

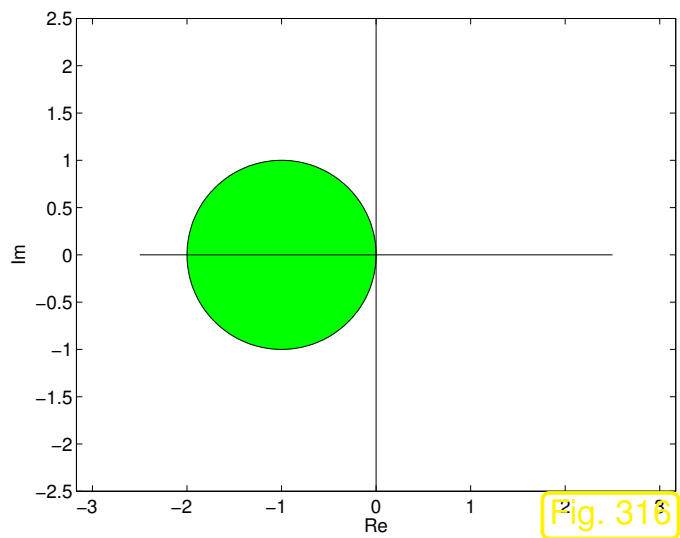
The linear stability analysis based on Fourier symbols of difference operators for Cauchy problems is often referred to as **von Neumann stability analysis**.

Remark 8.4.28 (Stability domains).

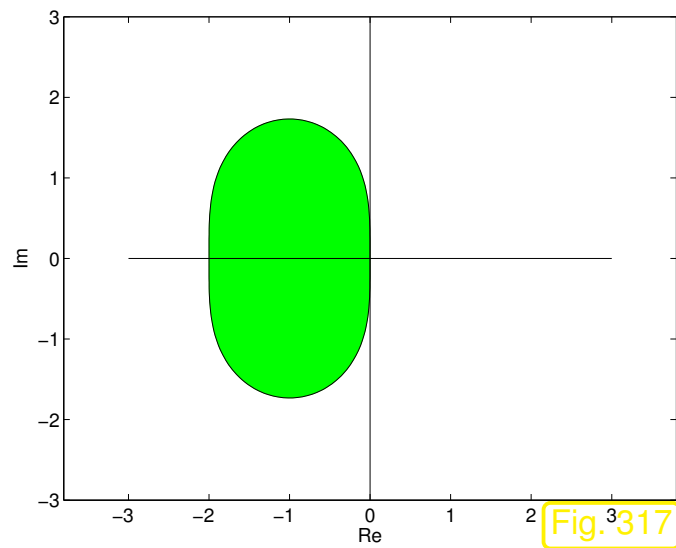
Terminology in the theory of Runge-Kutta single step methods

Stability domain: $\{z \in \mathbb{C} : |S(z)| \leq 1\}$.

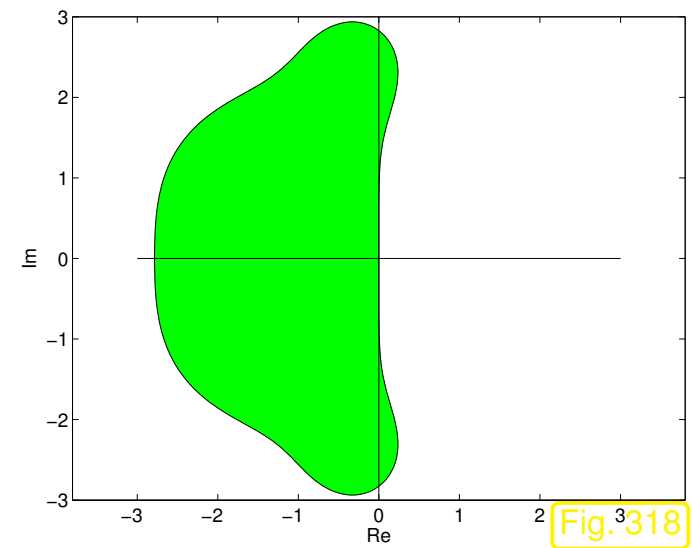
Stability domains:



explicit Euler method



explicit trapezoidal rule



Classical RK4-method

► Necessary stability condition:

$$\{\tau \hat{c}(\xi), -\pi < \xi \leq \pi\} \subset \text{stability domain of RK-method}$$



Example 8.4.29 (Stability and CFL condition).

Consider: upwind spatial discretization (8.4.17) & explicit Euler timestepping

➤ symbol of difference operator (\rightarrow Ex. 8.4.20): $\hat{c}_h(\xi) = \frac{v}{h}(\exp(-i\xi) - 1)$,

stability function:

$$S(z) = 1 + z.$$

Locus of

$$\Sigma := S(\tau \hat{c}(\xi)), \quad -\pi < \xi \leq \pi,$$

in the complex plane

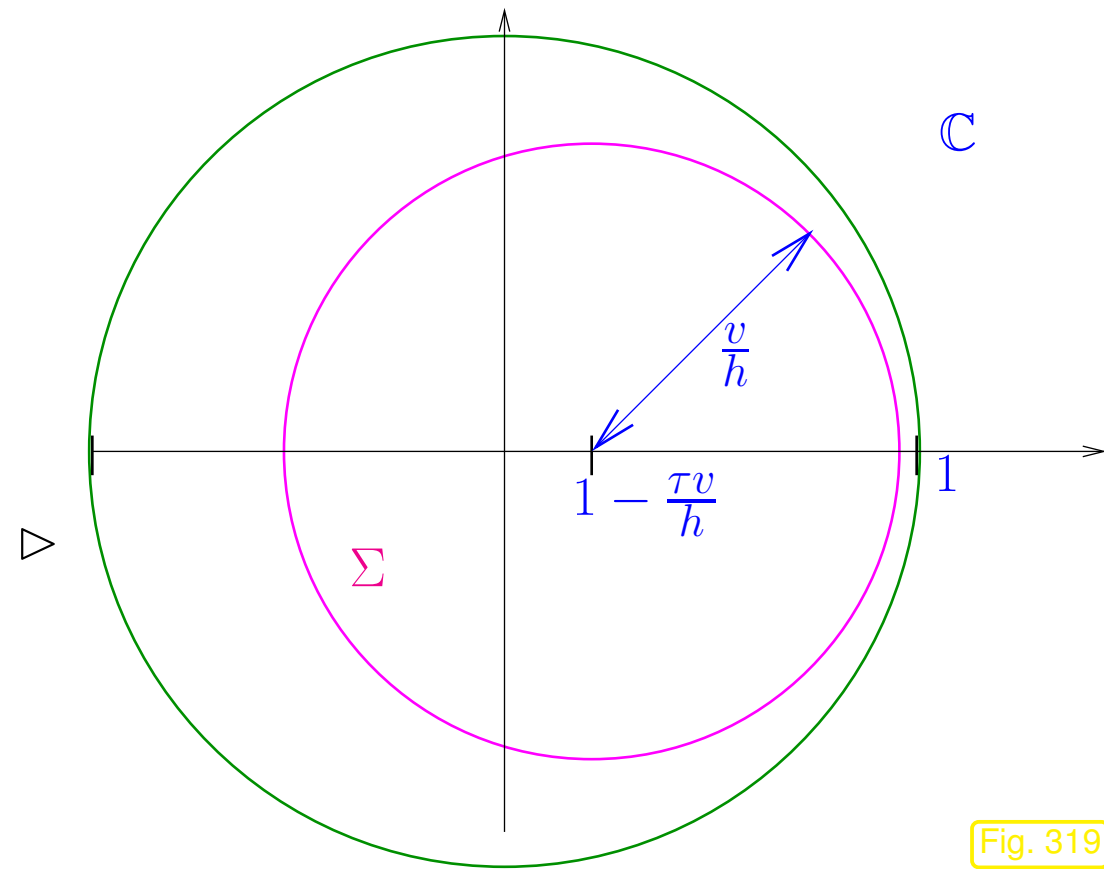


Fig. 319

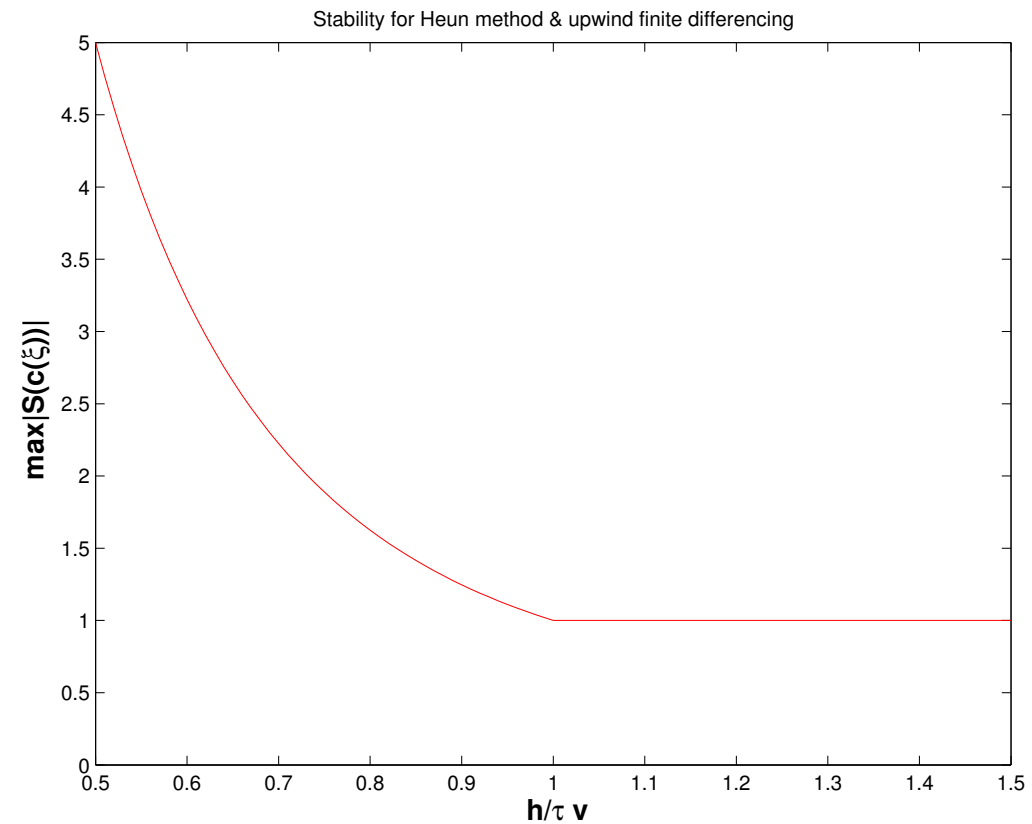
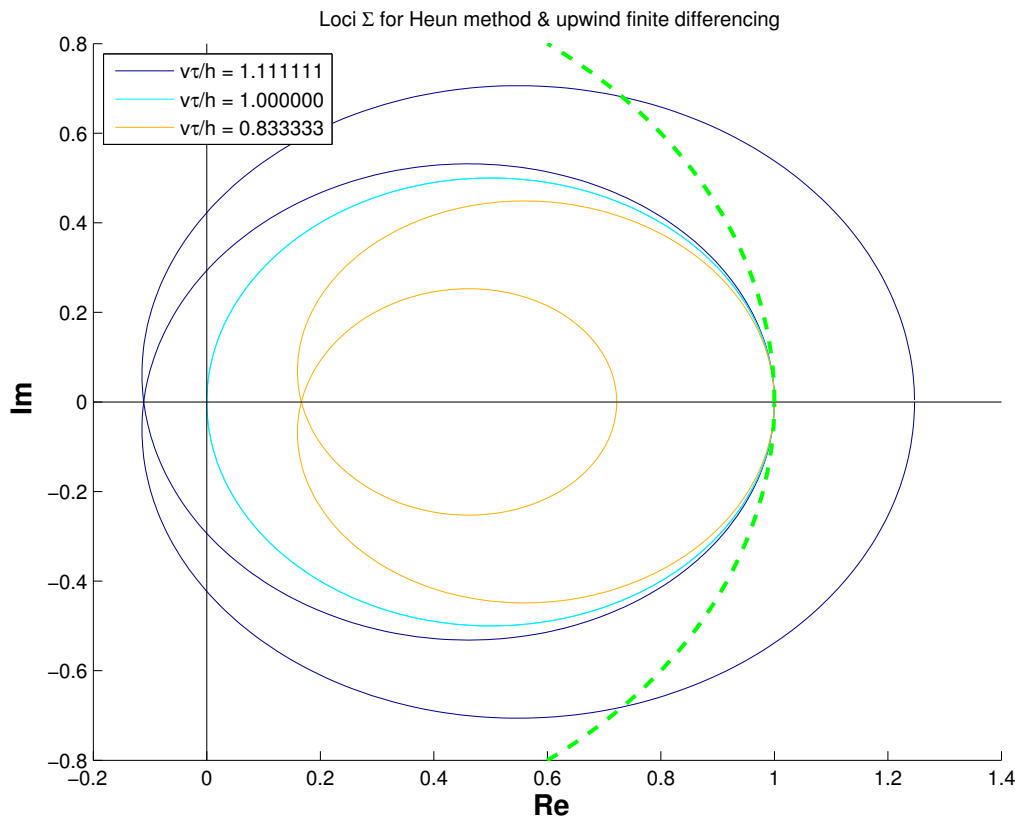
▶ $|S(\tau \hat{c}(\xi))| \leq 1 \quad \forall -\pi < \xi \leq \pi \iff v \frac{\tau}{h} \leq 1.$

= CFL-condition of Def. 8.4.13!

Note that the maximal analytic region of dependence for constant velocity v linear advection is merely a line with slope v in the $x - t$ -plane, see Ex. 8.2.12.

Consider: upwind spatial discretization (8.4.17) & explicit trapezoidal rule: stability function $S(z) = 1 + z + \frac{1}{2}z^2$

Plots for $v = 1, \tau = 1$



► $|S(\tau \hat{c}(\xi))| \leq 1 \quad \forall -\pi < \xi \leq \pi \iff v \frac{\tau}{h} \leq 1.$

= tighter timestep constraint than stipulated by mere CFL-condition (8.4.14).

To see this note that the explicit trapezoidal rule is a 2-stage Runge-Kutta method. Hence, the spatial stencil has width 2 in upwind direction, see Fig. 310.



8.4.3 Convergence

Example 8.4.30 (Convergence of fully discrete finite volume methods for Burgers equation).

- Cauchy problem for Burgers equation (8.1.60)

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{1}{2} u^2 \right) = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) , \quad x \in \mathbb{R} .$$

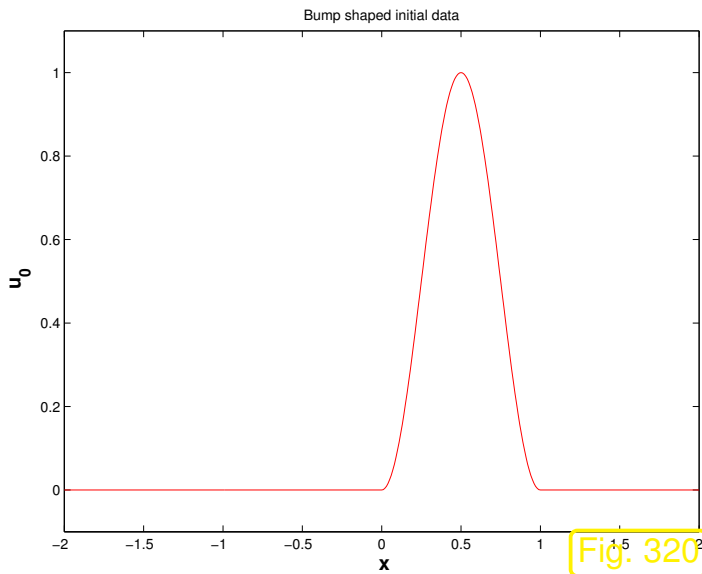
- smooth, non-smooth and discontinuous initial data, supported in $[0, 1]$:

$$u_0(x) = 1 - \cos^2(\pi x), \quad 0 \leq x \leq 1, \quad 0 \text{ elsewhere}, \quad \text{(BUMP)}$$

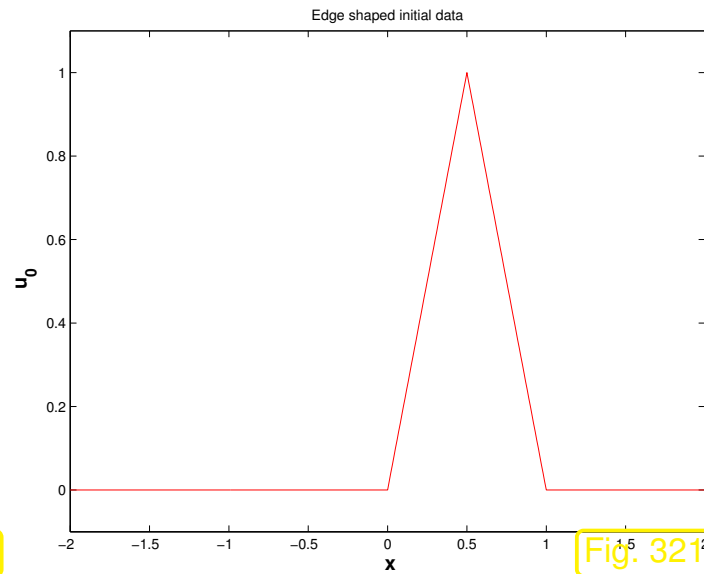
$$u_0(x) = 1 - 2 * |x - \frac{1}{2}|, \quad 0 \leq x \leq 1, \quad 0 \text{ elsewhere}, \quad \text{(WEDGE)}$$

$$u_0(x) = 1, \quad 0 \leq x \leq 1, \quad 0 \text{ elsewhere}. \quad \text{(BOX)}$$

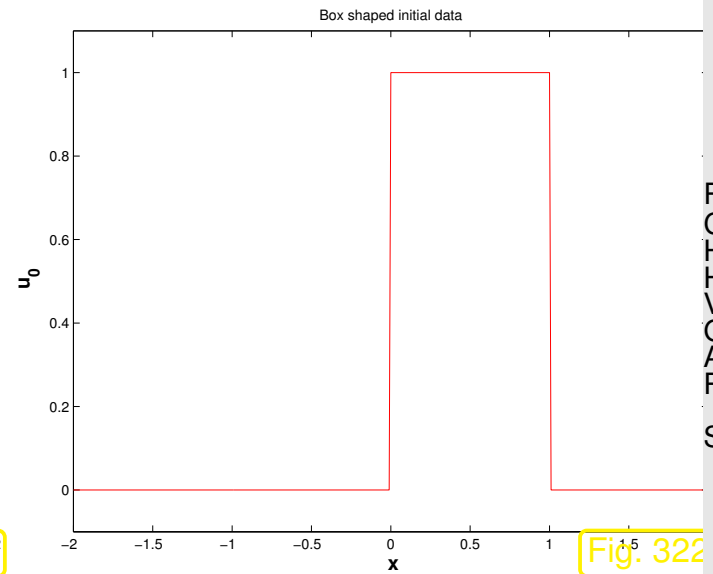
- maximum speed of propagation $\dot{s} = 1$.



(BUMP)



(WEDGE)



(BOX)

- Spatial discretization on equidistant mesh with meshwidth $h > 0$ based on finite volume method in conservation form with

❶ (local) Lax-Friedrichs numerical flux (8.3.29),

❷ Godunov numerical flux (8.3.51).

- Initial values $\vec{\mu}^{(0)}$ obtained from dual cell averages.

- Explicit Runge-Kutta (order 4) timestepping with uniform timestep $\tau > 0$.
- Fixed ratio: $\tau : h = 1$ (\triangleright CFL-condition satisfied)
- Monitored: error norms (log-log plots)

$$\text{err}_1(h) := \max_{k>0} h \sum_j |\mu_j^{(k)} - u(x_j, t_k)| \approx \max_{k>0} \left\| u_N^{(k)} - u(\cdot, t_k) \right\|_{L^1(\mathbb{R})}, \quad (8.4.31)$$

$$\text{err}_\infty(h) := \max_{k>0} \max_{j \in \mathbb{Z}} |\mu_j^{(k)} - u(x_j, t_k)| \approx \max_{k>0} \left\| u_N^{(k)} - u(\cdot, t_k) \right\|_{L^\infty(\mathbb{R})}. \quad (8.4.32)$$

for different final times $T = 0.3, 4$, $h \in \left\{ \frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}, \frac{1}{320}, \frac{1}{640}, \frac{1}{1280} \right\}$.

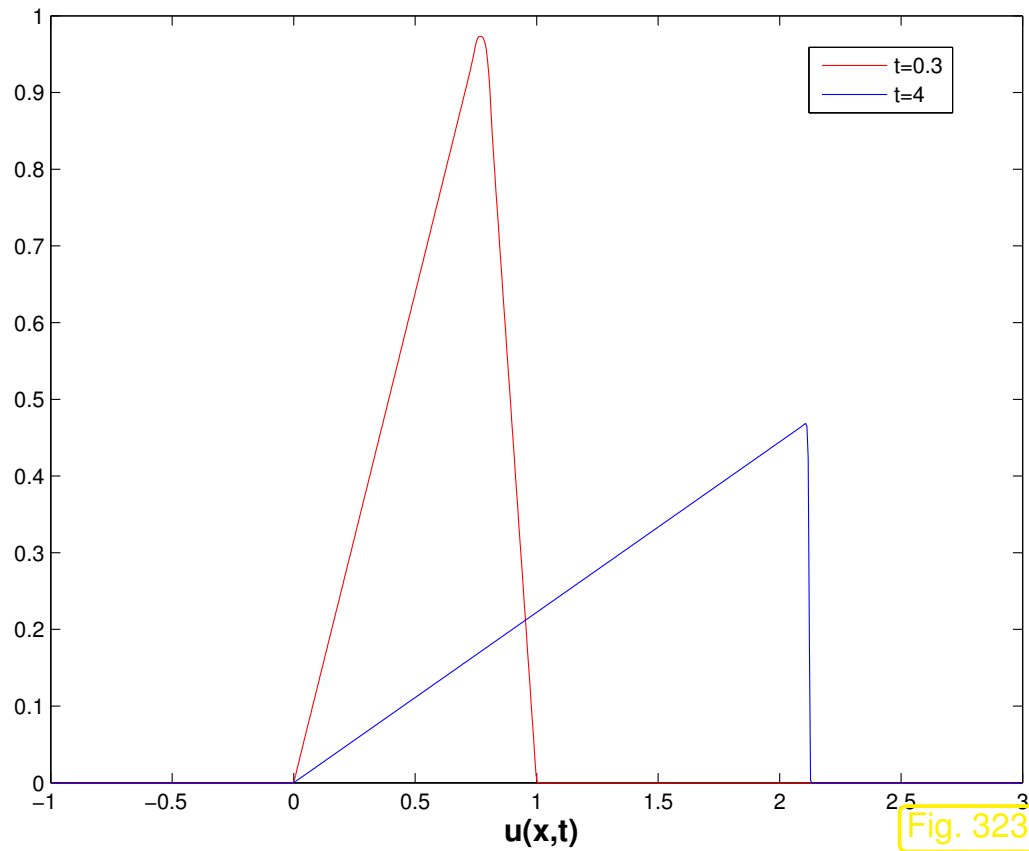


Fig. 323

“Exact solution”: Initial data (WEDGE)

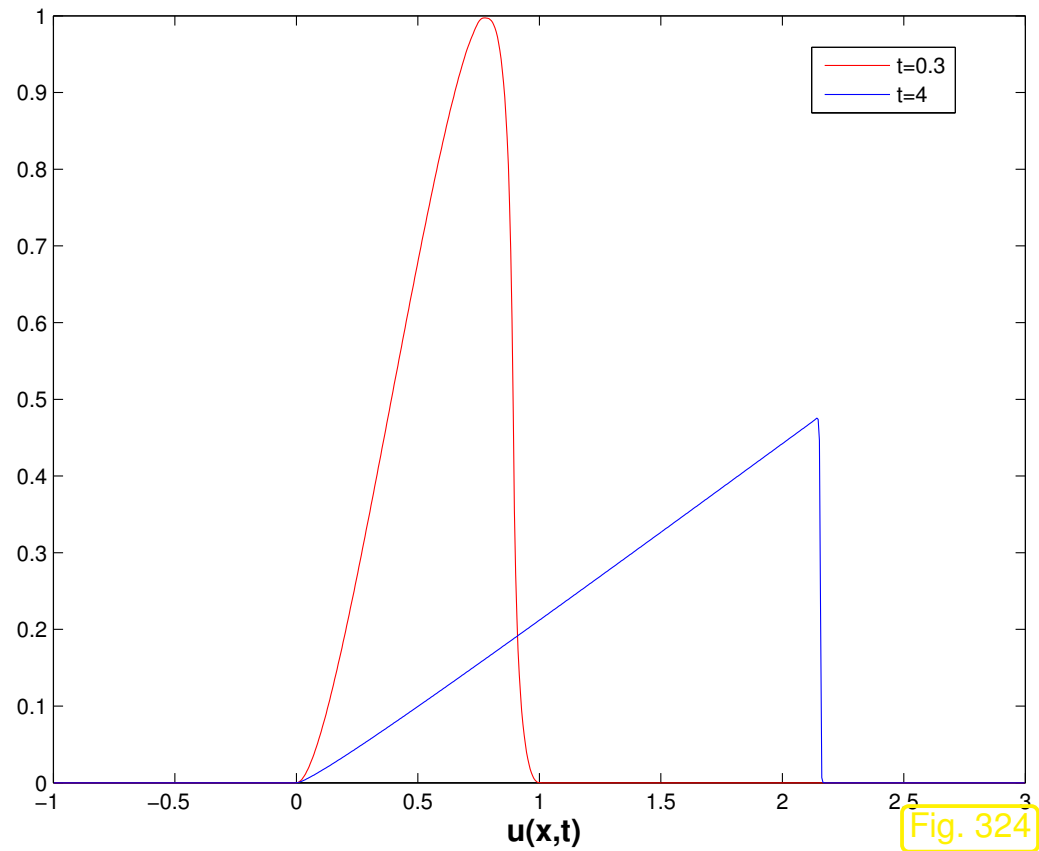


Fig. 324

“Exact solution”: Initial data (BUMP)

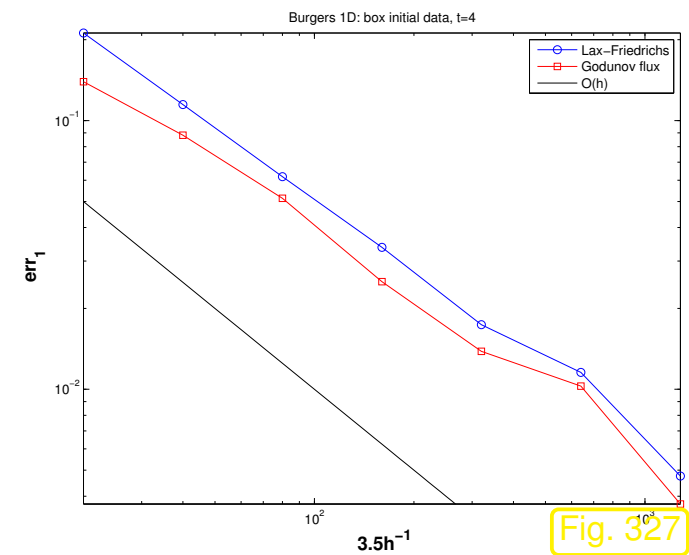
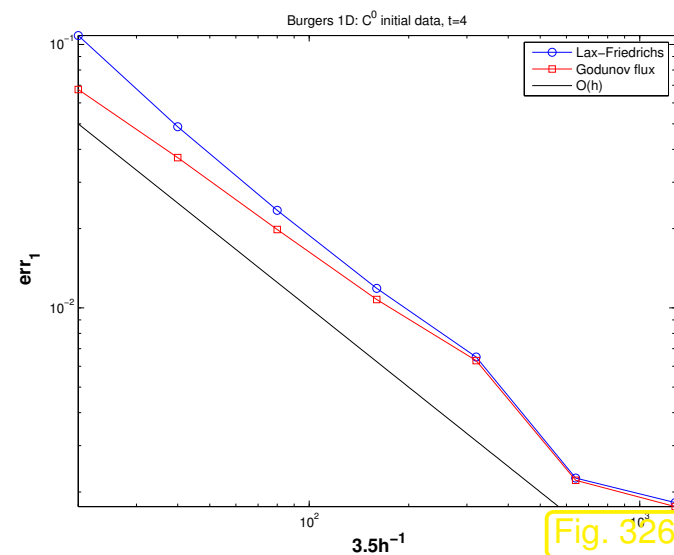
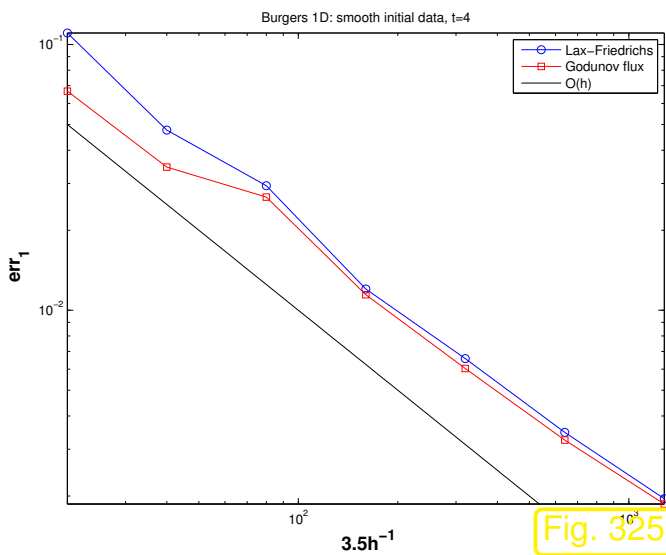
These “exact solutions” were computed with a MUSCL scheme (\rightarrow Sect. 8.5.3) on an equidistant mesh with $h = 10^{-4}$

Note: for bump initial data (BUMP) we can still expect $u(\cdot, 0.3)$ to be smooth, because characteristics will not intersect before that time, *cf.* (8.2.14) and Ex. 8.2.15.

Why do we study the particular error norms (8.4.31) and (8.4.32)?

From Thm. 8.2.45 and Thm. 8.2.47 we know that the evolution for a scalar conservation law in 1D enjoys stability on the norms $\|\cdot\|_{L^1(\mathbb{R})}$ and $\|\cdot\|_{L^\infty(\mathbb{R})}$. Hence, these norms are the natural norms for measuring discretization errors, *cf.* the use of the energy norm for measuring the finite element discretization error for 2nd order elliptic BVP.

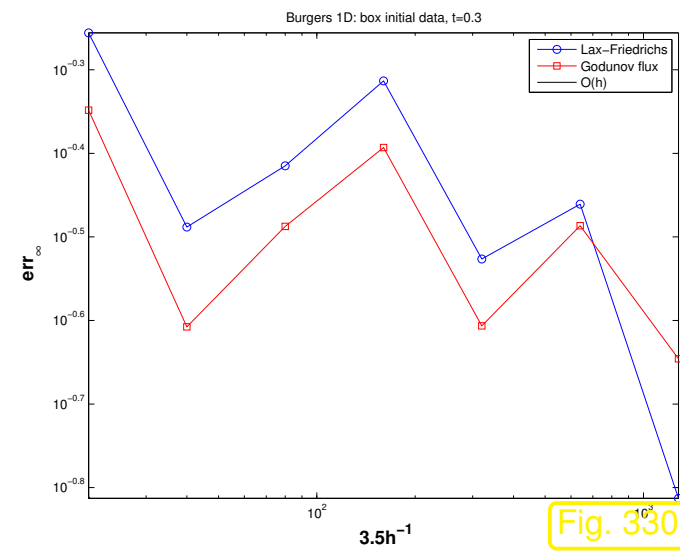
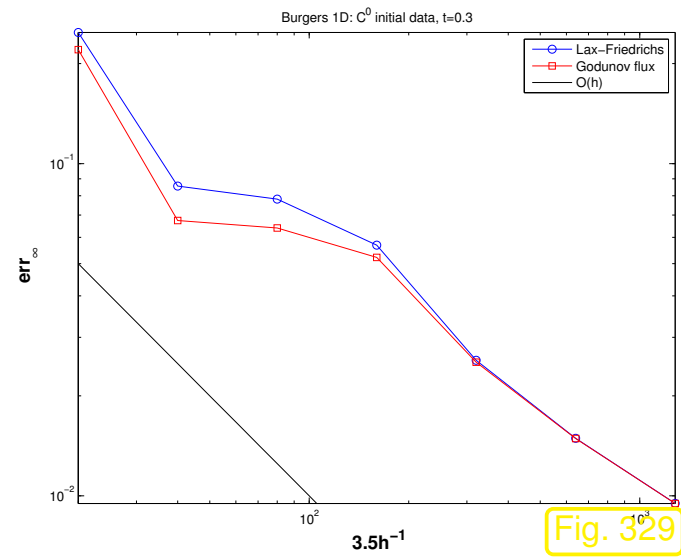
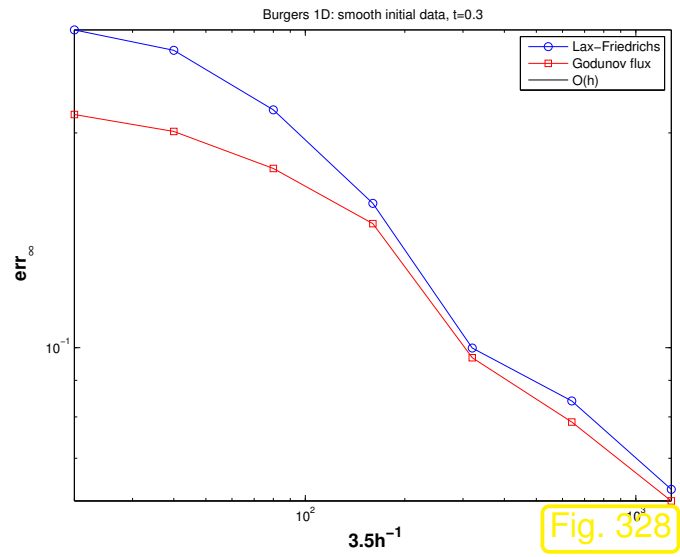
$T = 4$, error err_1



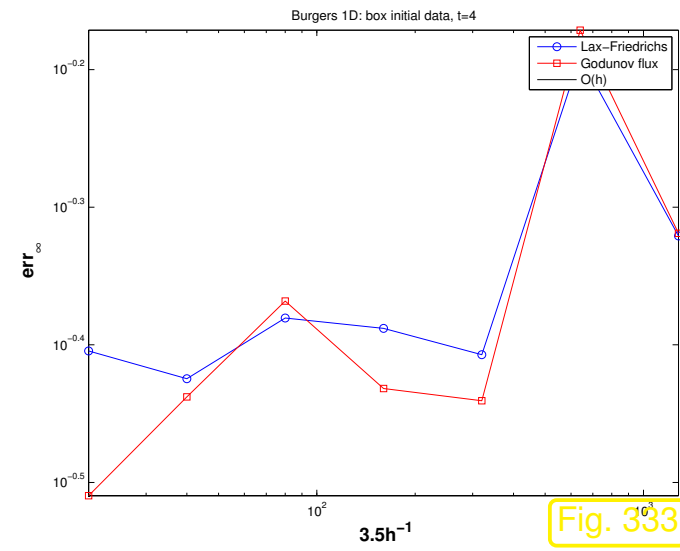
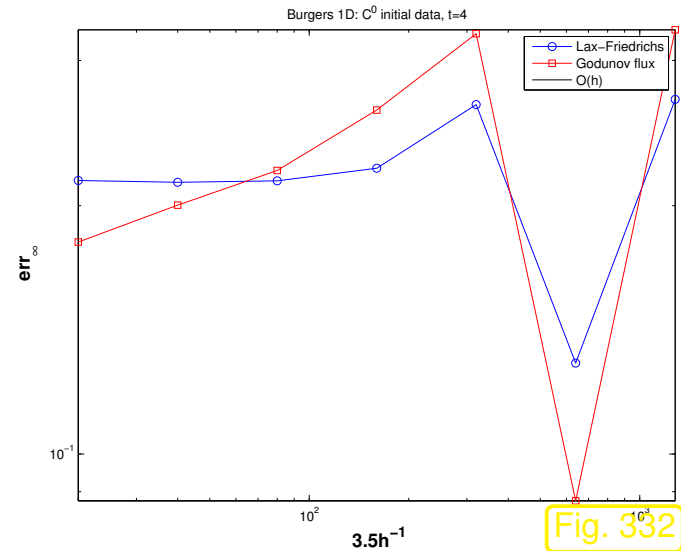
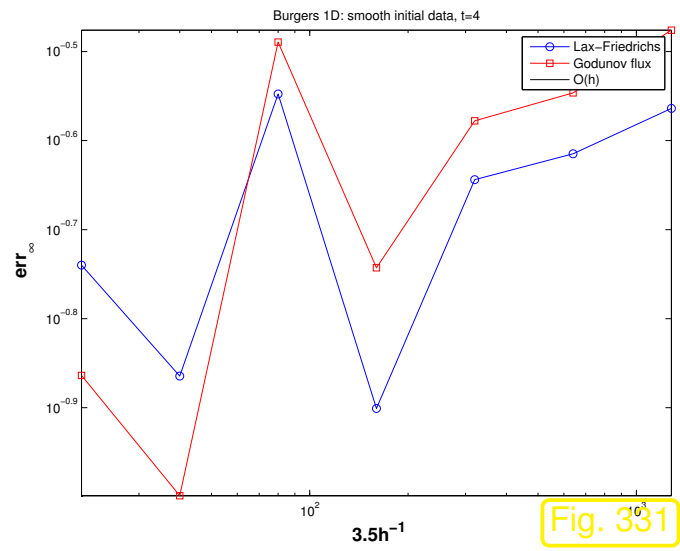
$T = 0.3$: error err_∞

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



$T = 4$: error err_{∞}



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Error obtained by comparison with numerical “reference solution” obtained on a very fine spatio-temporal grid.

Observations: for either numerical flux function

- (near) first order algebraic convergence (\rightarrow Def. 1.6.32) w.r.t. mesh width h in err_1 ,
- algebraic convergence w.r.t. mesh width h in err_∞ *before* the solution develops discontinuities (shocks),
- no convergence in norm err_∞ after shock formation.



Best we get: *merely first order* algebraic convergence $O(h)$

Heuristic explanation for limited order:

$u = u(x, t) \hat{=}$ *smooth* entropy solution of Cauchy problem

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times]0, T[\quad , \quad u(x, 0) = u_0(x) \quad , \quad x \in \mathbb{R} . \quad (8.2.9)$$

We study the so-called **consistency error** of the numerical flux $F = F(v, w)$

$$(\vec{\tau}_F(t))_j = F(u(x_j), u(x_{j+1}, t)) - f(u(x_{j+1/2}, t)) \quad , j \in \mathbb{Z} \quad ,$$

which measures the deviation of the approximate flux and the true flux, when the approximate solution agreed with the exact solution at the nodes of the mesh.

What we are interested in

behavior of $(\vec{\tau}_F(t))_j$ as mesh width $h \rightarrow 0$,

where an equidistant spatial mesh is assumed.

Terminology:

$$\max_{j \in \mathbb{Z}} (\vec{\tau}_F(t))_j = O(h^q) \quad \text{for } h \rightarrow 0 \quad \Leftrightarrow \quad \text{numerical flux consistent of order } q \in \mathbb{N} . \quad (8.4.34)$$

Rule of thumb: Order of consistency of numerical flux function limits (algebraic) order of convergence of (semi-discrete and fully discrete) finite volume schemes.

Example 8.4.35 (Consistency error of upwind numerical flux).

Assumption: f continuously differentiable $u_0 \geq 0$ and $f'(u) \geq 0$ for $u \geq 0$ \triangleright no transsonic rarefactions!

