ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Diploma Thesis

A Discontinuous Galerkin Scheme for Elastic Waves in Nearly Incompressible Materials

Christoph Winkelmann

March 12, 2004

advised by

Prof. Ralf Hiptmair
Seminar for Applied Mathematics
Department of Mathematics
Swiss Federal Institute of Technology
Zürich

Abstract

Two discontinuous Galerkin schemes for linear elastic waves in two and three dimensions with Dirichlet and optional Neumann boundary conditions are formulated: a pure displacement formulation and a mixed formulation with the displacement and a pressure parameter as variables. The schemes are implemented for two dimensions using *Concepts* with piecewise linear orthogonal basis functions for the displacements and piecewise constant functions for the pressure parameter, both on unstructured triangular meshes. Symmetric and nonsymmetric variants are tested in static and dynamic test cases, using implicit and explicit timestepping schemes and different linear solvers.

All symmetric schemes are found to be locking free, while the nonsymmetric ones behave badly for reasons made plausible. The non-mixed scheme has optimal convergence, but computation costs depend on compressibility. The mixed scheme can overcome this problem, but only in the stationary case.

Contents

1 Introduction

The phenomenon of elastic waves in nearly incompressible materials is of interest in technical applications for rubbers and, more recently, in biomechanical applications for biomaterials, e. g. in car crashes. For homogeneous isotropic materials, the waves can be described by the following linearized partial differential equation:

$$\rho\, \partial_t^2 \underline{u} - \mu \Delta \underline{u} - (\mu + \lambda)\, \underline{\nabla}\, (\underline{\nabla} \cdot \underline{u}) \;\; = \;\; \underline{f}$$

ρ stands for the mass density, $\underline{u}$ is the displacement, $\partial_t^2 \underline{u}$ is the acceleration; $\underline{f}$ is a volume force and μ and λ are the so called Lamé parameters. Nearly incompressible materials are characterized by $\mu \ll \lambda$.

Using standard continuous finite element methods (FEM) usually leads to a phenomenon called volume locking: The numerical displacement is much smaller than the exact displacement. In mathematical terms, volume locking means that the upper bound on the numerical error increases with λ/μ. Once a certain ratio is reached, the numerical results are more or less useless.

There are several possibilities to overcome this problem. The most known one is probably a mixed FEM with elements of higher order in space. Recent research [6], [11], shows that for stationary problems, the problem of volume locking does not occur with discontinuous Galerkin FEM (DGFEM) discretisations even of low order. Also, a mixed formulation is not necessary.

In this present work, we try to extend the DGFEM to the time dependent case and test experimentally how well this works and how expensive the calculations are. Chapter 2 formulates a non-mixed scheme and presents time discretizations. In chapter 4, the results for the non-mixed scheme are shown. In chapter 5 we formulate and test an alternative mixed formulation, and in chapter 6, we draw some conclusions.

2 Formulation

The entire formulation follows [6], [10] and [11] closely, generalizing to the time dependent, three dimensional problem.

2.1 Continuous Equations

2.1.1 Initial Boundary Value Problem of Elastic Wave Propagation

Let $\Omega \subset I\!R^d$ be a bounded polygon (polyhedron) for $d = 2$ ($d = 3$). We denote the boundary of the domain by Γ, and partition it into a Dirichlet boundary Γ_D and a Neumann boundary Γ_N. The length (area) of the Dirichlet boundary $|\Gamma_D|$ has to be strictly positive.

The linearized elastodynamic problem reads as follows:

$$\begin{aligned}
\rho\, \partial_t^2 \underline{u}\,(\underline{x},t) - \underline{\nabla} \cdot \underline{\underline{\sigma}}\,(\underline{u}\,(\underline{x},t)) &= \underline{f}\,(\underline{x},t) \quad (\underline{x},t) \in \Omega \times (0,T)\,, \quad T > 0 \qquad (2.1)\\
\partial_t \underline{u}\,(\underline{x},0) &= \underline{u}_1\,(\underline{x}) \qquad \underline{x} \in \Omega\\
\underline{u}\,(\underline{x},0) &= \underline{u}_0\,(\underline{x}) \qquad \underline{x} \in \Omega\\
\underline{u}\,(\underline{x},t) &= \underline{g}_D\,(\underline{x},t) \quad (\underline{x},t) \in \Gamma_D \times (0,T)\\
\underline{\underline{\sigma}}\,(\underline{u}\,(\underline{x},t)) \cdot \underline{n} &= \underline{g}_N\,(\underline{x},t) \quad (\underline{x},t) \in \Gamma_N \times (0,T)
\end{aligned}$$

ρ is the mass density, ∂_t denotes a partial derivative with respect to time $\partial/\partial t$, $\underline{u} : \Omega \times (0,T) \to I\!R^d$ is the displacement and $\underline{\underline{\sigma}}$ is the (Cauchy) stress tensor. Its divergence $\underline{\nabla} \cdot \underline{\underline{\sigma}}$ is defined as the column vector $\left\{\sum_j \partial_{x_j} \sigma_{ij}\right\}_i$. $\underline{f}$ is a given volume force density, $\underline{u}_0$ the initial displacement and $\underline{u}_1$ is the initial velocity. $\underline{g}_D$ denotes the prescribed boundary displacement, $\underline{g}_N$ is the prescribed boundary force, and $\underline{n}$ is the outer unit normal to Ω.

In linearized theory, $\underline{\underline{\sigma}}$ is given by

$$\underline{\underline{\sigma}}\,(\underline{u}) = \underline{\underline{\underline{\underline{A}}}}\, \underline{\underline{\varepsilon}}\,(\underline{u}) \qquad (2.2)$$

where $\underline{\underline{\varepsilon}}$ denotes the strain tensor. It is defined as the symmetric gradient of $\underline{u}$:

$$\underline{\underline{\varepsilon}}\,(\underline{u}) = {}^1\!/_2 \left(\underline{\nabla}\, \underline{u}^\top + \left(\underline{\nabla}\, \underline{u}^\top\right)^\top\right)$$

For homogeneous isotropic materials, the elasticity tensor $\underline{\underline{\underline{\underline{A}}}}$ is given by

$$A_{ijrs} = \mu\delta_{ir}\delta_{js} + \mu\delta_{is}\delta_{jr} + \lambda\delta_{ij}\delta_{rs}$$

where $\mu > 0$ and $\lambda > -2/3\mu$ are the Lamé coefficients. They can be expressed in terms of Young's modulus $E > 0$ and Poisson's ratio $\nu \in (-1, 1/2]$:

$$\lambda = E\,\frac{\nu}{(1+\nu)\,(1-2\nu)}, \qquad \mu = E\,\frac{1}{2\,(1+\nu)}$$

The expression for the strain tensor can then be written as

$$\underline{\underline{\sigma}}\,(\underline{u}) \;=\; 2\,\mu\,\underline{\underline{\varepsilon}}\,(\underline{u}) + \lambda \underline{\nabla} \cdot \underline{u}\;\underline{\underline{1}}_{d\times d}$$

and the differential equation can be simplified to

$$\rho\,\partial_t^2 \underline{u} - \mu \Delta \underline{u} - (\mu+\lambda)\,\underline{\nabla}\,(\underline{\nabla} \cdot \underline{u}) \;=\; \underline{f} \tag{2.3}$$

In order to stay more general, we only assume linear (2.2), but not necessarily homogeneous isotropic material, i. e. we will discretize equation (2.1).

In this setting, nearly incompressible materials are characterized by $\nu \to 1/2$, which implies $\lambda \to \infty$.