In this case the upwind numerical flux (8.3.33) agrees with the Godunov flux (8.3.51), see Rem. 8.3.52 and

$$F_{\text{uw}}(u(x_j, t), u(x_{j+1}, t)) = f(u(x_j), t), \quad j \in \mathbb{Z}.$$

$$\begin{aligned} \blacktriangleright \quad (\vec{\tau}_{F_{\text{uw}}}(t))_j &= f(u(x_j, t)) - f(u(x_{j+1/2}, t)) \\ &= f'(u(x_{j+1/2}, t))(u(x_j, t) - u(x_{j+1/2}, t)) + O(|u(x_j, t) - u(x_{j+1/2}, t)|^2) \\ &= -f'(u(x_{j+1/2}, t)) \frac{\partial u}{\partial x}(x_{j+1/2}, t) \frac{1}{2}h + O(h^2) \quad \text{for } h \rightarrow 0, \end{aligned}$$

by *Taylor expansion* of f and u .

This means that the upwind/Godunov numerical flux is (only) *first order consistent*.

Example 8.4.36 (Consistency error of Lax-Friedrichs numerical flux).

Assumption: smooth flux function

Recall: The (local) Lax-Friedrichs numerical flux

$$F_{\text{LF}}(v, w) = \frac{1}{2}(f(v) + f(w)) - \frac{1}{2} \max_{\min\{v,w\} \leq u \leq \max\{v,w\}} |f'(u)|(w - v), \quad (8.3.29)$$

is composed of the central flux and a diffusive flux.

We examine the consistency error for both parts separately, using Taylor expansion

❶ central flux:


$$\begin{aligned} & \frac{1}{2}(f(u(x_j, t)) + f(u(x_{j+1}, t))) - f(u(x_{j+1/2}, t)) \\ &= \frac{1}{2}f'(u(x_{j+1/2}, t))(u(x_j, t) - u(x_{j+1/2}, t) + u(x_{j+1}, t) - u(x_{j+1/2}, t)) + O(h^2) \quad (8.4.37) \\ &= \frac{1}{2}f'(u(x_{j+1/2}, t))\left(\frac{\partial u}{\partial x}(x_{j+1/2}, t)(-\frac{1}{2}h + \frac{1}{2}h) + O(h^2)\right) + O(h^2) \\ &= O(h^2) \quad \text{for } h \rightarrow 0. \end{aligned}$$



The central flux is *second order consistent*.

However, due to instability the central flux is useless, see Sect. 8.3.3.1.

② diffusive flux part:

$$u(x_{j+1}, t) - u(x_j, t) = \frac{\partial u}{\partial x}(x_{j+1/2}, t)h + O(h^2) \quad \text{for } h \rightarrow 0 .$$


$$F_{\text{LF}}(u(x_j, t), u(x_{j+1}, t)) - f(u(x_{j+1/2}, t)) = O(h) \quad \text{for } h \rightarrow 0 ,$$

because the consistency error is dominated by the diffusive flux.



The observations made in the above examples are linked to a general fact:

Monotone numerical fluxes (\rightarrow Def. 8.3.59) are at most first order consistent.

8.5 Higher-order conservative schemes

Formally, high-order conservative finite volume methods are distinguished by numerical flux functions that are consistent of order ≥ 2 , see (8.4.34).

However, solutions of (systems of) conservation laws will usually not even be continuous (because of shocks emerging even in the case of smooth u_0 , see (8.2.15)), let alone smooth, so that the formal order of consistency may not have any bearing for the (rate of) convergence observed for the method for a concrete Cauchy problem.

Therefore in the field of numerics of conservation laws “high-order” is desired not so much for the promise of higher rates of convergence, but for the following advantages:

- for the same spatial resolution. high-order methods frequently provide more accurate solutions in the sense of global error norms as first-order methods,
- high-order methods often provide *better resolution of local features* of the solution (shocks, etc.).

In standard semi-discrete finite volume schemes in conservation form for 2-point numerical flux function,

$$\frac{d\mu_j}{dt}(t) = -\frac{1}{h} \left(F(\mu_j(t), \mu_{j+1}(t)) - F(\mu_{j-1}(t), \mu_j(t)) \right), \quad j \in \mathbb{Z}, \quad (8.3.12)$$

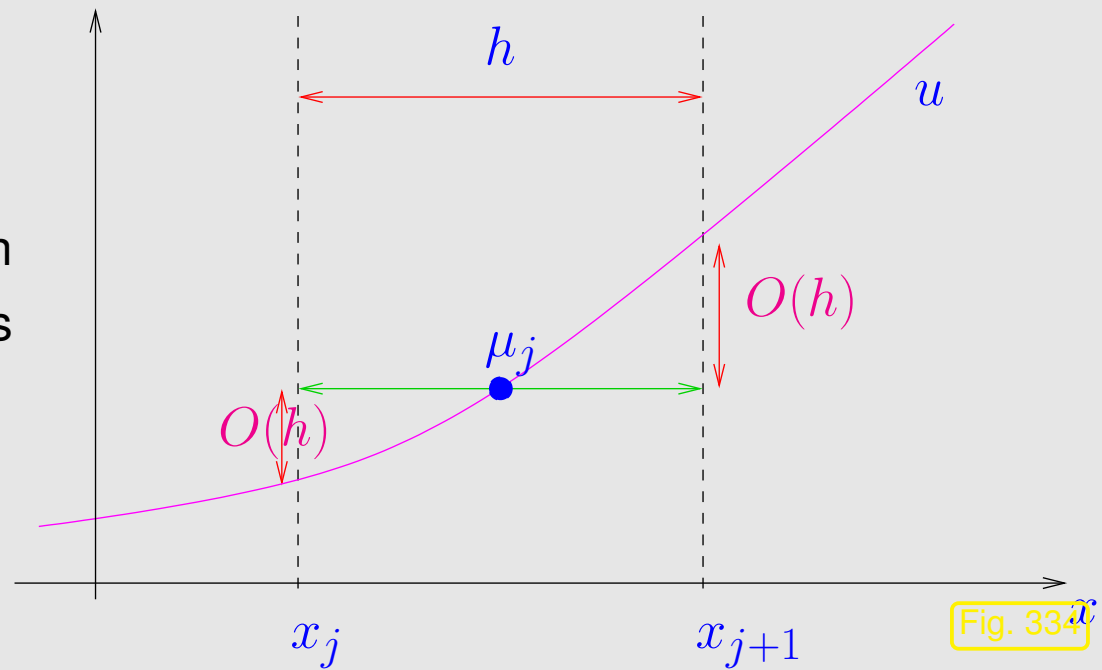
the numerical flux function is evaluated for the cell averages μ_j , which can be read as approximate values of a projection of the exact solution onto piecewise constant functions (on dual cells)

$$\mu_j(t) \approx \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t) \, dx. \quad (8.3.5)$$

By Taylor expansion we find for $u \in C^1$

$$u(x_{j+1/2}, t) - \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t) \, dx = O(h) \quad \text{for } h \rightarrow 0,$$

and, unless some lucky cancellation occurs as in the case of the central flux, see Ex. 8.4.36, this does not allow more than first order consistency.



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

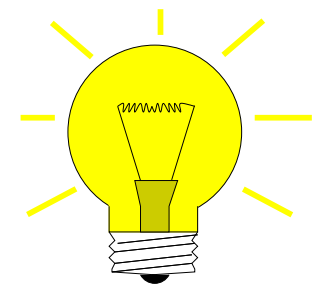
SAM, ETHZ

8.5.1 Piecewise linear reconstruction

Idea: Plug “better” approximations of $u(x_{j\pm 1/2}, t)$ into numerical flux function in (8.3.12)

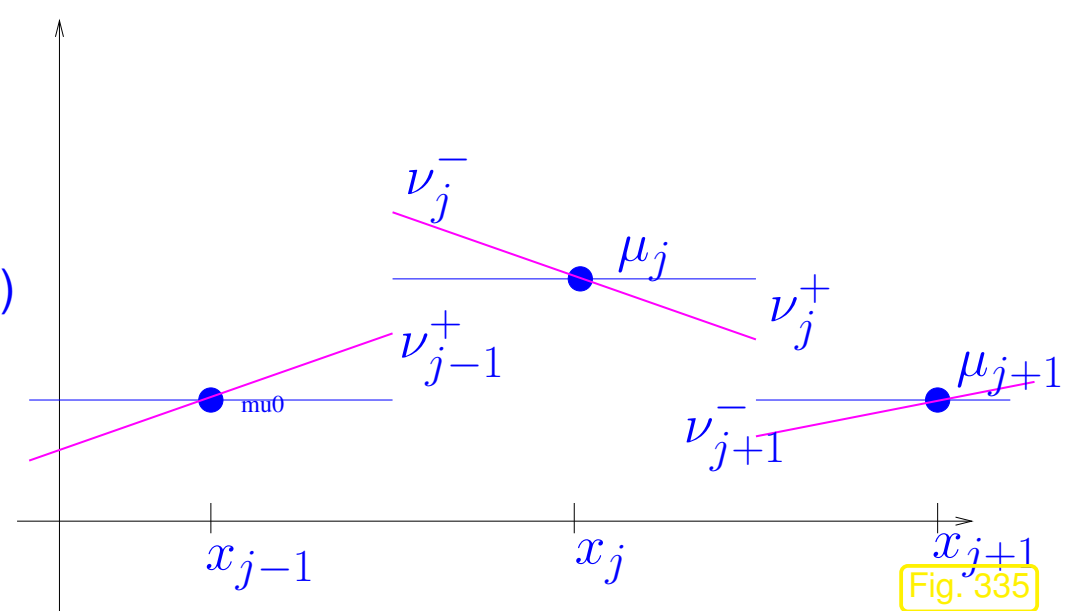
$$\frac{d\mu_j}{dt}(t) = -\frac{1}{h} (F(\nu_j^+(t), \nu_{j+1}^-(t)) - F(\nu_{j-1}^+(t), \nu_j^-(t))) , \quad j \in \mathbb{Z} , \quad (8.5.1)$$

where ν_j^\pm are obtained by **piecewise linear reconstruction** from the (dual) cell values μ_j .



$$\begin{aligned} \nu_j^-(t) &:= \mu_j(t) - \frac{1}{2}h\sigma_j(t), \\ \nu_j^+(t) &:= \mu_j(t) + \frac{1}{2}h\sigma_j(t), \end{aligned} \quad j \in \mathbb{Z}, \quad (8.5.2)$$

with suitable **slopes** $\sigma_j(t) = \sigma(\vec{\mu}(t))$.



Analogy: piecewise cubic Hermite interpolation with reconstructed slopes, discussed in the context of **shape preserving interpolation** in [21, Sect. 3.7.2]. However, we do not aim for smooth functions now.

Definition 8.5.5 (Linear reconstruction).

Given an (infinite) mesh $\mathcal{M} := \{]x_{j-1}, x_j[\}_{j \in \mathbb{Z}}$ ($x_{j-1} < x_j$), a **linear reconstruction operator** $R_{\mathcal{M}}$ is a mapping

$$R_{\mathcal{M}} : \mathbb{R}^{\mathbb{Z}} \mapsto \{v \in L^\infty(\mathbb{R}) : v \text{ linear on }]x_{j-1/2}, x_{j+1/2}[\forall j \in \mathbb{Z}\},$$

taking a sequence $\vec{\mu} \in \mathbb{R}^{\mathbb{Z}}$ of cell averages to a possibly discontinuous function $R_{\mathcal{M}}\vec{\mu}$ that is **piecewise linear on dual cells**.

Linear reconstruction & (8.5.1) \triangleright semi-discrete evolution in conservation form, cf. (8.3.11)

For 2-point numerical flux $F = F(u, w)$

$$\frac{d\mu_j}{dt}(t) = -\frac{1}{h} \left(F(\nu_j^+(t), \nu_{j+1}^-(t)) - F(\nu_{j-1}^+(t), \nu_j^-(t)) \right), \quad j \in \mathbb{Z}. \quad (8.5.6)$$

Code 8.5.8: Conservative FV with linear reconstruction: ode45 timestepping

```

1 function ufinal = highresevl(a,b,N,u0,T,F,slopes)
2 % finite volume discrete evolution in conservation form with linear
3 % reconstruction, see (8.5.6)
4 % Cauchy problem over time [0,T] restricted to finite interval [a,b],
5 % equidistant mesh with meshwidth N cells, meshwidth h := b-a/N.
6 % 2-point numerical flux function F = F(v,w) passed in handle F
7 % 3-point slope reconstruction rule passed as handle slopes = @(v,u,w) ...
8 % (Note: no division by h must be done in slope computation)
9 % returns cell averages for approximate solution at final time in a row vector
10 h = (b-a)/N; x = a+0.5*h:h:b-0.5*h; % cell centers
11 mu0 = h*u0(x)'; % vector of initial cell averages (column vector)
12 % right hand side function for MATLAB ode solvers

```

```

13 odefun = @(t,mu) (-1/h*fluxdiff(h,mu,F,slopes));
14 % timestepping by explicity Runge-Kutta method of order 5
15 options = odeset('abstol',1E-8,'reltol',1E-6,'stats','on');
16 [t,MU] = ode45(odefun,[0 T],mu0,options);
17 % Graphical output
18 [X,T] = meshgrid(x,t);
19 figure; surf(X,T,MU/h); colormap(copper);
20 xlabel('\bf x','fontsize',14);
21 ylabel('\bf t','fontsize',14);
22 zlabel('\bf u','fontsize',14);
23 ufinal = MU(end,:);
24 end

```

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Code 8.5.10: Operator \mathcal{L}_h for spatial semidiscretization with conservative FV with linear reconstruction and 2-point numerical flux

```

1 function fd = fluxdiff(h,mu,F,slopes)
2 % MATLAB function that realizes the right hand side operator  $\mathcal{L}_h$  for the ODE
3 % (8.4.1) arising from conservative finite volume semidiscretization of the
4 % Cauchy problem for a 1D scalar conservation law (8.2.9).
5 % h: meshwidth of equidistant spatial grid
6 % mu: (finite) vector  $\vec{\mu}$  of cell averages
7 % F: handle to 2-point numerical flux function  $F = F(v,w)$ 

```

```

8  % slope: handle to slope function  $\sigma_j = \text{slopes}(\mu_{j-1}, \mu_j, \mu_{j+1})$ 
9  n = length(mu); sigma = zeros(n,1); fd = zeros(n,1);
10 % Computation of slopes  $\sigma_j$ , uses  $\mu_0 = \mu_1$ ,
11 %  $m_{N+1} = \mu_N$ , which amounts to constant extension of state beyond domain of
12 % influence  $[a, b]$  of non-constant initial data.
13 sigma(1) = slopes(mu(1), mu(1), mu(2));
14 for j=2:n-1, sigma(j) = slopes(mu(j-1), mu(j), mu(j+1)); end
15 sigma(n) = slopes(mu(n-1), mu(n), mu(n));
16 % Compute linear reconstruction at endpoints of dual cells
17 nup = mu+0.5*sigma; %  $v_j^+$  at right endpoint
18 num = mu-0.5*sigma; %  $nu_j^-$  at left endpoint
19 % Also here: constant continuation of data outside  $[a, b]$  !
20 fd(1) = F(nup(1), num(2)) - F(mu(1), num(1));
21 for j=2:n-1
22     fd(j) = F(nup(j), num(j+1)) - F(nup(j-1), num(j)); % see (8.5.6)
23 end
24 fd(n) = F(nup(n), mu(n)) - F(nup(n-1), num(n));
25 end

```

“Natural” choice: **central slope** (averaged slope)

$$\sigma_j(t) = \frac{1}{2} \left(\frac{\mu_{j+1}(t) - \mu_j(t)}{h} + \frac{\mu_j(t) - \mu_{j-1}(t)}{h} \right) = \frac{1}{2} \frac{\mu_{j+1}(t) - \mu_{j-1}(t)}{h}. \quad (8.5.11)$$

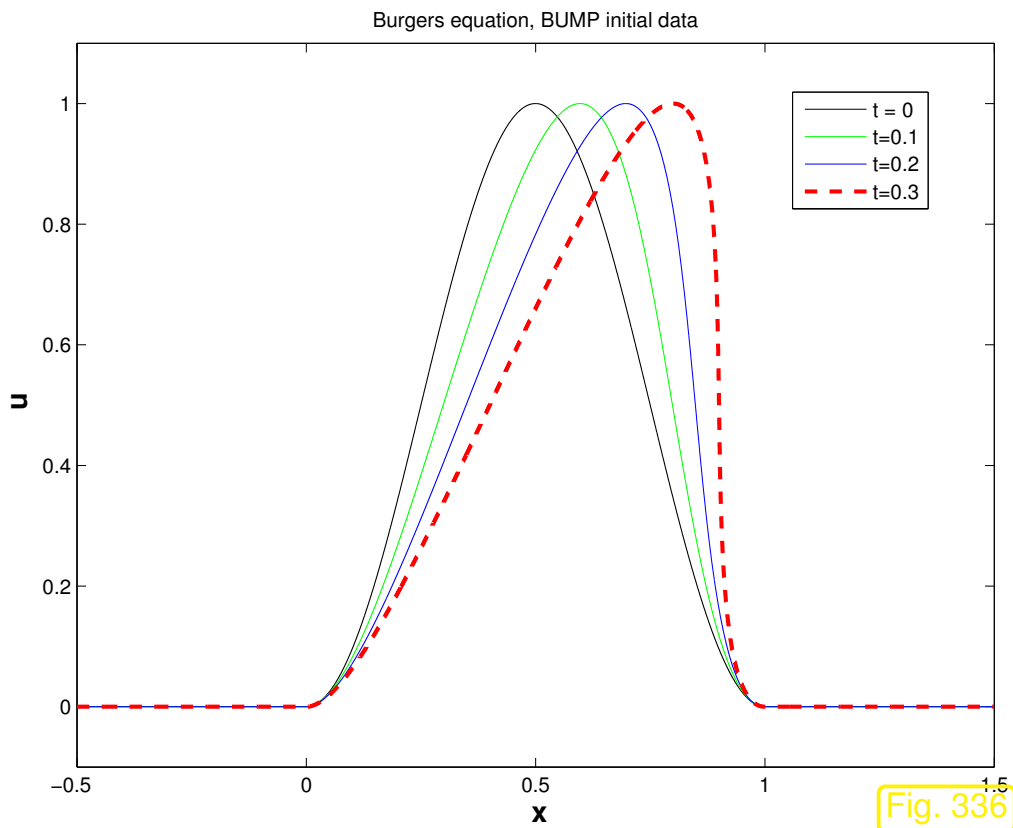
By Taylor expansion: for $u \in C^2$ (that is, u sufficiently smooth), central slope (8.5.11), ν_j^\pm according to (8.5.2)

$$|\nu_j^-(t) - u(x_{j-1/2}, t)|, |\nu_j^+(t) - u(x_{j+1/2}, t)| = O(h^2) .$$

Example 8.5.12 (Convergence of FV with linear reconstruction).

- Cauchy problem for Burgers equation (8.1.60) (flux function $f(u) = \frac{1}{2}u^2$) from Ex. 8.2.43 with C^1 bump initial data (BUMP)
- Equidistant spatial mesh with meshwidth $h =$
- Linear reconstruction with central slope (8.5.11)
- Godunov numerical flux (8.3.51): $F = F_{\text{GD}}$
- 2n-order Runge-Kutta timestepping (method of Heun), timestep $\tau = 0.5h$ (“CFL = 0.5”)

Monitored: Approximate L^1 - and L^∞ -norms of error at final time $T = 0.3$ (exact solution still *smooth* at this time, see Ex. 8.4.30)



◁ “exact solution”

computed by means of a high-order finite volume method (WENO) on a equidistant mesh with 2^{14} points., U. Fjordholm (SAM)

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Observation: 2nd-order convergence in both norms



Example 8.5.14 (Linear reconstruction with central slope (Burgers' equation)).

Cauchy problem of Ex. 8.3.21:

- Cauchy problem for Burgers equation (8.1.60) (flux function $f(u) = \frac{1}{2}u^2$) from Ex. 8.2.43 (“box” initial data)
- Equidistant spatial mesh with meshwidth $h =$
- Linear reconstruction with central slope (8.5.11)
- Godunov numerical flux (8.3.51): $F = F_{GD}$
- timestepping based on adaptive Runge-Kutta method `ode45` of MATLAB
(`opts = odeset('abstol',1E-7,'reltol',1E-6);`).

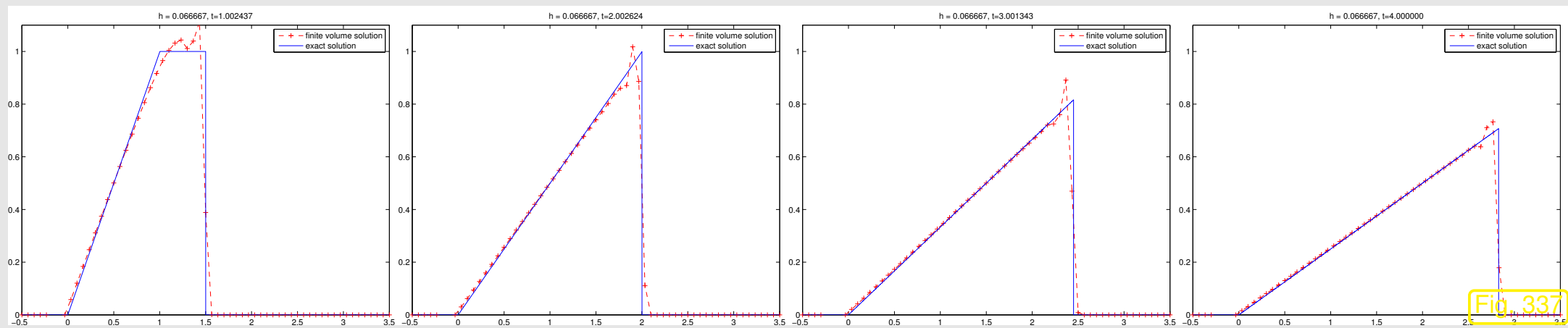
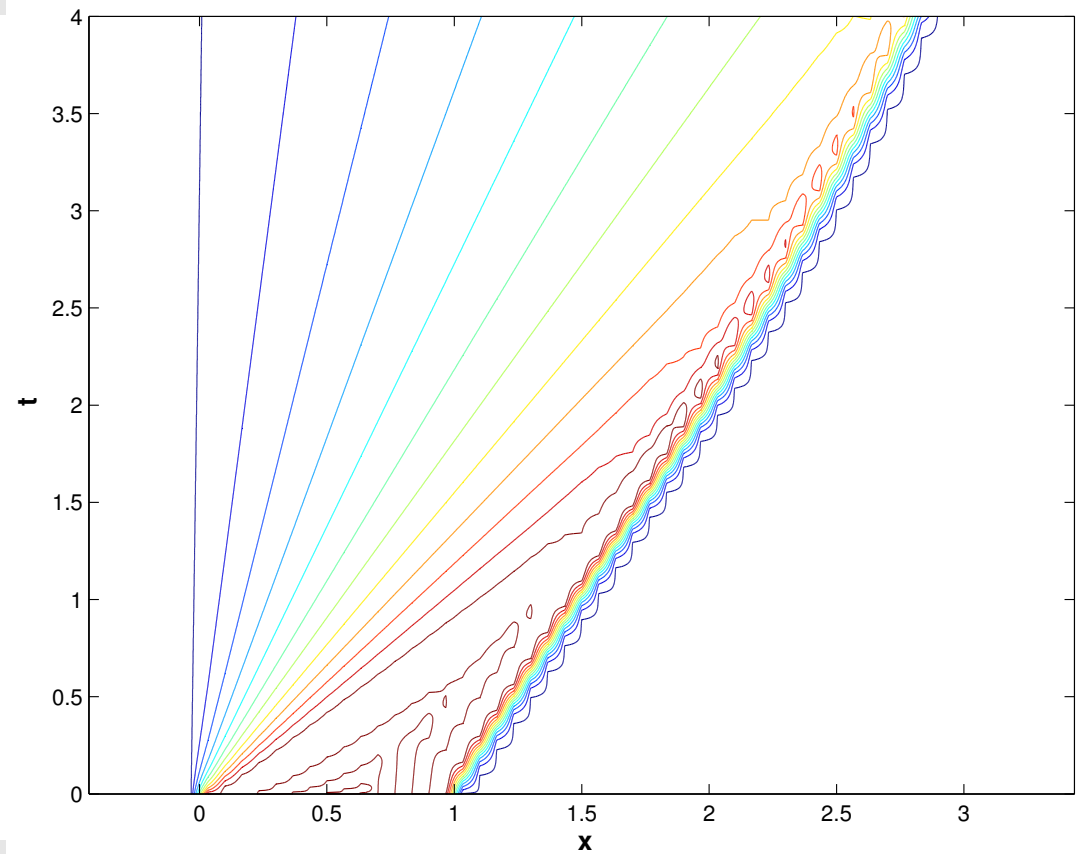
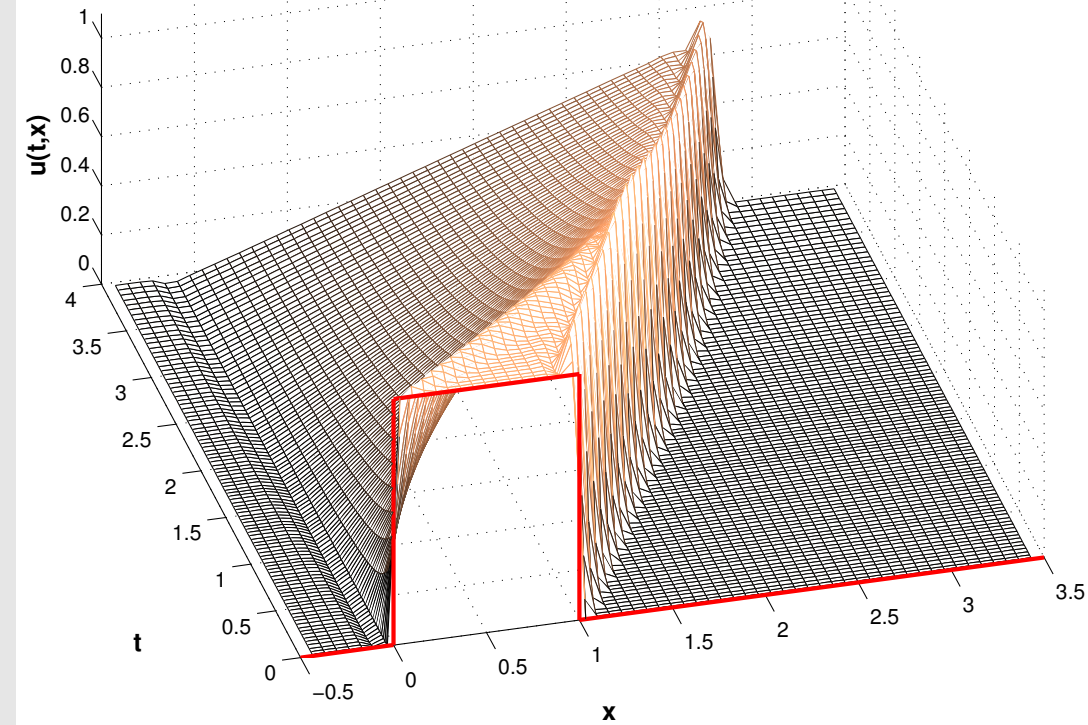


Fig. 337



Emergence of spurious oscillations in the vicinity of shock (in violation of structural properties of the exact solution, see (8.2.46).)

Compare: Oscillations occurring in FV schemes relying on central flux, see Ex. 8.3.21.



Example 8.5.15 (Linear reconstruction with central slope (traffic flow)).

Cauchy problem of Ex. 8.3.23:

- Cauchy problem for Traffic Flow equation (8.1.53) (flux function $f(u) = u(1-u)$) from Ex. 8.2.44 (“box” initial data)
- Equidistant spatial mesh with meshwidth $h =$
- Linear reconstruction with central slope (8.5.11)
- Godunov numerical flux (8.3.51): $F = F_{GD}$
- timestepping based on adaptive Runge-Kutta method `ode45` of MATLAB (`opts = odeset('abstol',1E-7,'reltol',1E-6);`).

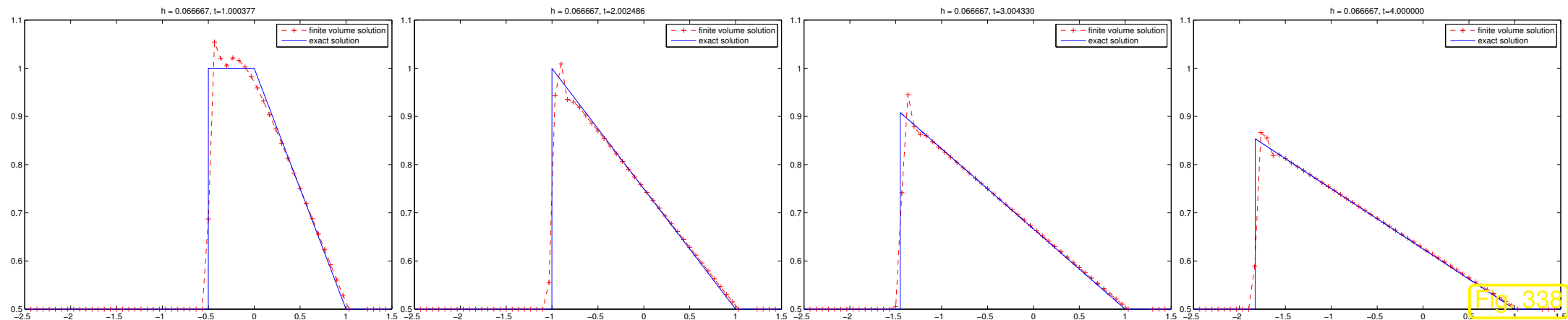
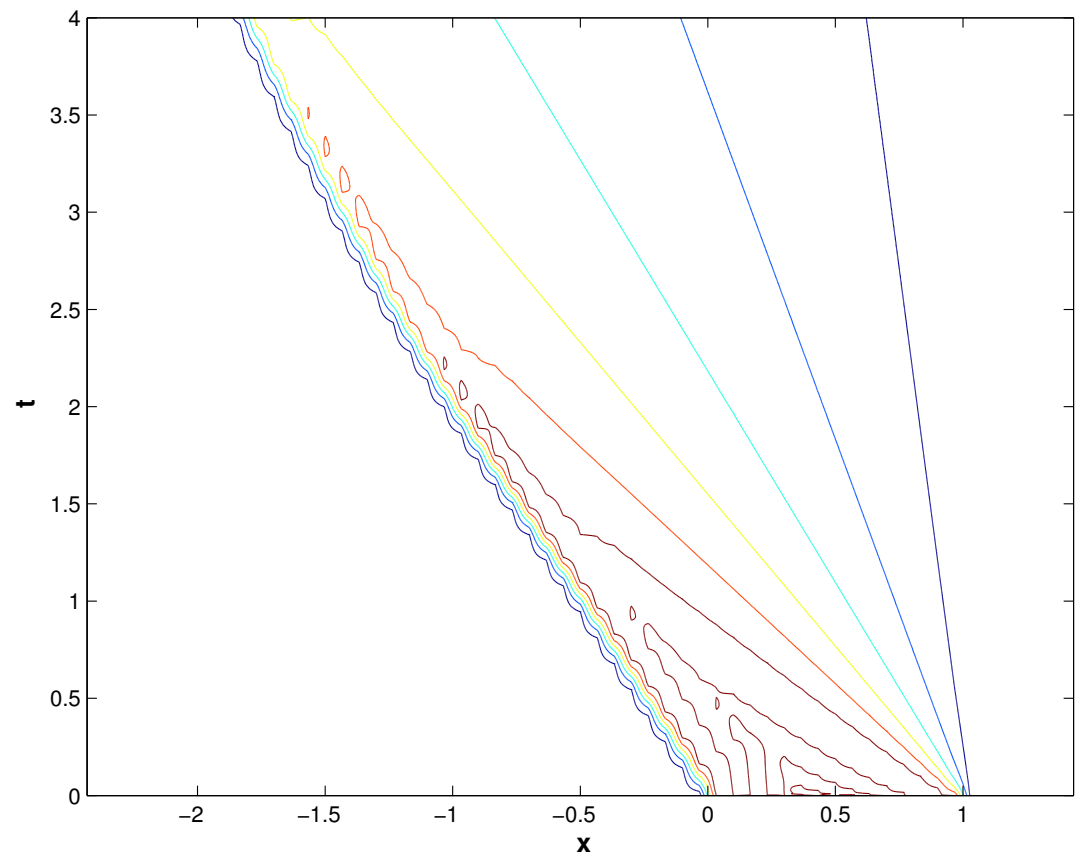
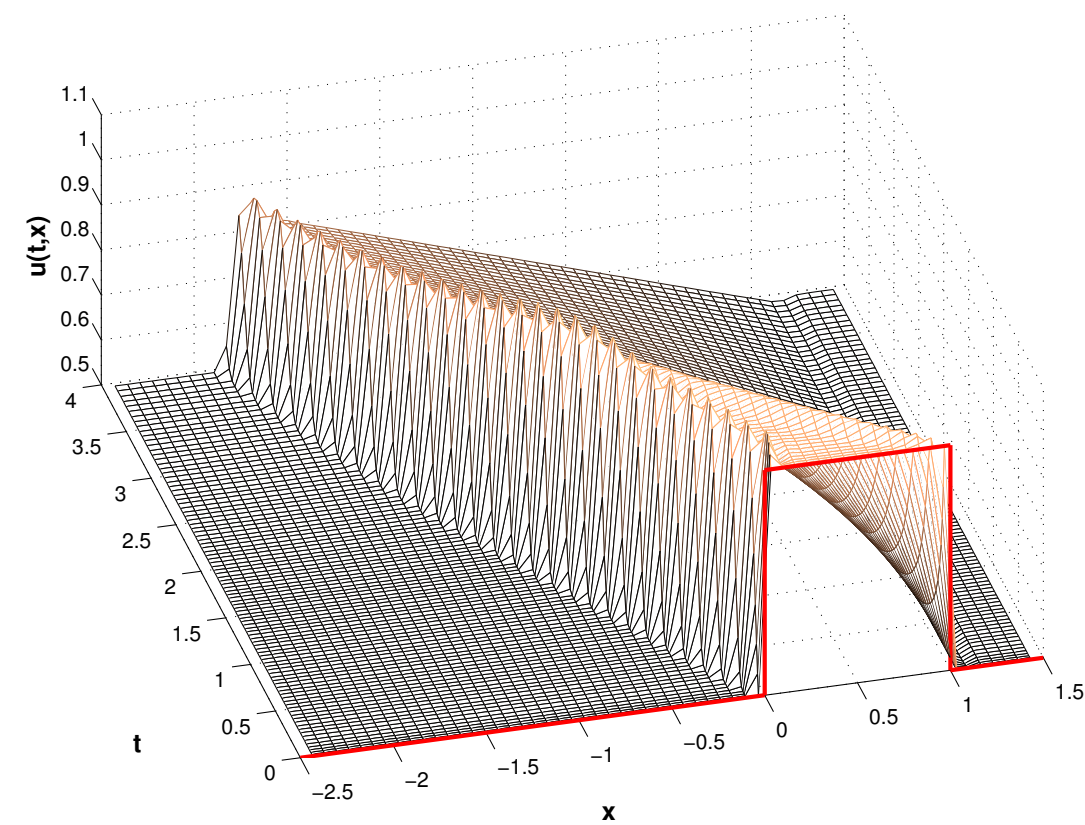


Fig. 338



Emergence of spurious oscillations in the vicinity of shock (in violation of structural properties of the exact solution, see (8.2.46).)

Compare: Oscillations occurring in FV schemes relying on central flux, see Ex. 8.3.23.

In Ex. 8.3.21, 7.2.17, the spurious oscillations can be blamed on the unstable central flux/central finite differences. Maybe, this time the central slope formula is the culprit. Thus, we investigate slope reconstruction connected with backward and forward difference quotients.

Example 8.5.19 (Linear reconstruction with one-sided slopes (Burgers' equation)).

One-sided slopes for use in (8.5.2)

$$\text{Right slope: } \sigma_j(t) = \frac{\mu_{j+1}(t) - \mu_j(t)}{h}, \quad (8.5.20)$$

$$\text{Left slope: } \sigma_j(t) = \frac{\mu_j(t) - \mu_{j-1}(t)}{h}. \quad (8.5.21)$$

Same setting as in Ex. 8.5.14, with central slope replaced with one-sided slopes.

Left slope:

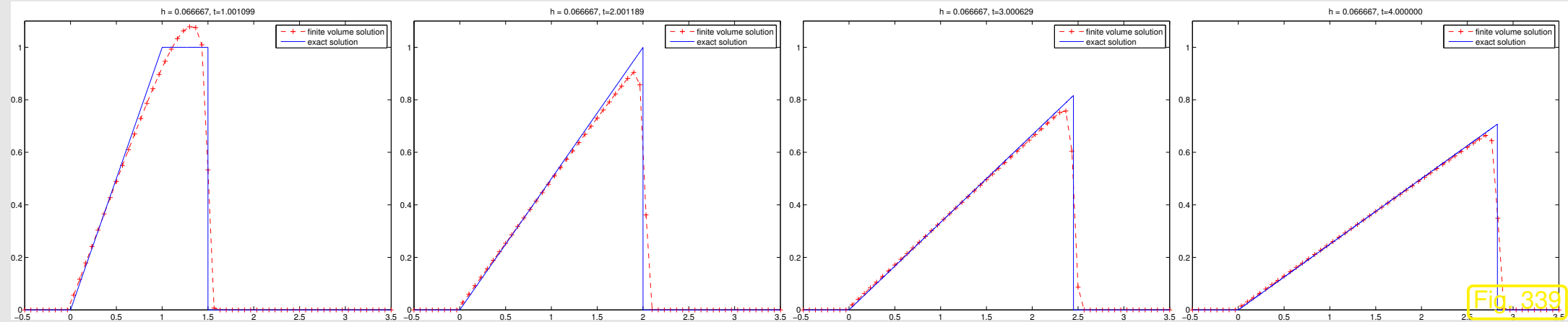


Fig. 339

Right slope:

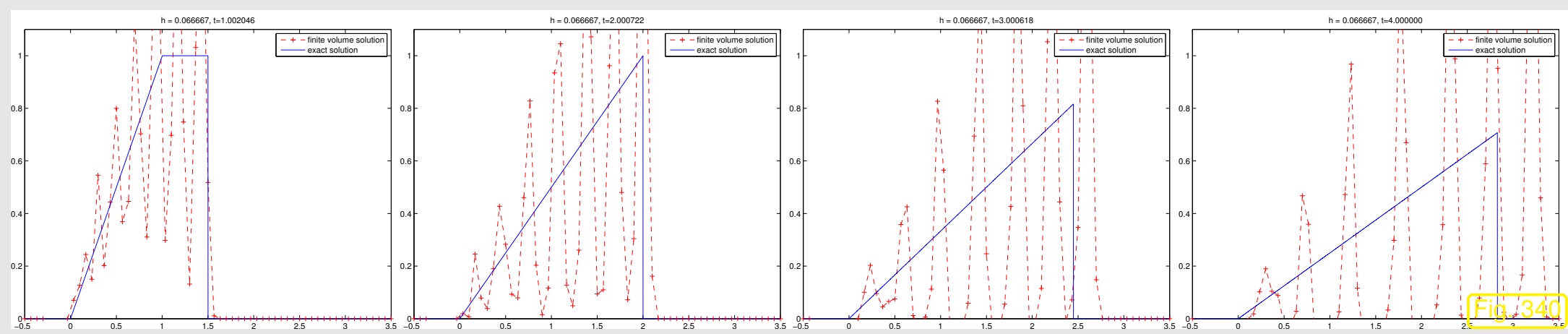


Fig. 340

Observation: spurious oscillations/overshoots, massive and global for (8.5.20), moderate close to shock for (8.5.21).



Example 8.5.22 (Linear reconstruction with one-sided slopes (traffic flow)).