2.1.2 Boundary Value Problem of Elastic Equilibrium

The numerical methods in consideration were developed for and can be tested with the static problem associated with (2.1). It follows from (2.1) by setting $\partial_t \underline{u} = 0$:

$$\begin{aligned} -\underline{\nabla} \cdot \underline{\underline{\sigma}}\,(\underline{u}) &= \underline{f} \quad \text{in } \Omega \\ \underline{u} &= \underline{g}_D \quad \text{on } \Gamma_D \\ \underline{\underline{\sigma}}\,(\underline{u}) \cdot \underline{n} &= \underline{g}_N \quad \text{on } \Gamma_N \end{aligned} \tag{2.4}$$

2.1.3 Eigenvalues and Eigenfunctions

Eigenvalues and eigenfunctions are of interest when considering the time dependent problem: Eigenvalues reveal information about caracteristic time scales, and knowing exact eigenfunctions for a given domain with given boundary conditions allows to construct exact time dependent solutions for this situation easily.

Substituting the ansatz $\underline{u}\,(\underline{x}, t) = c\,(t)\,\underline{w}\,(\underline{x})$ into (2.3) with $\underline{f} = 0$ immediately yields the eigenvalue problem:

$$-\rho\,\partial_t^2 c \;=\; \eta c \tag{2.5}$$

$$-\mu \Delta \underline{w} - (\mu+\lambda)\,\underline{\nabla}\,(\underline{\nabla} \cdot \underline{w}) \;=\; \eta \underline{w} \tag{2.6}$$

We denote the eigenvalues by η, as the standard symbol λ is already used for the second Lamé coefficient. Equation (2.5) has the solution

$$c\,(t) \;=\; c_0 \cos(\omega t) + \frac{c_1}{\omega} \sin(\omega t) \quad \text{with } \omega = \sqrt{\eta/\rho}$$

For the eigenfunctions in space, we set $\Omega = (-\pi, \pi)^d$ and assume periodic boundary conditions. We write the eigenfunctions $\underline{w}$ in a Fourier basis:

$$\underline{w}(\underline{x}) = \sum_{\underline{k} \in \mathbb{Z}^d} \underline{d}_{\underline{k}} \, e^{i \underline{k} \cdot \underline{x}} \tag{2.7}$$

This basis spans the space of square integrable functions $(L^2(\Omega))^d$ and is orthogonal in the L^2 scalar product on Ω. Substituting (2.7) into (2.6), we obtain

$$\begin{aligned} \sum_{\underline{k} \in \mathbb{Z}^d} -\mu \Delta \left(\underline{d}_{\underline{k}} \, e^{i \underline{k} \cdot \underline{x}} \right) - (\mu + \lambda) \, \underline{\nabla} \left(\underline{\nabla} \cdot \left(\underline{d}_{\underline{k}} \, e^{i \underline{k} \cdot \underline{x}} \right) \right) &= \\ \sum_{\underline{k} \in \mathbb{Z}^d} \left(\mu \left| \underline{k} \right|^2 \underline{\underline{1}} + (\mu + \lambda) \, \underline{k} \, \underline{k}^\top \right) \underline{d}_{\underline{k}} \, e^{i \underline{k} \cdot \underline{x}} &= \\ \sum_{\underline{k} \in \mathbb{Z}^d} \underline{\underline{K}}(\underline{k}) \; \underline{d}_{\underline{k}} \, e^{i \underline{k} \cdot \underline{x}} &= \eta \sum_{\underline{k} \in \mathbb{Z}^d} \underline{d}_{\underline{k}} \, e^{i \underline{k} \cdot \underline{x}} \end{aligned}$$

As this has to hold for all $\underline{x} \in \Omega$ and the basis functions are orthogonal, we have

$$\underline{\underline{K}}(\underline{k}) \; \underline{d}_{\underline{k}} = \eta \, \underline{d}_{\underline{k}} \quad \forall \underline{k} \in \mathbb{Z}^d$$

Thus, any $\underline{d}_{\underline{k}}$ satisfying the above equation defines an eigenfunction $\underline{d}_{\underline{k}} \, e^{i \underline{k} \cdot \underline{x}}$ with eigenvalue η.

For $d = 2$, we have for each $\underline{k} \in \mathbb{Z}^2 \setminus \{\underline{0}\}$

- a longitudinal mode: $\underline{d}_{\underline{k}} = \underline{k}$, $\eta = (2\mu + \lambda) \left| \underline{k} \right|^2$
- a transversal mode: $\underline{d}_{\underline{k}} = \underline{\underline{R}} \, \underline{k}$, $\eta = \mu \left| \underline{k} \right|^2$, $\underline{\underline{R}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

For $\underline{k} = \underline{0}$, we have the translation modes $\underline{d}_{\underline{k}} = (1, 0)^\top$ and $\underline{d}_{\underline{k}} = (0, 1)^\top$ with $\eta = 0$.

So the complete set of real valued eigenfunctions is

$$\left\{ (1, 0)^\top, (0, 1)^\top \right\} \cup \left\{ \underline{\underline{R}} \, \underline{k} \cos(\underline{k} \cdot \underline{x} + \varphi) \right\}_{\underline{k} \in \mathbb{Z}^2 \setminus \{\underline{0}\}} \cup \left\{ \underline{k} \cos(\underline{k} \cdot \underline{x} + \varphi) \right\}_{\underline{k} \in \mathbb{Z}^2 \setminus \{\underline{0}\}}$$

Let us consider the domain $\Omega = (0, \pi)^2$ with homogeneous Dirichlet boundary conditions, i. e. $\underline{w} = \underline{0}$ on $\partial\Omega$. For symmetry reasons, we can use the same Fourier ansatz (2.7). Under the assumption that $\mu / (2\mu + \lambda) \notin \mathbb{Q}$, we know that the eigenfunctions with nonzero eigenvalues are of the form

$$\underline{w}(\underline{x}) = \sum_{|\underline{k}| = k} a_{\underline{k}} \, \underline{k} \, e^{i \underline{k} \cdot \underline{x}} \text{ with } \eta = (2\mu + \lambda) \left| \underline{k} \right|^2, \, \underline{k} \in \mathbb{Z}^2$$

and

$$\underline{w}(\underline{x}) = \sum_{|\underline{k}| = k} a_{\underline{k}} \, \underline{\underline{R}} \, \underline{k} \, e^{i \underline{k} \cdot \underline{x}} \text{ with } \eta = \mu \left| \underline{k} \right|^2, \, \underline{k} \in \mathbb{Z}^2$$

It is however a nontrivial task to find coefficients $a_{\underline{k}}$ such that the boundary conditions are met.

2.1.4 Continuous Variational Formulation

In order to obtain a variational formulation, we multiply equation (2.1) with a test function $\underline{v}$ and integrate over the domain Ω by parts.

To this end, we introduce the following product operators:

$$\begin{aligned}
\underline{\underline{\alpha}} : \underline{\underline{\beta}} &:= \sum_{i=0}^{d-1}\sum_{j=0}^{d-1} \alpha_{ij}\beta_{ij} \\
\left(\underline{\underline{\alpha}}, \underline{\underline{\beta}}\right)_\Omega &:= \int_\Omega \underline{\underline{\alpha}} : \underline{\underline{\beta}} \, \mathrm{d}\underline{x} \\
\left(\underline{\alpha}, \underline{\beta}\right)_\Omega &:= \int_\Omega \underline{\alpha} \cdot \underline{\beta} \, \mathrm{d}\underline{x}
\end{aligned}$$

We will also use the following Green's formula:

$$-\left(\underline{v}, \underline{\nabla} \cdot \underline{\underline{\sigma}}\left(\underline{u}\right)\right)_\Omega = \left(\underline{\underline{\varepsilon}}\left(\underline{v}\right), \underline{\underline{\sigma}}\left(\underline{u}\right)\right)_\Omega - \left(\underline{v}, \underline{\underline{\sigma}}\left(\underline{u}\right)\underline{n}\right)_{\partial\Omega} \tag{2.8}$$

With this Green's formula and assuming $\underline{g}_D = \underline{0}$, the variational formulation reads:

Find $\underline{u} \in C^0\left([0,T], \underline{V}\right) \cap C^1\left([0,T], \underline{H}\right)$ such that

$$\begin{aligned}
\rho\, \partial_t^2(\underline{u}, \underline{v}) + B\left(\underline{u}, \underline{v}\right) &= L\left(\underline{v}\right)\ (t) \quad \forall \underline{v} \in \underline{V},\ t \in [0,T] \\
\partial_t(\underline{u}, \underline{v}) &= (\underline{u}_1, \underline{v}) \quad \forall \underline{v} \in \underline{H},\ t = 0 \\
(\underline{u}, \underline{v}) &= (\underline{u}_0, \underline{v}) \quad \forall \underline{v} \in \underline{H},\ t = 0
\end{aligned}$$

with

$$\begin{aligned}
B\left(\underline{u}, \underline{v}\right) &:= \left(\underline{\underline{\sigma}}\left(\underline{u}\right), \underline{\underline{\varepsilon}}\left(\underline{v}\right)\right)_\Omega \\
L\left(\underline{v}\right)\ (t) &:= \left(\underline{f}\left(t\right), \underline{v}\right)_\Omega + \left(\underline{g}_N\left(t\right), \underline{v}\right)_{\Gamma_N}
\end{aligned}$$

Here, $\underline{V}$ and $\underline{H}$ denote the spaces

$$\begin{aligned}
\underline{V} &= \left\{\underline{u} \in \left(H^1\left(\Omega\right)\right)^d : \ \underline{u}|_{\Gamma_D} = \underline{0}\right\} \\
\underline{H} &= \left(L^2\left(\Omega\right)\right)^d
\end{aligned}$$

where $L^2\left(\Omega\right)$ is the space of functions square integrable over Ω and $H^1\left(\Omega\right)$ is the space of functions in $L^2\left(\Omega\right)$ with gradient in $\left(L^2\left(\Omega\right)\right)^d$.

2.2 Space Discretization

2.2.1 Triangulation

Henceforth, we shall use the notation $I_n := \{0, \ldots, n-1\}$ for index sets. The triangulation $\mathcal{T}$ is a partition of the domain Ω into L triangular (tetrahedral) elements K_l:

$$\mathcal{T} = \left\{K_l\right\}_{l \in I_L}, \quad \bigcup_{l \in I_L} \overline{K}_l = \overline{\Omega}$$

Every element K_l is an affine image of the reference simplex $\hat{K}$:

$$\begin{aligned}
\hat{K} &= \left\{\underline{\hat{x}} \in I\!R^d \ : \ \hat{x}_i > 0, \ \textstyle\sum_i \hat{x}_i < 1, \ i \in I_d\right\} \\
K &= \underline{F}_K\left(\hat{K}\right) \\
\underline{F}_K\left(\underline{\hat{x}}\right) &= \underline{\underline{A}}_K \underline{\hat{x}} + \underline{b}_K, \quad \underline{\underline{A}}_K \in I\!R^{d\times d}, \quad \underline{b}_K \in I\!R^d \quad \forall K \in \mathcal{T}
\end{aligned}$$

We denote by $\Gamma_{ll'}$ the common boundary of elements K_l and $K_{l'}$ for $l, l' \in I_L$. $\{S_{l'}\}_{l'\in\mathcal{L}}$ is the set of boundary edges (faces) with $\mathcal{L} = \{L, L+1, \ldots, L+L'-1\}$. We extend the definition of $\Gamma_{ll'}$ for $l' \in \mathcal{L}$ to be $S_{l'}$ if $S_{l'}$ is part of ∂K_l and set:

$$\underline{n}_{ll'} \ : \ \text{normal on } \Gamma_{ll'}, \text{ points from } K_l \begin{cases} \text{to } K_{l'} & \text{if } (l,l') \in F \\ \text{out of } \Omega & \text{if } (l,l') \in G \end{cases}$$

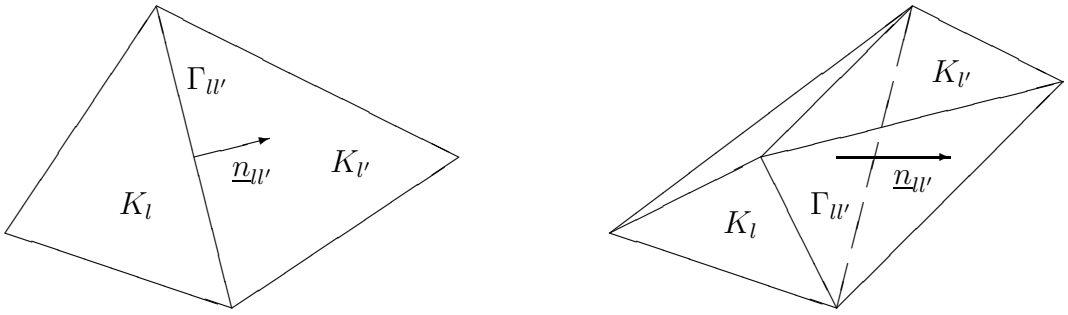

Figure 2.1: Neighbouring elements K_l, $K_{l'}$ in two (left) and three (right) dimensions

We introduce some sets of index pairs (l, l') associated with sets of interfaces $\Gamma_{ll'}$. F corresponds to the interior interfaces, G to the boundary faces, G_D to the Dirichlet boundary and G_N to the Neumann boundary:

$$\begin{aligned}
F &= \{(l,l') : \ l, l' \in I_L, \ K_{l'} \text{ is a neighbour of } K_l, l < l'\} \\
G_D &= \{(l,l') : l \in I_L, \ l' \in \mathcal{L}, S_{l'} \subset (\partial K_l \cap \Gamma_D)\} \\
G_N &= \{(l,l') : l \in I_L, \ l' \in \mathcal{L}, S_{l'} \subset (\partial K_l \cap \Gamma_N)\} \\
G &= G_D \cup G_N \\
F_D &= F \cup G_D
\end{aligned}$$

Then, we need some trace operators:

$$\begin{aligned}\underline{v}|_{\Gamma_{ll'}}(\underline{x}) &= \lim_{\varepsilon\downarrow 0}\underline{v}(\underline{x}-\varepsilon\underline{n}_{ll'})\\ \underline{v}|_{\Gamma_{l'l}}(\underline{x}) &= \lim_{\varepsilon\downarrow 0}\underline{v}(\underline{x}+\varepsilon\underline{n}_{ll'})\end{aligned}$$

Now, we define the average operator $\langle\cdot\rangle$ and the jump operator $[\cdot]$. For interior interfaces, i. e. $(l,l')\in F$, we set:

$$\begin{aligned}\langle\underline{v}\rangle_{\Gamma_{ll'}} &= {}^1\!/_2\left(\underline{v}|_{\Gamma_{ll'}}+\underline{v}|_{\Gamma_{l'l}}\right)\\ [\underline{v}]_{\Gamma_{ll'}} &= \underline{v}|_{\Gamma_{ll'}}-\underline{v}|_{\Gamma_{l'l}}\end{aligned}$$