Left slope:

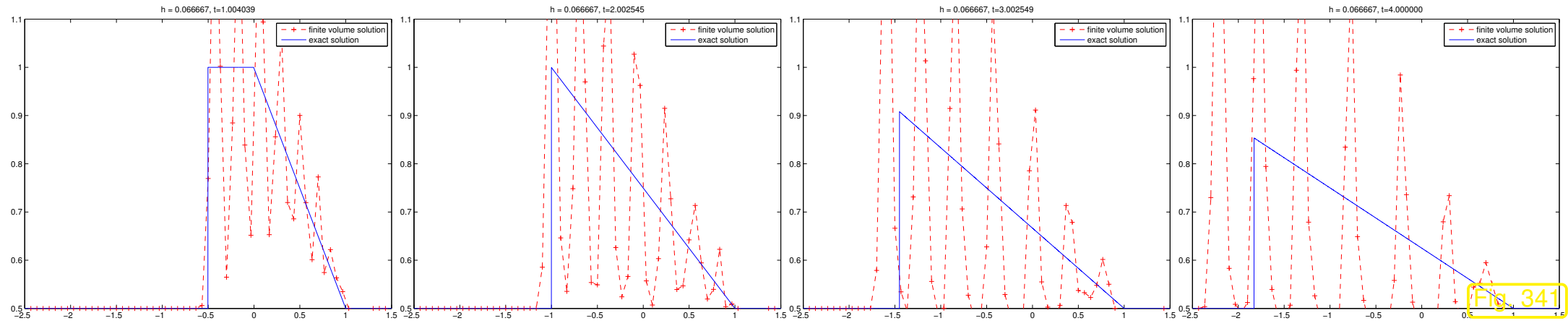


Fig. 341

Right slope:

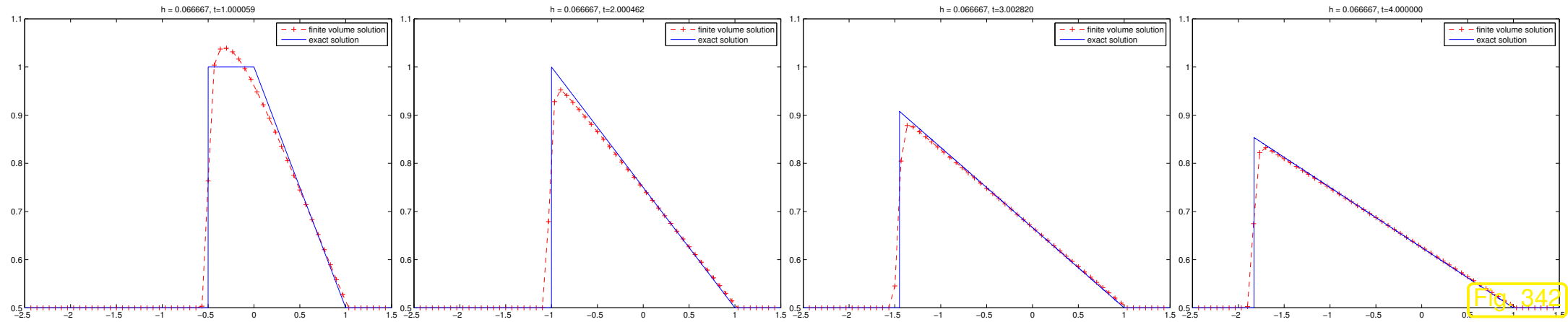


Fig. 342

Observation: spurious oscillations/overshoots, massive and global for (8.5.20), moderate close to shock for (8.5.21).

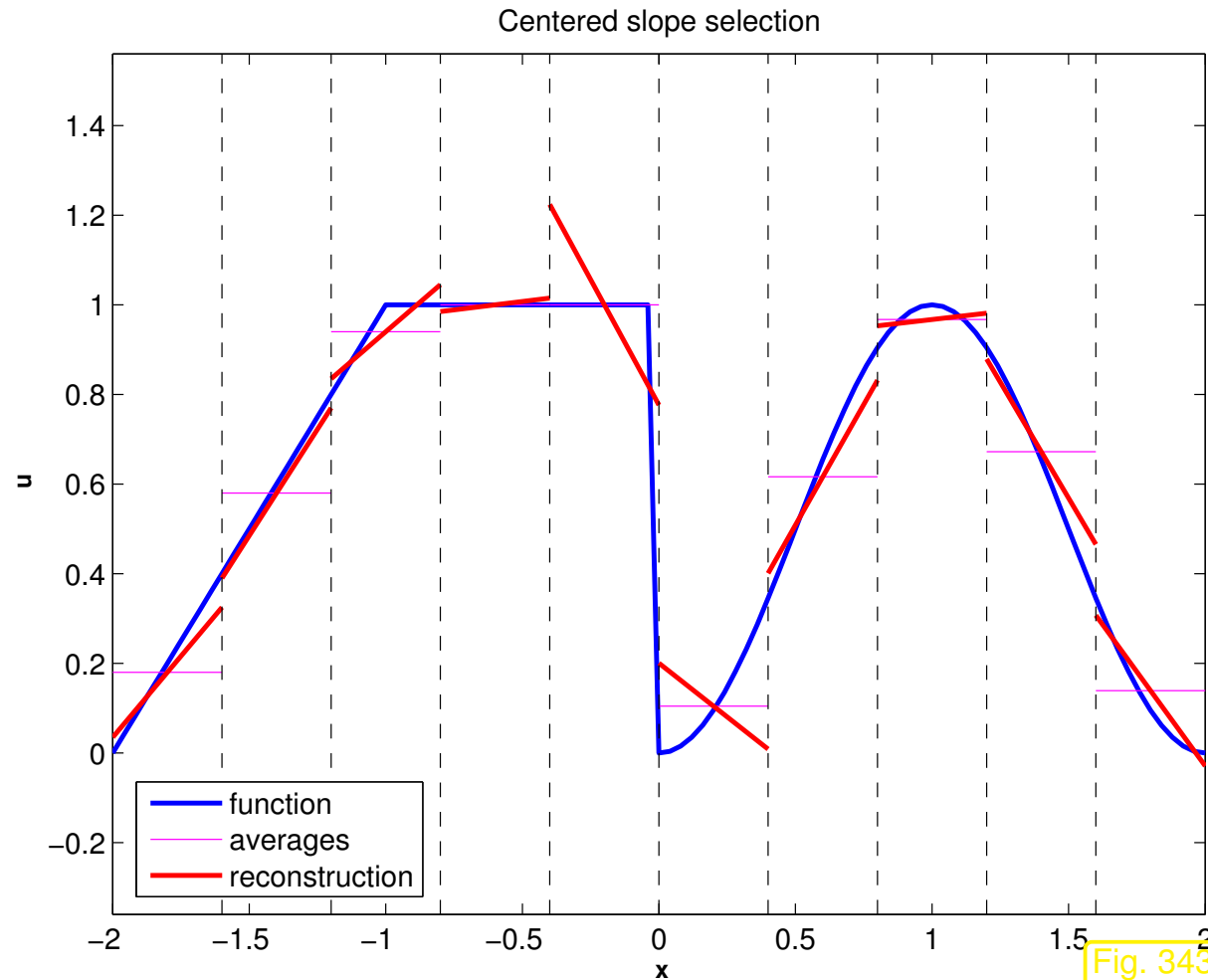
R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

It seems to be the very process of linear reconstruction that triggers oscillations near shocks. These oscillations can be traced back to “overshooting” of linear reconstruction at jumps.

Slope from central differencing:

$$\sigma_j = \frac{1}{2h}(\mu_{j+1} - \mu_{j-1}) . \quad (8.5.11)$$



Slope from forward differencing:

$$\sigma_j = \frac{1}{h}(\mu_{j+1} - \mu_j) . \quad (8.5.20)$$

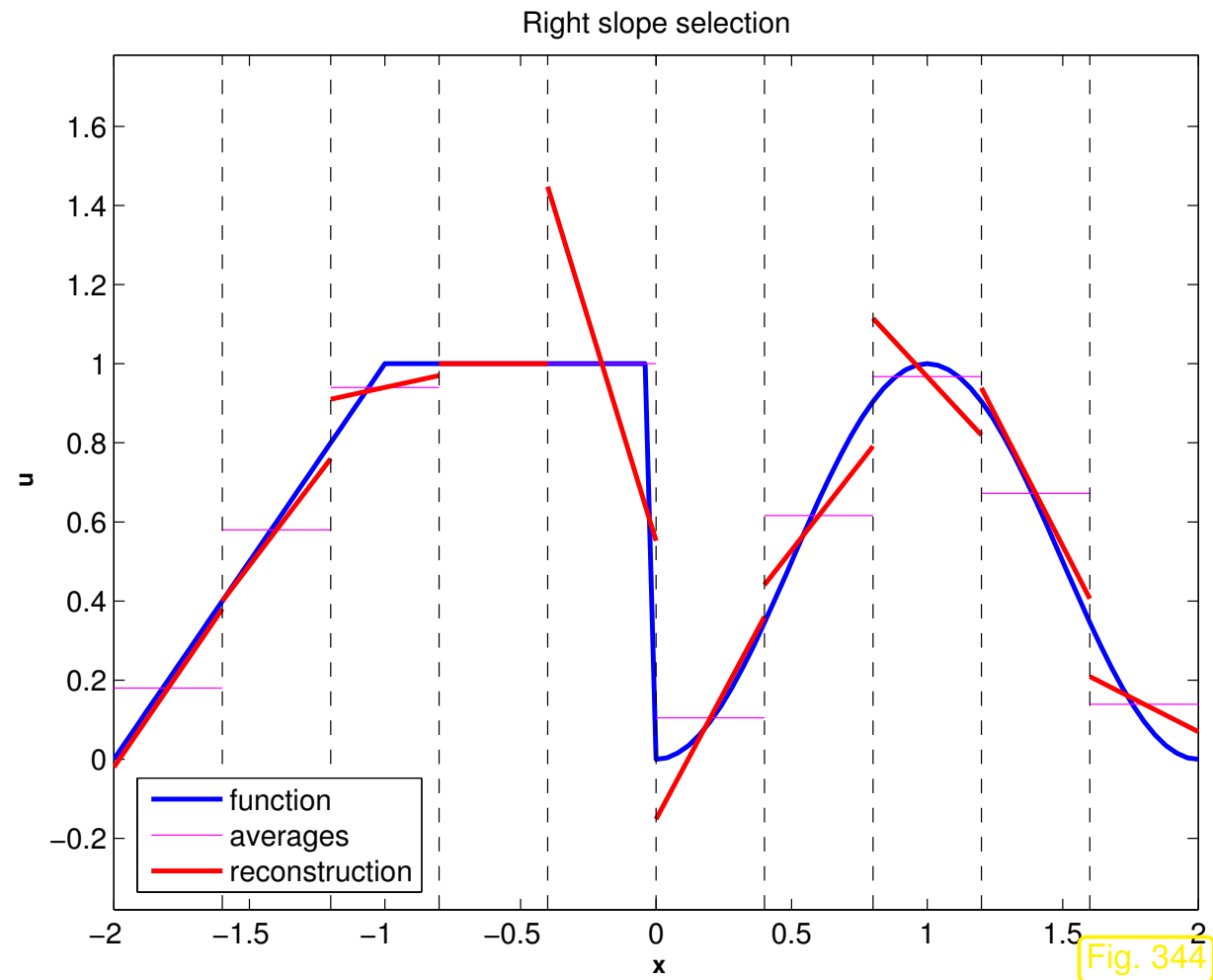
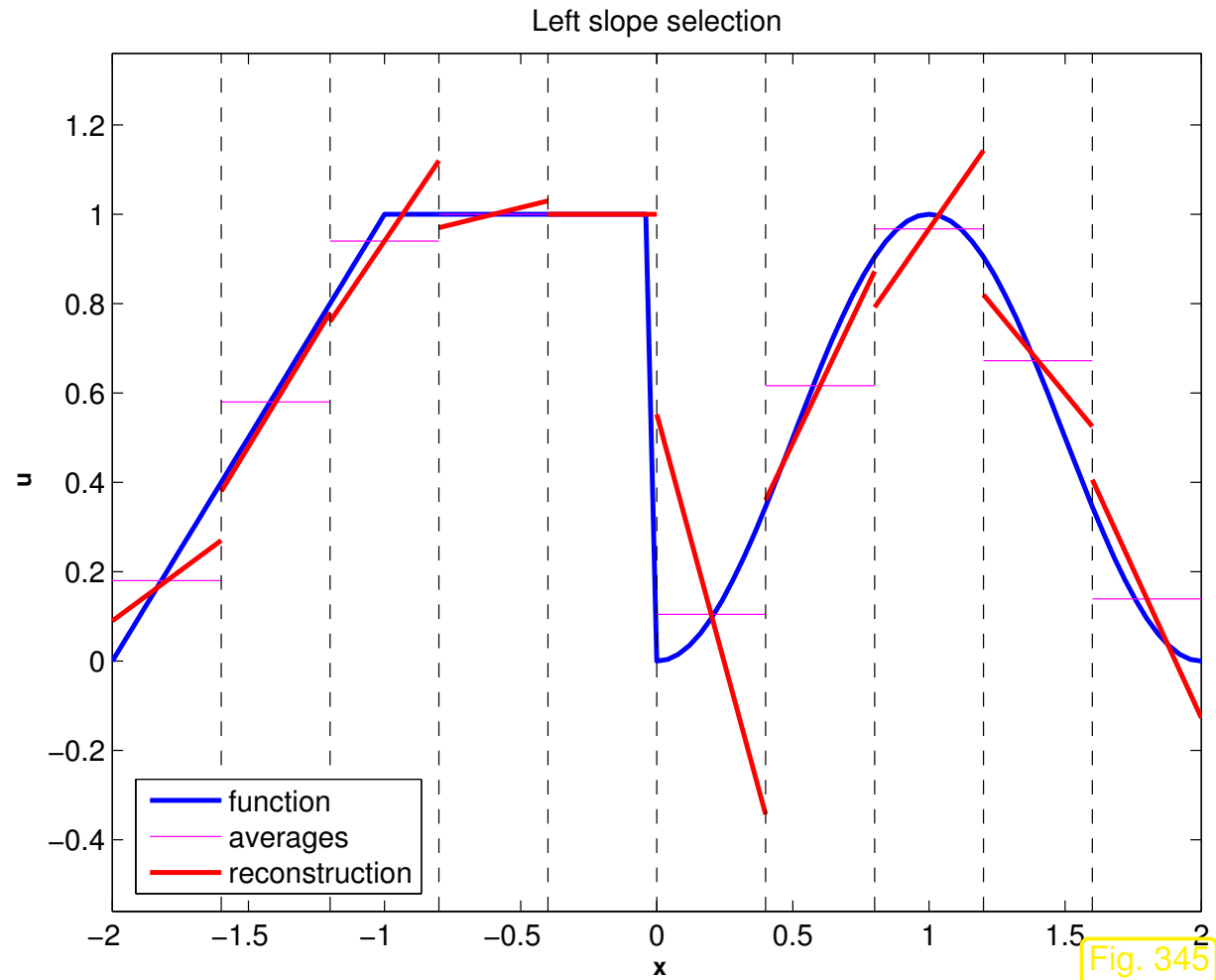


Fig. 344

Slope from backward differencing:

$$\sigma_j = \frac{1}{h}(\mu_j - \mu_{j-1}) . \quad (8.5.21)$$



8.5.2 Slope limiting

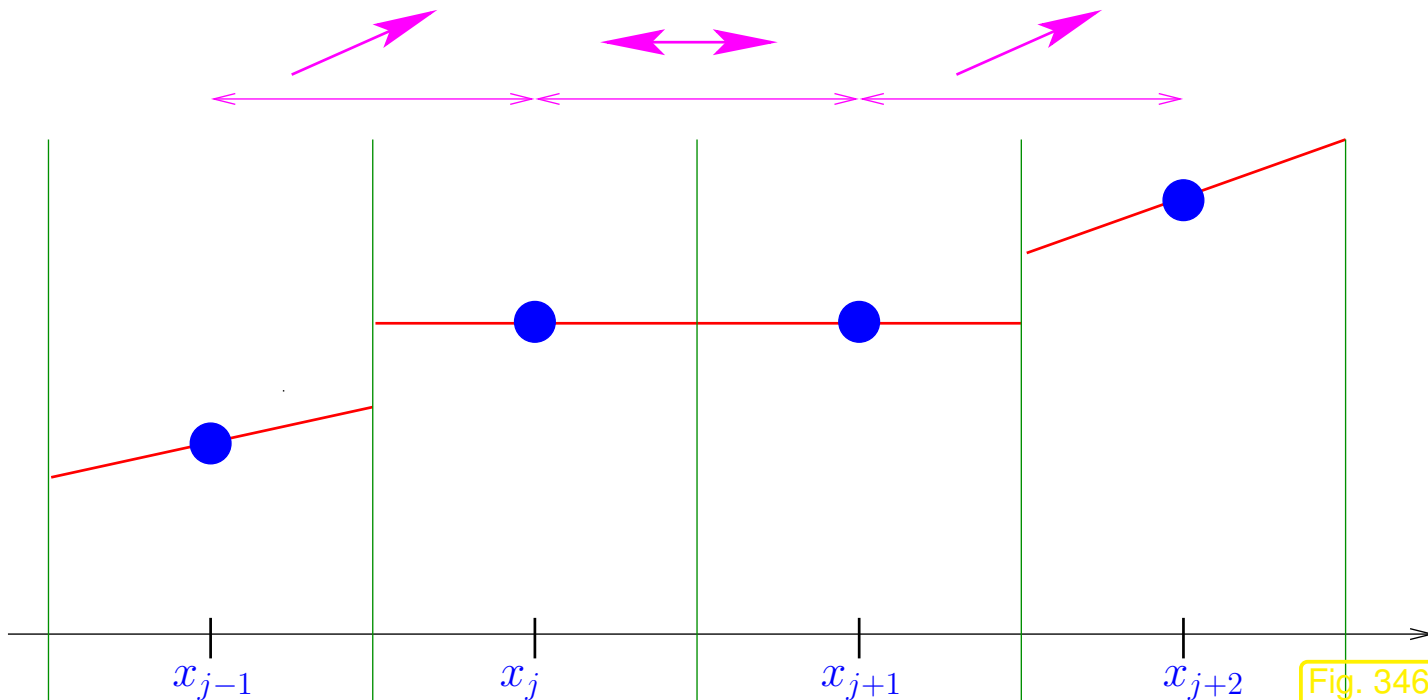
Guarantee for suppression of “overshoots” (→ Figs. 343, 344, 345)

local **monotonicity preservation** of linear reconstruction

Definition 8.5.24 (Monotonicity preserving linear interpolation).

An linear reconstruction operator $R_{\mathcal{M}}$ (→ Def. 8.5.5) is **monotonicity preserving**, if

$$(R_{\mathcal{M}}\vec{\mu})(x_j) = \mu_j \quad \wedge \quad \begin{aligned} \mu_j \leq \mu_{j+1} &\Rightarrow R_{\mathcal{M}}\vec{\mu} \text{ non-decreasing in }]x_j, x_{j+1}[, \\ \mu_j \geq \mu_{j+1} &\Rightarrow R_{\mathcal{M}}\vec{\mu} \text{ non-increasing in }]x_j, x_{j+1}[. \end{aligned}$$



Monotonicity preserving linear reconstruction:

- constant at plateaus
- constant at (local) extrema

Fig. 346

Related: **shape preserving** Hermite interpolation, see [21, Sect. 3.7.2], achieved by using

- zero slope, in case of local slopes with opposite sign, see [21, (3.7.7)],
- harmonic averaging of local slopes, see [21, (3.7.9)].

Remark 8.5.28 (Consequence of monotonicity preservation).

A monotonicity preserving linear reconstruction operator $R_{\mathcal{M}}$ (\rightarrow Def. 8.5.24)

- respects the range of cell averages

$$\min\{\mu_k, \mu_{k+1}, \dots, \mu_m\} \leq (R_{\mathcal{M}}\vec{\mu})(x) \leq \max\{\mu_k, \mu_{k+1}, \dots, \mu_m\}, \quad x_k < x < x_m. \quad (8.5.29)$$

\Leftrightarrow “range preservation” by entropy solutions, see Thm. 8.2.45.

- does not allow the creation of new extrema

$$\#\{\text{extrema of } R_{\mathcal{M}}\vec{\mu}\} \leq \#\{\text{extrema of } \vec{\mu}\}. \quad (8.5.30)$$

\Leftrightarrow preservation of number of extrema in entropy solution, Sect. 8.2.7.



Remark 8.5.34 (Linearity and monotonicity preservation).

The linear reconstruction operators (\rightarrow Def. 8.5.5) based on the slope formulas (8.5.11) (central slope), (8.5.20) (forward slope), (8.5.21) (backward slope) are *linear* in the sense that

$$R_{\mathcal{M}}(\alpha \vec{\mu} + \beta \vec{\nu}) = \alpha R_{\mathcal{M}}(\vec{\mu}) + \beta R_{\mathcal{M}}(\vec{\nu}) \quad \forall \vec{\mu}, \vec{\nu} \in \mathbb{R}^{\mathbb{Z}}, \alpha, \beta \in \mathbb{R}. \quad (8.5.35)$$

Lemma 8.5.36 (Linear monotonicity preserving reconstruction trivial).

Every linear, monotonicity preserving (\rightarrow Def. 8.5.24) linear reconstruction yields piecewise constant functions.

Proof. Define $\vec{\epsilon}^k \in \mathbb{R}^{\mathbb{Z}}$, $k \in \mathbb{Z}$, by


$$\epsilon_j^k = \begin{cases} 1 & \text{for } k = j, \\ 0 & \text{else.} \end{cases}$$

The $\vec{\epsilon}^k$ form a basis of $\mathbb{R}^{\mathbb{Z}}$. Thus, due to linearity, $R_{\mathcal{M}}$ is fixed by its action on the basis vectors $\vec{\epsilon}^k$ and its image is spanned by $\{R_{\mathcal{M}}\vec{\epsilon}^k\}_{k \in \mathbb{Z}}$.

However, monotonicity preservation entails that $R_{\mathcal{M}}\vec{\epsilon}^k$ is piecewise constant, see Fig. 346. □

► Necessary (for monotonicity preservation): Non-linear linear reconstruction

!?



A simple consideration, see Fig. 346

$$\mu_{j-1} \leq \mu_j \quad \text{and} \quad \mu_j \geq \mu_{j+1} \quad \Rightarrow \quad R_{\mathcal{M}}\vec{\mu} \equiv \text{const} \quad \text{on} \quad]x_{j-1/2}, x_{j+1/2}[, \quad (8.5.38)$$

for any monotonicity preserving (\rightarrow Def. 8.5.24) linear reconstruction operator $R_{\mathcal{M}}$ (\rightarrow Def. 8.5.5).

➤ monotonicity preserving linear reconstruction $R_{\mathcal{M}}\vec{\mu}$ must be constant at local extrema of $\vec{\mu}$!



Definition 8.5.39 (Minmod reconstruction).

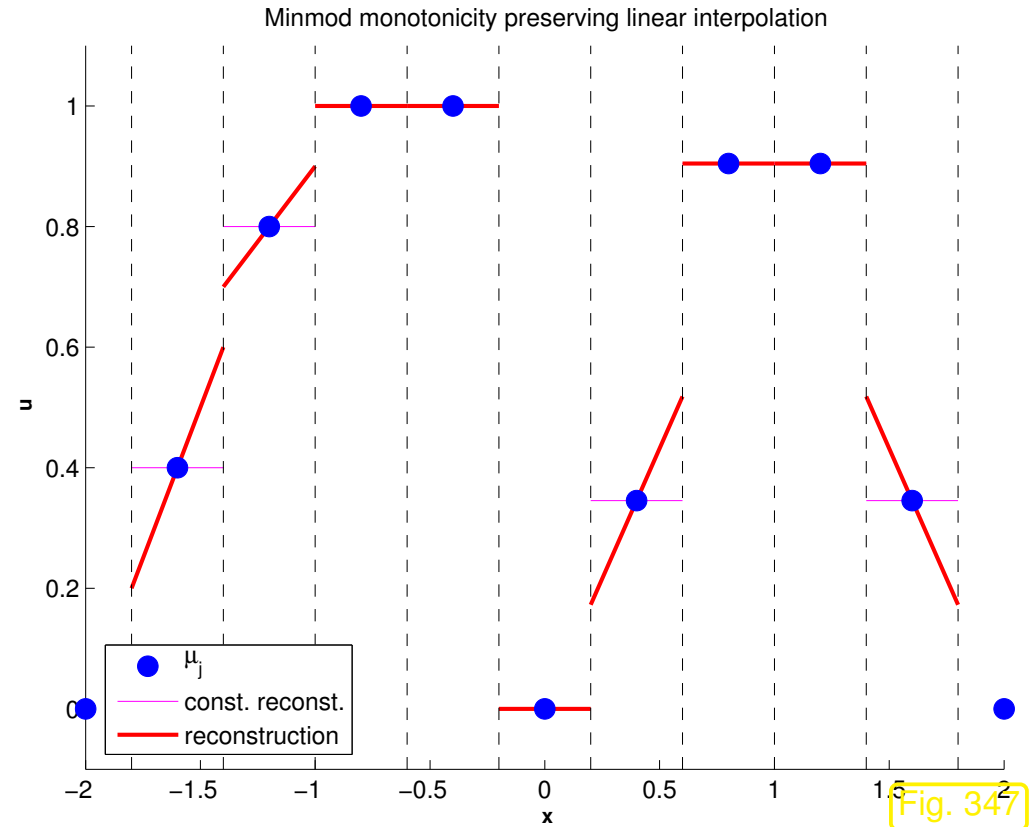
The *minmod reconstruction* R_{mm} is a piecewise linear reconstruction (\rightarrow Def. 8.5.5) defined by

$$(R_{\text{mm}}\bar{\mu})(x) = \mu_j + \sigma_j(x - x_j)$$

for $x_{j-1/2} < x < x_{j+1/2}, j \in \mathbb{Z}$,

$$\sigma_j := \text{minmod} \left(\frac{\mu_{j+1} - \mu_j}{x_{j+1} - x_j}, \frac{\mu_j - \mu_{j-1}}{x_j - x_{j-1}} \right),$$

$$\text{minmod}(v, w) := \begin{cases} v & , vw > 0, |v| < |w|, \\ w & , vw > 0, |w| < |v|, \\ 0 & , vw \leq 0. \end{cases}$$



R. Hiptmair
 C. Schwab,
 H. Harbrecht
 V. Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

Lemma 8.5.40 (Monotonicity preservation of minmod reconstruction).

Minmod reconstruction (\rightarrow Def. 8.5.39) is monotonicity preserving (\rightarrow Def. 8.5.24)

Proof. w.l.o.g. assume $\mu_{j+1} \geq \mu_j \Rightarrow \sigma_j \geq 0 \wedge \sigma_{j+1} \geq 0$
 $\Rightarrow \mu_j + \frac{1}{2}h\sigma_j \leq \frac{1}{2}(\mu_j + \mu_{j+1}) \leq \mu_{j+1} - \frac{1}{2}h\sigma_{j+1}$ □

Terminology: effect of minmod-function in R_{mm} : **slope limiting**: minmod = **slope limiter**

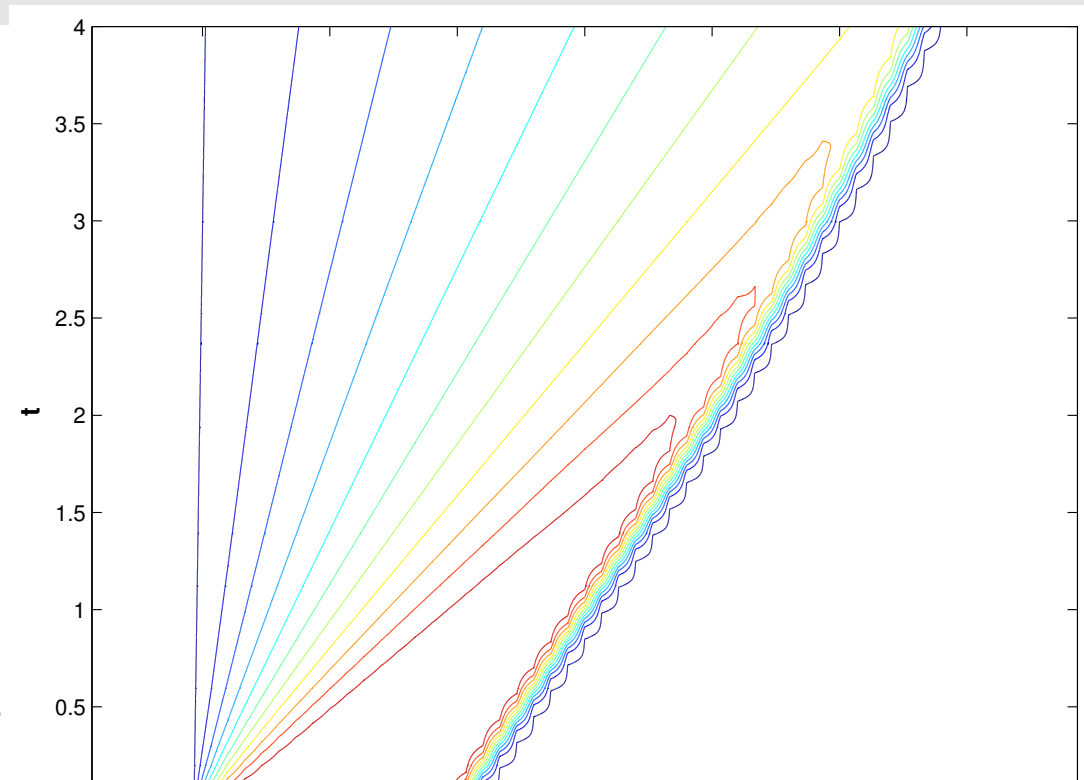
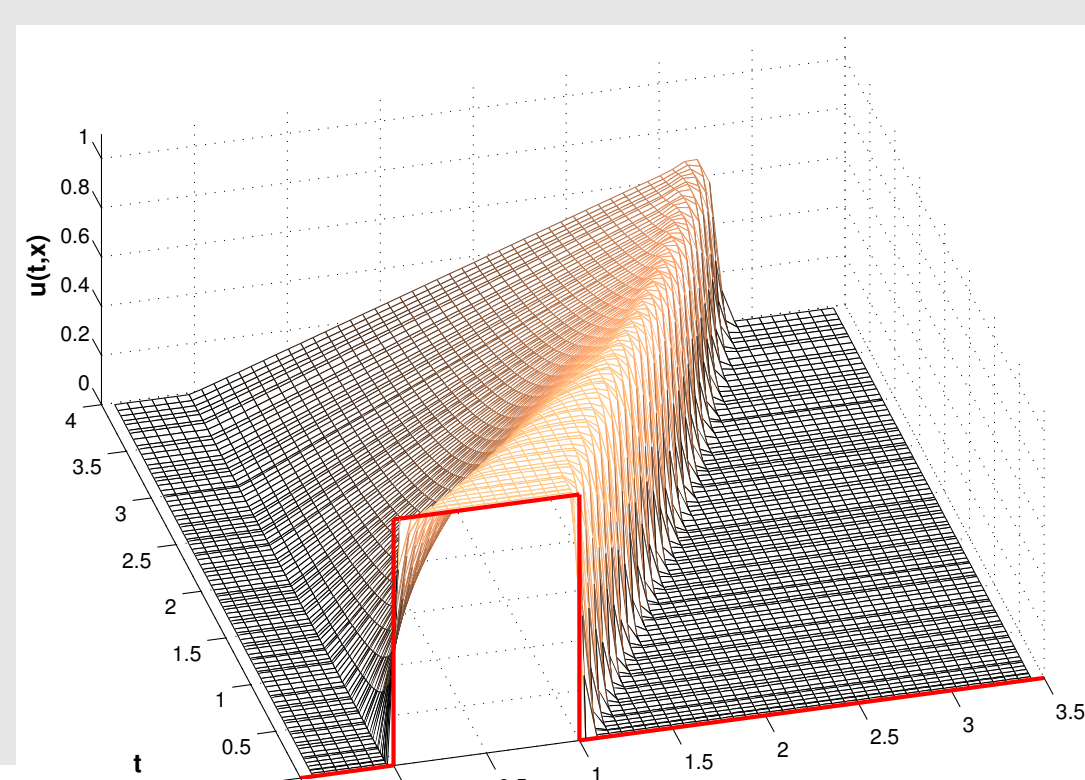
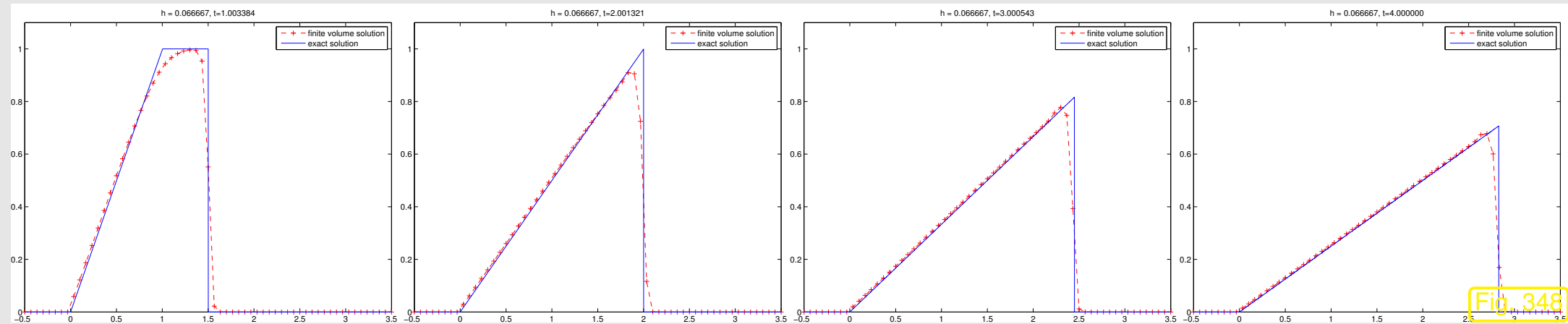
Example 8.5.42 (Linear reconstruction with minmod limiter (Burgers' equation)).

Same setting as in Ex. 8.5.14, Cauchy problem as in Ex. 8.3.21:

- Cauchy problem for Burgers equation (8.1.60) (flux function $f(u) = \frac{1}{2}u^2$) from Ex. 8.2.43 (“box” initial data)
- Equidistant spatial mesh with meshwidth $h = \frac{1}{15}$
- Linear reconstruction with minmod limited slope (\rightarrow Def. 8.5.39)

$$\sigma_j := \text{minmod} \left(\frac{\mu_j - \mu_{j-1}}{h}, \frac{\mu_{j+1} - \mu_j}{h} \right) .$$

- Godunov numerical flux (8.3.51): $F = F_{GD}$
- timestepping based on adaptive Runge-Kutta method `ode45` of MATLAB (`opts = odeset('abstol', 1E-7, 'reltol', 1E-6);`).



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Observation: spurious oscillations successfully suppressed!

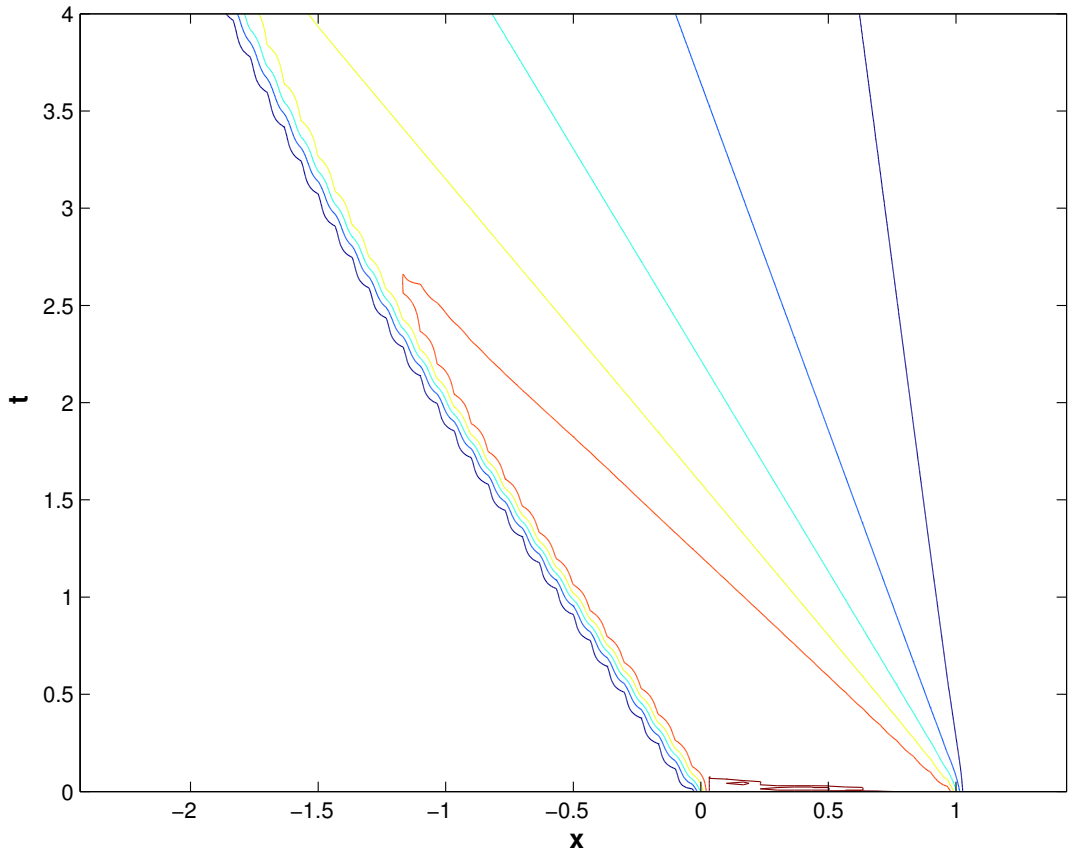
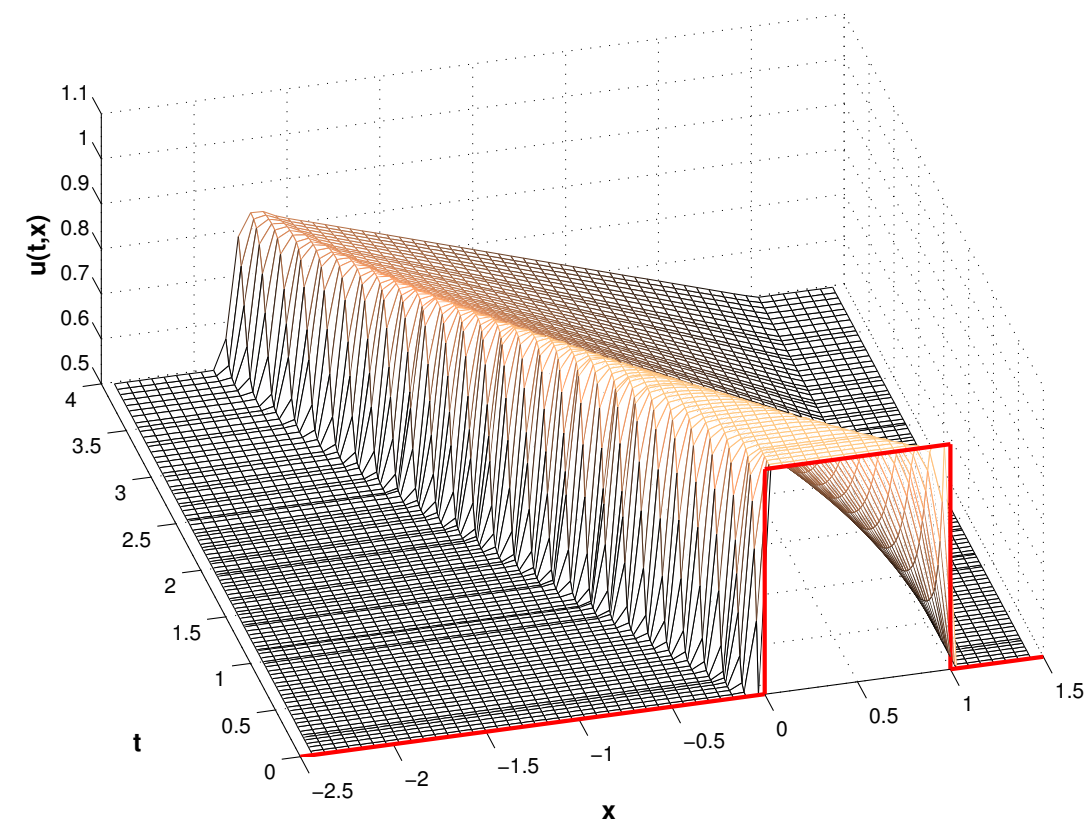
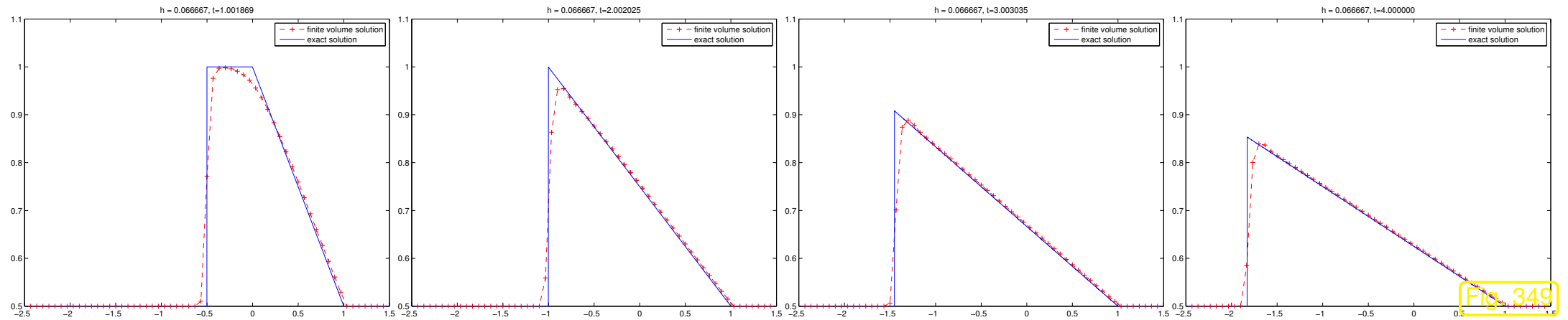
Example 8.5.43 (Linear reconstruction with minmod limiter).