For boundary faces, i. e. $(l,l')\in G$, we set:

$$\begin{aligned}\langle\underline{v}\rangle_{\Gamma_{ll'}} &= \underline{v}|_{\Gamma_{ll'}}\\ [\underline{v}]_{\Gamma_{ll'}} &= \underline{v}|_{\Gamma_{ll'}}\end{aligned}$$

2.2.2 Finite Element Space

On the triangulation $\mathcal{T}$, we define the space of piecewise linear discontinuous functions $\mathcal{S}_1$:

$$\begin{aligned}\mathcal{S}_1=\mathcal{S}^{1,0}(\Omega,\mathcal{T}) &:= \left\{u\in L^2(\Omega)\ :\ u|_K\in\mathcal{P}_1(K),\ K\in\mathcal{T}\right\}\\ \mathcal{P}_1(K) &:= \left\{u(\underline{x})=\underline{a}\cdot\underline{x}+b,\ \underline{a}\in I\!R^d,\ b\in I\!R\right\}\\ \underline{\mathcal{S}}_1 &:= (\mathcal{S}_1)^d\end{aligned}$$

Note: As $\dim(\mathcal{P}_1(K))=d+1$, we have $\dim(\mathcal{S}_1)=(d+1)L$, where L denotes the number of elements in $\mathcal{T}$.

2.2.3 Discontinuous Variational Formulation

In order to obtain the discontinuous variational formulation, we multiply equation (2.1) with a test function $\underline{v}$, integrate it over each element $K\in\mathcal{T}$ by parts and sum these integrals. We apply a Galerkin discretization by restricting $\underline{u}$ and $\underline{v}$ to be elements of $\underline{\mathcal{S}}_1$. With the Green's formula (2.8), the variational formulation reads:

Find $\underline{u}\in C^2([0,T],\underline{\mathcal{S}}_1)$ such that

$$\begin{aligned}\rho\,\partial_t^2(\underline{u},\underline{v})+B(\underline{u},\underline{v}) &= L(\underline{v})(t) \qquad \forall\underline{v}\in\underline{\mathcal{S}}_1,\ t\in[0,T]\\ \partial_t(\underline{u},\underline{v}) &= (\underline{u}_1,\underline{v}) \qquad \forall\underline{v}\in\underline{\mathcal{S}}_1,\ t=0\\ (\underline{u},\underline{v}) &= (\underline{u}_0,\underline{v}) \qquad \forall\underline{v}\in\underline{\mathcal{S}}_1,\ t=0\end{aligned}$$

with:

$$
\begin{aligned}
B\left(\underline{u},\underline{v}\right) &:= \sum_{l\in I_L}\left(\underline{\underline{\sigma}}\left(\underline{u}\right),\underline{\underline{\varepsilon}}\left(\underline{v}\right)\right)_{K_l} \\
&\quad -\sum_{(l,l')\in F_D}\left(\left\langle\underline{\underline{\sigma}}\left(\underline{u}\right)\cdot\underline{n}_{ll'}\right\rangle,\left[\underline{v}\right]\right)_{\Gamma_{ll'}}-\tau\left(\left[\underline{u}\right],\left\langle\underline{\underline{\sigma}}\left(\underline{v}\right)\cdot\underline{n}_{ll'}\right\rangle\right)_{\Gamma_{ll'}} \\
&\quad +\mu\sum_{(l,l')\in F_D}\frac{\gamma_\mu}{h_{ll'}}\left(\left[\underline{u}\right],\left[\underline{v}\right]\right)_{\Gamma_{ll'}}+\lambda\sum_{(l,l')\in F_D}\frac{\gamma_\lambda}{h_{ll'}}\left(\left[\underline{u}\cdot\underline{n}_{ll'}\right],\left[\underline{v}\cdot\underline{n}_{ll'}\right]\right)_{\Gamma_{ll'}} \\
L\left(\underline{v}\right)\left(t\right) &:= \sum_{l\in I_L}\left(\underline{f}\left(t\right),\underline{v}\right)_{K_l} \\
&\quad +\sum_{(l,l')\in G_N}\left(\underline{g}_N\left(t\right),\underline{v}\right)_{\Gamma_{ll'}}+\ \tau\sum_{(l,l')\in G_D}\left(\underline{g}_D\left(t\right),\underline{\underline{\sigma}}\left(\underline{v}\right)\cdot\underline{n}_{ll'}\right)_{\Gamma_{ll'}} \\
&\quad +\mu\sum_{(l,l')\in G_D}\frac{\gamma_\mu}{h_{ll'}}\left(\underline{g}_D\left(t\right),\underline{v}\right)_{\Gamma_{ll'}}+\lambda\sum_{(l,l')\in F_D}\frac{\gamma_\lambda}{h_{ll'}}\left(\underline{g}_D\cdot\underline{n}_{ll'},\underline{v}\cdot\underline{n}_{ll'}\right)_{\Gamma_{ll'}}
\end{aligned}
$$

where

$$
\begin{aligned}
&\tau\in\{-1,1\}: \quad \text{symmetry parameter} \\
&\gamma_{\mu,\lambda}: \quad \text{stabilization coefficients} \\
&h_{ll'}: \quad \text{suitable measure of mesh size on } \Gamma_{ll'}
\end{aligned}
$$

Note that we have added some terms consistently. The terms with coefficent τ make the bilinearform symmetric for $\tau=-1$. The terms with γ_μ and γ_λ are stabilization or penalty terms.

For $\tau=1$ and $d=2$, Wihler proves in [10] coercivity of $B\left(\underline{u},\underline{v}\right)$ and absence of volume locking in the static problem, using the following parameters:

$$
\begin{aligned}
&\gamma_\mu=1 \\
&\gamma_\lambda=0 \\
&h_{ll'}=\operatorname{diam}\left(\Gamma_{ll'}\right)
\end{aligned}
$$

For $\tau=-1$ and $d=2$, Hansbo and Larson prove in [6] coercivity of $B\left(\underline{u},\underline{v}\right)$ and absence of volume locking in the static problem, using the following parameters:

$$
\begin{aligned}
&\gamma_\mu\geq 2 \\
&\gamma_\lambda\geq 2 \\
&h_{ll'}=\begin{cases} 2\left(1/\left|K_l\right|+1/\left|K_{l'}\right|\right)^{-1}/\operatorname{diam}\left(\Gamma_{ll'}\right) & (l,l')\in F \\ \left|K_l\right|/\operatorname{diam}\left(\Gamma_{ll'}\right) & (l,l')\in G \end{cases}
\end{aligned}
$$

We will use the later definition of $h_{ll'}$ in both cases, as the choice in the nonsymmetric case does not matter a lot.

2.3 DGFEM Implementation

This section gives a broad overview of the DGFEM implementation. A more detailed presentation is provided in appendix A.

2.3.1 Vectorial FE Basis

We choose a suitable basis for $\underline{\mathcal{S}}_1$:

$$\underline{\mathcal{S}}_1 = \operatorname{span}\left\{\underline{\varphi}_n\right\}_{n\in I_N}, \quad N = d\,(d+1)\,L$$

where L is the number of elements in $\mathcal{T}$ and N is the number of basis functions, i. e. the number of degrees of freedom or the dimension of $\underline{\mathcal{S}}_1$.

We write the trial and test functions $\underline{u}$ and $\underline{v}$ as linear combinations of basis functions:

$$\begin{aligned} \underline{u} &= \sum_{n\in I_N} U_n \underline{\varphi}_n, \quad \underline{v} = \sum_{n\in I_N} V_n \underline{\varphi}_n \\ \underline{U} &= \{U_n\}_{n\in I_N}, \quad \underline{V} = \{V_n\}_{n\in I_N} \end{aligned}$$

2.3.2 Element Computations

With this setup, we now carry out the element computations.

Mass Matrix

$$\begin{aligned} (\underline{u}, \underline{v}) &= \underline{V}^\top \underline{\underline{M}}\, \underline{U} \\ \left(\underline{\underline{M}}\right)_{nn'} &= \left(\underline{\varphi}'_n, \underline{\varphi}_n\right) \end{aligned}$$

Note: As no continuity has to be enforced, the basis functions can be chosen to have support on only one element. Those having support on the same element can be chosen L^2-orthogonal, which causes the consistent mass matrix to be diagonal without applying special mass lumping techniques.

Stiffness Matrix

$$\begin{aligned} B\,(\underline{u}, \underline{v}) &= \underline{V}^\top \underline{\underline{B}}\, \underline{U} \\ \left(\underline{\underline{B}}\right)_{nn'} &= B\left(\underline{\varphi}_{n'}, \underline{\varphi}_n\right) \end{aligned}$$

Load Vector

$$\begin{aligned} L\,(\underline{v})\,(t) &= \underline{V}^\top \underline{L}\,(t) \\ \left(\underline{L}\,(t)\right)_n &= L\left(\underline{\varphi}_n\right)(t) \end{aligned}$$

Initial Conditions

$$\begin{aligned}(\underline{u}_i, \underline{v}) &= \underline{V}^\top \underline{U}_i, \quad i \in \{0,1\} \\ (\underline{U}_i)_n &= \left(\underline{u}_i, \underline{\varphi}_n\right)\end{aligned}$$

2.3.3 System of Ordinary Differential Equations

By inserting the expressions found in the element computation section into the variational formulation, we get a system of ordinary differential equations:
Find $\underline{U} \in C^2\left([0,T], I\!R^N\right)$ such that $\forall\ \underline{V} \in I\!R^N$

$$\begin{aligned}\rho\, \underline{V}^\top \underline{\underline{M}}\, \ddot{\underline{U}}(t) + \underline{V}^\top \underline{\underline{B}}\, \underline{U}(t) &= \underline{V}^\top \underline{L}(t) \\ \underline{V}^\top \underline{\underline{M}}\, \dot{\underline{U}}^\top(0) &= \underline{V}^\top \underline{U}_1 \\ \underline{V}^\top \underline{\underline{M}}\, \underline{U}^\top(0) &= \underline{V}^\top \underline{U}_0\end{aligned}$$

As this has to hold $\forall\ \underline{V} \in I\!R^N$, we can drop $\underline{V}^\top$ and obtain:
Find $\underline{U} \in C^2\left([0,T], I\!R^N\right)$ such that

$$\rho\, \underline{\underline{M}}\, \ddot{\underline{U}}(t) + \underline{\underline{B}}\, \underline{U}(t) = \underline{L}(t) \tag{2.9}$$

$$\underline{\underline{M}}\, \dot{\underline{U}}(0) = \underline{U}_1 \tag{2.10}$$

$$\underline{\underline{M}}\, \underline{U}(0) = \underline{U}_0 \tag{2.11}$$

2.4 Time Discretization

2.4.1 Requirements

The system of ordinary differential equations (2.9)-(2.11), which is linear and of second order, can be discretized in time using a suitable timestepper for equations of the type

$$\underline{\underline{M}}\, \ddot{\underline{y}}(t) + \underline{\underline{B}}\, \underline{y}(t) = \underline{l}(t) \tag{2.12}$$

$$\dot{\underline{y}}(0) = \underline{z}_0 \tag{2.13}$$

$$\underline{y}(0) = \underline{y}_0 \tag{2.14}$$

In our case, $\underline{\underline{B}}$ is real valued, but not necessarily symmetric. Thus it may have complex (conjugate) eigenvalues. As for $\mu \ll \lambda$ the spectrum of $\underline{\underline{B}}$ gets large, the chosen timestepper is preferably absolutely stable.

2.4.2 Stability of the Semidiscrete Problem

The related eigenproblem to (2.12), which can also be seen as a discretization of (2.6), reads:

$$\underline{\underline{B}}\,\underline{W}_k \;=\; \eta_k \underline{\underline{M}}\,\underline{W}_k, \quad k \in I_N$$

Assuming that $\underline{\underline{B}}$ has a full eigenvector basis, we write the state vector $\underline{y}$ as a linear combination of eigenvectors and substitute into (2.12), with $\underline{l}\,(t) = 0$:

$$\begin{aligned} \underline{y}\,(t) &= \sum_{k \in I_N} c_k\,(t)\,\underline{W}_k \\ \sum_{k \in I_N} \ddot{c}_k \underline{\underline{M}}\,\underline{W}_k &= -\sum_{k \in I_N} c_k \underline{\underline{B}}\,\underline{W}_k \end{aligned}$$

The following scalar equation holds for the eigencoefficents:

$$\begin{aligned} \ddot{c}_k\,(t) &= -\eta_k c_k\,(t)\,, \quad \eta_k \in \mathbb{C} \\ \dot{c}_k\,(0) &= c_{k,1} \\ c_k\,(0) &= c_{k,0} \end{aligned}$$

The equation has the solution

$$c_k\,(t) \;=\; c_{k,0} \cos\left(\omega_k t\right) + \frac{c_{k,1}}{\omega_k} \sin\left(\omega_k t\right), \quad \omega_k = \sqrt{\eta_k}$$

Note: $c_k\,(t)$ is bounded if and only if $\omega_k \in I\!R$, i. e. $\eta_k > 0$. This implies that the semidiscrete problem has a bounded solution if all eigenvalues of $\underline{\underline{B}}$ are real and positive.

2.4.3 Timestepping Schemes

In the following paragraphs, we will evaluate some possible timesteppers. We will denote by $\underline{y}_n$, $\underline{z}_n$ and $\underline{l}_n$ approximations to $\underline{y}\,(n\Delta t)$, $\underline{\dot{y}}\,(n\Delta t)$ and $\underline{l}\,(n\Delta t)$, respectively.

Newmark The Newmark scheme is the standard choice in conforming FEM. The Newmark scheme for problem (2.12) – (2.14) reads:

$$\begin{aligned} \left(\underline{\underline{M}} + \Delta t^2 \beta \underline{\underline{B}}\right) \underline{y}_{n+1} &= \left(\underline{\underline{M}} - \Delta t^2 \left(1/2 - \beta\right) \underline{\underline{B}}\right) \underline{y}_n \\ &\quad + \Delta t \underline{\underline{M}}\,\underline{z}_n + \beta\,\underline{l}_{n+1} + \left(1/2 - \beta\right)\,\underline{l}_n \end{aligned}$$

$$\underline{\underline{M}}\,\underline{z}_{n+1} \;=\; \underline{\underline{M}}\,\underline{z}_n - \Delta t \left[\gamma \left(\underline{\underline{B}}\,\underline{y}_{n+1} + \underline{l}_{n+1}\right) + (1-\gamma)\left(\underline{\underline{B}}\,\underline{y}_n + \underline{l}_n\right)\right]$$

As shown in [8], it is of second order accuracy and unconditionally stable for the real valued model problem if e. g. $\beta = 1/4$ and $\gamma = 1/2$. The scheme resulting from these parameter values is also called trapezoidal scheme.