Same setting as in Ex. 8.5.14, Cauchy problem as in Ex. 8.3.23:

- Cauchy problem for Traffic Flow equation (8.1.53) (flux function $f(u) = u(1-u)$) from Ex. 8.2.44 (“box” initial data)
- Equidistant spatial mesh with meshwidth $h = \frac{1}{15}$
- Linear reconstruction with minmod limited slope (\rightarrow Def. 8.5.39)

$$\sigma_j := \text{minmod} \left(\frac{\mu_j - \mu_{j-1}}{h}, \frac{\mu_{j+1} - \mu_j}{h} \right) .$$

- Godunov numerical flux (8.3.51): $F = F_{\text{GD}}$
- timestepping based on adaptive Runge-Kutta method `ode45` of MATLAB (`opts = odeset('abstol', 1E-7, 'reltol', 1E-6);`).



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs
SAM, ETHZ

Observation: spurious oscillations successfully suppressed!



Example 8.5.44 (Improved resolution by limited linear reconstruction).

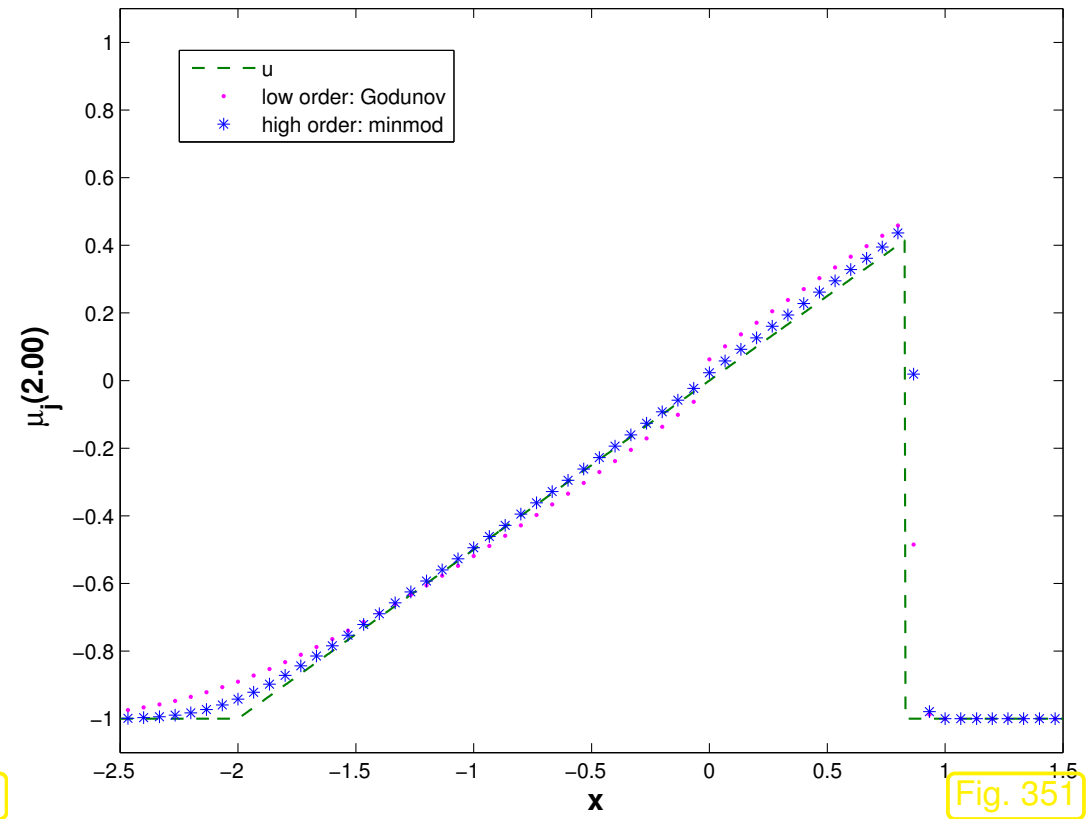
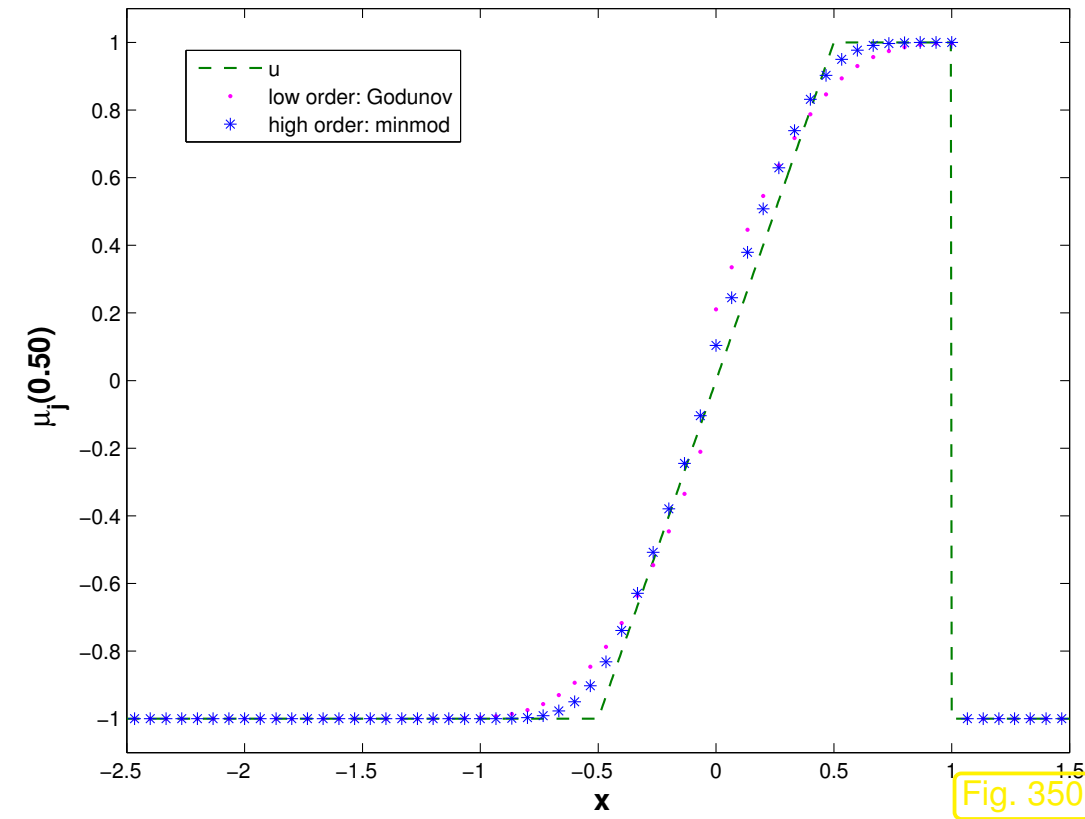
- Same setting as in Ex. 8.3.37: Cauchy problem for Burgers equation (8.1.60) (flux function $f(u) = \frac{1}{2}u^2$) from Ex. 8.2.43 (shifted “box” initial data, $u_0(x) = -1$ for $x \notin [0, 1]$, $u_0(x) = 1$ for $x \in [0, 1]$)
- Equidistant spatial mesh with meshwidth $h = \frac{1}{15}$
- “High-order” method based on linear reconstruction with minmod limited slope (\rightarrow Def. 8.5.39)

$$\sigma_j := \text{minmod} \left(\frac{\mu_j - \mu_{j-1}}{h}, \frac{\mu_{j+1} - \mu_j}{h} \right) .$$

- Godunov numerical flux (8.3.51): $F = F_{\text{GD}}$
- timestepping based on adaptive Runge-Kutta method `ode45` of MATLAB (`opts = odeset('abstol', 1E-10, 'reltol', 1E-8);`).

Burgers equation (transsonic rarefaction), N = 60

Burgers equation (transsonic rarefaction), N = 60



Observation: *Better resolution* of rarefaction fan compared with the conservative finite volume method based on of Godunov numerical flux without linear reconstruction. Good resolution of shock.

This improved resolution is the main rationale for the use of piecewise linear reconstruction.

8.5.3 MUSCL scheme

= Monotone Upwind Scheme for Conservation Laws

Case of equidistant spatial mesh with meshwidth $h > 0$:

- Conservative finite volume spatial discretization (8.5.1) with monotone consistent 2-point flux, e.g., Godunov numerical flux (8.3.51)
- Piecewise linear reconstruction (\rightarrow Def. 8.5.5) with **minmod** slope limiting (\rightarrow Def. 8.5.39):

$$\nu_j^\pm := \mu_j \pm \frac{1}{2} \text{minmod}(\mu_{j+1} - \mu_j, \mu_j - \mu_{j-1}) . \quad (8.5.47)$$

- 2nd-order Runge-Kutta timestepping for (8.5.1): **method of Heun**, cf. (8.4.6):

If the right hand side of (8.5.1) is abbreviated by

$$\mathcal{L}_h(\vec{\mu}) := -\frac{1}{h} (F(\nu_j^+(t), \nu_{j+1}^-(t)) - F(\nu_{j-1}^+(t), \nu_j^-(t))) ,$$

then the fully discrete scheme (uniform timestep $\tau > 0$) reads 8.5.1

$$\begin{aligned}\vec{\kappa} &:= \vec{\mu}^{(k)} + \frac{1}{2}\tau\mathcal{L}_h(\vec{\mu}^{(k)}) , \\ \vec{\mu}^{(k+1)} &:= \vec{\mu}^{(k)} + \tau h\mathcal{L}_h(\vec{\kappa}) .\end{aligned}\tag{8.5.48}$$

Example 8.5.49 (Adequacy of 2nd-order timestepping).

- Same setting as in Ex. 8.3.37: Cauchy problem for Burgers equation (8.1.60) (flux function $f(u) = \frac{1}{2}u^2$) from Ex. 8.2.43 (shifted “box” initial data, $u_0(x) = -1$ for $x \notin [0, 1]$, $u_0(x) = 1$ for $x \in [0, 1]$)
- Equidistant spatial mesh with meshwidth $h = \frac{1}{15}$
- Linear reconstruction with minmod limited slope (\rightarrow Def. 8.5.39)

$$\sigma_j := \text{minmod} \left(\frac{\mu_j - \mu_{j-1}}{h}, \frac{\mu_{j+1} - \mu_j}{h} \right) .$$

- Godunov numerical flux (8.3.51): $F = F_{\text{GD}}$

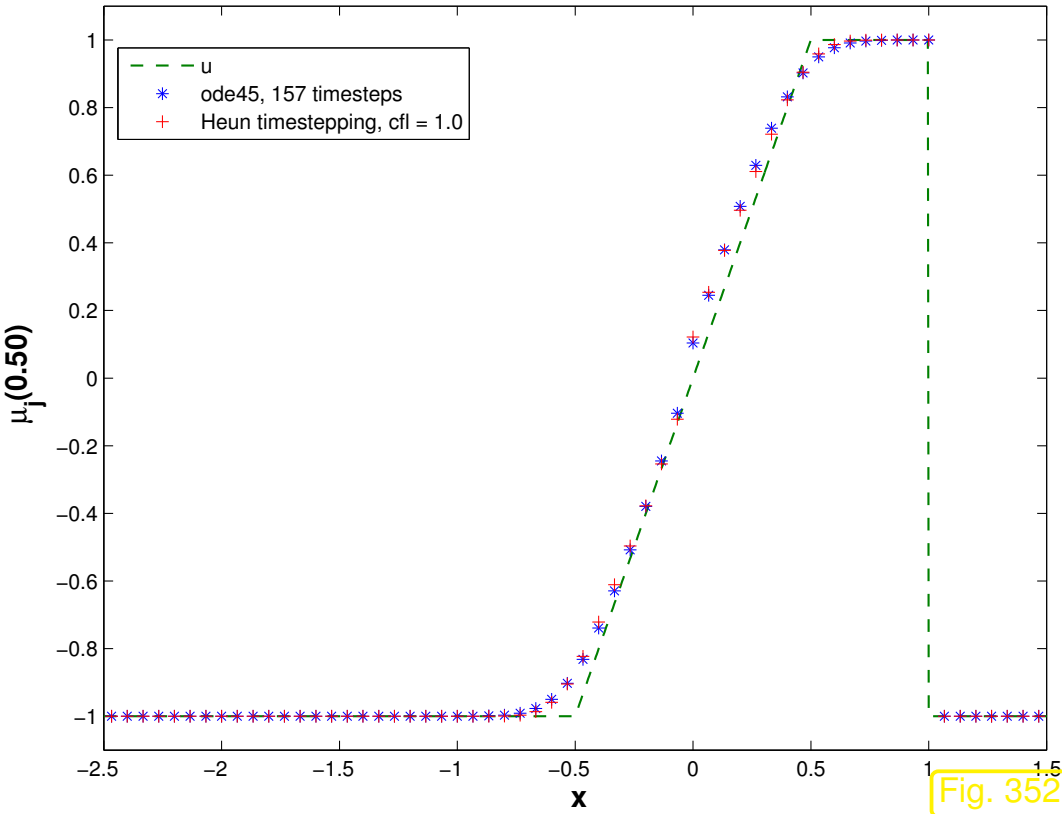
Two options for timestepping

1. timestepping based on adaptive Runge-Kutta method `ode45` of MATLAB

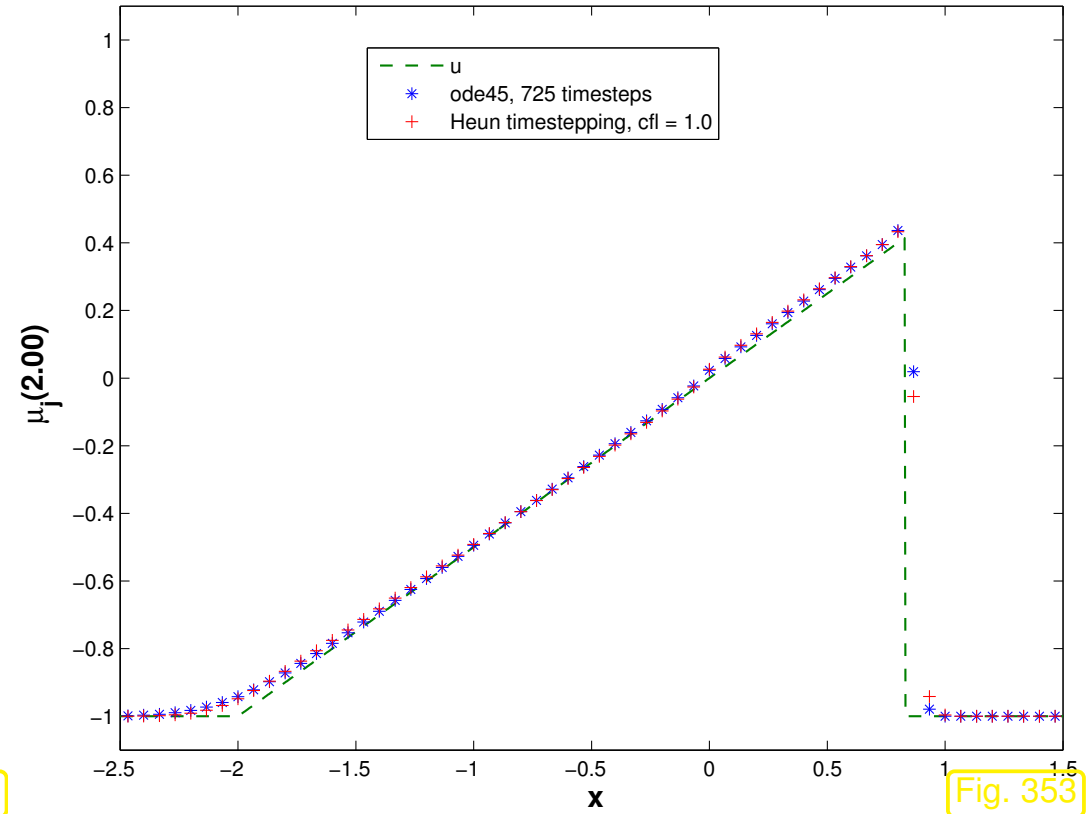
```
(opts = odeset('abstol', 1E-10, 'reltol', 1E-8);).
```

2. Heun timestepping (8.5.48) with uniform timestep $\tau = h$

Burgers equation (transsonic rarefaction), N = 60



Burgers equation (transsonic rarefaction), N = 60



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Observation: 2nd-order Runge-Kutta method (8.5.48) provides same accuracy as “overkill integration” by means of `ode45` with tight tolerances.

➤ For the sake of efficiency balance order of spatial and temporal discretizations and use Heun timestepping.



Example 8.5.50 (Convergence of MUSCL scheme).

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Numerical experiments of Ex. 8.4.30 repeated for

- conservative finite volume discretization with Godunov numerical flux and `minmod`-limited linear reconstruction, see Ex. 8.5.42 (`ode45` timestepping),
- MUSCL scheme as introduced above with fixed timestep $\tau = 0.5h$.

Monitored: “discrete” error norms (8.4.31), (8.4.32)

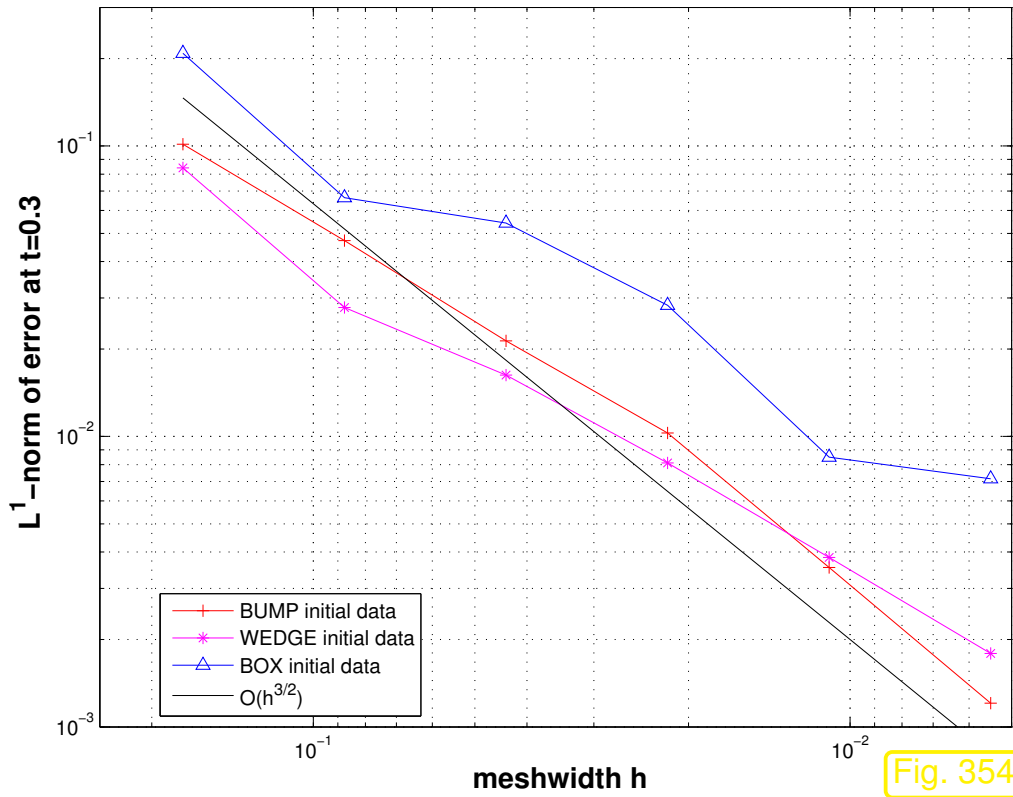


Fig. 354

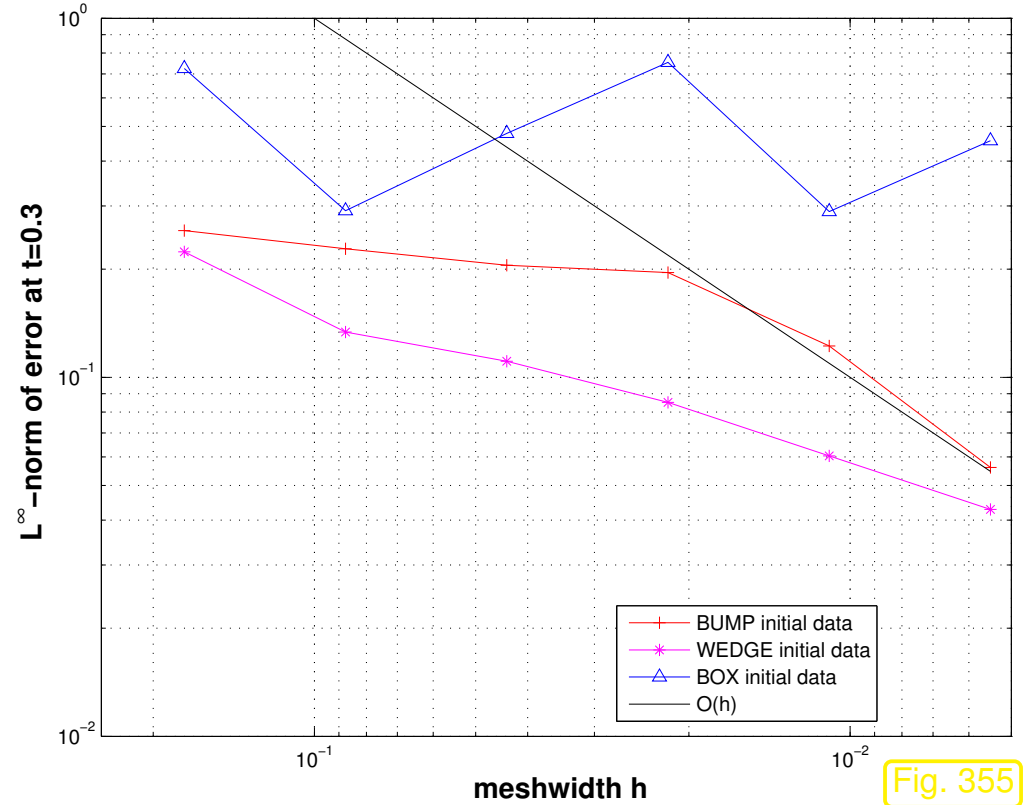


Fig. 355

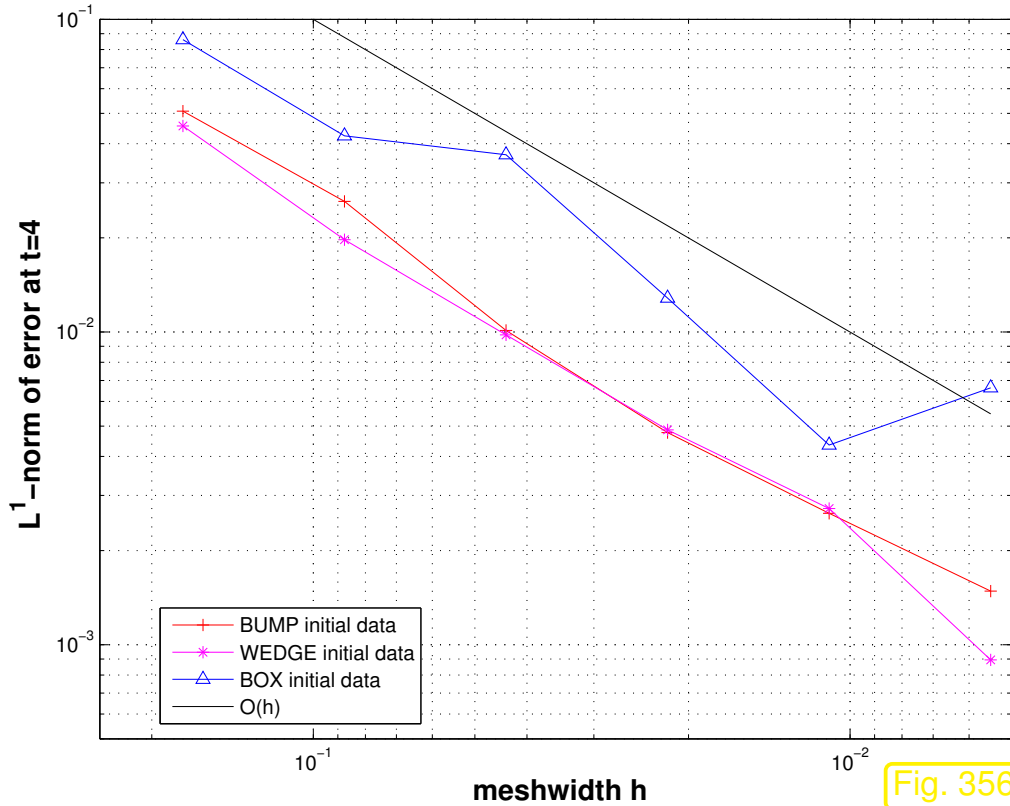


Fig. 356

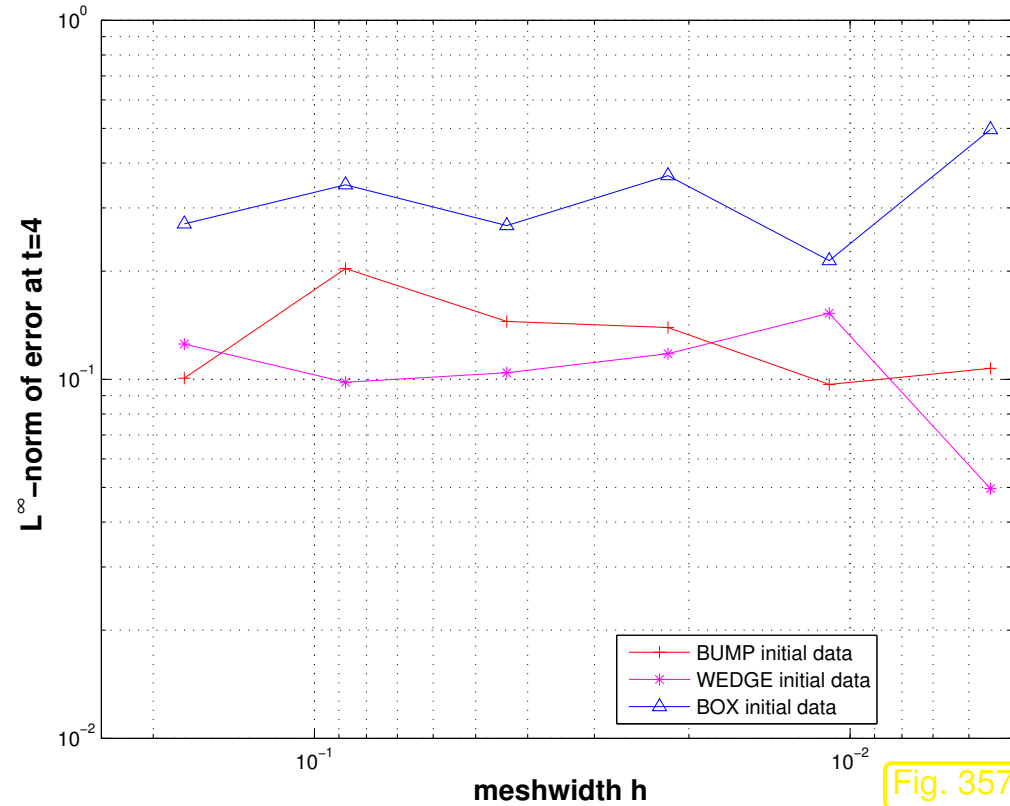


Fig. 357

Observation: 2nd-order Heun method produces solutions whose convergence and accuracy matches those of solutions obtained by highly accurate high-order Runge-Kutta timestepping.



8.6 Outlook: systems of conservation laws

9

Finite Elements for the Stokes Equations



Supplementary and further reading:

Books (chapters) dealing with the (mathematical foundations of) discretization of the Stokes boundary value problem:

[18, Ch. 12]: Concise introduction to the Stokes PDEs and Galerkin discretization. Modelling is not discussed.

[6, Ch. 12]: Numerical analysis of variational saddle point problems

[4, III.§5]: Concise presentation of principles of finite element discretization of the Stokes problem

9.1 Viscous fluid flow

Task: simulation of *stationary fluid flow*

computation of the velocity $\mathbf{v} = \mathbf{v}(\mathbf{x})$ of a fluid moving in a container $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, under the influence of an external force field $\mathbf{f} : \Omega \mapsto \mathbb{R}^d$.

($d = 2?$ \leftrightarrow translational symmetry \rightarrow dimensionally reduced model)

 notation: as before, bold typeface for vector valued functions

Recall: description of fluid motion through a velocity field \rightarrow Sect. 7.1.1

We restrict ourselves to *incompressible fluids* \rightarrow Def. 7.1.7

Thm. 7.1.12 \blacktriangleright **Constraint** $\operatorname{div} \mathbf{v} = 0$. (9.1.2)

$$V := \left\{ \mathbf{v} : \bar{\Omega} \mapsto \mathbb{R}^d \text{ continuous, } \operatorname{div} \mathbf{v} = 0 \right\}. \quad (9.1.3)$$

Flow regimes of an incompressible **Newtonian fluid** (a fluid, for which stress is linearly proportional to strain) are distinguished by the size of a fundamental *non-dimensional* quantity, the

$$\text{Reynolds number} \quad \operatorname{Re} := \frac{\rho V L}{\mu},$$

- where (for $d = 3$)
- $\rho \hat{=}$ density ($[\rho] = \text{kg m}^{-3}$)
 - $V \hat{=}$ mean velocity ($[V] = \text{m s}^{-1}$)
 - $L \hat{=}$ characteristic length of region of interest ($[L] = \text{m}$)
 - $\mu \hat{=}$ dynamic viscosity ($[\mu] = \text{kg m}^{-1} \text{s}^{-1}$)

Reynolds number = ratio of **inertia forces** : **viscous (friction) forces**

The Reynolds number becomes small, if

- the speed of the flow is very small (slowly flowing fluids), or
- the flow is studied at tiny length scales (micro flows), or
- the fluid is highly viscous (“sticky”).

In this case acceptably accurate modelling can **neglect inertia forces** ➤ **creeping flow**

Viscous fluids “stick to the walls of the container”

$$\text{no-slip boundary conditions: } \mathbf{v} = 0 \quad \text{on } \partial\Omega . \quad (9.1.5)$$

► **configuration space** for viscous incompressible fluid

$$V := \left\{ \begin{array}{l} \mathbf{v} : \bar{\Omega} \mapsto \mathbb{R}^d \text{ continuous,} \\ \operatorname{div} \mathbf{v} = 0, \quad \mathbf{v}|_{\partial\Omega} = 0 \end{array} \right\} . \quad (9.1.6)$$

We appeal to an extremal principle to derive governing equations for incompressible creeping flow: the state of the system renders a physical quantity minimal.

For the elastic string (\rightarrow Sect. 1.2), taut membrane (\rightarrow Sect. 2.1.1), electrostatic field (\rightarrow Sect. 2.1.2) this quantity was the total potential energy. For stationary viscous fluid flow, this role is played by the energy dissipation:

energy dissipation = conversion of kinetic energy into internal energy (heat)
(\leftrightarrow entropy production)

AXIOM: **energy dissipation** functional for viscous fluid ($[P_{\text{diss}}] = W$)

$$P_{\text{diss}}(\mathbf{v}) = \int_{\Omega} \mu \|\mathbf{curl} \mathbf{v}(\mathbf{x})\|^2 \, d\mathbf{x} \quad (9.1.7)$$

rotation/curl $\hat{=}$ first-order differential operator

$$\mathbf{curl} \mathbf{v} := \begin{pmatrix} \frac{\partial v_2}{\partial x_3} - \frac{\partial v_3}{\partial x_2} \\ \frac{\partial v_3}{\partial x_1} - \frac{\partial v_1}{\partial x_3} \\ \frac{\partial v_1}{\partial x_2} - \frac{\partial v_2}{\partial x_1} \end{pmatrix} \quad \text{for } d = 3, \quad \mathbf{curl} \mathbf{v} := \frac{\partial v_1}{\partial x_2} - \frac{\partial v_2}{\partial x_1} \quad \text{for } d = 2. \quad (9.1.8)$$

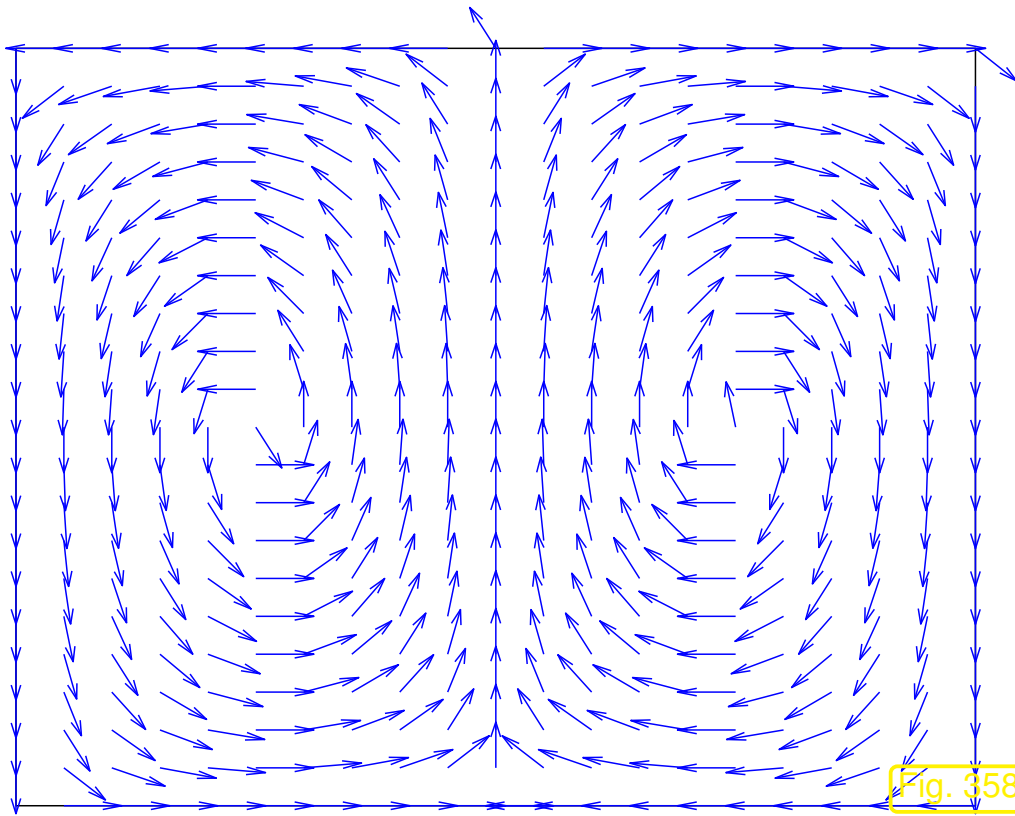


Fig. 358

“eddy field”, $\text{div } \mathbf{v} = 0$

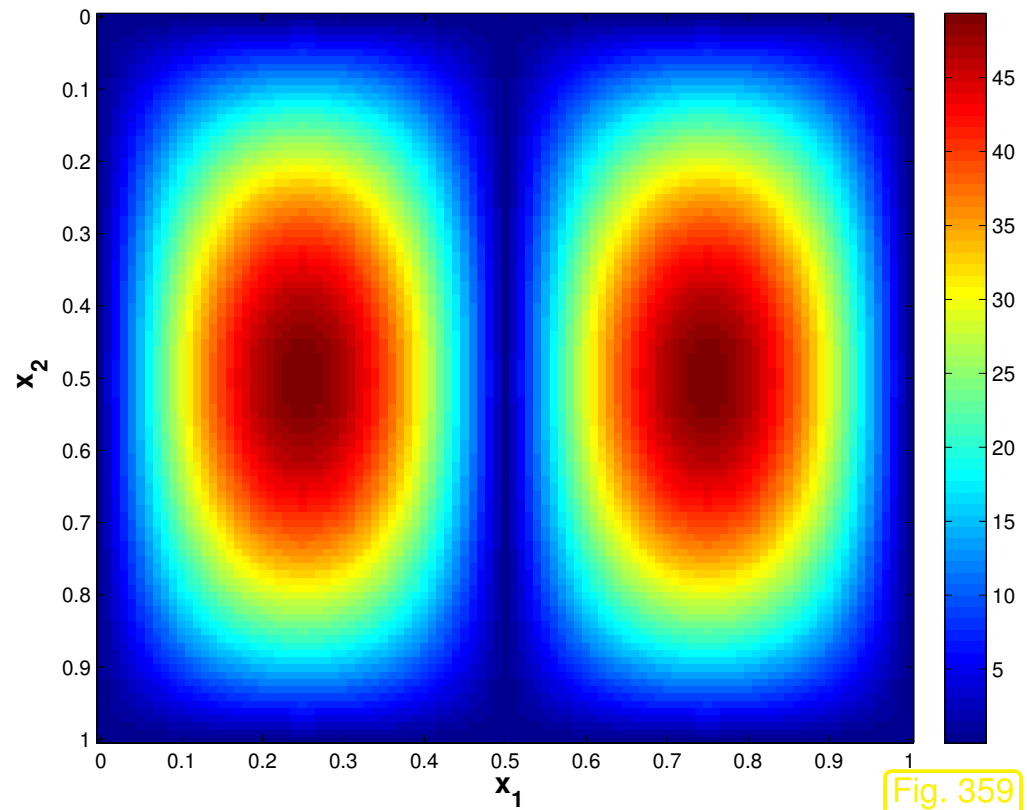


Fig. 359

Plot of $\|\text{curl } \mathbf{v}\|$ for eddy field

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

$\text{curl } \mathbf{v}$ = “density of eddies/vortices in flow field”

Thus, in viscous fluid flow the conversion of kinetic energy into heat due to friction presumably happens in vortical flow patterns (eddies).

Second law of thermodynamics for creeping flow:

$$\text{Maximization of energy dissipation in flow} \quad (9.1.9)$$

↕

$$\text{entropy production}$$

First law of thermodynamics: conservation of energy/power balance

$$\int_{\Omega} \mu \|\mathbf{curl} \mathbf{v}(\mathbf{x})\|^2 \, d\mathbf{x} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x} . \quad (9.1.10)$$

↗ ↖

dissipated energy energy injected through forces

► First equilibrium condition for viscous stationary flow:

$$\mathbf{v}^* = \operatorname{argmax} \left\{ \int_{\Omega} \mu \|\mathbf{curl} \mathbf{v}(\mathbf{x})\|^2 \, d\mathbf{x} : \mathbf{v} \in V, \mathbf{v} \text{ satisfies (9.1.10)} \right\} \quad (9.1.14)$$

≐ *constrained optimization problem* with constraint (9.1.10).