Numerical dissipation may be introduced by setting $\gamma > 1/2$. In this case, the order of accuracy drops to 1 and substantial undesired damping of low frequency modes occurs.

α-Method The α-method, as described in [7], was introduced by Hilber, Hughes and Taylor as a modification to the Newmark scheme. It allows numerical damping without losing second order accuracy. Also, the damping mainly affects high frequency modes.

The α-method for problem (2.12) – (2.14) can be formulated as follows:

$$\begin{aligned}
\left(\underline{\underline{M}} + \Delta t^2 (1-\alpha)\, \beta \underline{\underline{B}}\right) \underline{k}_{n+1} &= \underline{\underline{B}} \left(\underline{y}_n + (1+\alpha) \left(\Delta t \underline{z}_n + \Delta t^2 \left({}^1\!/_2 - \beta \right) \underline{k}_n \right) \right) \\
&\quad + \underline{l}\left((n+1+\alpha)\, \Delta t \right) \\
\underline{y}_{n+1} &= \underline{y}_n + \Delta t \underline{z}_n + \Delta t^2 \left({}^1\!/_2 - \beta \right) \underline{k}_n + \Delta t^2 \beta \underline{k}_{n+1} \\
\underline{z}_{n+1} &= \underline{z}_n + \Delta t \left(1-\gamma\right) \underline{k}_n + \Delta t \gamma \underline{k}_{n+1}
\end{aligned}$$

with $\beta = (1-\alpha)^2/4$ and $\gamma = {}^1\!/_2 - \alpha$. For $\alpha = 0$, the scheme reduces to the respective Newmark method, i. e. the trapezoidal scheme. For $\alpha \in [-1/3, 0)$, numerical dissipation is introduced. As shown in [7], it is of second order accuracy and unconditionally stable for the real valued model problem for all $\alpha \in [-1/3, 0]$.

Nyström Methods Nyström methods are direct applications of Runge's method to second order differential equations. Examples of fourth and fifth order accuracy can be found in [5], although there is no statement about stability. As these methods are explicit and of high order, disadvantages in stability have to be expected. Symplectic explicit Nyström methods of second and fourth order accuracy can be found in [5] as well.

The symplectic Nyström method of second order reads:

$$\begin{aligned}
\underline{\underline{M}}\ \underline{k}_n &= -\underline{\underline{B}} \left(\underline{y}_n + {}^1\!/_2 \Delta t \underline{z}_n \right) + \underline{l}\left(\left(n + {}^1\!/_2 \right) \Delta t \right) \\
\underline{y}_{n+1} &= \underline{y}_n + \Delta t \underline{z}_n + {}^1\!/_2 \Delta t^2 \underline{k}_n \\
\underline{z}_{n+1} &= \underline{z}_n + \Delta t \underline{k}_n
\end{aligned}$$

For linear operators and without right hand side, this method is equivalent to the well known leap frog scheme! Therefore, a stability analysis for the real valued model problem

$$\ddot{y} = -\omega^2 y, \quad \omega > 0$$

would show that this method is conditionally stable for

$$\Delta t < 2/\omega$$

like the leap frog scheme.

3 Implementation

In this chapter, some of the implementation details are presented. The code written for this thesis is part of the *Concepts* library, which is presented in [3]. The main idea behind the library is 'design by concept'. This means that every mathematical concept like a finite element, a FE space or a bilinear form is represented by a class.

3.1 Class Design

3.1.1 DGFEM

In order to perform DGFEM, the concepts of interfaces and index pairs introduced in chapter 2 need to be cast into classes.

The class `EdgeInfo` represents an interface $\Gamma_{ll'}$. The class is able to return the neighbour cells K_l and K_l', the normal $\underline{n}_{ll'}$, the edge length $|\Gamma_{ll'}|$ and more properties associated with the respective edge.

The class `MeshInfo` represents the set of all edges of a triangulation $\mathcal{T}$. The class `MeshInfo` builds all `EdgeInfo` objects efficiently and allows to scan over all of them, as well as direct access to the `EdgeInfo` object associated to a specific edge.

The index sets F, G and the like are represented by the class `ElementPairList`. It consists of objects of type `ElementPair`, each of which having a reference to both elements involved and the `EdgeInfo` object of the underlying edge. The `ElementPairList` can easily be built up by scanning over all `EdgeInfo` objects in the `MeshInfo` object.

3.1.2 Vector Valued DGFEM

In order to simplify solving problems where the unknown is a vector valued function, like in our case, a set of classes exists in *Concepts*. If for example the scalar space $\mathcal{S}_1$ is represented by the new class `SpaceP1`, then the vectorial space $(\mathcal{S}_1)^d$ can be represented by the existing class `vectorial::Space`, representing the action of $(\cdot)^d$ on any space.

This idea has been extended during this work to be applicable to the concepts of DGFEM, introducing namely a vectorial element pair `vectorial::ElementPair` and the class `ElementPairListIdentic`. This later class allows for a simple construction of the list of `vectorial::ElementPair`s, when the list of scalar `ElementPair`s is already constructed and the spaces in each vectorial dimension are identical. This would not be

the case, if we would use e. g. the space $\mathcal{S}_1 \times \mathcal{S}_2$ for the displacements, where $\mathcal{S}_2$ would have to be defined.

3.2 Mesh Generation

3.2.1 General Nonperiodic Domains

For the generation of meshes of general domains, we use *Netgen* [9]. The data from *Netgen* is written to three files using a minimal plugin to *Netgen*. The first file contains a numbered list of nodes with their coordinates. The second file contains a list of triangles, defined by three node numbers. The order of the nodes is strictly counterclockwise throughout the whole file. The third file contains a list of boundary edges, defined by two node numbers and its attribute, i. e. the boundary condition. The data in these three files can be imported to *Concepts* using existing mesh import classes.

3.2.2 Periodic Unit Square

For the periodic unit square, we use our own class representing geometrically the unit square $(0, 1)^2$, while having the topology of a torus. Like this, the rest of the code does not require any modifications.

3.3 Shape Functions

3.3.1 Orthogonal Basis

The basis functions are chosen to be orthogonal in the L^2 scalar product used for the mass matrix, causing this matrix to become diagonal. This sounds more difficult than it actually is.

First of all, as we have discontinuous functions, we can choose a basis where each function has support exactly on one element. Like this, all pairs of basis functions who do not have the same element as support, have a trivially vanishing L^2 scalar product and are therefore orthogonal.

Then, we use the same scalar basis in each vectorial dimension using a tensor product. So basis functions not associated with the same vectorial dimension are orthogonal because the respective unit vectors are orthogonal.

Finally, we have an affine mapping from each element to the reference element. If the L^2 scalar product as an integral is mapped, the determinant of the Jacobian to be introduced is constant. So functions orthogonal on the reference element affinely mapped to any other element remain orthogonal. Thus any scalar basis orthogonal on the reference triangle can be used to construct a globally orthogonal vectorial basis!

We use the three functions

$$\begin{aligned} N_1 &= 1 - 2x \\ N_2 &= -1 + 2\,(x - y) \\ N_3 &= -1 + 2y \end{aligned}$$

which are L^2-orthogonal on the reference triangle

$$\hat{K} = \{(x, y) : 0 < y < x < 1\}$$

i. e.

$$(N_i, N_j)_{\hat{K}} = {}^1\!/_6\ \delta_{ij}$$

as such a basis.

3.3.2 Quadrature

All integrals in the bilinear forms and the linear forms, except the one for the mass matrix, are calculated using quadrature. The integral on triangles are transformed to a square where a tensor product Gauss quadrature is applied. This procedure seems unfavorable in terms of order of polynomials exactly integrated. A closer look shows that this is not a problem: The original coordinates $\underline{\xi}$ can be expressed as a linear function of the transformed coordinates $\underline{\eta}$. If the function to integrate is a polynom of order p in $\underline{\xi}$, then the transformed function is a polynomial of order $p + 1$ in $\underline{\eta}$. The determinant of the Jacobian introduced is also linear in $\underline{\eta}$. So the function finally integrated on the square is a polynomial of order $p + 2$ in $\underline{\eta}$, which can be integrated exactly using a suitable Gauss quadrature rule. All basis functions are polynomials, so all matrices can be calculated exactly using this quadrature.

4 Numerical Results

All numerical tests are carried out for $d = 2$, as the case $d = 3$ does not change the problem qualitatively. Also, we set $\mu = 1$ and $\rho = 1$ in all tests, as we want to study the behaviour with λ.

4.1 Static Tests

4.1.1 Setup

In order to test the spacial discretization separately, we consider the static problem (2.4):

$$\begin{aligned} -\underline{\nabla} \cdot \underline{\underline{\sigma}}(\underline{u}) &= \underline{f} \quad \text{in } \Omega \\ \underline{u} &= \underline{g}_D \quad \text{on } \Gamma_D \\ \underline{\underline{\sigma}}(\underline{u}) \cdot \underline{n} &= \underline{g}_N \quad \text{on } \Gamma_N \end{aligned}$$

Using the same discretization techniques as for the dynamic problem yields

$$\underline{\underline{B}}\,\underline{U} = \underline{L}$$

where $\underline{\underline{B}}$, $\underline{U}$ and $\underline{L}$ are defined as in section 2.3.2, except that neither $\underline{U}$ nor $\underline{L}$ depend on time. This linear system can be solved with a suitable solver.

4.1.2 Regular Problem

We first consider the following regular problem:

$$\begin{aligned} \Omega &= (0,1)^2 \\ \Gamma_D &= \partial\Omega \\ \underline{u}^*(\underline{x}) &= \begin{pmatrix} -\sin(\pi x)^2 \sin(2\pi y) \\ \sin(2\pi x)\sin(\pi y)^2 \end{pmatrix} \\ \underline{g}_D &= \underline{u}^*|_{\Gamma_D} \\ \underline{f} &= -\underline{\nabla} \cdot \underline{\underline{\sigma}}(\underline{u}^*) = \mu \begin{pmatrix} 2\pi^2 \sin(2\pi y)\left(1 - 4\sin(\pi x)^2\right) \\ -2\pi^2 \sin(2\pi x)\left(1 - 4\sin(\pi y)^2\right) \end{pmatrix} \end{aligned}$$

with the exact solution $\underline{u}^*$. We assess the performance of the method by calculating the L^2 norm of the error:

$$e(h) = \|\underline{u}_h - \underline{u}^*\|_{L^2(\Omega)}$$

where $\underline{u}_h$ is the finite element solution. In the optimal case, we can expect $e(h) = \mathcal{O}(h^2)$.

The coarsest triangulation of the computational domain is shown in figure 4.1. This triangulation is refined uniformly for the convergence study, i. e. each triangle is split into 4 similar triangles recursively.

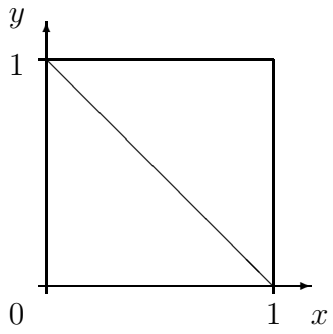

Figure 4.1: Computational domain and coarsest mesh for the regular problem

We test two cases, the symmetric ($\tau = -1$) and the nonsymmetric ($\tau = 1$) case.

symmetric

In the symmetric case, we set $\gamma_\mu = \gamma_\lambda = 3$. So we are on the safe side, as Hansbo and Larson prove in [6] that values ≥ 2 are sufficient for robust convergence, i. e. convergence independent of λ. As can be seen in figure 4.2, the error $e(h)$ is $\mathcal{O}(h^2)$, i. e. the convergence is robust and optimal.

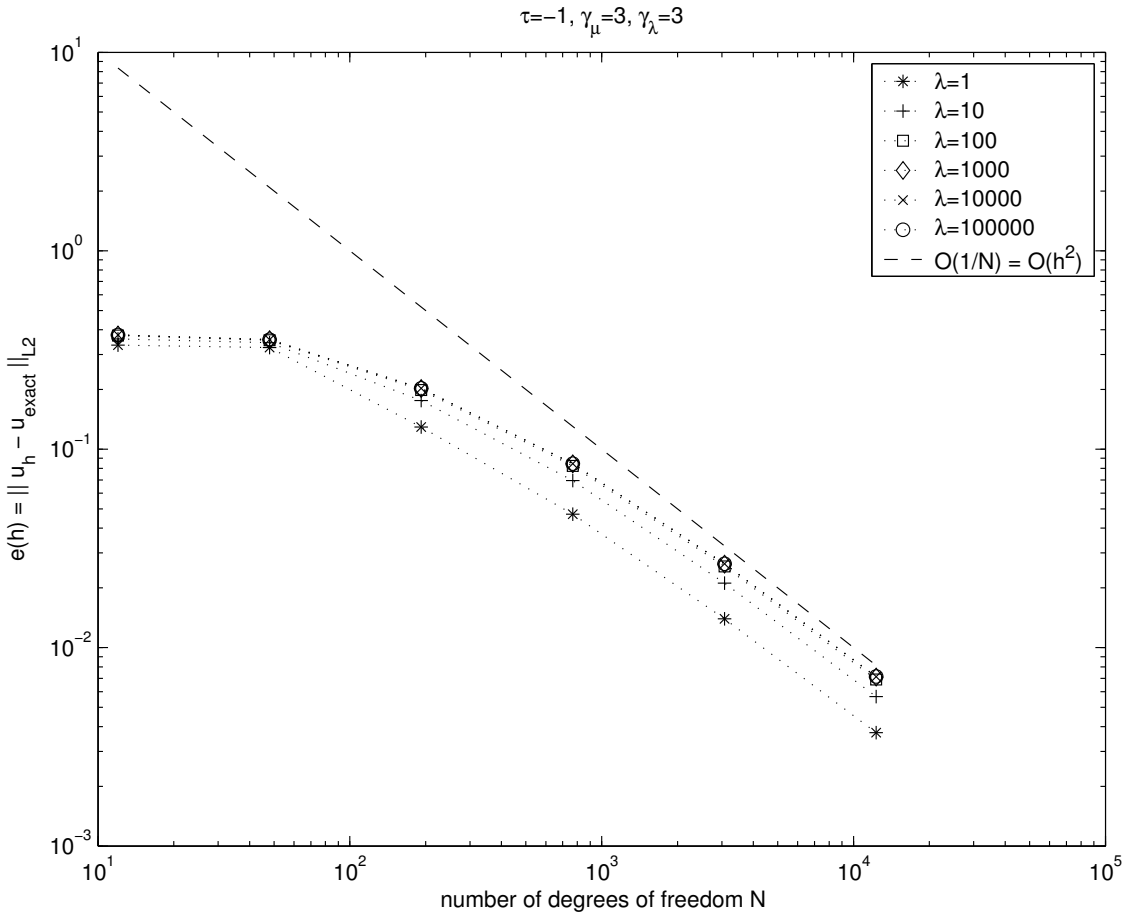

Figure 4.2: Mesh convergence for regular static problem, symmetric case.

nonsymmetric

In the nonsymmetric case, we set $\gamma_\mu = 1$, and $\gamma_\lambda = 0$. This is proven to be sufficient for robust convergence in [10], i. e. we can expect convergence independent of λ. As can be seen in figure 4.3, the error $e(h)$ is $\mathcal{O}(h^2)$, i. e. the convergence is robust and optimal.