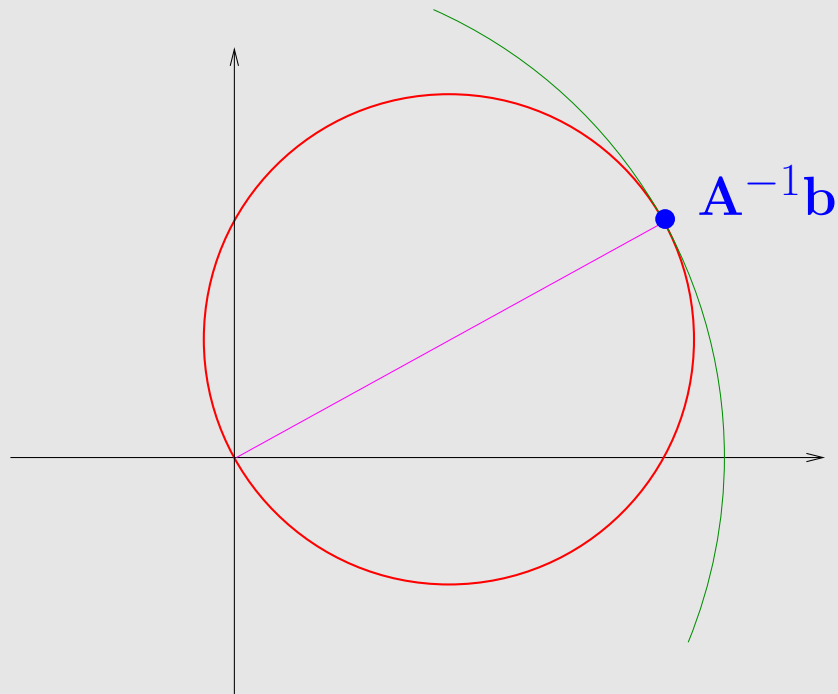
Goal: Convert (9.1.14) into a “more standard” optimization problem.

To that end we study a related problem in finite dimensional context \mathbb{R}^n :

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{b}^T \mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{x}, \quad (9.1.15)$$

with s.p.d. $\mathbf{A} \in \mathbb{R}^{n,n}$, $\mathbf{b} \in \mathbb{R}^n$. With the transformation $\mathbf{y} = \mathbf{A}^{-1/2} \mathbf{x}$ (\rightarrow [21, Rem. 5.3.2]) we arrive at the equivalent maximization problem

$$\mathbf{y}^* = \operatorname{argmax}_{\|\mathbf{y}\|^2 = (\mathbf{A}^{-1/2} \mathbf{b})^T \mathbf{y}} \|\mathbf{y}\|^2.$$



The set $\{\mathbf{y} : \|\mathbf{y}\|^2 = (\mathbf{A}^{-1/2} \mathbf{b})^T \mathbf{y}\}$ is a sphere through 0 around $\frac{1}{2} \mathbf{A}^{-1/2} \mathbf{b}$ and we are looking for its point farthest away from 0 . By “geometric considerations” this will be the point $\mathbf{y}^* = \mathbf{A}^{-1/2} \mathbf{b} \succ \mathbf{x}^* = \mathbf{A}^{-1} \mathbf{b}$.

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Recall: relationship between linear systems of equations and quadratic minimization problems, see [21, Sect. 5.1.1] and Sect. 2.1.3.

► $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$ can be obtained as solution of

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} . \quad (9.1.16)$$

To have faith that this reasoning applies to (9.1.14) as well, the bilinear form $(\mathbf{u}, \mathbf{v}) \mapsto \int_{\Omega} \mathbf{curl} \mathbf{u} \cdot \mathbf{curl} \mathbf{v} \, d\mathbf{x}$ should be positive definite (\rightarrow Def. 2.1.32) ► see Lemma 9.2.1 below.

Another issue, of course, is, whether the above arguments remain true for (infinite dimensional) function spaces ► theory of variational calculus [34, Ch. 49], not elaborated here.

► Second **equilibrium condition** for viscous stationary flow, *cf.* (2.1.4), (2.1.15):

$$\mathbf{v}^* = \operatorname{argmin}_{\mathbf{v} \in V} \frac{1}{2} \int_{\Omega} \mu \|\mathbf{curl} \mathbf{v}(\mathbf{x})\|^2 \, d\mathbf{x} - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x} . \quad (9.1.17)$$

9.2 The Stokes equations

9.2.1 Constrained variational formulation

Lemma 9.2.1 ($-\Delta = \mathbf{curl\,curl} - \mathbf{grad\,div}$).

For $\mathbf{v} \in C^2(\bar{\Omega})$, $\mathbf{v}|_{\partial\Omega} = 0$, holds

$$\int_{\Omega} \|\mathbf{curl\,v}\|^2 \, d\mathbf{x} + \int_{\Omega} |\mathbf{div\,v}|^2 \, d\mathbf{x} = \int_{\Omega} \|D\mathbf{v}\|_F^2 \, d\mathbf{x} .$$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

✎ notations: $D\mathbf{v} := \left(\frac{\partial v_i}{\partial x_j} \right)_{i,j=1}^d : \Omega \mapsto \mathbb{R}^{d,d}$ Jacobian,
 $\|\mathbf{M}\|_F \hat{=} \text{Frobenius matrix norm}$ (\rightarrow [21, Def. 6.5.35])

Proof (of Lemma 9.2.1)

Use the variant of Green's first formula Thm. 2.4.11

$$\int_{\Omega} \frac{\partial u}{\partial x_j} v \, d\mathbf{x} = - \int_{\Omega} \frac{\partial v}{\partial x_j} u \, d\mathbf{x} \quad \forall u, v \in C^1(\bar{\Omega}), \quad u, v = 0 \quad \text{on } \partial\Omega, \quad (9.2.3)$$

and the fact that different partial derivatives can be interchanged, which implies

$$\int_{\Omega} \frac{\partial u}{\partial x_j} \frac{\partial v}{\partial x_k} \, d\mathbf{x} = \int_{\Omega} \frac{\partial u}{\partial x_k} \frac{\partial v}{\partial x_j} \, d\mathbf{x}, \quad k, j = 1, \dots, d.$$

Then use the definitions of **curl** and **div**. □

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

In light of the properties $\operatorname{div} \mathbf{v} = 0$, $\mathbf{v} = 0$ on $\partial\Omega$, for eligible fluid velocity fields, see (9.1.6), we have the equivalence:

$$(9.1.17) \quad \stackrel{\text{Lemma 9.2.1}}{\iff} \quad \mathbf{v}^* = \operatorname{argmin}_{\mathbf{v} \in V} \underbrace{\frac{1}{2} \int_{\Omega} \mu \|D\mathbf{v}\|_F^2 \, d\mathbf{x}}_{=: a(\mathbf{v}, \mathbf{v})} - \underbrace{\int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x}}_{=: \ell(\mathbf{v})}. \quad (9.2.4)$$

$\hat{=}$ **quadratic minimization problem** (\rightarrow Def. 2.1.26) on function space V .

Rewrite quadratic form ($\mu \equiv \text{const}$)

$$\mathbf{v} = (v_1, \dots, v_d)^T: \quad \int_{\Omega} \mu \|D\mathbf{v}\|_F^2 \, d\mathbf{x} = \mu \sum_{i=1}^d \|\mathbf{grad} v_i\|^2 \, d\mathbf{x} .$$

By the first Poincaré-Friedrichs inequality of Thm. 2.2.25

$$\|\mathbf{v}\|_{L^2(\Omega)}^2 \leq \text{diam}(\Omega)^2 \int_{\Omega} \|D\mathbf{v}\|_F^2 \, d\mathbf{x} \quad \forall \mathbf{v} \in V \subset (H_0^1(\Omega))^3 .$$

► Bilinear form \mathbf{a} from (9.2.4) is positive definite (\rightarrow Def. 2.1.32).

Remark 9.2.7 (Decoupling of velocity components ?).

Rewrite (9.2.4) in terms of components v_i of velocity (with force field $\mathbf{f} = (f_1, f_2, f_3)^T$):

$$(9.2.4) \quad \Leftrightarrow \quad \underset{\mathbf{v} \in V}{\operatorname{argmin}} \sum_{i=1}^3 \left(\frac{1}{2} \int_{\Omega} \mu \|\mathbf{grad} v_i\|^2 \, d\mathbf{x} - \int_{\Omega} f_i v_i \, d\mathbf{x} \right) . \quad (9.2.8)$$

Well, three copies of (2.1.15) ?!

NO! $\operatorname{div} \mathbf{v} = 0$ constraint (9.1.2) links components of velocity field \mathbf{v} .

This constraint in the space V represents the crucial difference compared to minimization problems (2.1.4), (2.1.15) underlying scalar 2nd-order elliptic variational equations.

As in Sect. 2.2: put (9.2.4) into Hilbert space (more precisely, Sobolev space) framework, where we have existence and uniqueness of solutions.

(9.2.8) offers hint on how to choose suitable Sobolev spaces.

Remember: function spaces for a (linear) variational problem are chosen as the largest (Hilbert) spaces on which the involved bilinear forms and linear forms are still *continuous*, cf. (2.2.3), (3.1.2).

appropriate Sobolev space for (9.2.4):

(9.2.8), (9.1.5)



$$\mathbf{H}_0^1(\operatorname{div} 0, \Omega) := \left\{ \mathbf{v} \in (H_0^1(\Omega))^3 : \operatorname{div} \mathbf{v} = 0 \right\}$$

$((H_0^1(\Omega))^3 \hat{=}$ space of vector fields with components in $H_0^1(\Omega)$, alternative notation $\mathbf{H}_0^1(\Omega)$).

► As in Sect. 2.3.1 derive the linear variational problem

$$\mathbf{v} \in \mathbf{H}_0^1(\text{div } 0, \Omega): \quad \mathbf{a}(\mathbf{v}, \mathbf{w}) = \ell(\mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{H}_0^1(\text{div } 0, \Omega),$$

from (9.2.8), which reads in concrete terms:

Seek $\mathbf{v} \in \mathbf{H}_0^1(\text{div } 0, \Omega) := \left\{ \mathbf{v} \in (H_0^1(\Omega))^3: \text{div } \mathbf{v} = 0 \right\}$ such that

$$\int_{\Omega} \mathbf{grad} v_i \cdot \mathbf{grad} w_i \, d\mathbf{x} = \int_{\Omega} f_i w_i \, d\mathbf{x} \quad \forall \mathbf{w} \in \mathbf{H}_0^1(\text{div } 0, \Omega), \quad i = 1, 2, 3,$$

\Leftrightarrow

$$\int_{\Omega} D\mathbf{v} : D\mathbf{w} \, d\mathbf{x} = \int_{\Omega} \mathbf{f} \cdot \mathbf{w} \, d\mathbf{x} \quad \forall \mathbf{w} \in \mathbf{H}_0^1(\text{div } 0, \Omega).$$

(9.2.9)

✎ notation: $\mathbf{A} : \mathbf{B} := \sum_{i,j} a_{ij} b_{ij}$ for matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m,n}$ (“componentwise dot product”).

For this linear variational problem we verify

- Assumption 5.1.1 from Poincaré-Friedrichs inequality, see above,
- Assumption 5.1.2 for $\mathbf{f} \in (L^2(\Omega))^d$ by Cauchy-Schwarz inequality, see (2.2.24), (2.3.24),
- Assumption 5.1.3, since $\mathbf{H}_0^1(\operatorname{div} 0, \Omega)$ is a closed subspace of $\mathbf{H}^1(\Omega)$.

Thm. 5.1.4 \Rightarrow existence & uniqueness of solutions of (9.2.9)

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 9.2.11 ($\mathbf{H}_0^1(\operatorname{div} 0, \Omega)$ -conforming finite elements).

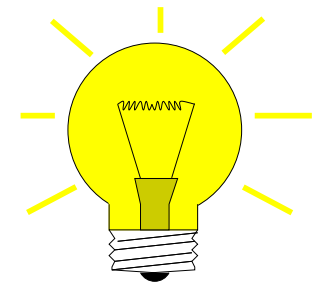
In principle, the linear variational problem could be tackled by means of a finite element Galerkin discretization.

However, finding finite element spaces $\subset \mathbf{H}_0^1(\operatorname{div} 0, \Omega)$ is complicated [30]: Continuous, piecewise polynomial, locally supported, and divergence free basis fields exist only for polynomial degree $\geq 4!$



This remark motivates an approach that removes the constraint from trial and test space (and incorporates it into the variational formulation).

9.2.2 Saddle point problem



Idea:

weak enforcement of divergence constraint (9.1.2)
through **Lagrange multiplier**

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Remark 9.2.20 (Heuristics behind Lagrangian multipliers).

Setting:

- $U, Q \hat{=}$ real Hilbert spaces with inner products $(\cdot, \cdot)_U, (\cdot, \cdot)_Q$,
- $J : U \mapsto \mathbb{R}$ convex and differentiable functional,
- $B : U \mapsto Q$ linear operator (defining constraint)

Linearly **constrained minimization problem**

$$v^* = \operatorname{argmin}_{v \in U, Bv=0} J(v) . \quad (9.2.21)$$

Introduce **Lagrangian functional**:

$$L(v, p) := J(v) + (p, Bv)_Q \quad \blacktriangleright \quad v^* = \operatorname{argmin}_{v \in U} \sup_{p \in Q} L(v, p) , \quad (9.2.22)$$

because, if $Bv \neq 0$, the value of the inner supremum will be $+\infty$, and, thus, such a v can never be a candidate for a minimizer.

Terminology: p is called a **Lagrange multiplier**, Q the multiplier space.

Terminology: a min-max problem like (9.2.22) = **saddle point problem**

Lemma 9.2.23 (Necessary conditions for solution of saddle point problem). \rightarrow [34, Ch. 50]

Any solution v^ of (9.2.22) will be the first component of a zero (v^*, p^*) of the derivative (“gradient”) of the Lagrangian functional L .*

► (v^*, p^*) will satisfy

$$\begin{aligned} \lim_{t \rightarrow 0} \frac{L(v^* + tw, p^*) - L(v^*, p^*)}{t} &= 0 \quad \forall w \in U, \\ \lim_{t \rightarrow 0} \frac{L(v^*, p^* + tq) - L(v^*, p^*)}{t} &= 0 \quad \forall q \in Q. \end{aligned} \tag{9.2.24}$$

because by the very structure of the saddle point problem, see Fig. 360 for illustration,

$$L(v^*, p) \leq L(v^*, p^*) \leq L(v, p^*) \quad \forall v \in U, p \in Q. \tag{9.2.25}$$

Computing these “directional derivatives” as in Sect. 1.3.1 (for the elastic string energy functional there), we obtain

$$\begin{aligned} \langle DJ(v^*), w \rangle + (p^*, Bw)_Q &= 0 \quad \forall w \in U, \\ (q, Bv^*)_Q &= 0 \quad \forall q \in Q. \end{aligned} \tag{9.2.26}$$

This is a **variational saddle point problem**.

Special case: quadratic functional $J : U \mapsto \mathbb{R} \rightarrow$ Def. 2.1.18

$$J(v) := \frac{1}{2} \mathbf{a}(v, v) - \ell(v),$$

with a positive definite, symmetric bilinear form $\mathbf{a} : U \times U \mapsto \mathbb{R}$ (\rightarrow Defs. 1.3.23, 2.1.32), continuous linear form $\ell : U \mapsto \mathbb{R}$.

(2.3.12)



$$\langle DJ(v^*), w \rangle = \mathbf{a}(v^*, w) - \ell(w), \quad w \in U.$$

In this special case (9.2.26) becomes a **linear variational saddle point problem**:

Seek $v^* \in U, p^* \in Q$

$$\begin{aligned} \mathbf{a}(v^*, w) + (p^*, \mathbf{B}w)_Q &= \ell(w) \quad \forall w \in U, \\ (q, \mathbf{B}v^*)_Q &= 0 \quad \forall q \in Q. \end{aligned} \tag{9.2.27}$$

For rigorous mathematical treatment of constrained optimization in Banach spaces refer to [34, Ch. 49 & Ch. 50]. A discussion in finite-dimensional setting is given in [21, Sect. 7.4.1].

Solution of min-max problem:

saddle point
(non-extremal critical point)

The saddle point is a minimum when approached from the “ U -direction”, and a maximum, when approached from the “ Q -direction”.

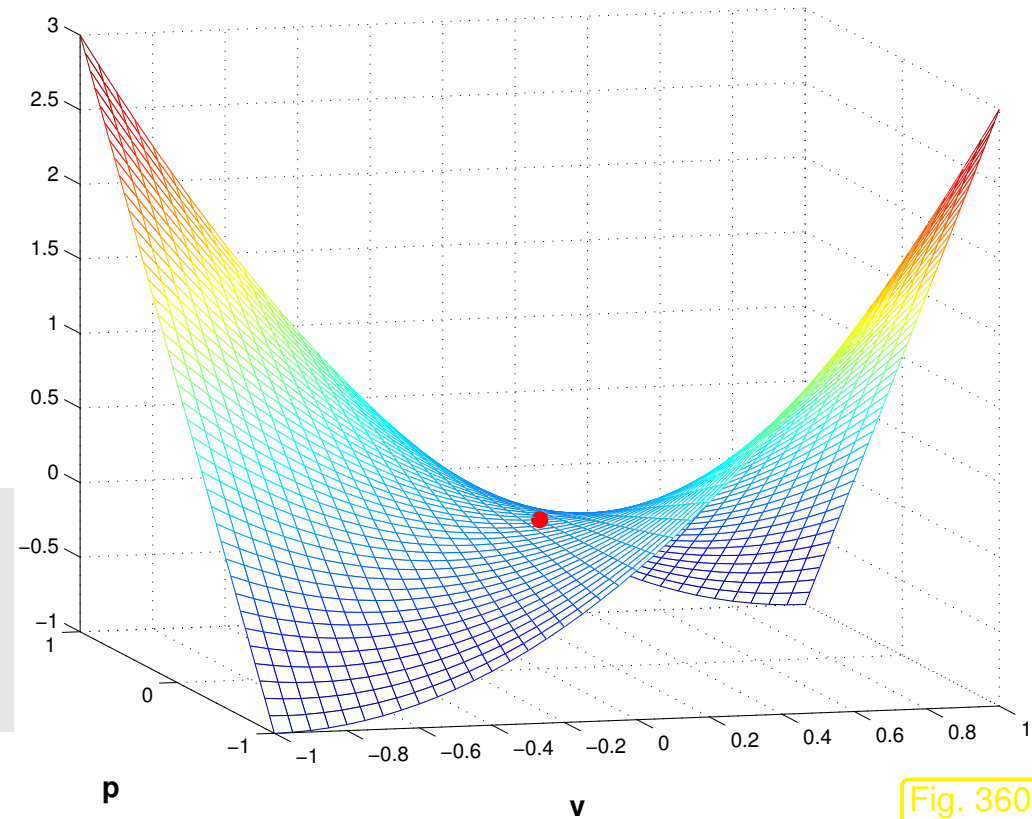


Fig. 360

Adapt abstract approach outline in Rem. 9.2.20 to (9.2.9):

- Hilbert spaces: $U = \mathbf{H}_0^1(\Omega)$, $Q = L^2(\Omega)$,
- Constraint $\operatorname{div} \mathbf{v} = 0 \quad \triangleright \quad \mathbf{B} := \operatorname{div} : U \mapsto Q$ continuous,
- $J \leftrightarrow \mathbf{v} \mapsto \frac{1}{2} \int_{\Omega} \mu \|D\mathbf{v}\|^2 \, d\mathbf{x} - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x}$, a strictly convex quadratic functional (\rightarrow Def. 2.1.18)

Lagrangian functional for (9.2.9)

$$L(\mathbf{v}, p) = \frac{1}{2} \int_{\Omega} \mu \|D\mathbf{v}\|_F^2 \, d\mathbf{x} - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x} + \int_{\Omega} \operatorname{div} \mathbf{v} p \, d\mathbf{x}, \quad \mathbf{v} \in \mathbf{H}_0^1(\Omega), \quad p \in L^2(\Omega). \quad (9.2.28)$$

Next use formula for derivative of quadratic functionals, see Sect. 2.3.1, (2.3.12), which yields a concrete specimen of (9.2.27).



Stokes problem: Linear variational saddle point problem for viscous flow (preliminary version)

seek velocity $\mathbf{v} \in \mathbf{H}_0^1(\Omega)$, Lagrange multiplier $p \in L^2(\Omega)$

$$\begin{aligned} \int_{\Omega} \mu D\mathbf{v} : D\mathbf{w} \, d\mathbf{x} + \int_{\Omega} \operatorname{div} \mathbf{w} p \, d\mathbf{x} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{w} \, d\mathbf{x} & \forall \mathbf{w} \in \mathbf{H}_0^1(\Omega), \\ \int_{\Omega} \operatorname{div} \mathbf{v} q \, d\mathbf{x} &= 0 & \forall q \in L^2(\Omega). \end{aligned}$$

Lagrange multiplier p = **pressure** ($[p] = \text{N m}^{-2}$)

No *differential constraints* in test/trial spaces for (9.2.33)!

Remark 9.2.31 (Ensuring uniqueness of pressure).

Notice:
$$\int_{\Omega} \operatorname{div} \mathbf{v} \, d\mathbf{x} = \int_{\partial\Omega} \mathbf{v} \cdot \mathbf{n} \, dS = 0, \text{ since } \mathbf{v}|_{\partial\Omega} = 0.$$

► Pressure solution p in (9.2.33) can be unique only up to a constant!

Compare: Non-uniqueness of solution of 2nd-order elliptic Neumann problem, Rem. 2.8.20.

Remedy, cf. (2.8.23)

Choose
$$p \in L_*^2(\Omega) := \left\{ q \in L^2(\Omega) : \int_{\Omega} q \, d\mathbf{x} = 0 \right\} . \quad (9.2.32)$$

↔ constraint on trial/test space $L^2(\Omega)$



Stokes problem: Variational saddle point problem for viscous flow

seek velocity $\mathbf{v} \in \mathbf{H}_0^1(\Omega)$, Lagrange multiplier $p \in L_*^2(\Omega)$

$$\begin{aligned} \int_{\Omega} \mu D\mathbf{v} : D\mathbf{w} \, d\mathbf{x} + \int_{\Omega} \operatorname{div} \mathbf{w} p \, d\mathbf{x} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{w} \, d\mathbf{x} & \forall \mathbf{w} \in \mathbf{H}_0^1(\Omega), \\ \int_{\Omega} \operatorname{div} \mathbf{v} q \, d\mathbf{x} &= 0 & \forall q \in L_*^2(\Omega). \end{aligned} \quad (9.2.33)$$

Theorem 9.2.34 (Existence and uniqueness of weak solutions of Stokes problem).

The linear variational saddle point problem (9.2.33) (“Stokes problem”) has a unique solution.

Proof. (crude outline; this sketch of the proof is included, because its ideas carry over to the discrete setting.)

Preparatory considerations: $\mathbf{a}(\mathbf{v}, \mathbf{w}) := \int_{\Omega} \mu D\mathbf{v} : D\mathbf{w} \, d\mathbf{x}$ is an inner product on $\mathbf{H}_0^1(\Omega)$.

\mathbf{a} -orthogonal decomposition $\mathbf{H}_0^1(\Omega) = \mathbf{H}_0^1(\operatorname{div} 0, \Omega) \oplus V^\perp$

① Unique solution $\mathbf{v} \in \mathbf{H}_0^1(\operatorname{div} 0, \Omega)$ of (9.2.9) \Rightarrow unique \mathbf{v} -solution for (9.2.33)
(first test with $\mathbf{w} \in \mathbf{H}_0^1(\operatorname{div} 0, \Omega)$, then with $\mathbf{w} \in V^\perp$.)

② Use the following profound result from functional analysis [4, Thm. 5.3]:

Theorem 9.2.36 (Existence of stable velocity potentials).

$$\exists C = C(\Omega) > 0: \quad \forall q \in L_*^2(\Omega): \quad \exists \mathbf{v} \in \mathbf{H}_0^1(\Omega): \quad q = \operatorname{div} \mathbf{v} \quad \wedge \quad \|\mathbf{v}\|_{H^1(\Omega)} \leq C \|q\|_{L^2(\Omega)} .$$

Idea: Assume $\mathbf{f} = 0$, test first equation with $\mathbf{w} \in V^\perp$ satisfying $\operatorname{div} \mathbf{w} = p \Rightarrow \|p\|_{L^2(\Omega)} = 0 \Leftrightarrow p = 0$, for any pressure solution $p \in L_*^2(\Omega)$.

uniqueness of pressure solution

③ Existence of pressure solution from Riesz representation theorem (\rightarrow functional analysis) and Thm. 9.2.36, not elaborated here. \square

Remaining issue: (9.2.32) introduces another *constraint* into (9.2.33)!

Relax, Lagrangian multipliers can deal with this, too. Now we study their use to enforce a zero mean constraint in the simpler setting of 2nd-order elliptic Neumann BVPs.

Remark 9.2.39 (Enforcing zero mean). \rightarrow [3]

As in Sect. 2.4, Rem. 2.8.20, we consider a 2nd-order linear Neumann BVP (with zero Neumann boundary conditions, $h = 0$), cf. (2.8.24),

$$u \in H_*^1(\Omega): \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H_*^1(\Omega) .$$

with the *constrained* trial/test space

$$H_*^1(\Omega) := \{v \in H^1(\Omega): \int_{\Omega} v(\mathbf{x}) \, d\mathbf{x} = 0\} . \quad (2.8.23)$$

The related quadratic minimization problem reads (\rightarrow Sect. 2.1.3)

$$u = \operatorname{argmin}_{v \in H_*^1(\Omega)} J(v) \quad , \quad J(v) := \frac{1}{2} \int_{\Omega} \kappa(\mathbf{x}) \|\mathbf{grad} v\|^2 \, d\mathbf{x} - \int_{\Omega} f v \, d\mathbf{x} .$$

Idea: enforce linear constraint $\int_{\Omega} v(\mathbf{x}) \, d\mathbf{x} = 0$ by means of **Lagrangian multiplier**,
see Rem. 9.2.20

Here: scalar constraint ($Q = \mathbb{R}$) \blacktriangleright scalar multiplier $p \in \mathbb{R}$

\blacktriangleright Lagrangian functional:

$$L(v, p) = J(v) + p \int_{\Omega} v(\mathbf{x}) \, d\mathbf{x}, \quad v \in H^1(\Omega), \quad p \in \mathbb{R}.$$

\blacktriangleright related (augmented) linear variational saddle point problem, specialization of (9.2.27):

seek $u \in H^1(\Omega), p \in \mathbb{R}$

$$\begin{aligned} \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} + p \int_{\Omega} v \, d\mathbf{x} &= \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H^1(\Omega), \\ \int_{\Omega} v \, d\mathbf{x} &= 0. \end{aligned} \tag{9.2.40}$$

The same technique can be applied to (9.2.33).

\blacktriangleright Stokes variational saddle point problem with pressure normalization:

seek velocity $\mathbf{v} \in \mathbf{H}_0^1(\Omega)$, pressure $p \in L^2(\Omega)$, multiplier $\lambda \in \mathbb{R}$

$$\begin{aligned}
 \int_{\Omega} \mu \nabla \mathbf{v} : \nabla \mathbf{w} \, d\mathbf{x} + \int_{\Omega} \operatorname{div} \mathbf{w} p \, d\mathbf{x} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{w} \, d\mathbf{x} & \forall \mathbf{w} \in \mathbf{H}_0^1(\Omega), \\
 \int_{\Omega} \operatorname{div} \mathbf{v} q \, d\mathbf{x} + \lambda \int_{\Omega} q \, d\mathbf{x} &= 0 & \forall q \in L^2(\Omega), \\
 \int_{\Omega} p \, d\mathbf{x} &= 0 & .
 \end{aligned} \tag{9.2.41}$$

9.2.3 Stokes system

As in Sect. 2.4: derivation of the BVP in PDE form corresponding to (9.2.41).

Approach: Remove spatial derivatives from test functions by **integration by parts** (1.3.36)

Assuming sufficient smoothness of solution (\mathbf{v}, p) , constant μ and (9.2.41) and taking into account boundary conditions, apply Green's formula of Thm 2.4.11:

$$\int_{\Omega} \mu \nabla \mathbf{v} : \nabla \mathbf{w} \, d\mathbf{x} = \mu \sum_{i=1}^d \int_{\Omega} \mathbf{grad} v_i \cdot \mathbf{grad} w_i \, d\mathbf{x} = -\mu \sum_{i=1}^d \int_{\Omega} \Delta v_i w_i \, d\mathbf{x} ,$$

$$\int_{\Omega} \operatorname{div} \mathbf{w} p \, d\mathbf{x} = - \int_{\Omega} \mathbf{grad} p \cdot \mathbf{w} \, d\mathbf{x} .$$



$$(9.2.41) \Rightarrow \begin{cases} -\mu \Delta \mathbf{v} - \mathbf{grad} p = \mathbf{f} \\ \operatorname{div} \mathbf{v} = 0 \text{ in } \Omega , \\ \int_{\Omega} p \, d\mathbf{x} = 0 \\ \mathbf{v} = 0 \text{ on } \partial\Omega . \end{cases} \quad (9.2.42)$$



Remark 9.2.44 (Pressure Poisson equation).

Manipulating the PDEs in (9.2.42):

$$\begin{array}{l}
 \text{div} \cdot (9.2.42) \quad \blacktriangleright \quad -\mu \operatorname{div} \Delta \mathbf{v} + \operatorname{div} \operatorname{grad} p = \operatorname{div} \mathbf{f} \quad \text{in } \Omega, \\
 \quad \quad \quad \quad \quad \quad \blacktriangleright \quad -\mu \Delta(\operatorname{div} \mathbf{v}) + \Delta p = \operatorname{div} \mathbf{f} \quad \text{in } \Omega, \\
 \quad \quad \quad \quad \quad \quad \text{div } \mathbf{v} = 0 \quad \quad \quad \quad \quad \quad \blacktriangleright \quad \Delta p = \operatorname{div} \mathbf{f}.
 \end{array}$$

Appearance: (9.2.42) can be solved by solving $d + 1$ Poisson equations,

- first solve **pressure Poisson** equation $\Delta p = \operatorname{div} \mathbf{f}$
- then solve Dirichlet boundary value problems for velocity components

$$-\Delta v_i = f_i + \frac{\partial p}{\partial x_i} \quad \text{in } \Omega, \quad v_i = 0 \quad \text{on } \partial\Omega.$$

☞ above manipulations only valid for *sufficiently smooth* \mathbf{u} (not guaranteed).

Problems

☞ we cannot solve a “Poisson equation”, we also need boundary conditions for p : not available!

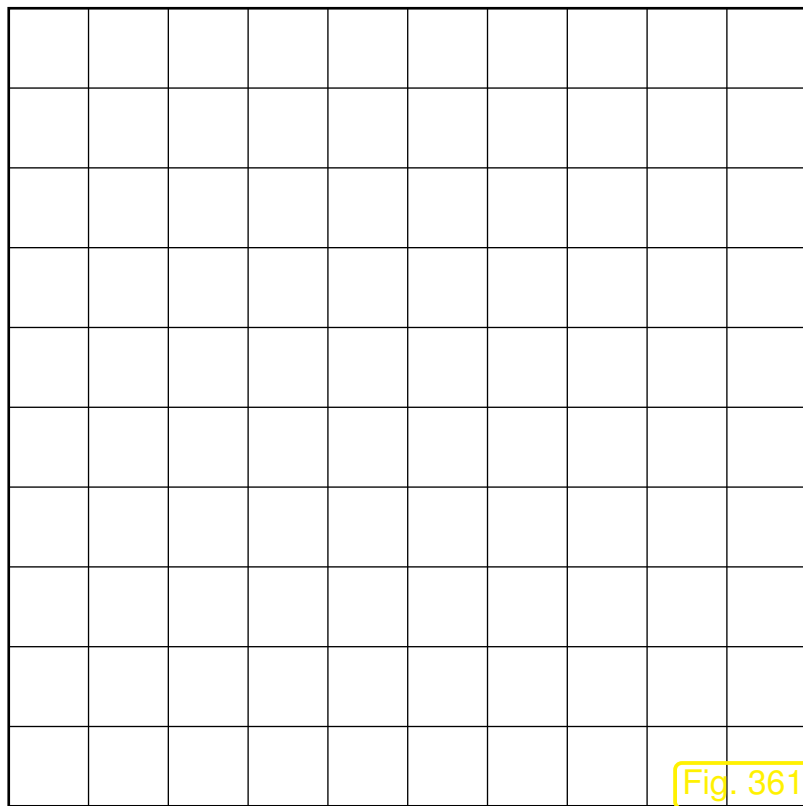


9.3 Saddle point problems: Galerkin discretization

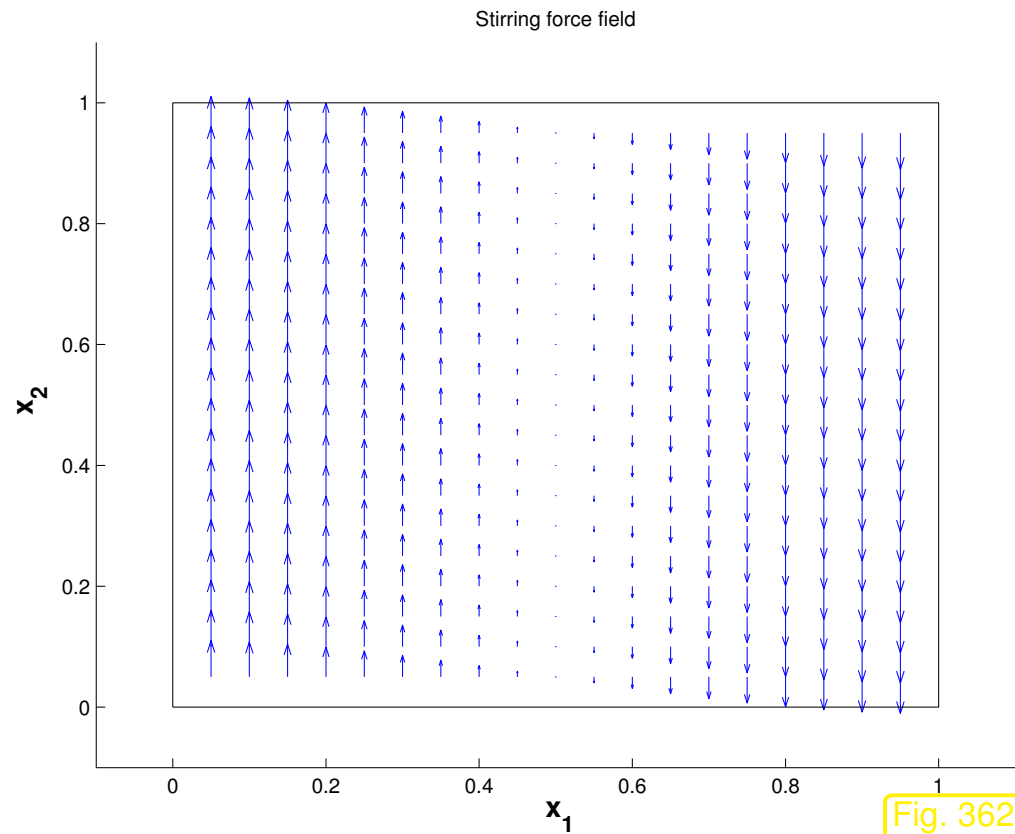
Example 9.3.1 (Naive finite difference discretization of Stokes system).

- BVP (9.2.42) on $\Omega =]0, 1[^2$, $\mu \equiv 1$, $\mathbf{f} = \cos(\pi x_1) \begin{pmatrix} 0 \\ 1 \end{pmatrix}$,
- **Finite difference** discretization on (\rightarrow Sect. 4.1) equidistant tensor product grid ▷
Unknowns: $v_{1,ij}$, $v_{2,ij}$, $p_{ij} \hat{=}$ approximations of $v_1(ih, jh)$, $v_2(ih, jh)$, $p(ih, jh)$, $0 < i, j < N$.
- Zero boundary values for v_1 , v_2 , **and** p
- 5-point stencil discretization of $-\Delta$, see (4.1.1)
- **Central** finite difference approximation of **grad** p , e.g.,

$$\frac{\partial p}{\partial x_1} \Big|_{(ih, jh)} \approx \frac{1}{2h} (p_{i+1, j} - p_{i-1, j}) \quad , \quad 1 \leq i, j < N .$$



finite difference grid



force field \mathbf{f}

Code 9.3.3: Central finite difference discretization of Stokes system

```

1 function [u1,u2,p] = StokesFD(N,f)
2 % Naive finite difference discretization of the Stokes system (9.2.42)
3 % N: number of grid cells in each direction.
4 % f: handle to a (vector valued!) function implementing the force field  $\mathbf{f}$ 
5 % Return values u1, u2 give the velocity components  $\mathbf{v} = (v_1, v_2)^T$ 
6 % in a matrix whose entries correspond to the vertices of the mesh,
7 % p returns the preassure.
    
```

```
8 h = 1/N;      % mesh width
9 x = h:h:1-h; % coordinates of interior grid points
10 unk = N-1;   % number of interior points in each direction
11 n = 3 * unk^2; % total number of unknowns for v and p
12 % A line-by-line numbering (lexikographic numbering) of the grid points is
   % assumed,
13 % see Sect. 4.1, Fig. 365.
14 A = gallery ('poisson', unk); % Matrix for 5-point stencil discretization of  $-\Delta$ 
15
16 % Build matrix representation of grad p. Note the efficient assembly based on
   % the
17 % special structure of the matrices.
18 % Auxiliary 1D central finite difference matrix
19 e = ones(unk, 1); CD = spdiags ([-h/2*e h/2*e], [-1 1], unk, unk);
20 % Central difference matrix for  $\frac{\partial}{\partial x_1}$ : This matrix is a block
21 % diagonal matrix with  $N-1$  diagonal blocks corresponding to the grid rows.
   % Its diagonal
22 % blocks are skew-symmetric and bidiagonal with non-zero first off-diagonals
   % only.
23 P1 = kron (speye (unk), CD);
24 % Central difference matrix for  $\frac{\partial}{\partial x_2}$ : This matrix is
25 % a block matrix with non-zero first off-diagonal blocks only. Each non-zero
   % block is
26 % a multiple of the identity.
27 P2 = kron (CD, speye (unk));
28 % Build the complete  $n \times n$  system matrix and make sure that it is a sparse
   % matrix.
29 Z = sparse (unk^2, unk^2); % Major mistake would be z = zeros(unk^2, unk^2);
30 H = [A Z P1; Z A P2; P1' P2' Z];
31
32 % Assemble the right hand side (sampling of f at interior grid points)
```

```
33 F = zeros (n,1);
34 pidx1 = 1; pidx2 = n/3+1;
35 for j = 1:size (x), for i = 1:size (x)
36     frc = h^2*f(x(i),x(j));
37     F(pidx1) = frc(1); F(pidx2) = frc(2);
38     pidx1 = pidx1+1; pidx2 = pidx2 + 1;
39 end, end
40 % Direct solution of sparse indefinite symmetric system
41 X = H\F;
42
43 % Convert vectors of nodal values into matrix representations of grid functions
44 u1 = rot90 (reshape (X (1:unk^2) ,unk,unk) );
45 u2 = rot90 (reshape (X (unk^2+1:2*unk^2) ,unk,unk) );
46 p = rot90 (reshape (X (2*unk^2+1:end) ,unk,unk) );
47
48 end
```

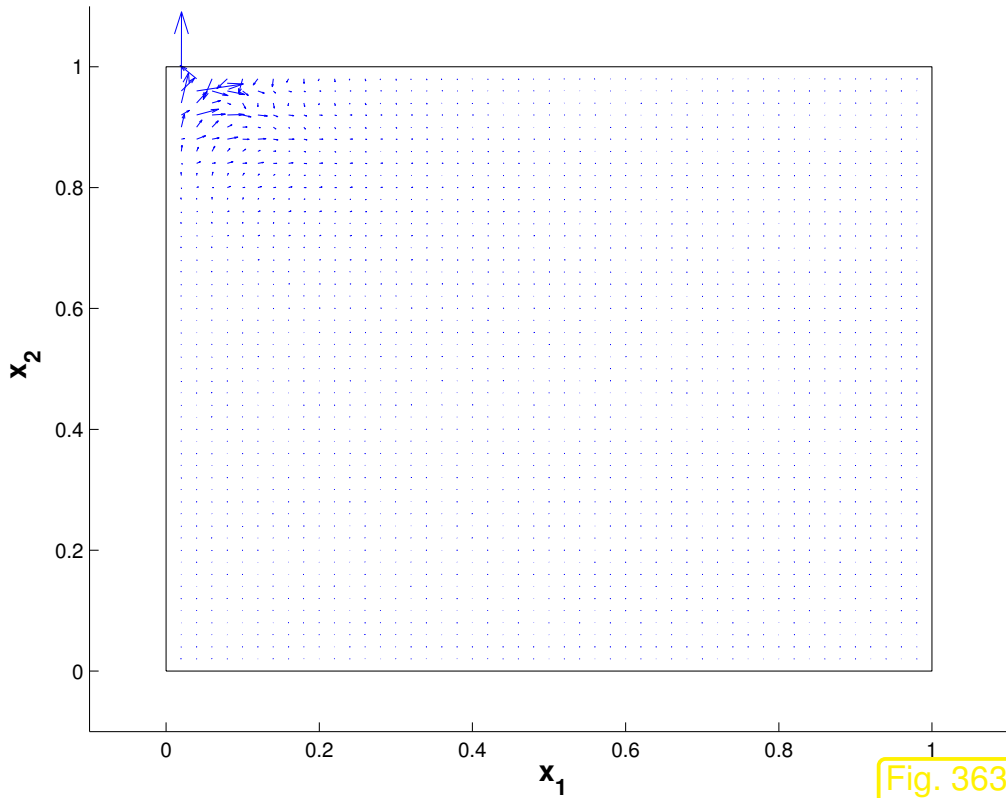


Fig. 363

velocity solution, $N = 50$

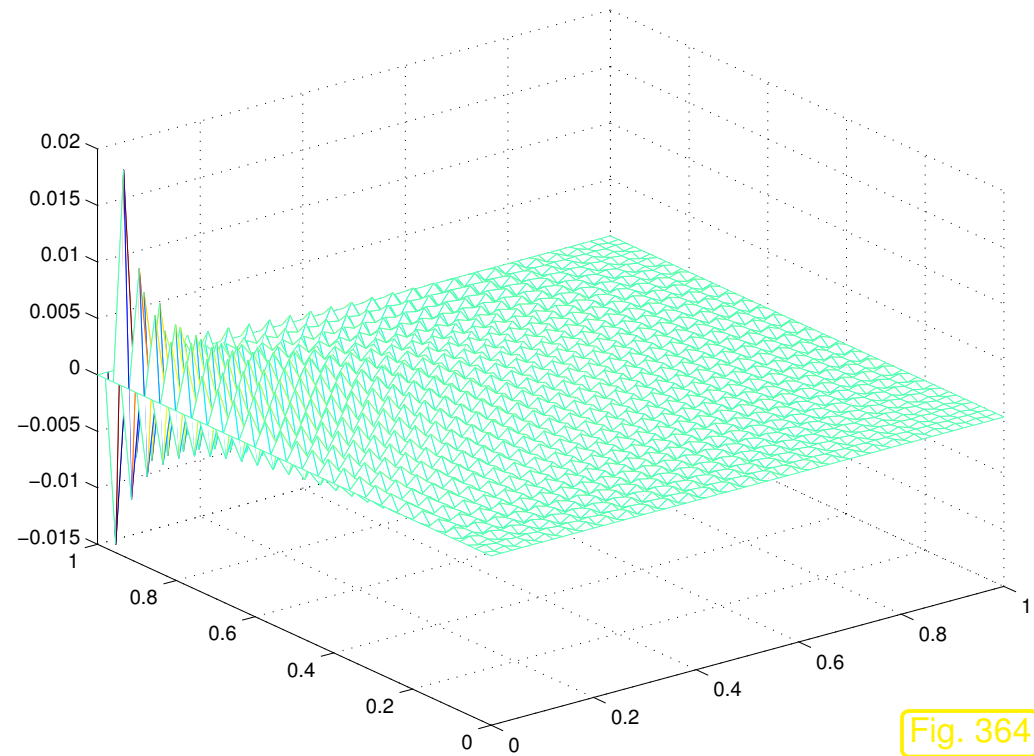


Fig. 364

pressure solution, $N = 50$

Physically meaningless solution marred by massive spurious oscillations of the pressure.



9.3.1 Pressure instability

Lesson learned: discretizing saddle point problems can be tricky!

Now, we examine the *Galerkin discretization* (\rightarrow Sect. 3.1) of the linear variational problem (9.2.33) (Practical schemes will rely on (9.2.41), but here, for the sake of simplicity, we skirt the treatment of zero mean constraint.)

Shorthand notation for (9.2.33) (\leftrightarrow abstract linear variational saddle point problem, see (9.2.27))

$$\begin{aligned} \mathbf{v} \in U &:= \mathbf{H}_0^1(\Omega), & \cdot & \quad \mathbf{a}(\mathbf{v}, \mathbf{w}) + \mathbf{b}(\mathbf{w}, p) = \ell(\mathbf{w}) \quad \forall \mathbf{w} \in U, \\ p \in Q &:= L_*^2(\Omega) & \cdot & \quad \mathbf{b}(\mathbf{v}, q) = 0 \quad \forall q \in Q. \end{aligned} \tag{9.3.13}$$

with concrete *bilinear forms*

$$\mathbf{a}(\mathbf{v}, \mathbf{w}) := \int_{\Omega} \mu D\mathbf{v} : D\mathbf{w} \, d\mathbf{x} \quad , \quad \mathbf{b}(\mathbf{v}, q) := \int_{\Omega} \operatorname{div} \mathbf{v} \, q \, d\mathbf{x} . \tag{9.3.14}$$

First step of Galerkin discretization:

Replace	$\mathbf{H}_0^1(\Omega)$	with finite dimensional <i>subspaces</i>	$U_N \subset \mathbf{H}_0^1(\Omega)$
	$L_*^2(\Omega)$		$Q_N \subset L_*^2(\Omega)$



Discrete linear variational saddle point problem:

$$\begin{aligned} \mathbf{v}_N \in U_N & : & \mathbf{a}(\mathbf{v}_N, \mathbf{w}_N) + \mathbf{b}(\mathbf{w}_N, p_N) &= \ell(\mathbf{w}_N) \quad \forall \mathbf{w}_N \in U_N, \\ p_N \in Q_N & : & \mathbf{b}(\mathbf{v}_N, q_N) &= 0 \quad \forall q_N \in Q_N. \end{aligned} \tag{9.3.15}$$

Second step of Galerkin discretization:

Introduce ordered bases $\mathfrak{B}_U := \{\mathbf{b}_N^1, \dots, \mathbf{b}_N^N\}$, of U_N , $N := \dim U_N$,
 $\mathfrak{B}_Q := \{\beta_N^1, \dots, \beta_N^M\}$ of Q_N , $M := \dim Q_N$.



$(N + M) \times (N + M)$ linear system of equations

symmetric indefinite matrix!

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix} \begin{pmatrix} \vec{\mathcal{U}} \\ \vec{\pi} \end{pmatrix} = \begin{pmatrix} \vec{\varphi} \\ 0 \end{pmatrix}, \tag{9.3.16}$$

$$\tag{9.3.17}$$

with Galerkin matrices, right hand side vectors

$$\mathbf{A} := \left(\mathbf{a}(\mathbf{b}_N^j, \mathbf{b}_N^i) \right)_{i,j=1}^N = \left(\int_{\Omega} \mu D\mathbf{b}_N^j(\mathbf{x}) : D\mathbf{b}_N^i(\mathbf{x}) d\mathbf{x} \right)_{i,j=1}^N \in \mathbb{R}^{N,N}, \tag{9.3.18}$$

$$\mathbf{B} := \left(\mathbf{b}(\mathbf{b}_N^j, \beta_N^i) \right)_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} = \left(\int_{\Omega} \operatorname{div} \mathbf{b}_N^j(\mathbf{x}) \beta_N^i(\mathbf{x}) \, d\mathbf{x} \right)_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \in \mathbb{R}^{M,N}, \quad (9.3.19)$$

$$\vec{\varphi} := \left(\ell(\mathbf{b}_N^j) \right)_{j=1}^N = \left(\int_{\Omega} \mathbf{f}(\mathbf{x}) \cdot \mathbf{b}_N^j(\mathbf{x}) \, d\mathbf{x} \right)_{j=1}^N \in \mathbb{R}^N, \quad (9.3.20)$$

and basis expansions

$$\mathbf{v}_N = \sum_{j=1}^N \nu_j \mathbf{b}_N^j, \quad p_N = \sum_{j=1}^M \pi_j \beta_N^j. \quad (9.3.21)$$

Issue: existence, uniqueness and stability of solutions of (9.3.15)

Existence, uniqueness and stability of solutions of discrete variational saddle point problems cannot be inferred from these properties for the continuous saddle point problem (\rightarrow Thm. 9.2.34).

(Unlike in the case of linear variational problems with s.p.d. bilinear forms, *cf.* Thm. 3.1.5)

A simple consideration:

$$M > N \Rightarrow \text{Ker}(\mathbf{B}) \neq \{0\} \Rightarrow \text{non-uniqueness of } p_N .$$

➤ $\dim U_N \geq \dim Q_N$ is a *necessary* condition for uniqueness of solution p_N of (9.3.15)

Some “natural” finite element Galerkin schemes for (9.2.33) \leftrightarrow (9.3.13) fail to meet this condition:

Example 9.3.22 (Unstable P1-P0 finite element pair on triangular mesh).

Notation: (cf. $\mathcal{S}_p^0(\mathcal{M})$): $\mathcal{S}_p^{-1}(\mathcal{M})$ discontinuous functions
locally polynomials of degree p , cf. $\mathcal{P}_p(\mathbb{R}^d)$

The spaces $\mathcal{S}_p^{-1}(\mathcal{M})$ are the natural finite element spaces for test/trial functions $\in L^2(\Omega)$, because this function space does not enforce any continuity conditions on piecewise smooth functions. Conversely, $H^1(\Omega)$ does, see Thm. 2.2.26.

Regular triangular mesh of $]0, 1[^2$



Finite element spaces for (9.2.33)

$$U_N := (\mathcal{S}_{1,0}^0(\mathcal{M}))^2,$$

$$Q_N := \mathcal{S}_0^{-1}(\mathcal{M}) \cap L_*^2(\Omega) \quad (\mathcal{M}\text{-piecewise constants}).$$

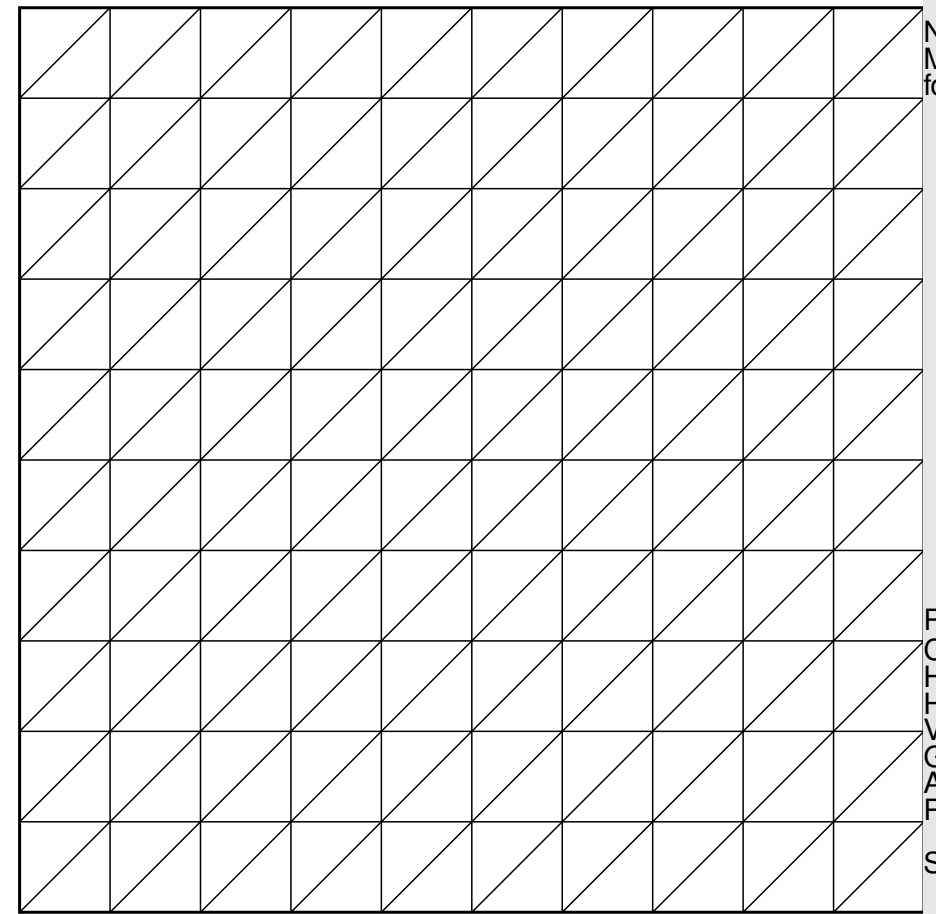
$K \in \mathbb{N} \hat{=}$ no. of mesh cells in one coordinate direction,

$$\dim U_N = 2(K - 1)^2, \quad \dim Q_N = 2K^2 - 1.$$



$$\dim Q_N > \dim U_N$$

Fig. 365



R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs
SAM, ETHZ

In this case we end up with a *singular* linear system (9.3.16), which will make the linear solver bail out or produce a pressure solution, which is polluted by “noise” from $\text{Ker}(\mathbf{B})$.

But $\dim U_N \geq \dim Q_N$ is not enough: even if this condition is satisfied, the pressure may not be unique:

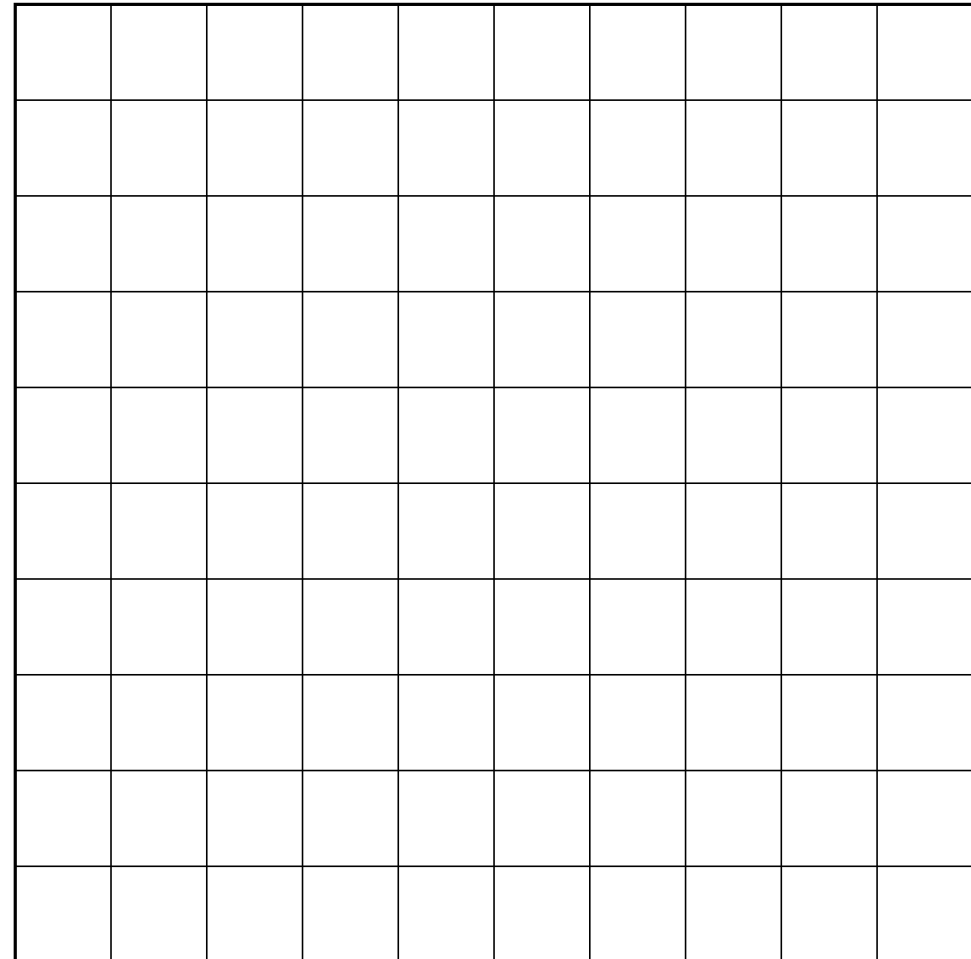
Example 9.3.23 (Checkerboard instability for quadrilateral P1-P0 pair). (\rightarrow [4, § 6])

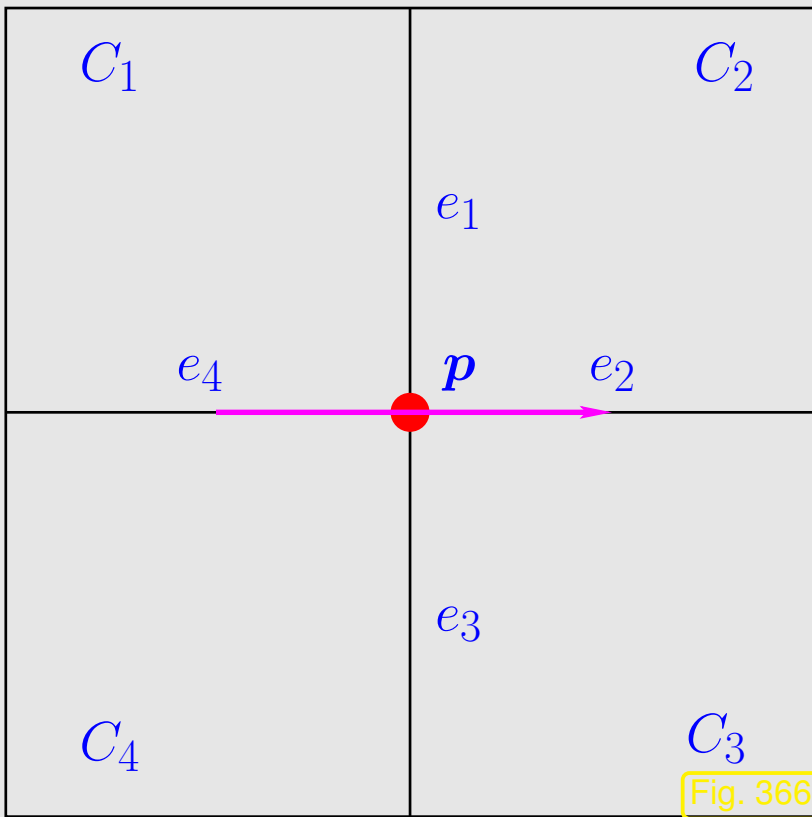
- \mathcal{M} = uniform tensor product mesh of $]0, 1[^2 \triangleright$
- velocity space $U_N = (\mathcal{S}_{1,0}^0(\mathcal{M}))^2$
- pressure space $Q_N = \mathcal{S}_0^{-1}(\mathcal{M}) \cap L_*^2(\Omega)$

If $K \in \mathbb{N}$ mesh cells in one coordinate direction,

$$\dim U_N = 2(K - 1)^2 \quad , \quad \dim Q_N = K^2 - 1 .$$

$$\blacktriangleright \quad \dim Q_N < \dim U_N \quad \text{for } K \geq 4 .$$





Consider interior grid point $\mathbf{p} = (ih, jh)$, $1 \leq i, j \leq K$, with adjacent quadratic cells C_1, C_2, C_3, C_4 , see figure.

Denote by p_i the piecewise constant values of p_N on C_i , $i = 1, 2, 3, 4$.

R. Hiptmair
C. Schwab,
H. Harbrecht
V. Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

$\mathbf{b}_{N,1}^{\mathbf{p}} \hat{=}$ nodal basis function for x_1 velocity component at vertex \mathbf{p} : $\mathbf{b}_{N,1}^{\mathbf{p}} = b_N^{\mathbf{p}} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, where $b_N^{\mathbf{p}}$ is the 2D “tent function” (\rightarrow Fig. 88) associated with \mathbf{p} .

$$\text{supp}(\mathbf{b}_{N,1}^{\mathbf{p}}) = C_1 \cup C_2 \cup C_3 \cup C_4$$

Apply Gauss’ theorem Thm. 2.4.9 on C_i taking into account that $\mathbf{b}_N^{\mathbf{p}} \perp$ normals at e_2, e_4 , and $\mathbf{b}_N^{\mathbf{p}} \parallel$

normals at e_1, e_3 ,

$$\begin{aligned} \int_{\Omega} \operatorname{div} \mathbf{b}_{N,1}^{\mathbf{p}} p_N \, d\mathbf{x} &= p_1 \int_{e_1} b_N^{\mathbf{p}} \, d\mathbf{x} - p_2 \int_{e_1} b_N^{\mathbf{p}} \, d\mathbf{x} + p_3 \int_{e_3} b_N^{\mathbf{p}} \, d\mathbf{x} - p_4 \int_{e_3} b_N^{\mathbf{p}} \, d\mathbf{x} \\ &= \frac{1}{2}(p_1 - p_2 + p_3 - p_4) . \end{aligned}$$

Similarly, if $\mathbf{b}_{N,2}^{\mathbf{p}}$ is the nodal basis function at \mathbf{p} for the x_2 -component of the velocity \mathbf{v}_N , then

$$\int_{\Omega} \operatorname{div} \mathbf{b}_{N,2}^{\mathbf{p}} p_N \, d\mathbf{x} = \frac{1}{2}(p_1 + p_2 - p_3 - p_4) .$$

$$p_1 = 1, p_2 = -1, p_3 = 1, p_4 = -1 \quad \Rightarrow \quad \int_{\Omega} \operatorname{div} \mathbf{b}_{N,1}^{\mathbf{p}} p_N \, d\mathbf{x} = \int_{\Omega} \operatorname{div} \mathbf{b}_{N,2}^{\mathbf{p}} p_N \, d\mathbf{x} = 0 . \quad (9.3.25)$$

Now, realize that the setting is translation invariant!

+1	-1	+1	-1	+1	-1	+1	-1	+1	-1
-1	+1	-1	+1	-1	+1	-1	+1	-1	+1
+1	-1	+1	-1	+1	-1	+1	-1	+1	-1
-1	+1	-1	+1	-1	+1	-1	+1	-1	+1
+1	-1	+1	-1	+1	-1	+1	-1	+1	-1
-1	+1	-1	+1	-1	+1	-1	+1	-1	+1
+1	-1	+1	-1	+1	-1	+1	-1	+1	-1
-1	+1	-1	+1	-1	+1	-1	+1	-1	+1
+1	-1	+1	-1	+1	-1	+1	-1	+1	-1
-1	+1	-1	+1	-1	+1	-1	+1	-1	+1

Fig. 367

By (9.3.25) the discrete pressure with alternating values ± 1 in checkerboard fashion will belong to $\text{Ker}(\mathbf{B})$ for this finite element Galerkin method (for odd K).

Observation:

$$\{p_N \in Q_N : \mathbf{b}(\mathbf{v}_N, p_N) = 0 \quad \forall \mathbf{v}_N \in U_N\} \neq \emptyset .$$

= 1-dimensional space of checkerboard modes

◁ p.w. constant checkerboard mode



Example 9.3.26 (P1-P0 quadrilateral finite elements for Stokes problem).

- BVP (9.2.42) on $\Omega =]0, 1[^2$, $\mu \equiv 1$, $\mathbf{f} = \cos(\pi x_1) \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, see Ex. 9.3.1
- P1-P0 finite element Galerkin discretization on equidistant tensor product quadrilateral mesh, as in Ex. 9.3.23

Code 9.3.28: P1-P0 finite difference discretization of augmented Stokes problem

```

1 function [v1,v2,p] = stokesP1P0FD(N,frc)
2 % P1-P0 finite element discretization of Stokes problem (9.2.41) on a
3 % quadrilateral tensor product mesh, see Ex. 9.3.23.
4 % N: number of mesh cells in one coordinate direction.
5 % f: function handle of type symbol64(x1,x2) to right hand side
6 % force field f
7 h = 1/N; nv = (N-1)^2; nc = N^2; % meshwidth,  $\#\mathcal{V}(\mathcal{M})$ ,  $\#\mathcal{M}$ 
8 % Assemble system matrix from Kronecker products of 1D Galerkin matrices
9 % Tridiagonal 1D mass matrix for linear finite elements
10 M = h*spdiags(ones(N-1,3)*diag([1/6 2/3 1/6]),[-1 0 1],N-1,N-1);
11 % Tridiagonal 1D Galerkin matrix for  $\frac{d^2}{dx^2}$ , see (1.5.86)
12 D = spdiags(ones(N-1,3)*diag([-1 2 -1]),[-1 0 1],N-1,N-1)/h;
13 % 1D Galerkin matrix for p.w. linear/p.w. constant finite elements and the
    bilinear
14 % form  $\int_0^1 \frac{du}{dx} q dx$ 

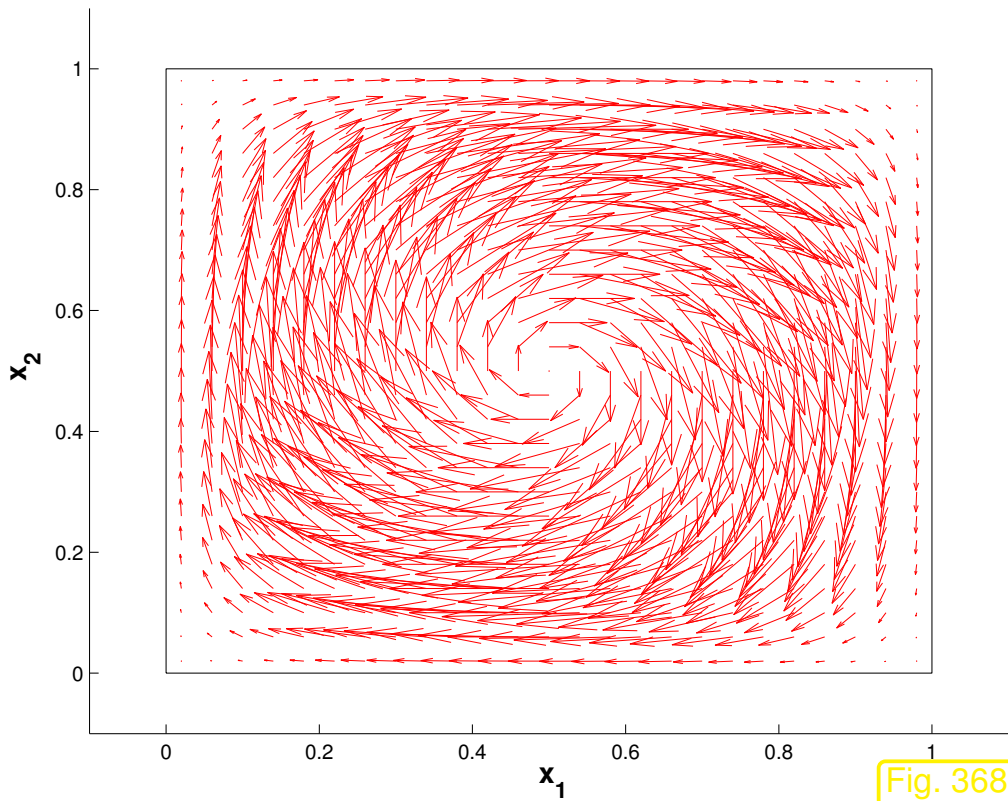
```

```

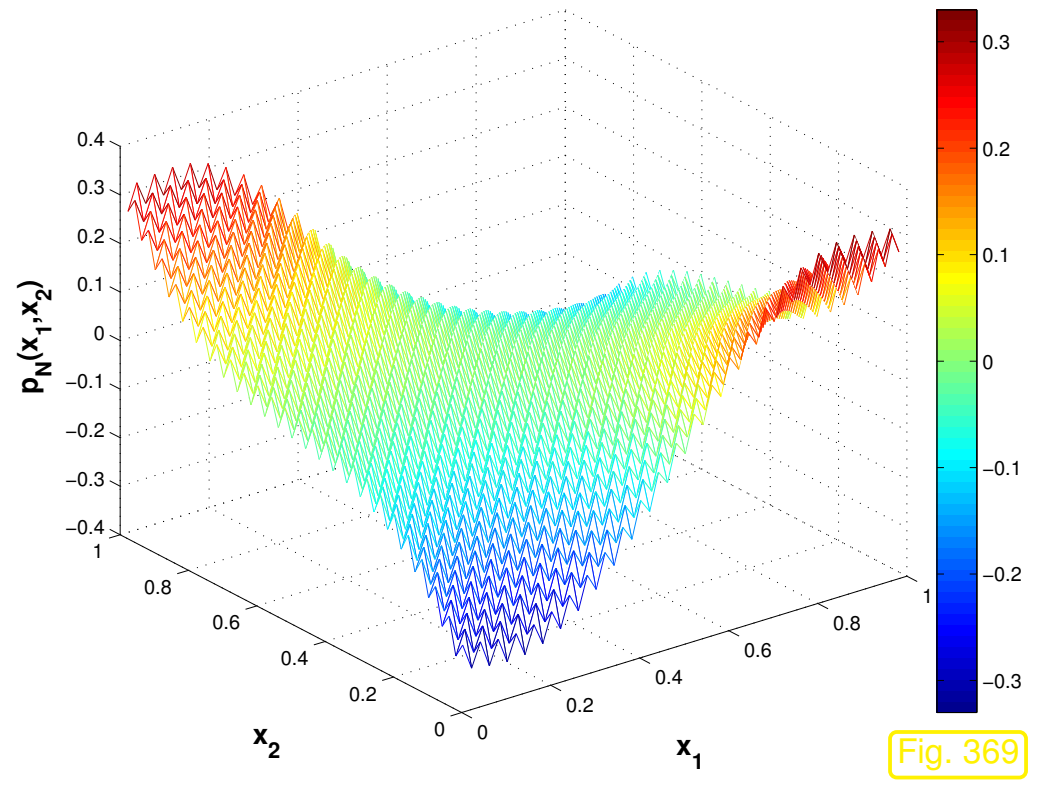
15 G = spdiags (ones (N,2) * diag ([-1 1]), [-1 0], N, N-1);
16 % 1D mass matrix for p.w. linear and p.w. constant finite elements
17 C = 0.5*h*spdiags (ones (N,2), [-1 0], N, N-1);
18 % constraint on pressure, see Rem. 9.2.39
19 Delta = kron (M,D) + kron (D,M); % 9-point stencil matrix for discrete Laplacian
20 divx = kron (C,G); divy = kron (G,C); % discrete divergence
21 % Complete saddle point system matrix including Lagrangian multiplier for
    enforcing mean zero
22 A = [      Delta          , sparse (nv,nv) ,      divx'          , sparse (nv,1) ; ...
23       sparse (nv,nv)      ,      Delta          ,      divy'          , sparse (nv,1) ; ...
24       divx                ,      divy          , sparse (nc,nc) ,      ones (nc,1) ; ...
25       sparse (1,nv)      , sparse (1,nv) , ones (1,nc) ,      0          ];
26 % Assembly of right hand side
27 phi = zeros (2*nv+nc+1,1); x = h:h:1-h; idx = 1;
28 for j=1:N-1, for i=1:N-1,
29     phi([idx idx+nv]) = h*h*frc(x(i),x(j)); idx = idx+1;
30 end, end;
31 % Direct solve of (singular for even N) linear saddle point system
32 u = A\phi;
33 % Recover velocity and pressure values on the grid
34 v1 = rot90 (reshape (u(1:nv), N-1, N-1));
35 v2 = rot90 (reshape (u(nv+1:2*nv), N-1, N-1));
36 p = rot90 (reshape (u(2*nv+1:end-1), N, N));



```

P1-P0 velocity vector field, N=50



P1-P0 pressure field, N=50



Observation:  pressure solution marred by checkerboard mode
 computed velocity field ok!



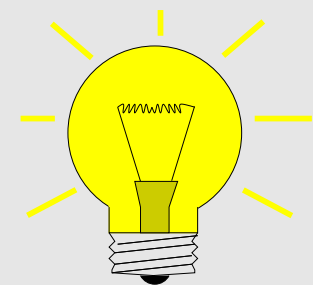
In the previous examples we found a subspace of Q_N , which dodges $\operatorname{div} \mathbf{v}_N$ in the bilinear form b .

We arrive at the important heuristic insight:

$\operatorname{div} \mathbf{v}_N$ must be “large enough to fix the pressure” $p_N \in Q_N$

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ



Idea:

Use larger velocity trial/test spaces \mathbf{v}_N

- larger space $\operatorname{div} \mathbf{v}_N$
- more control of p_N

How to get a larger trial space for the velocity? Raise polynomial degree!

Example 9.3.29 (P2-P0 finite element scheme for the Stokes problem).

- $\Omega =]0, 1[^2$, $\mathbf{u}(\mathbf{x}) = (\cos(\pi/2(x_1 + x_2)), -\cos(\pi/2(x_1 + x_2)))^T$, $p(\mathbf{x}) = \sin(\pi/2(x_1 - x_2))$, \mathbf{f} and inhomogeneous Dirichlet boundary values for \mathbf{u} accordingly
- Sequence of (a) uniform triangular meshes, created by regular refinement, (b) randomly perturbed meshes from (a) (still uniformly shape-regular & quasi-uniform).
- “P2-P0-scheme” velocity finite element space $U_N = (\mathcal{S}_{2,0}^0(\mathcal{M}))^2$ (continuous, piecewise quadratic \rightarrow Sect. 3.4.1, Ex. 3.4.2), pressure finite element space $Q_N = \mathcal{S}_0^{-1}(\mathcal{M}) \cap L_*^2(\Omega)$ (piecewise constant)

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Monitored: Error norms $\|\mathbf{u} - \mathbf{u}_N\|_1$, $\|\mathbf{u} - \mathbf{u}_N\|_0$, $\|p - p_N\|_0$

Code 9.3.31: LehrFEM driver script P2-P0 finite element method for Stokes problem

```
1 % LehrFEM driver script for computing solutions of the steady Stokes problem on
  the unit square
2 % using piecewise quadratic finite elements for the velocity and piecewise
  constants for the
3 % pressure.
4 NREFS = 4; % Number of red refinements
5 NU = 1; % Viscosity
6 % Dirichlet boundary data
7 GD_HANDLE = @(x,varargin)[cos(pi/2*(x(:,1)+x(:,2)))
  -cos(pi/2*(x(:,1)+x(:,2)))];
8 % Right hand side source (force field)
9 F_HANDLE = @(x,varargin)[sin(pi*x(:,1)) zeros(size(x(:,1))) ] ;
10
11 % Initialize mesh
12 Mesh = load_Mesh('Coord_Sqr.dat','Elem_Sqr.dat');
13 Mesh.ElemFlag = ones(size(Mesh.Elements,1),1);
14 Mesh = add_Edges(Mesh);
15 Loc = get_BdEdges(Mesh);
16 Mesh.BdFlags = zeros(size(Mesh.Edges,1),1);
17 Mesh.BdFlags(Loc) = -1;
18 for i = 1:NREFS, Mesh = refine_REG(Mesh); end
19
20 % Assemble Galerkin matrix and load vector
21 A = assemMat_Stokes_P2P0(Mesh,@STIMA_Stokes_P2P0,NU,P706());
22 L = assemLoad_Stokes_P2P0(Mesh,P706(),F_HANDLE);
23
24 % Incorporate Dirichlet boundary data
25 [U,FreeDofs] = assemDir_Stokes_P2P0(Mesh,-1,GD_HANDLE); L = L - A*U;
```

```
27 % Solve the linear system (direct solver)
28 U(FreeDofs) = A(FreeDofs,FreeDofs)\L(FreeDofs);
29
30 % Plot and print solution
31 plot_Stokes(U,Mesh,'P2P0');
32 title ('\bf Steady Stokes equation (P2 elements)');
33 xlabel(['\bf # Dofs : ' int2str(size(U,1)) '']);
34 colorbar;
35 print('-depsc','func_P2P0.eps')
```

Code 9.3.35: Assembly of global Galerkin matrix for P2-P0 finite element method for Stokes problem

```
1 function varargout = assemMat_Stokes_P2P0(Mesh,EHandle,varargin)
2 % Assemble Galerking matrix for P2-P0 finite element discretization of Stokes
  problem
3 % (9.2.41): piecewise quadratic continuous velocity components and piecewise
4 % constant pressure approximation.
5 %mesh LehrFEM mesh data structure, complete with edge information,
6 %Sect. 3.5.2 The struct MESH must at least contain the following fields:
7 % COORDINATES M-by-2 matrix specifying the vertices of the mesh.
8 % ELEMENTS N-by-3 or matrix specifying the elements of the mesh.
9 % EDGES P-by-2 matrix specifying the edges of the mesh.
10 % ELEMFLAG N-by-1 matrix specifying additional elementinformation.
```

```
11 % EHandle passes function for computation of element matrix.
12 % See Sect. 3.5.3 for a discussion of the generic assembly algorithm
13 nCoordinates = size (Mesh.Coordinates,1);
14 nElements = size (Mesh.Elements,1);
15 nEdges = size (Mesh.Edges,1);
16 % Preallocate memory for the efficient initialization of sparse matrix,
   Ex. 3.5.18
17 I = zeros (196*nElements,1); J = zeros (196*nElements,1); A =
   zeros (196*nElements,1);
18 % Local assembly: loop over all cells of the mesh
19 loc = 1:196;
20 for i = 1:nElements
21     % Extract vertex coordinates
22     vidx = Mesh.Elements(i,:);
23     Vertices = Mesh.Coordinates(vidx,:);
24     % Compute 14x14 element matrix: there are 6 local shape functions for the
       finite
25     % element space  $\mathcal{S}_2^0(\mathcal{M})$ , and 1 (constant) local shape function for
26     %  $\mathcal{S}_0^{-1}(\mathcal{M})$ : 6+6+1=13 local shape functions for the P2-P0 scheme
27     Aloc = EHandle(Vertices,Mesh.ElemFlag(i),varargin{:});
28     % Add contributions to global Galerkin matrix: the numbering convention is a
       follows:
29     % d.o.f. for  $x_1$ -components of the velocity are numbered first, then
30     %  $x_2$ -components of the velocity, then the pressure d.o.f.
31     eidx = [Mesh.Vert2Edge(vidx(1),vidx(2)) ...
32             Mesh.Vert2Edge(vidx(2),vidx(3)) ...
33             Mesh.Vert2Edge(vidx(3),vidx(1))];
34     % Note: entries of an extra last row/column of the Galerkin matrix
       corresponding to
35     % pressure d.o.f. are filled with one to enforce zero mean pressure, see
36     % Ex. 9.2.39
```

```

37 idx = [vidx,eidx+nCoordinates,... % ↔  $v_1$ 
38         vidx+nCoordinates+nEdges,eidx+2*nCoordinates+nEdges, % ↔  $v_2$ 
39         i+2*(nEdges+nCoordinates), % ↔  $p$ 
40         2*(nEdges+nCoordinates)+nElements+1]; % ↔ zero mean multiplier
41 I(loc) = set_Rows(idx,14); J(loc) = set_Cols(idx,14); A(loc) = Aloc(:);
42 loc = loc+196;
43 end
44
45 % Assign output arguments for creation of sparse matrix
46 if (nargout > 1), varargout{1} = I; varargout{2} = J; varargout{3} = A;
47 else, varargout{1} = sparse(I,J,A); end
48 return

```

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Code 9.3.37: Computation of element matrix for P2-P0 finite element method for Stokes problem

```

1 function Aloc = STIMA_Stokes_P2P0(Vertices,ElemInfo,nu,QuadRule,varargin)
2 % Computation of element matrix for P2-P0 finite element discretization of 2D
3 % Stokes problem
4 % Vertices passes the location of the vertices of the triangle
5 % nu is the viscosity parameter
6 % QuadRule specifies local quadrature rule, see Rem. 3.5.39
7 % The function returns a 14x14 dense matrix
8 Aloc = zeros(14,14); % Preallocate memory
9 % Compute element mapping
10 bK = Vertices(1,:); BK = [Vertices(2,:)-bK; Vertices(3,:)-bK];

```

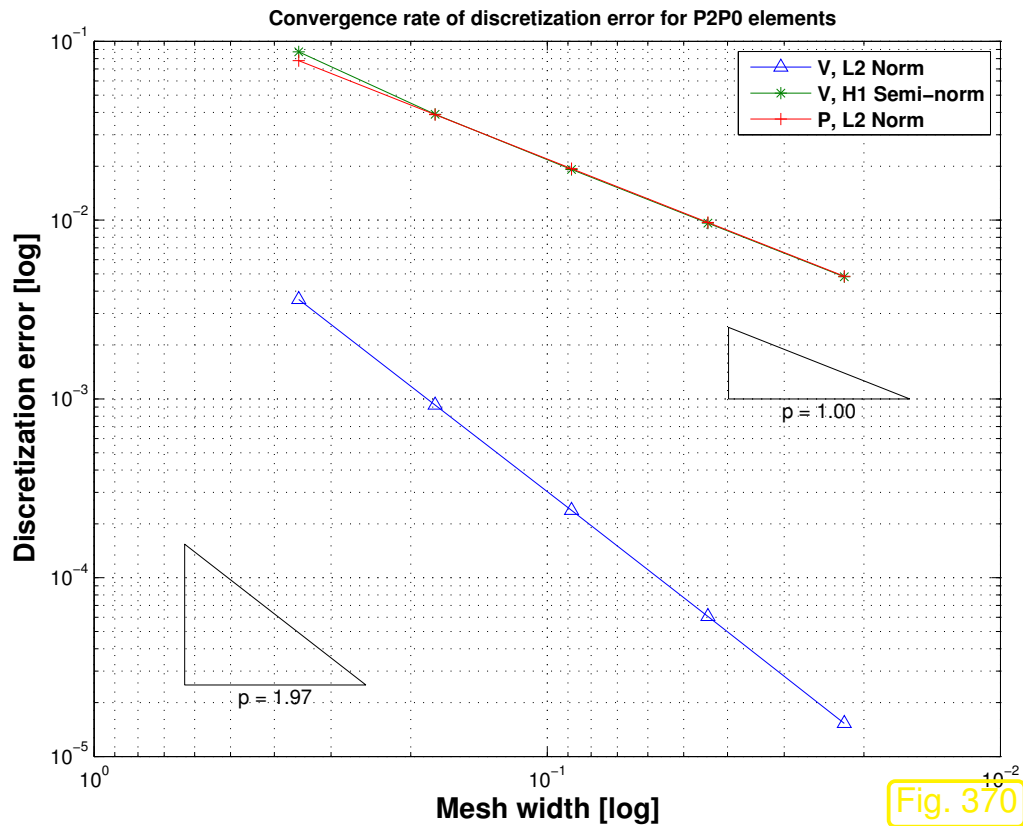
```
10 inv_BK_t = transpose(inv(BK)); det_BK = abs(det(BK));
11 % Compute gradients element shape functions and their values at quadrature
    points
12 grad_N = grad_shap_QFE(QuadRule.x);
13 grad_N(:,1:2) = grad_N(:,1:2)*inv_BK_t;
14 grad_N(:,3:4) = grad_N(:,3:4)*inv_BK_t;
15 grad_N(:,5:6) = grad_N(:,5:6)*inv_BK_t;
16 grad_N(:,7:8) = grad_N(:,7:8)*inv_BK_t;
17 grad_N(:,9:10) = grad_N(:,9:10)*inv_BK_t;
18 grad_N(:,11:12) = grad_N(:,11:12)*inv_BK_t;
19 % The first 6 rows/columns of the element matrix correspond to the  $x_1$ -component
    of the
20 % velocity. the corresponding block of the element matrix agrees with that for
     $-\Delta$ 
21 % discretized by means of quadratic Lagrangian finite elements. The local
    shape functions are
22 % described in Ex. 3.4.2.
23 Aloc(1,1) = nu*sum(QuadRule.w.*sum(grad_N(:,1:2).*grad_N(:,1:2),2))*det_BK;
24 Aloc(1,2) = nu*sum(QuadRule.w.*sum(grad_N(:,1:2).*grad_N(:,3:4),2))*det_BK;
25 Aloc(1,3) = nu*sum(QuadRule.w.*sum(grad_N(:,1:2).*grad_N(:,5:6),2))*det_BK;
26 Aloc(1,4) = nu*sum(QuadRule.w.*sum(grad_N(:,1:2).*grad_N(:,7:8),2))*det_BK;
27 Aloc(1,5) = nu*sum(QuadRule.w.*sum(grad_N(:,1:2).*grad_N(:,9:10),2))*det_BK;
28 Aloc(1,6) = nu*sum(QuadRule.w.*sum(grad_N(:,1:2).*grad_N(:,11:12),2))*det_BK;
29 Aloc(2,2) = nu*sum(QuadRule.w.*sum(grad_N(:,3:4).*grad_N(:,3:4),2))*det_BK;
30 Aloc(2,3) = nu*sum(QuadRule.w.*sum(grad_N(:,3:4).*grad_N(:,5:6),2))*det_BK;
31 Aloc(2,4) = nu*sum(QuadRule.w.*sum(grad_N(:,3:4).*grad_N(:,7:8),2))*det_BK;
32 Aloc(2,5) = nu*sum(QuadRule.w.*sum(grad_N(:,3:4).*grad_N(:,9:10),2))*det_BK;
33 Aloc(2,6) = nu*sum(QuadRule.w.*sum(grad_N(:,3:4).*grad_N(:,11:12),2))*det_BK;
34 Aloc(3,3) = nu*sum(QuadRule.w.*sum(grad_N(:,5:6).*grad_N(:,5:6),2))*det_BK;
35 Aloc(3,4) = nu*sum(QuadRule.w.*sum(grad_N(:,5:6).*grad_N(:,7:8),2))*det_BK;
36 Aloc(3,5) = nu*sum(QuadRule.w.*sum(grad_N(:,5:6).*grad_N(:,9:10),2))*det_BK;
```

```
37 Aloc(3,6) = nu*sum(QuadRule.w.*sum(grad_N(:,5:6).*grad_N(:,11:12),2))*det_BK;
38 Aloc(4,4) = nu*sum(QuadRule.w.*sum(grad_N(:,7:8).*grad_N(:,7:8),2))*det_BK;
39 Aloc(4,5) = nu*sum(QuadRule.w.*sum(grad_N(:,7:8).*grad_N(:,9:10),2))*det_BK;
40 Aloc(4,6) = nu*sum(QuadRule.w.*sum(grad_N(:,7:8).*grad_N(:,11:12),2))*det_BK;
41 Aloc(5,5) = nu*sum(QuadRule.w.*sum(grad_N(:,9:10).*grad_N(:,9:10),2))*det_BK;
42 Aloc(5,6) = nu*sum(QuadRule.w.*sum(grad_N(:,9:10).*grad_N(:,11:12),2))*det_BK;
43 Aloc(6,6) = nu*sum(QuadRule.w.*sum(grad_N(:,11:12).*grad_N(:,11:12),2))*det_BK;
44 % the same for the  $x_2$ -component of the velocity
45 Aloc(7,7) = Aloc(1,1); Aloc(7,8) = Aloc(1,2); Aloc(7,9) = Aloc(1,3);
46 Aloc(7,10) = Aloc(1,4); Aloc(7,11) = Aloc(1,5); Aloc(7,12) = Aloc(1,6);
47 Aloc(8,8) = Aloc(2,2); Aloc(8,9) = Aloc(2,3); Aloc(8,10) = Aloc(2,4);
48 Aloc(8,11) = Aloc(2,5); Aloc(8,12) = Aloc(2,6); Aloc(9,9) = Aloc(3,3);
49 Aloc(9,10) = Aloc(3,4); Aloc(9,11) = Aloc(3,5); Aloc(9,12) = Aloc(3,6);
50 Aloc(10,10) = Aloc(4,4); Aloc(10,11) = Aloc(4,5); Aloc(10,12) = Aloc(4,6);
51 Aloc(11,11) = Aloc(5,5); Aloc(11,12) = Aloc(5,6); Aloc(12,12) = Aloc(6,6);
52 % Interaction of pressure shape function (constant  $\equiv 1$ ) with velocity:
    evaluation of
53 % local bilinear form  $b_K$ .
54 % First for  $x_1$ -components, then for
55 Aloc(1,13) = sum(QuadRule.w.*grad_N(:,1))*det_BK;
56 Aloc(2,13) = sum(QuadRule.w.*grad_N(:,3))*det_BK;
57 Aloc(3,13) = sum(QuadRule.w.*grad_N(:,5))*det_BK;
58 Aloc(4,13) = sum(QuadRule.w.*grad_N(:,7))*det_BK;
59 Aloc(5,13) = sum(QuadRule.w.*grad_N(:,9))*det_BK;
60 Aloc(6,13) = sum(QuadRule.w.*grad_N(:,11))*det_BK;
61 % Next for  $x_2$ -components of velocity
62 Aloc(7,13) = sum(QuadRule.w.*grad_N(:,2))*det_BK;
63 Aloc(8,13) = sum(QuadRule.w.*grad_N(:,4))*det_BK;
64 Aloc(9,13) = sum(QuadRule.w.*grad_N(:,6))*det_BK;
65 Aloc(10,13) = sum(QuadRule.w.*grad_N(:,8))*det_BK;
```

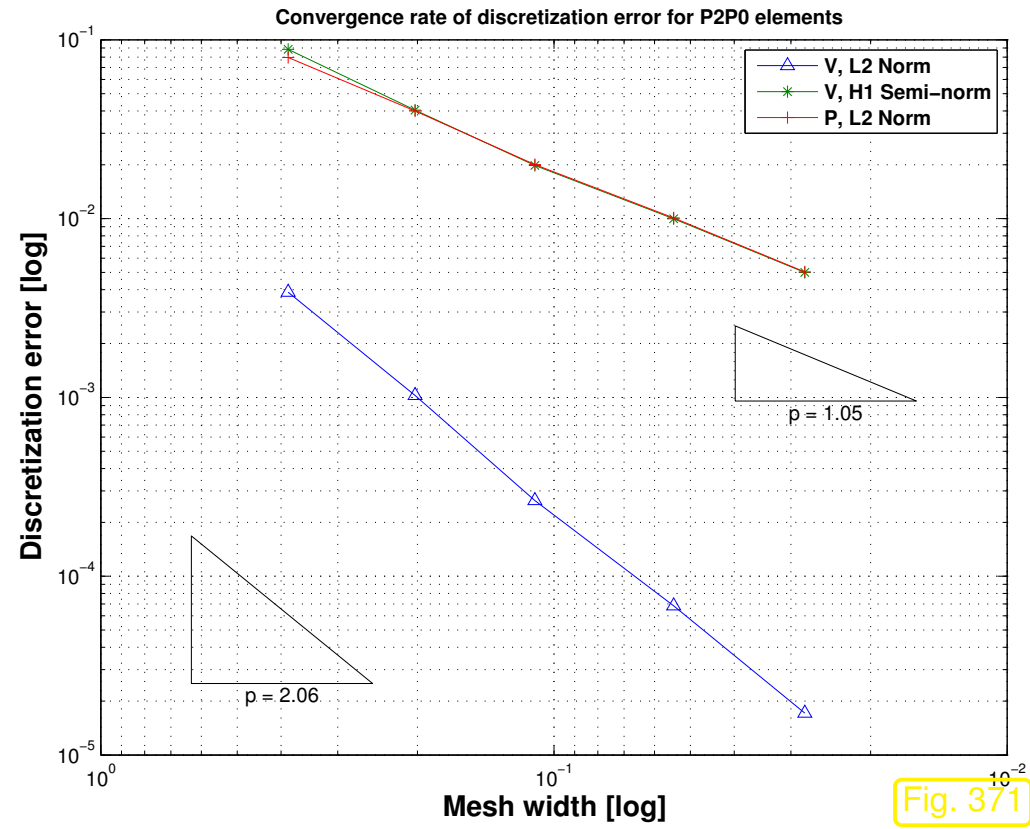


```

66 Aloc(11,13) = sum(QuadRule.w.*grad_N(:,10))*det_BK;
67 Aloc(12,13) = sum(QuadRule.w.*grad_N(:,12))*det_BK;
68 % Entry corresponding to zero mean multiplier
69 Aloc(13,14) = det_BK;
70 % Fill in lower triangular part
71 tri = triu(Aloc); Aloc = tri+tril(tri',-1);
72 return
    
```



Structured meshes



Randomly perturbed meshes

Raising the polynomial degree has cured the instability!

Observation: algebraic convergence

$$\begin{aligned}\|\mathbf{u} - \mathbf{u}_N\|_1 &= O(h_{\mathcal{M}}), \\ \|\mathbf{u} - \mathbf{u}_N\|_0 &= O(h_{\mathcal{M}}^2), \\ \|p - p_N\|_0 &= O(h_{\mathcal{M}}).\end{aligned}$$


R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

The pair $U_N = \mathcal{S}_{2,0}^0(\mathcal{M})$, $Q_N = \mathcal{S}_0^{-1}(\mathcal{M})$ is the first combination of finite element spaces that we find to provide a **stable** Galerkin discretization of the variational Stokes problem (9.2.33) \leftrightarrow (9.2.26).

Recall the concept of **stability/well-posedness** for linear problems, see Sect. 2.3.2, “stability estimate” of Thm. 3.1.5,

$$\|\text{solution}\| \leq C \|\text{right hand side}\| \quad \text{for all data,}$$

where **relevant norms** have to be considered.

For the Stokes problem: relevant norms = norms of Sobolev spaces fitting (9.2.33)

For velocity \mathbf{v} : use “energy norm” $\|\cdot\|_a := \mathbf{a}(\cdot, \cdot)^{1/2} \sim \|\cdot\|_{\mathbf{H}^1(\Omega)}$, cf. Def. 2.1.35

For pressure p : use $\|\cdot\|_{L^2(\Omega)}$.

Definition 9.3.38 (Stable finite element pair).

A pair of finite element spaces $U_N \subset \mathbf{H}_0^1(\Omega)$, $Q_N \subset L_*^2(\Omega)$ is a **stable finite element pair**, if the solution (\mathbf{v}_N, p_N) of the discrete saddle point problem (9.3.15) satisfies

$$|\ell(\mathbf{w}_N)| \leq C_\ell \|\mathbf{w}_N\|_a \quad \forall \mathbf{w}_N \implies \exists C > 0: \|\mathbf{v}_N\|_a + \|p_N\|_{L^2(\Omega)} \leq CC_\ell,$$

where $C > 0$ may depend only on Ω , the coefficient μ , and the shape regularity measure (\rightarrow Def. 5.3.26) of \mathcal{M} .

We have already encountered an estimate like

$$|\ell(\mathbf{w}_N)| \leq C_\ell \|\mathbf{w}_N\|_a \quad \forall \mathbf{w}_N \in U_N, \quad (9.3.40)$$

when finding that the existence of solutions of quadratic minimization problems (\rightarrow Def. 2.1.26) hinges on the **continuity** of the involved linear form, see (2.2.3).

Let us embark on a mathematical analysis of the stability issue, which turns out to be much simpler than expected.

Remark 9.3.43 (Stable velocity solution).

Consider (9.2.33) \leftrightarrow (9.2.26), and Galerkin discretization (9.3.15), define the *subspace*

$$\mathcal{N}(\mathbf{b}_N) := \{\mathbf{w}_N \in U_N : \mathbf{b}(\mathbf{w}_N, q_N) = 0 \quad \forall q_N \in Q_N\} \subset U_N. \quad (9.3.44)$$

From 2nd equation \triangleright for any solution (\mathbf{v}_N, p_N) of (9.3.15): $\mathbf{v}_N \in \mathcal{N}(\mathbf{b}_N)$

Test the first equation of (9.3.15) with $\mathbf{w}_N \in \mathcal{N}(\mathbf{b}_N)$

$$\blacktriangleright \quad \mathbf{a}(\mathbf{v}_N, \mathbf{w}_N) = \ell(\mathbf{w}_N) \quad \xrightarrow{\mathbf{w}_N := \mathbf{v}_N} \quad \|\mathbf{v}_N\|_a^2 \leq \ell(\mathbf{v}_N) \stackrel{(9.3.40)}{\leq} C_\ell \|\mathbf{v}_N\|_a.$$

perfect stability of *any* velocity Galerkin solution

This explains the observation made in Ex. 9.3.26: reasonable approximation for velocity \mathbf{v} despite pressure instability.

Remark 9.3.50 (Stability of pressure solution: inf-sup condition).

Goal: stability of pressure solution $p_N \in Q_N$ of (9.3.15)

$$\|p_N\|_{L^2(\Omega)} \leq C \sup_{\mathbf{w}_N \in U_N} \frac{\ell(\mathbf{w}_N)}{\|\mathbf{w}_N\|_a} \quad (9.3.51)$$

best constant in (9.3.40)

From the first equation of (9.3.15)

$$\mathbf{a}(\mathbf{v}_N, \mathbf{w}_N) + \mathbf{b}(\mathbf{w}_N, p_N) = \ell(\mathbf{w}_N) \quad \forall \mathbf{w}_N \in U_N,$$

and the stability of the velocity solution (\rightarrow Rem. 9.3.43) we conclude (9.3.51), once we know

$$\mathbf{b}(\mathbf{w}_N, p_N) = g(\mathbf{w}_N) \quad \forall \mathbf{w}_N \in U_N \quad \Rightarrow \quad \|p_N\|_{L^2(\Omega)} \leq C \sup_{\mathbf{w}_N \in U_N} \frac{|g(\mathbf{w}_N)|}{\|\mathbf{w}_N\|_a}. \quad (9.3.52)$$

Theorem 9.3.53 (inf-sup condition).

The finite element spaces $U_N \subset \mathbf{H}_0^1(\Omega)$, $Q_N \subset L_^2(\Omega)$ provide a stable finite element pair (\rightarrow Def. 9.3.38) for the Stokes problem (9.2.33)/(9.2.26) if there is a constant $\beta > 0$ depending only on Ω and the shape regularity measure (\rightarrow Def. 5.3.26) of \mathcal{M} such that*

$$\sup_{\mathbf{w}_N \in U_N} \frac{|\mathbf{b}(\mathbf{w}_N, q_N)|}{\|\mathbf{w}_N\|_a} \geq \beta \|q_N\|_{L^2(\Omega)} \quad \forall q_N \in Q_N. \quad (9.3.54)$$

inf-sup condition

The estimate (9.3.54) is known as

LBB (Ladyzhenskaya-Babuska-Brezzi) condition

It is the linchpin of the numerical analysis of finite element methods for the Stokes problem, see [16].

Abstract considerations (easier this way!):

- $H \hat{=}$ normed vector space, norm $\|\cdot\|$ (think of a function space),
- $c : H \times H \mapsto \mathbb{R}$ bilinear form on H , *not necessarily s.p.d.* (\rightarrow Def. 2.1.32),
- $\ell : H \mapsto \mathbb{R}$ linear form on H ,
- Assumption: c is **continuous**, cf. Rem. 7.2.2, (3.1.2)

$$\exists C_c > 0: |c(u, v)| \leq C_c \|u\| \|v\| \quad \forall u, v \in H . \quad (9.3.61)$$

We consider the linear variational problem (\rightarrow Rem. 1.4.6)

$$u \in H: c(u, v) = \ell(v) \quad \forall v \in H , \quad (9.3.62)$$

and its Galerkin discretization, based on finite-dimensional subspace $H_N \subset H$, cf. (3.1.4),

$$u_N \in H_N: \quad \mathbf{c}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in H_N. \quad (9.3.63)$$

Assumption: **stability**

$$u_N \text{ solves (9.3.63)} \implies \exists C_s > 0: \quad \|u_N\| \leq \sup_{w_N \in H_N} \frac{|\ell(w_N)|}{\|w_N\|}. \quad (9.3.64)$$

Trick! For any $v_N \in H_N$ the difference $u_N - v_N$ (u_N solution of (9.3.63)) solves

$$\mathbf{c}(u_N - v_N, w_N) = \ell(w_N) - \mathbf{c}(v_N, w_N) \quad \forall w_N \in H_N.$$

$$\begin{aligned} (9.3.64) \implies \|u_N - v_N\| &\leq C_s \sup_{w_N \in H_N} \frac{|\ell(w_N) - \mathbf{c}(v_N, w_N)|}{\|w_N\|} \\ &\stackrel{(9.3.62)}{=} C_s \sup_{w_N \in H_N} \frac{|\mathbf{c}(u - v_N, w_N)|}{\|w_N\|} \\ &\stackrel{(9.3.61)}{\leq} C_c C_s \|u - v_N\|. \end{aligned} \quad (9.3.65)$$

”Trick” Triangle inequality

$$\|u - u_N\| \leq \|u - v_N\| + \|u_N - v_N\| \stackrel{(9.3.65)}{\leq} (1 + C_c C_s) \|u - v_N\| \quad \forall v_N \in H_N.$$



$$\|u - u_N\| \leq (1 + C_c C_s) \inf_{v_N \in H_N} \|u - v_N\|. \quad (9.3.66)$$

(9.3.66) is a fundamental insight into the properties of Galerkin discretizations, *cf.* Thm. 5.1.10 that was confined to s.p.d. bilinear forms:

For the Galerkin discretization of linear variational problems:

$$\text{Stability} \quad \Rightarrow \quad \text{Quasi-optimality} (*)$$

Terminology: **Quasi-optimality** of Galerkin solutions: with $C > 0$ *independent* of data and discretization parameters

$$\underbrace{\|u - u_N\|}_{\substack{\uparrow \\ \text{(norm of) discretization error}}} \leq C \underbrace{\inf_{v_N \in H_N} \|u - v_N\|}_{\substack{\uparrow \\ \text{best approximation error}}} , \quad (9.3.67)$$

Application of abstract theory to finite element discretization of Stokes problem (9.2.33):

- $H := \mathbf{H}_0^1(\Omega) \times L^2(\Omega)$ (combination of two function spaces!)

- Role of \mathbf{c} played by

$$\mathbf{c} \left(\begin{pmatrix} \mathbf{v} \\ p \end{pmatrix}, \begin{pmatrix} \mathbf{w} \\ q \end{pmatrix} \right) := \mathbf{a}(\mathbf{v}, \mathbf{w}) + \mathbf{b}(\mathbf{w}, p) + \mathbf{b}(\mathbf{v}, q) . \quad (9.3.69)$$

- Right hand side functional " $\ell \left(\begin{pmatrix} \mathbf{w} \\ q \end{pmatrix} \right) = \ell(\mathbf{w})$ "

- Galerkin trial/test space $H_N := U_N \times Q_N$.

Then, along the lines of the above abstract considerations, one can show the following a priori error estimate:

Theorem 9.3.70 (Convergence of stable FE for Stokes problem).

If U_N, Q_N is a stable finite element pair (\rightarrow Def. 9.3.38) for the Stokes problem (9.2.33), then the corresponding finite element Galerkin solution (\mathbf{v}_N, p_N) satisfies

$$\|\mathbf{v} - \mathbf{v}_N\|_{H^1(\Omega)} + \|p - p_N\|_{L^2(\Omega)} \leq C \left(\inf_{\mathbf{w}_N \in U_N} \|\mathbf{v} - \mathbf{w}_N\|_{H^1(\Omega)} + \inf_{q_N \in Q_N} \|p - q_N\|_{L^2(\Omega)} \right),$$

with a constant $C > 0$ that depends only on Ω, μ , and the shape regularity of the finite element mesh.

Note: the a priori error bound of Thm. 9.3.70 involves the *sum* of the best approximation errors for both the velocity and pressure trial/test spaces.

Example 9.3.72 (Convergence of P2-P0 scheme for Stokes equation).

Interpretation of error curves observed in Ex. 9.3.29:

Smooth solutions for both \mathbf{v} and p :

Sect. 5.3.5 ➤
$$\inf_{\mathbf{w}_N \in \mathcal{S}_{2,0}^0(\mathcal{M})} \|\mathbf{v} - \mathbf{w}_N\|_{H^1(\Omega)} \leq Ch_{\mathcal{M}}^2 \|\mathbf{v}\|_{H^3(\Omega)} \quad (\text{Thm. 5.3.42}),$$
$$\inf_{q_N \in \mathcal{S}_0^{-1}(\mathcal{M})} \|p - q_N\|_{L^2(\Omega)} \leq Ch_{\mathcal{M}} \|p\|_{H^1(\Omega)},$$

with constants depending *only* on the shape regularity measure (\rightarrow Def. 5.3.26) of triangulation \mathcal{M} .

The observed $O(h)$ algebraic convergence in the $\mathbf{H}^1(\Omega)$ -norm (for \mathbf{v}_N) and $L^2(\Omega)$ -norm (for p_N) results, because

the larger best approximation error of $\mathcal{S}_0^{-1}(\mathcal{M})$ dominates.



9.4 The Taylor-Hood element

A: The ultimate cure for instability

chose trial/test space for velocity large enough \rightarrow very large (to play safe).

B: Well, but a large finite element space leads to a large system of linear equations, that is, high computational cost.

A: Never mind, a large space buys good accuracy, which is what we also want!

Remark 9.4.2 (Efficient finite element discretization of Stokes problem).

Thm. 9.3.70, *cf.* discussion in Ex. 9.3.72: the finite element discretization error for a stable finite element pair (U_N, Q_N) (\rightarrow Def. 9.3.38) for the Stokes problem (9.2.33) is the *sum* of approximation errors for the velocity \mathbf{v} in U_N and the pressure p in Q_N .

➤ Excellent approximation of either \mathbf{v} or p alone may not lead to an accurate solution.

Recall similar situation for method of lines, where errors of spatial discretization and timestepping add up, see “Meta-Thms.” 6.1.70, 6.2.42.

For the sake of **efficiency**

$$\text{balance } \inf_{\mathbf{w}_N \in U_N} \|\mathbf{v} - \mathbf{w}_N\|_{H^1(\Omega)} \text{ and } + \inf_{q_N \in Q_N} \|p - q_N\|_{L^2(\Omega)}$$

Too ambitious: we have no chance of guessing the best approximation errors a priori.

Thus we settle for a more modest *asymptotic* balance condition, *cf.* the considerations in Sect. 6.1.5.



Guideline for **viable** and **efficient** choice of Galerkin finite element spaces for Stokes problem:

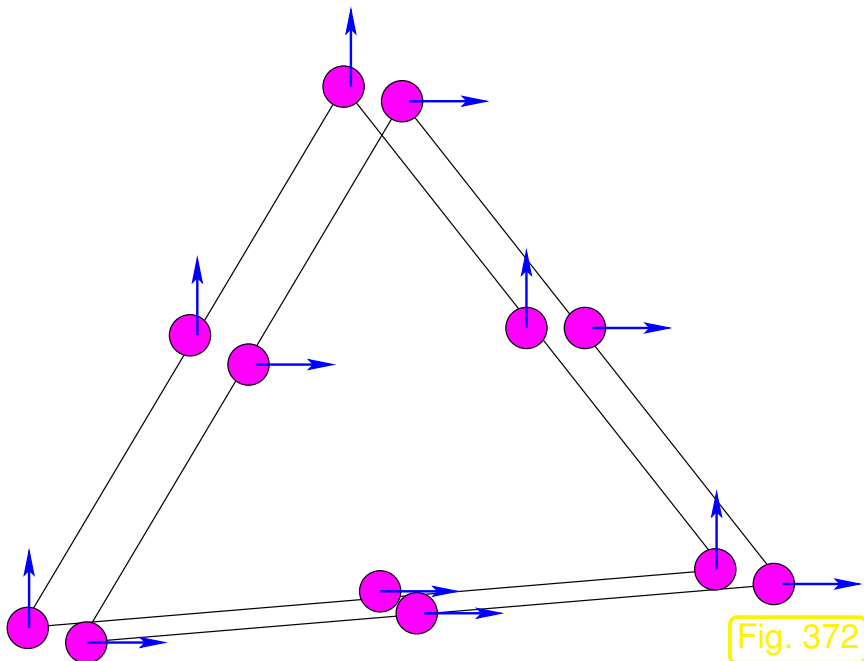
- ❶ The pair (U_N, Q_N) of finite element spaces must be **stable** (\rightarrow Def. 9.3.38)
- ❷ The velocity finite element space U_N should provide the **same rate of algebraic convergence** of the $H^1(\Omega)$ -best approximation error w.r.t. $h_{\mathcal{M}} \rightarrow 0$ as the pressure space in $L^2(\Omega)$.
- ❸ The velocity finite element space U_N should guarantee ❶ and ❷ with as few degrees of freedom as possible.

Note that the stable finite element pair $(\mathcal{S}_{2,0}^0(\mathcal{M}), \mathcal{S}_0^{-1}(\mathcal{M}))$ does not meet the efficiency criterion, because the velocity space offers a better asymptotic rate of convergence than the pressure space, see Ex. 9.3.72.

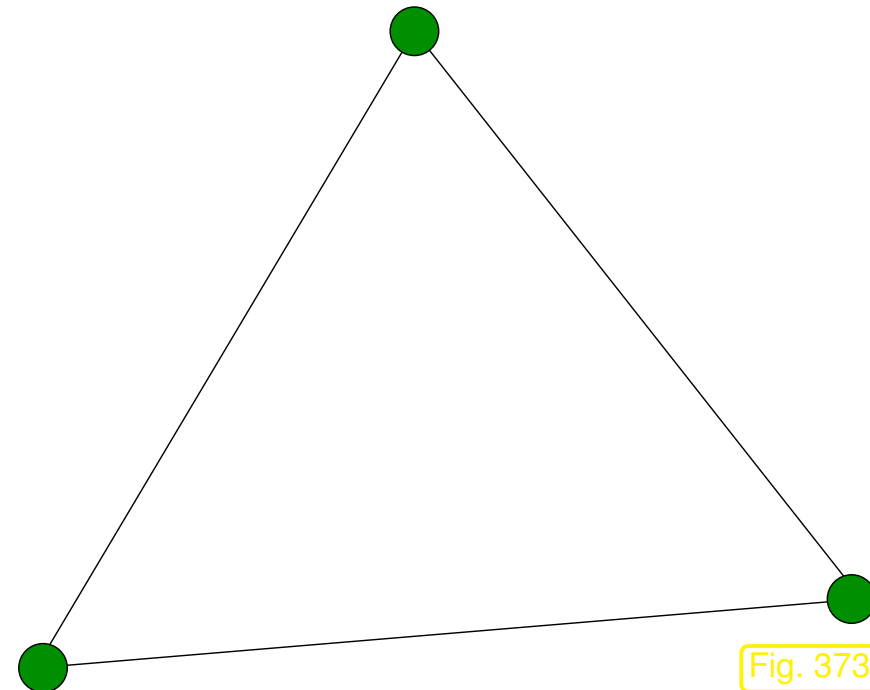
There is a stable, perfectly balanced pair of spaces:

Taylor-Hood finite element method for Stokes problem:

- \mathcal{M} : triangular/tetrahedral or rectangular/hexahedral mesh of Ω , may be hybrid, see Sect. 3.3.1
- Velocity space: $U_N := \mathcal{S}_{2,0}^0(\mathcal{M}) \subset \mathbf{H}_0^1(\Omega)$
- Pressure space: $Q_N := \mathcal{S}_1^0(\mathcal{M})$ (continuous pressure)



sites of velocity local shape functions



sites of pressure local shape functions

Balanced approximation properties of finite element spaces (for sufficiently smooth velocity and pressure solution):

$$\begin{aligned} \text{velocity:} & \quad \inf_{\mathbf{w}_N \in U_N} \|\mathbf{v} - \mathbf{w}_N\|_{H^1(\Omega)} \leq Ch_{\mathcal{M}}^2 \|\mathbf{v}\|_{H^3(\Omega)} && \text{by Thm. 5.3.42,} \\ \text{pressure:} & \quad \inf_{q_N \in \mathcal{S}_0^{-1}(\mathcal{M})} \|p - q_N\|_{L^2(\Omega)} \leq Ch_{\mathcal{M}}^2 \|p\|_{H^2(\Omega)} && \text{by Thm. 5.3.27.} \end{aligned}$$

Theorem 9.4.3 (Stability and convergence of Taylor-Hood finite element). \rightarrow [31]

The Taylor-Hood element provides a *stable* finite element pair for the Stokes problem (\rightarrow Def. 9.3.38) and for sufficiently smooth velocity and pressure solution

$$\|\mathbf{v} - \mathbf{v}_N\|_{H^1(\Omega)} + \|p - p_N\|_{L^2(\Omega)} \leq Ch_{\mathcal{M}}^2 \left(\|\mathbf{v}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)} \right),$$

with a constant $C > 0$ that depends only on Ω , μ , and the shape regularity of the finite element mesh.

Example 9.4.4 (Convergence of Taylor-Hood method for Stokes problem).

- Stokes problem (9.2.41) as in Ex. 9.3.29
- perturbed triangular meshes as in Ex. 9.3.29
- Taylor-Hood finite element Galerkin discretization

Monitored: Error norms $\|\mathbf{u} - \mathbf{u}_N\|_{H^1(\Omega)}$,
 $\|\mathbf{u} - \mathbf{u}_N\|_{L^2(\Omega)}$, $\|p - p_N\|_{L^2(\Omega)}$
 Observation: algebraic convergence

$$\|\mathbf{u} - \mathbf{u}_N\|_{H^1(\Omega)} = O(h_{\mathcal{M}}^2),$$

$$\|\mathbf{u} - \mathbf{u}_N\|_{L^2(\Omega)} = O(h_{\mathcal{M}}^3),$$

$$\|p - p_N\|_{L^2(\Omega)} = O(h_{\mathcal{M}}^2).$$

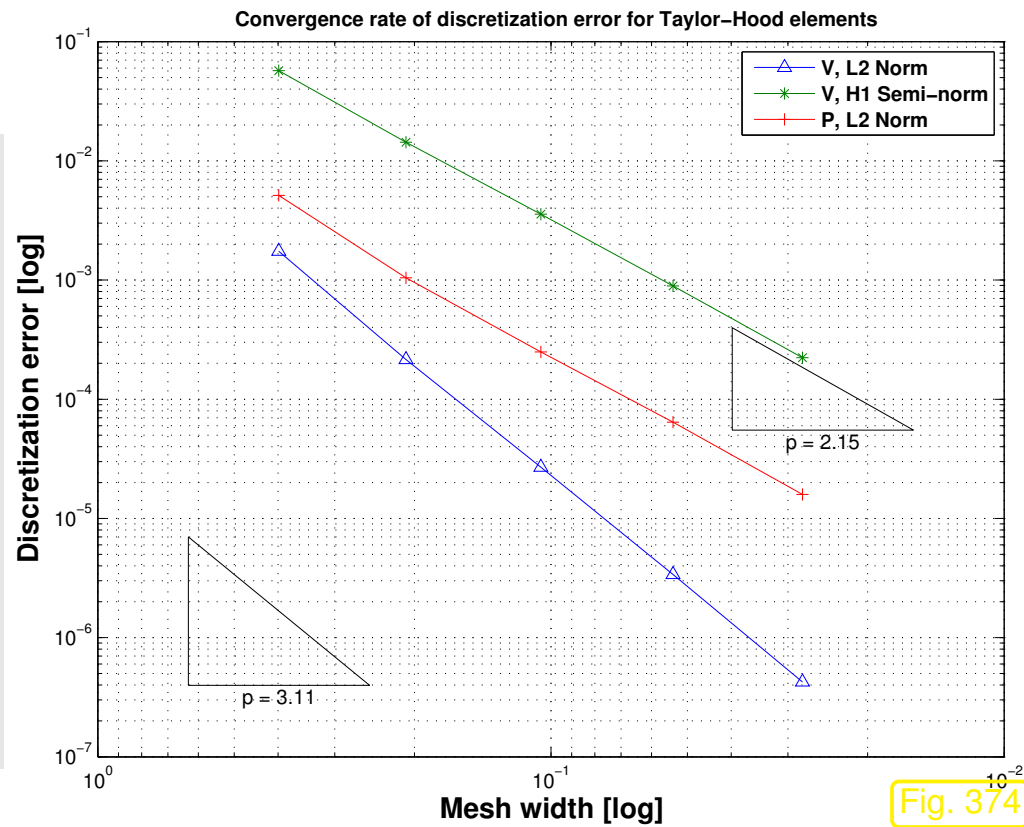


Fig. 374



Adaptive Finite Element Discretization

10

10.1 Concept of adaptivity

10.2 A priori hp-adaptivity

10.2.1 Graded meshes in 1D

10.2.2 Triangular graded meshes

10.2.3 hp-approximation in 1D

11

Multilevel iterative solvers

11.1 Solving finite element linear systems

11.2 Subspace correction

11.2.1 Successive subspace correction algorithm (SSC)

11.2.2 Gauss-Seidel iteration

11.2.3 Hierarchical basis multigrid

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

12

Sparse Grids Galerkin Methods

12.1 The curse of dimension

12.2 Hierarchical basis

12.3 Sparse grids

12.4 Approximation on sparse grids

12.5 Sparse grids algorithms

Index

- H^1 -semi-norm, 150
- L^2 -norm, 147
- 2-regularity
 - of Dirichlet problem, 594
- convergence
 - exponential, 159
- a priori estimates, 497
- a-orthogonal, 476
- affine linear function
 - in 2D, 295
- affine transformation, 389
- algebraic convergence, 159
- algorithm
 - numerical, 75
- analytic solution, 74
- angle condition
 - for Delaunay mesh, 461
- anisotropic diffusion, 768
- artificial diffusion, 763
- artificial viscosity, 763
- assembly, 308
 - cell oriented, 375
 - in FEM, 369
 - linear finite elements, 315
- balance law, 850
- barycentric coordinate representation
 - of local shape functions, 382
- barycentric coordinates, 309
- basis
 - change of, 287
- best approximation error, 478
- beta function, 386
- bilinear form, 52, 68
 - continuous, 734
 - positive definite, 190
- bilinear transformation, 418
- boundary conditions, 27, 71, 239
 - Dirichlet, 239, 250
 - homogeneous, 252
 - Neumann, 242, 251
 - no slip, 1038

- radiation, 251
- boundary fitting, 429
 - parabolic, 429
- boundary flux
 - computation of, 578
- boundary layer, 737
- boundary value, 171
- boundary value problem
 - elliptic, 253
- boundary value problem (BVP), 171
- bounding box, 361
- Burger's equation, 890
- Burgers equation, 856
- calculus of variations, 46
- Cauchy problem, 832, 859, 876
 - for one-dimensional conservation law, 864
 - for wave equation, 682
- cell, 325
- cell contributions, 370
- central flux, 920, 923
- central slope, 1004
- CFL-condition, 708, 971
- characteristic curve, 867
- characteristic method, 797
- Chebyshev nodes, 134
- checkerboard mode, 1077
- circumcenter, 460
- classical solution, 63, 262
- coefficient vector, 285
- collocation, 127, 128
 - spline, 137
- compatibility condition, 263
- compatibility conditions
 - for $H^1(\Omega)$, 213
- complete space, 473
- composite midpoint rule, 117
- composite trapezoidal rule, 117
- computational domain, 244
- computational effort, 541
- condition number, 101
- conditionally stable, 669, 670
- configuration space, 27
 - for incompressible fluid, 1036
 - for taut membrane, 178
- congruent matrices, 288
- conservation
 - of energy, 692
- conservation form, 913
- conservation law, 860
 - differential form, 863
 - integral form, 862
 - one-dimensional, 863
 - scalar, 863
- conservation of energy, 245
- consistency
 - of a variational problem, 769
- consistency error, 994
- continuity
 - of a linear form, 210
 - of linear functional, 571
- control volume, 456, 860
- convection-diffusion equation, 725

convective cooling, 251
convective terms, 725
convergence, 143, 482
 algebraic , 159
 asymptotic, 153
convex function, 194
corner singular function, 553
Courant-Friedrichs-Levy condition (CFL), 708
Crank-Nicolson method, 636
creeping flow, 1038
curve, 27, 28
 length, 29
 parameterization, 28
cut-off function, 585
d'Alembert solution, 683
Delaunay mesh
 angle condition, 461
delta distribution, 227
dielectric tensor, 182
difference quotient, 140
differential operator, 132, 171
diffusion tensor, 768
diffusive flux, 862
diffusive terms, 725
Dirac delta function, 227
Dirichlet boundary conditions, 239, 250
 for linear FE, 401
Dirichlet data, 220, 268
Dirichlet problem, 239
 variational formulation, 219
discrete maximum principle, 604
discrete model, 34
discrete variational problem, 281, 285
discretization, 75, 279
discretization error, 143, 478
discretization parameter, 482
displacement, 71
displacement function, 71
dissipation, 629
 in fluid, 1041
DistMesh, 361
divergence
 of a vectorfield, 233
domain, 171
 computational, 244
 spatial, 73, 176
domain of dependence, 685
domain of influence, 685
dual mesh, 459
dual problem, 575
duality estimate, 575
dynamic viscosity, 1037
eddy
 in a fluid, 1040
edge, 325
elastic energy, 33
 mass-spring model, 34
elastic string, 25
electric field, 181
electric scalar potential, 182
electromagnetic field energy, 182
electrostatic field energy, 182

electrostatics, 181
element, 325
element load vector, 371
element stiffness matrix, 371
elliptic
 linear scalar second order PDE, 249
elliptic boundary value problem, 253
energy
 conservation, 245
 of electrostatic field, 182
energy conservation
 for wave equation, 688
energy norm, 191
entropy, 1039
equidistant mesh, 110, 140, 333
equilibrium length
 of spring, 33
equilibrium principle, 35
equivalence
 of norms, 518
Euler equations, 856
Euler method, 634
evolution operator
 fully discrete, 966
 semi-discrete, 964
evolution problem, 615
 semi-discrete, 632
 spatial variational formulation, 623
expansion shock, 886, 887
explicit Euler method, 634
exponential convergence, 159
face, 325
field energy
 electromagnetic, 182
finite difference methods, 443
finite differences
 1D, 139
 in 2D, 443
finite elements
 parametric, 422
finite volume methods, 455
flow field, 718
flow map, 721
flux function, 860, 865
force density, 179
Fourier's law, 246
 if fluid, 724
Frobenius norm, 513
functional, 569
 linear, 571
fundamental lemma of calculus of variations, 237

Galerkin discretization, 280
Galerkin matrix, 369
Galerkin orthogonality, 476
Galerkin solution, 80
 quasi-optimality, 478
Galerkin test space, 80
Galerkin trial space, 80
Gamma function, 386
Gauss' theorem, 234, 248, 723, 874
Gauss-Legendre quadrature, 396
Gauss-Lobatto quadrature, 396

General entropy solution for 1D scalar Riemann problem, 900
generic constants, 539
global shape functions, 332
GMSH, 361
Godunov numerical flux, 947
gradient, 178
 of a function, 179
gravitational force, 31
Green's first formula, 234
grid
 1D, 110, 140
grid function, 144
h-refinement, 547
hanging node, 328
hat function, 216, 299
heat capacity, 618
heat conductivity, 247
heat equation, 618
heat flux, 244, 247
 computation of, 578
 convective, 724
 diffusive, 724
heat source, 246
Hessian, 509
Heun method, 966
Hilbert space, 473
homogeneous boundary conditions, 252
Hooke's law, 32
hyperbolic evolution problem, 679
 discrete case, 689

implicit Euler method, 634
implicit midpoint rule, 636
increments
 Runge-Kutta, 637
index mapping matrix, 373
inf-sup condition, 1094
inflow, 860
inflow boundary, 739
initial conditions, 615
initial value problem
 stiff, 646
initial-boundary value problems (IBVP), 616
 parabolic, 620
integrated Legendre polynomials, 90
integration by parts
 in 1D, 60
 multidimensional, 234
intermediate state, 943
interpolant
 piecewise linear, 464
interpolation error, 508
interpolation error estimates
 anisotropic, 524
 in 1D, 499
interpolation nodes, 340
inviscid, 854

kinetic energy, 688

L(π)-stability, 660
L-shaped domain, 489
L-stability, 660

Lagrange functional
 for zero mean constraint, 1060
 Lagrange multiplier, 1050
 Lagrangian finite elements, 339, 377
 on quadrilaterals, 415
 Lagrangian functional, 1051
 Lagrangian method, 797
 for advection, 789
 Laplace equation, 238
 Lax entropy condition, 899
 Lax-Friedrichs flux, 928
 layer
 boundary, 737
 layers
 internal, 765
 LBB condition, 1094
 leapfrog, 698
 Legendre polynomials, 91
 integrated, 90
 LehrFEM, 355
 lexicographic ordering, 444
 lifting theorem, 551, 558
 linear boundary fitting, 565
 linear form, 52
 continuity, 210
 linear function
 in 2D, 295
 linear functional, 52
 linear interpolation
 in 1D, 499
 in 2D, 507, 508
 linear variational problem, 67
 Linearity, 254
 linearization
 of variational problems, 430
 load vector, 285, 370
 local linearization, 435
 local operations, 376
 local shape function, 335
 barycentric representation, 413
 local shape functions
 quadratic, 341
 macroscopic quantities, 845
 mass lumping, 700
 mass-spring model, 32
 elastic energy, 34
 material coordinate, 30
 material derivative, 819
 material tensor, 220
 mathematical modelling, 23
 maximum principle, 600, 730
 discrete, 604
 Maxwell's equations
 static case, 182
 mean value formula, 509
 membrane, 174
 potential energy, 178, 179
 membrane problem
 variational formulation, 219
 mesh, 325
 1D, 110, 140
 data structures, 362

equidistant, 110, 140
node, 292
non-conforming, 328
quadrilateral, 326
simplicial, 328
triangular, 326
mesh data structure, 362
mesh file format, 356
 triangular mesh, 357
mesh generation, 361
mesh generator, 356
mesh width, 485
method of characteristics, 738
method of lines, 632
midpoint rule
 composite, 117
minmod, 1021
mixed boundary conditions, 253
Mixed Neumann–Dirichlet problem, 243
model
 continuous, 75
 discrete, 34, 75
monomial basis, 89
monotonicity preserving linear interpolation, 1017
multi-index notation, 329
multiplicative trace inequality, 271
MUSCL scheme, 1028

NETGEN, 361
Neumann boundary conditions, 242, 251
Neumann data
 admissibility conditions, 269

Neumann problem, 263
 compatibility condition, 263
 variational form, 263
Newton update, 438
Newton’s method, 434
 in function space, 434
 termination, 439
Newton’s second law of motion, 676
Newton-Cotes formula, 396
Newton-Galerkin iteration, 438
nodal basis, 298
nodal interpolation operators, 535
nodal value, 299
node, 292
 1D, 110, 140
 quadrature, 96
norm, 145
 on function space, 146
numerical domain of dependence, 969
numerical flux, 457, 913
numerical flux function, 457
numerical quadrature, 95
 nodex, 387
 weights, 387

offset function, 53, 398
 for linear FE, 399
order of quadrature rule, 388
outflow, 860
outflow boundary, 739
output functional, 569

- p-refinement, 548
- parameterization
 - of curve, 28
- parametric finite elements, 422, 423
- parametric quadrature rule, 387
- particle method, 797
- particle model
 - of traffic flow, 838
- PDE
 - Linear scalar second order elliptic, 249
- perpendicular bisector, 460
- phase space, 859
- Pythagoras' theorem, 477
- piecewise linear interpolant, 464
- piecewise linear reconstruction, 1000
- piecewise quadratic interpolation, 536
- Poincaré-Friedrichs inequality, 208, 270
- point force, 58
- Poisson equation, 238, 483
- Poisson matrix, 447
- polar coordinates, 228
- polynomials
 - degree, 329
 - multivariate, 329
 - univariate, 87
- positive definite
 - bilinear form, 190
 - uniformly, 184
- postprocessing, 144
- potential energy, 688
 - of taut membrane, 178
- pressure, 1055
- pressure Poisson equation, 1063
- problem parameters
 - for elastic string, 31
- problem size, 541
- procedural form
 - of functions, 278
- product rule, 627
 - in higher dimensions, 233
- production term, 860
- pullback, 407
- quadratic functional, 187
- quadratic local shape functions, 341
- quadratic minimization problem, 66
- quadratic minimization problems, 186
- quadrature formula, 96
- quadrature node, 96
- quadrature nodes, 387
- quadrature rule, 387
 - on triangle, 392
 - order, 388
 - parametric, 387
- quadrature rules
 - Gauss-Legendre, 396
 - Gauss-Lobatto, 396
- quadrature weight, 96
- quadrature weights, 387
- quadrilateral mesh, 326
- quasi-optimality, 478, 1097
- Radau timestepping, 660

radiation boundary conditions, 251, 261
rarefaction
 subsonic, 946
 supersonic, 946
 transonic, 946
rarefaction wave/fan, 895
reaction term
 in 2nd-order BVP, 290
recirculating flow, 740
reference elements, 423
regular refinement, 480
reversibility, 692
Reynolds number, 1037
Riemann problem, 882
Riesz representation theorem, 473
right hand side vector, 370
Ritz-Galerkin discretization, 78
Robin boundary conditions, 251
rubber band, 25
Runge-Kutta
 increments, 637
Runge-Kutta method, 637
Runge-Kutta methods
 stability function, 981
saddle point problem, 1051
 linear, 1053, 1069
 variational, 1052
SDIRK timestepping, 660
semi-discrete evolution problem, 632
semi-norm, 150
sensitivity

 of a problem, 223
shape functions
 global, 332
shape regularity measure, 516
shock, 884
 physical, 899
 subsonic, 946
 supersonic, 946
shock speed, 884
similarity solution, 893
simplicial mesh, 328
singular perturbation, 741
slope limiter, 1022
slope limiting, 1016
Sobolev norms, 520
Sobolev semi-norms, 522
Sobolev space $H^1(\Omega)$, 206
Sobolev space $H_0^1(\Omega)$, 204
Sobolev spaces, 197, 520
solution
 analytic, 74
 approximate, 75
source term, 171
space-time-cylinder, 615
sparsity pattern, 306
spatial domain, 176
spectrum, 975
spline
 cubic, 138
spline collocation, 137
spring constant, 33

Störmer scheme, 697
 stability, 1090
 of linear variational problem, 223
 stability domain, 983
 stability function
 of explicit Runge-Kutta methods, 981
 of RK-SSM, 660
 Stable finite element pair, 1091
 stiff IVP, 646
 stiffness
 of spring, 33
 stiffness matrix, 285, 369
 sparsity, 335
 Stokes problem
 variational form, 1056
 Strang splitting, 789
 streamline, 719, 739
 streamline diffusion, 762
 strong form, 63
 subsonic rarefaction, 946
 subsonic shock, 946
 supersonic rarefaction, 946
 supersonic shock, 946
 supremum norm, 146
 symbol
 of a difference operator, 975
 T-matrix, 373
 Taylor expansion, 48
 Taylor-Hood finite element, 1103
 tensor product polynomials, 331
 tensor-product grid, 444
 tent function, 111, 299
 test function, 53
 test space, 53
 TETGEN, 361
 thermodynamics
 2nd law, 1041
 trace theorem, 271
 traffic flow
 velocity model, 839
 trajectory, 719
 transformation of functions, 407
 transformation techniques, 422
 translation-invariant, 968
 transonic rarefaction, 946
 transport equation, 785
 transsonic rarefaction fan, 937
 trapezoidal rule
 composite, 116, 117
 global, 756
 trial space, 53, 128
 triangle inequality, 145
 triangular mesh, 326
 triangular mesh: file format, 357
 triangulation, 325
 two-point boundary value problem, 63
 two-step method, 697

 uniformly positive, 248
 upwind quadrature, 753, 755, 758
 upwinding, 751
 variational crime, 559

variational equation
 linear, 67
 non-linear, 50
 variational formulation
 spatial, 623
 variational problem
 discrete, 281, 285
 linear, 67
 non-linear, 431
 perturbed, 560
 vector Laplacian, 1062
 vertex, 325
 virtual work principle, 51
 von Neumann stability analysis, 983
 Voronoi cell, 459
 Voronoi dual mesh, 459
 vortex, 1040

 wave equation, 679
 weak form, 63
 weak solution, 876
 weight
 quadrature, 96
 well-posedness, 1090
 width
 of a mesh, 485

R. Hiptmair
 C. Schwab,
 H.
 Harbrecht
 V.
 Gradinaru
 A. Chernov
 P. Grohs

SAM, ETHZ

Examples and Remarks

- L^2 interpolation error, 592
- L^2 -convergence of FE solutions, 589
- L^2 -estimates on non-convex domain, 596
- L^∞ interpolation error estimate in 1D, 534
- $\mathbf{H}_0^1(\operatorname{div} 0, \Omega)$ -conforming finite elements, 1049
- $|\cdot|_{H^1(\Omega)}$ -semi-norm, 207
- (Bi)-linear Lagrangian finite elements on hybrid meshes, 353
- [Membrane with free boundary values, 240
- ode45 for discrete parabolic evolution, 643
- “PDEs” for univariate functions, 24
- “Physics based” discretization, 76
- 1D convection-diffusion boundary value problem, 737
- Acceleration based traffic modeling, 840
- Adequacy of 2nd-order timestepping, 1029
- Admissible Dirichlet data, 268
- Admissible Neumann data, 269
- Affine transformation of triangles, 388
- Approximate computation of norms, 152
- Approximate Dirichlet boundary conditions, 400
- Approximate sub-steps for Strang splitting time, 794
- Approximation of mean temperature, 572, 575
- Arrays storing 2D triangular mesh, 364
- Assembly algorithm for linear Lagrangian finite elements, 315
- Assembly for linear Lagrangian finite elements on triangular mes, 374
- Assembly for quadratic Lagrangian FEM, 377
- Assembly of right hand side vector for linear finite elements, 322
- Asymptotic nature of a priori estimates, 539
- Barycentric representation of local shape functions, 413
- Bases for polynomial spectral collocation, 133
- Behavior of generalized eigenvalues of $\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu}$, 649
- Benefit of variational formulation of BVPs, 112
- Bilinear Lagrangian finite elements, 346
- Blow-up for leapfrog timestepping, 704
- Boundary conditions and $L^2(\Omega)$, 202
- Boundary conditions for 2nd-order parabolic IBVPs, 621
- Boundary conditions for linear advection, 836

Boundary conditions for wave equation, 680	Convergence of Euler timestepping, 638	Numerical Methods for PDEs
Boundary conditions in $H_0^1(\Omega)$, 205	Convergence of fully discrete finite volume methods for Burgers equation, 987	
Boundary value problems, 171	Convergence of fully discrete timestepping in one spatial dimension, 662	
Boundary values for conservation laws, 864	Convergence of FV with linear reconstruction, 1005	
Breakdown of characteristic solution formula, 870	Convergence of Lagrangian FEM for p -refinement, 494	
Causes for non-smoothness of solutions of elliptic BVPs, 558	Convergence of linear and quadratic Lagrangian finite elements in L^2 -norm, 487	
Central flux for Burgers equation, 920	Convergence of linear and quadratic Lagrangian finite elements in energy norm, 483	
Central flux for linear advection, 925	Convergence of MUSCL scheme, 1031	
Central flux for Traffic Flow equation, 923	Convergence of P2-P0 scheme for Stokes equation, 1099	
Characteristics for advection, 868	Convergence of SUPG and upwind quadrature FEM, 776	
Checkerboard instability for quadrilateral P1-P0 pair, 1074	Convergence of Taylor-Hood method for Stokes problem, 1105	R. Hiptmair C. Schwab, H. Harbrecht V. Gradinaru A. Chernov P. Grohs
Choice of basis for polynomial spectral Galerkin methods, 89	Corner singular functions, 552	SAM, ETHZ
Choice of timestepping for m.o.I. for transient convection-diffusion, 784	Crank-Nicolson timestepping, 636	
Coefficients/data in procedural form, 77	Decoupling of velocity components ?, 1046	
Collocation approach on “complicated” domains, 442	Delaunay-remeshing in 2D, 806	
Collocation points for polynomial spectral collocation, 133	Derivative of non-linear $u \mapsto a(u; \cdot)$, 436	
Collocation: smoothness requirements for coefficients, 130	Difference stencils, 967	
Compatible boundary and initial data, 620	Differentiating a functional on a space of curves, 50	
Computation of heat flux, 579, 586	Differentiating bilinear forms with time-dependent arguments 627	
Conditioning of linear variational problems, 223	Diffusive flux, 862	
Conditioning of spectral Galerkin system matrices, 101	Dimensionless equations, 30	
Consequence of monotonicity preservation, 1018	Dimensions of Lagrangian finite element spaces on triangular meshes, 543	12.5
Consistency error of Lax-Friedrichs numerical flux, 996	Discontinuity connecting constant states, 880	p. 1125
Consistency error of upwind numerical flux, 995		
Continuity of interpolation operators, 523		
Convective cooling, 251		
Convergence for conditionally stable Runge-Kutta timestepping, 670		

- Domain of dependence/influence for 1D wave equation, constant coefficient case, 684
- Effect of added diffusion, 764
- Efficient finite element discretization of Stokes problem, 1101
- Efficient implementation of assembly, 381
- Elastic string shape by finite element discretization, 126
- Elliptic lifting result in 1D, 550
- Energy conservation for leapfrog, 701
- Energy norm, 148
- Energy norm and $H^1(\Omega)$ -norm, 518
- Enforcing zero mean, 1059
- Ensuring uniqueness of pressure, 1056
- Entropy solution of Burgers equation, 901
- Entropy solution of Traffic Flow equation, 902
- Euler equations, 856
- Euler timestepping, 634
- Euler timestepping for 1st-order form of semi-discrete wave equation, 692
- Evaluation of local shape functions at quadrature points, 412
- Explicit Euler in Fourier domain, 979
- Exploring convergence experimentally, 163
- Extended MATLAB mesh data structure, 366
- Extra regularity requirements, 64
- Extra smoothness requirement for PDE formulation, 239
- Fan patterns in traffic flow, 888
- Finding continuous replacement functionals, 588
- Finite differences for convection-diffusion equation in 1D, 743
- First-order semidiscrete hyperbolic evolution problem, 691
- Fourier series, 977
- Fully discrete evolutions, 966
- Gap between interpolation error and best approximation error, 532
- General asymptotic estimates, 547
- Generic constants, 538
- Geometric interpretation of CFL condition in 1D, 708
- Geometric obstruction to Voronoi dual meshes, 460
- Godunov flux for Burgers equation, 950
- Godunov flux for traffic flow equation, 951
- Good accuracy on “bad” meshes, 529
- Graph description of string shape, 71
- Grid functions, 144
- heat conduction, 253
- Heat conduction with radiation boundary conditions, 432
- Heuristics behind Lagrangian multipliers, 1050
- Higher order timestepping for 1D heat equation, 665
- Impact of choice of basis, 286
- Impact of linear boundary fitting on FE convergence, 565
- Impact of numerical quadrature on finite element discretization error, 562
- Implementation of non-homogeneous Dirichlet b.c. for linear FE, 401
- Implementation of spectral Galerkin discretization for elastic string problem, 103
- Implementation of spectral Galerkin discretization for linear 2nd-order two-point BVP, 97
- Implicit Euler method of lines for transient convection-diffusion, 781

Imposing homogeneous Dirichlet boundary conditions, 352
 Improved resolution by limited linear reconstruction, 1026
 Inefficiency of conditionally stable single step methods, 669
 Initial time, 616
 Internal layers, 765
 Interpolation nodes for cubic and quartic Lagrangian FE in 2D, 344

 $L(\pi)$ -stable Runge-Kutta single step methods, 660
 Lagrangian finite elements on hybrid meshes, 355
 Lagrangian method for convection-diffusion in 1D, 810
 Lagrangian method for convection-diffusion in 2D, 814
 Laplace operator, 238
 Lax-Friedrichs flux for Burgers equation, 929
 Lax-Friedrichs flux for traffic flow equation, 930
 Leapfrog timestepping, 698
 Linear FE discretization of 1D convection-diffusion problem, 744
 Linear finite element Galerkin discretization for elastic string model, 120
 Linear finite element space for homogeneous Dirichlet problem, 299
 Linear reconstruction with central slope, 1007, 1008
 Linear reconstruction with minmod limiter, 1022, 1024
 Linear reconstruction with one-sided slopes, 1011, 1012
 Linear variational problems, 67
 Linearity and monotonicity preservation, 1019
 Local interpolation onto higher degree Lagrangian finite element spaces, 535
 Local quadrature rules on quadrilaterals, 395
 Local quadrature rules on triangles, 392

Mass lumping, 700
 Material coordinate, 30
 Mathematical modelling, 23
 Maximum principle for finite difference discretization, 604
 Maximum principle for higher order Lagrangian FEM, 612
 Maximum principle for linear FE for 2nd-order elliptic BVPs, 611
 Mesh file format for MATLAB code, 359
 Minimal regularity of membrane displacement, 180
 Mixed boundary conditions, 253

 Naive finite difference discretization of Stokes system, 1064
 Naive finite difference scheme, 908
 Non-continuity of boundary flux functional, 583
 Non-differentiable function in $H_0^1(]0, 1[)$, 215
 Non-existence of solutions of positive definite quadratic minimization problem, 195
 Non-homogeneous Dirichlet boundary conditions in LehrFEM, 403
 Non-linear variational equation, 51
 Non-polynomial “bilinear” local shape functions, 421
 Non-smooth external forcing, 57
 Norms on grid function spaces, 151
 Numerical quadrature in LehrFEM, 394

 Offset function for finite element Galerkin discretization, 118
 Offset functions and Ritz-Galerkin discretization, 82
 offset functions for linear Lagrangian FE, 399
 One-sided difference approximation of convective terms, 748
 Ordered basis of test space, 84

Output functionals, 570	Semi-Lagrangian method for convection-diffusion in 2D, 827	Numerical Methods for PDEs
P1-P0 quadrilateral finite elements for Stokes problem, 1077	Shock patterns in traffic flow, 887	
P2-P0 finite element scheme for the Stokes problem, 1082	Smoothness of solution of scalar elliptic boundary value problem, 256	
Parameterization of a curve, 28	Smoothness requirements for collocation trial space, 130	
Particle simulation of traffic flow, 840, 845	Solution formula for sourceless transport, 786	
Piecewise gradient, 296	Space of square integrable functions, 200	
Piecewise linear functions (not) in H^1 , 213	Sparse stiffness matrices, 305	
Piecewise quadratic interpolation, 536	Spatial difference operators for linear advection, 974	
Point charge, 227	Spatial discretization options, 632	
Point particle method for pure advection, 798	Spatial domains, 176	
Pressure Poisson equation, 1063	Specification of local quadrature rules, 391	R. Hiptmair C. Schwab, H. Harbrecht V. Gradinaru A. Chernov P. Grohs
Properties of weak solutions, 876	Spectral Galerkin discretization of non-linear variational problem, 107	
Pullback of functions, 407	Spectral Galerkin discretization with quadrature, 95	
Quadratic functionals with positive definite bilinear form in 2D, 192	Spectrum of elliptic operators, 654	
Quadratic minimization problem, 66	Spectrum of upwind difference operator, 975	
Quadratic minimization problems on Hilbert spaces, 203	Spurious Galerking solution for 2D convection-diffusion BVP, 751	
Quadratic tensor product Lagrangian finite elements, 350	Stability and CFL condition, 984	
Quasi-locality of solution of scalar elliptic boundary value problem, 257	Stability domains, 983	
Radiative cooling, 252	Stability functions of explicit RK-methods, 981	
Relationship between discrete minimization problem and discrete variational problem, 80	Stability of pressure solution: inf-sup condition, 1093	
Scalar elliptic boundary value problem in one space dimension, 255	Stable velocity solution, 1092	SAM, ETHZ
Scaling of entries of element matrix for $-\Delta$, 312	Streamline-diffusion discretization, 771	
Semi-Lagrangian method for convection-diffusion in 1D, 825	Streamlines, 739	
	Suitability of macroscopic models for traffic flow, 848	
	Supports of global shape functions in 1D, 333	
	Supports of global shape functions on triangular mesh, 333	

Tense string without external forcing, 43
Timestepping for ODEs, 77
Transformation of basis functions, 94
Transformation techniques for bilinear transformations, 427
Triangular mesh: file format, 357
Triangular quadratic Lagrangian finite elements, 339

Uniqueness of solutions of Neumann problem, 264
Unstable P1-P0 finite element pair on triangular mesh,
1072
Upwind flux and expansion shocks, 948
Upwind flux and transsonic rarefaction, 935
Upwind flux for Burgers equation, 933
Upwind flux for traffic flow equation, 934
Upwind quadrature discretization, 760

Vanishing viscosity for Burgers equation, 890
Variational formulation for convection-diffusion BVP, 734
Variational formulation for heat conduction with Dirichlet
boundary conditions, 259
Variational formulation for pure Neumann problem, 263
Variational formulation: heat conduction with general radi-
ation boundary conditions, 261
Virtual work principle, 51

Wave equation as first order system in time, 681
Well-posed 2nd-order linear elliptic variational problems,
474

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

Definitions

- H^1 -semi-norm, 150
- Affine transformation, 389
- Characteristic curve for one-dimensional scalar conservation law, 867
- congruent matrices, 288
- Consistent modifications of variational problems, 769
- Consistent numerical flux function, 917
- Courant-Friedrichs=Levy (CFL-)condition, 971
- Cubic spline, 138
- element load vector, 371
- element stiffness matrix, 371
- Energy norm, 191
- Higher order Lagrangian finite element spaces, 339
- Higher order Sobolev norms, 520
- Higher order Sobolev semi-norms, 522
- Higher order Sobolev spaces, 520
- Incompressible flow field, 726
- $L(\pi)$ -stability, 660
- Lax entropy condition, 899
- Legendre polynomials, 91
- Linear interpolation in 2D, 508
- Linear reconstruction, 1001
- Local shape functions, 336
- Material derivative, 819
- Mean square norm/ L^2 -norm, 147
- mesh, 325
- mesh width, 485
- Minmod reconstruction, 1021
- Monotone numerical flux function, 956
- Monotonicity preserving linear interpolation, 1017
- multivariate polynomials, 329
- norm, 145
- Numerical domain of dependence, 969
- parametric finite elements, 423
- Positive definite bilinear form, 190
- pullback, 407
- Quadratic functional, 187

Quadratic minimization problem, 188
Riemann problem, 882
Runge-Kutta method, 637
Shape regularity measure for simplex, 516
shock, 884
Singularly perturbed problem, 741
Sobolev space $H^1(\Omega)$, 206
Sobolev space $H_0^1(\Omega)$, 204
Space $L^2(\Omega)$, 201
sparse matrix, 305
Stable finite element pair, 1091
Support of a function, 114
Supremum norm, 146
Tensor product Langrangian finite element spaces, 350
tensor product polynomials, 331
Uniformly positive definite tensor field, 184
Weak solution of Cauchy problem for conservation law,
876

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

MATLAB codes

assemMat_QFE, 371
sparse (MATLAB-function), 371
add_Edge2Elem, 358
add_Edges, 358
init_Mesh, 358
Loading a mesh from file, 353

R. Hiptmair
C. Schwab,
H.
Harbrecht
V.
Gradinaru
A. Chernov
P. Grohs

SAM, ETHZ

List of Symbols

- $C_0^2([0, 1]) := \{v \in C^2([0, 1]): v(0) = v(1) = 0\}$, 47
 $C_0^\infty(\Omega) \hat{=}$ smooth functions with support inside Ω , 209
 $C^k([a, b]) \hat{=}$ k -times continuously differentiable functions
 on $[a, b] \subset \mathbb{R}$, 28
 $C_{\text{pw}}^k([a, b])$, 57
 $D^-(\bar{x}, \bar{t}) \hat{=}$ maximal analytical domain of dependence of
 (\bar{x}, \bar{t}) , 970
 $D^\alpha u \hat{=}$ multiple partial derivatives, 520
 $L_*^2(\Omega) := \{q \in L^2(\Omega): \int_\Omega q \, d\mathbf{x} = 0\}$, 1056
 $M_i \hat{=}$ i -th integrated Legendre polynomial, 90
 $O(f(N)) \hat{=}$ Landau- O for $N \rightarrow \infty$, 159
 $S(z) \hat{=}$ stability function of Runge-Kutta method, 981
 \mathbf{n} , 250
 $\mathbf{n} \hat{=}$ exterior unit normal vectorfield, 234
 $\mathcal{H}_h \hat{=}$ fully discrete evolution operator, 966
 $\mathcal{L}_h \hat{=}$ semi-discrete evolution operator doe 1D conserva-
 tion law, 964
 $\mathcal{P}_p(\mathbb{R}) \hat{=}$ space of univariate polynomials of degree $\leq p$,
 87
 $\mathcal{P}_p(\mathbb{R}^d)$, 329
 $\mathcal{P}_p(\mathbb{R}^d) \hat{=}$ space of d -variate polynomials, 329
 $\mathcal{Q}_p(\mathbb{R}^d)$, 331
 $\mathcal{V}(\mathcal{M}) \hat{=}$ set of vertices of a mesh, 292
 $\Delta \hat{=}$ Laplace operator, 238
 $\Delta \hat{=}$ vector Laplacian, 1062
 $\text{div } \mathbf{j} \hat{=}$ divergence of a vectorfield, 233
 $\mathcal{E}(\mathcal{M})$, 366
 l_1 , 508
 $\Gamma_{\text{in}} \hat{=}$ inflow boundary for advection BVP, 739
 $H^m(\Omega) \hat{=}$ m -th order Sobolev space, 520
 $\mathbf{H}_0^1(\text{div } 0, \Omega) \hat{=}$ componentwise $H_0^1(\Omega)$ -vectorfields with
 vanishing divergence., 1048
 $\mathcal{S}_1^0(\mathcal{M})$, 294
 $l_1 \hat{=}$ piecewise linear interpolation on finite element mesh,
 464
 $P_n \hat{=}$ n -th Legendre polynomial, 91
 $\mathcal{S}_p^0(\mathcal{M}) \hat{=}$ $H^1(\Omega)$ -conforming Lagrangian FE space, 339
 $L^\infty(\Omega) \hat{=}$ space of (essentially) bounded functions on Ω ,
 146
 $L^2(\Omega) \hat{=}$ space of square-integrable functions on Ω , 201

$\|\cdot\|_0 \hat{=}$ norm on $L^2(\Omega)$, 201
 $\|\cdot\|_\infty \hat{=}$ supremum norm of a function/maximum norm of a vector, 146
 $\|u\|_{H^m(\Omega)} \hat{=}$ m -th order Sobolev norm, 520
 $\|u\|_{L^\infty(\Omega)} \hat{=}$ supremum norm of $u : \Omega \mapsto \mathbb{R}^n$, 146
 $\|\cdot\|_{L^2(\Omega)} \hat{=}$ L^2 -norm of a function, 147
 $\|\cdot\|_{L^2(\Omega)} \hat{=}$ norm on $L^2(\Omega)$, 201
 $\|\cdot\|_0 \hat{=}$ L^2 -norm of a function, 147
 $\mathcal{V}(\mathcal{M})$, 325
 Ω , 171
 $\Omega \hat{=}$ spatial domain or parameter domain, 28
 Φ^* , 407
 $|u|_{H^m(\Omega)}$ m -th order Sobolev semi-norm, 522
 $|\cdot|_{H^1(\Omega)} \hat{=}$ H^1 -semi-norm of a function, 150
 $\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$ (matrices), 285
 $\mathbf{A} : \mathbf{B} \hat{=}$ componentwise dot product of matrices, 1048
 \mathbf{M}^{-T} *hat*= inverse transposed of matrix, 426
 $\mathbf{a}_K \hat{=}$ restriction of bilinear form \mathbf{a} to cell K , 308
 $\cdot \hat{=}$ inner product of vectors in \mathbb{R}^n , 35
 $\chi_I \hat{=}$ characteristic function of an interval $I \subset \mathbb{R}$, 912
 $\mathbf{curl} \hat{=}$ rotation/curl of a vector field, 1040
 $\ddot{u} := \frac{\partial u}{\partial t^2}$, 677
 $\dot{u}(t) \hat{=}$ (partial) derivative w.r.t. time, 624
 ℓ_K restriction of linear form ℓ to cell K , 319
 $\frac{Df}{D\mathbf{v}}(t) \hat{=}$ material derivative w.r.t. velocity field \mathbf{v} , 819
 $\mathbf{grad} \hat{=}$ gradient of a scalar valued function, 178
 $\hat{c}(\xi) \hat{=}$ symbol of a finite difference operator, 975
 $\mathbf{1} = (1, \dots, 1)^T$, 981
 $dS \hat{=}$ integration over a surface, 234
 \mathcal{M} , 325

$\nabla F(\mathbf{x}) := \mathbf{grad} F(\mathbf{x}) \hat{=}$ nabla notation for gradient, 179
 $\text{diam}(\Omega) \hat{=}$ diameter of $\Omega \subset \mathbb{R}^d$, 177
 nnz , 305
 $\partial\Omega \hat{=}$ boundary of domain Ω , 177
 $\rho_K \hat{=}$ shape regularity measure of cell K , 516
 $\rho_{\mathcal{M}} \hat{=}$ shape regularity measure of a mesh \mathcal{M} , 516
 $\vec{\mu}, \vec{\varphi}, \vec{\xi}, \dots$ (coefficient vectors), 285
 $\mathcal{S}_{p,0}^0(\mathcal{M}) \hat{=}$ Degree p Lagrangian finite element space with zero Dirichlet boundary conditions., 352
 $\mathcal{S}_{1,0}^0(\mathcal{M}) \hat{=}$ space of p.w. linear C^0 -finite elements, 111
 $H_0^1(\Omega)$ Sobolev space, 204
 $\mathbf{H}_0^1(\Omega) \hat{=}$ componentwise $H_0^1(\Omega)$ -vectorfields, 1048
 $h_{\mathcal{M}} \hat{=}$ mesh width of mesh \mathcal{M} , 485
 $h_{\mathcal{M}} \hat{=}$ meshwidth of a grid, 110
 $x_{j-1/2} := \frac{1}{2}(x_j + x_{j-1}) \hat{=}$ midpoint of cell in 1D, 912
 $\|\cdot\| \hat{=}$ Euclidean norm of a vector $\in \mathbb{R}^n$, 29

Bibliography

- [1] I. Babuška and M. Suri. The optimal convergence rate of the p-version of the finite element method. *SIAM J. Numer. Anal.*, 24(4):750–769, 1987.
- [2] Nicola Bellomo and Christian Dogbe. On the Modeling of Traffic and Crowds: A Survey of Models, Speculations, and Perspectives. *SIAM REVIEW*, 53(3):409–463, 2011.
- [3] P. Bochev and R. B. Lehoucq. On the finite element solution of the pure neumann problem. *SIAM Review*, 47(1):50–66, 2005.
- [4] D. Braess. *Finite Elements*. Cambridge University Press, 2nd edition, 2001.
- [5] S. Brenner and R. Scott. *Mathematical theory of finite element methods*. Texts in Applied Mathematics. Springer–Verlag, New York, 1994.
- [6] S. Brenner and R. Scott. *Mathematical theory of finite element methods*. Texts in Applied Mathematics. Springer–Verlag, New York, 2nd edition, 2002.

- [7] A. Bartscher, E. Fonn, P. Meury, and C. Wiesmayr. *LehrFEM - A 2D Finite Element Toolbox*. SAM, ETH Zürich, Zürich, Switzerland, 2010. <http://www.sam.math.ethz.ch/~hiptmair/tmp/LehrFEM-Manual.pdf>.
- [8] S. Childress. Notes on traffic flow. Online notes, <http://chaton.inf.ethz.ch/cselabJoomla/images/NPI2005>.
- [9] D. Christodoulou. The Euler equations of compressible fluid flow. *Bull. American Math. Soc.*, 44(4):581–602, 2007.
- [10] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 4 of *Studies in Mathematics and its Applications*. North-Holland, Amsterdam, 1978.
- [11] B. Cockburn and J. Gopalakrishnan. New hybridization techniques. *GAMM-Mitteilungen*, 2:28, 2005.
- [12] C.M. Dafermos. *Hyperbolic conservation laws in continuum physics*, volume 325 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, 2000.
- [13] D.A. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle. *Int. J. Numer. Meth. Engr.*, 21:1129–1148, 1985.
- [14] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*, volume 159 of *Applied Mathematical Sciences*. Springer, New York, 2004.
- [15] L.C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1998.

- [16] V. Girault and P.A. Raviart. *Finite element methods for Navier–Stokes equations*. Springer, Berlin, 1986.
- [17] Ch. Großmann and H.-G. Roos. *Numerik partieller Differentialgleichungen*. Teubner, Stuttgart, 3rd edition, 2005.
- [18] W. Hackbusch. *Elliptic Differential Equations. Theory and Numerical Treatment*, volume 18 of *Springer Series in Computational Mathematics*. Springer, Berlin, 1992.
- [19] W. Hackbusch. *Integral equations. Theory and numerical treatment.*, volume 120 of *International Series of Numerical Mathematics*. Birkhäuser, Basel, 1995.
- [20] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2 edition, 2006.
- [21] R. Hiptmair. Numerische mathematik für studiengang rechnergestützte wissenschaften. Lecture Slides, 2005. <http://www.sam.math.ethz.ch/~hiptmair/tmp/NCSE.pdf>.
- [22] P. Knabner and L. Angermann. *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*, volume 44 of *Texts in Applied Mathematics*. Springer, Heidelberg, 2003.
- [23] D. Kröner. *Numerical Schemes for Conservation Laws*. Wiley-Teubner, Chichester, 1997.
- [24] S. Larsson and V. Thomée. *Partial Differential Equations with Numerical Methods*, volume 45 of *Texts in Applied Mathematics*. Springer, Heidelberg, 2003.
- [25] R. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, UK, 2002.

- [26] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.*, 21(2):217–235, 1984.
- [27] P.-O. Persson and G. Strang. A simple mesh generator in matlab. *SIAM Review*, 46(2):329–345, 2004.
- [28] H.-G. Roos, M. Stynes, and L. Tobiska. *Numerical methods for singularly perturbed differential equations. Convection-diffusion-reaction and flow problems*, volume 24 of *Springer Series in Computational Mathematics*. Springer, Berlin, 2nd edition, 2008.
- [29] C. Schwab. *p - and hp -Finite Element Methods. Theory and Applications in Solid and Fluid Mechanics*. Numerical Mathematics and Scientific Computation. Clarendon Press, Oxford, 1998.
- [30] L.R. Scrott and M. Vogelius. Conforming finite element methods for incompressible and nearly incompressible continua. In B. Engquist, S. Osher, and R. Sommerville, editors, *Large-scale computations in fluid mechanics. Proc. 15th AMS-SIAM Summer Semin. Appl. Math., La Jolla/Calif. 1983*, volume 22 of *Lect. Appl. Math.*, pages 221–243. AMS, Providence, RI, 1985.
- [31] R. Stenberg. Error analysis of some finite element methods for the stokes problem. *Math. Comp.*, 54(190):495–508, 1990.
- [32] M. Struwe. *Analysis für Informatiker*. Lecture notes, ETH Zürich, 2009. <https://moodle-app1.net.ethz.ch/lms/mod/resource/index.php?id=145>.
- [33] J. Xu and L. Zikatanov. A monotone finite element scheme for convection diffusion equations. *Math. Comp.*, 68(228):1429–1446, May 1999.

- [34] E. Zeidler. *Nonlinear Functional Analysis and its Applications. III: Variational Methods and Optimization*. Springer–Verlag, New York, Berlin, Heidelberg, 1990.
- [35] S. Zlotnik and P. DÃÑez. Assembling sparse matrices in MATLAB. *Communications in Numerical Methods in Engineering*, 2008. Published Online: 1 Sep 2008.