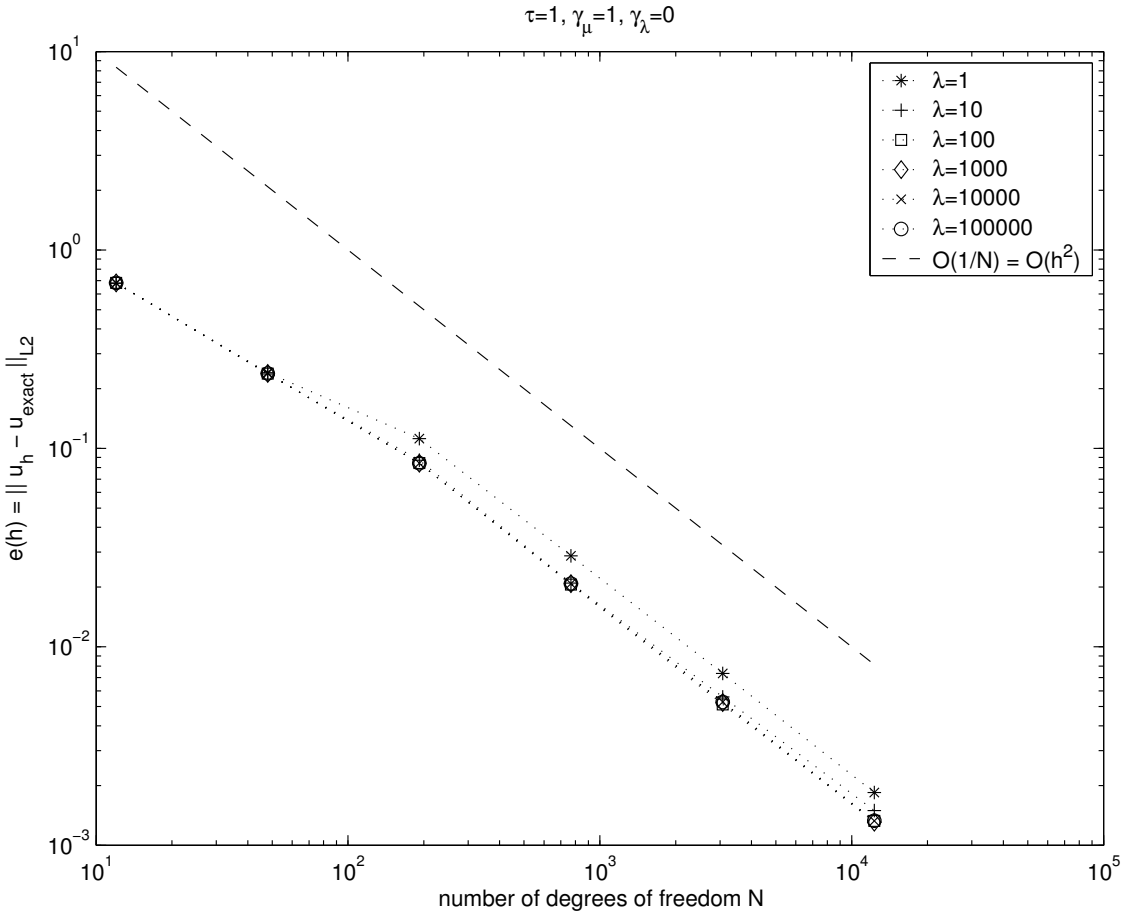

Figure 4.3: Mesh convergence for regular static problem, nonsymmetric case.

4.1.3 Singular Problem

We now consider a problem from [10], which has a singular solution. The computational domain and its coarsest triangulation is shown in figure 4.4. This triangulation is refined uniformly for the convergence study, i. e. each triangle is split into four similar triangles recursively.

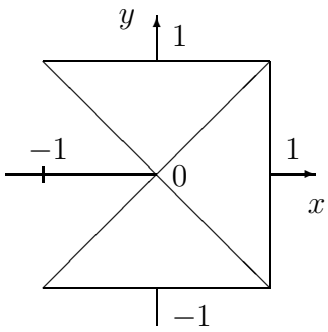

Figure 4.4: Computational domain and coarsest mesh for the singular problem

The exact solution is given in polar coordinates:

$$\begin{aligned}
\underline{u}\,(r,\varphi) &= \begin{pmatrix} \cos\varphi\; u_r\,(r,\varphi) - \sin\varphi\; u_\varphi\,(r,\varphi) \\ \sin\varphi\; u_r\,(r,\varphi) + \cos\varphi\; u_\varphi\,(r,\varphi) \end{pmatrix} \\
u_r\,(r,\varphi) &= \frac{1}{2\mu} r^\alpha \left(-\left(\alpha+1\right)\cos\left(\left(\alpha+1\right)\varphi\right) + \left(C_2-\alpha-1\right) C_1 \cos\left(\left(\alpha-1\right)\varphi\right)\right) \\
u_\varphi\,(r,\varphi) &= \frac{1}{2\mu} r^\alpha \left(\left(\alpha+1\right)\sin\left(\left(\alpha+1\right)\varphi\right) + \left(C_2+\alpha-1\right) C_1 \sin\left(\left(\alpha-1\right)\varphi\right)\right)
\end{aligned}$$

where $\alpha \approx 0.544484$ is the solution of the equation

$$\alpha \sin\left(2\omega\right) + \sin\left(2\omega\alpha\right) = 0$$

with $\omega = 3\pi/4$ and

$$C_1 = -\frac{\cos\left(\left(\alpha+1\right)\omega\right)}{\cos\left(\left(\alpha-1\right)\omega\right)}, \qquad C_2 = \frac{2\left(\lambda+2\mu\right)}{\lambda+\mu}$$

The other data are given as

$$\begin{aligned}
\Gamma_D &= \partial\Omega \\
\underline{g}_D &= \underline{u}^*|_{\Gamma_D} \\
\underline{f} &= 0
\end{aligned}$$

We assess the performance of the method by calculating the L^2 norm of the error:

$$e\,(h) = \|\underline{u}_h - \underline{u}^*\|_{L^2(\Omega)}$$

where $\underline{u}_h$ is the finite element solution. In the optimal case, we could expect $e(h) = \mathcal{O}(h^2)$. As the solution is singular and we use a uniformly refined mesh, a suboptimal convergence will be observed. However it should still be robust.

We again test the symmetric ($\tau = -1$) and the nonsymmetric($\tau = 1$) case. In the symmetric case, we set $\gamma_\mu = \gamma_\lambda = 2$. In the nonsymmetric case, we set $\gamma_\mu = 1$, and $\gamma_\lambda = 0$. In both cases, the error $e(h)$ is $\approx \mathcal{O}(h^{1.4})$, i. e. the convergence is robust but suboptimal. This can be seen in figure 4.5 for the symmetric case, and in figure 4.6 for the nonsymmetric case.

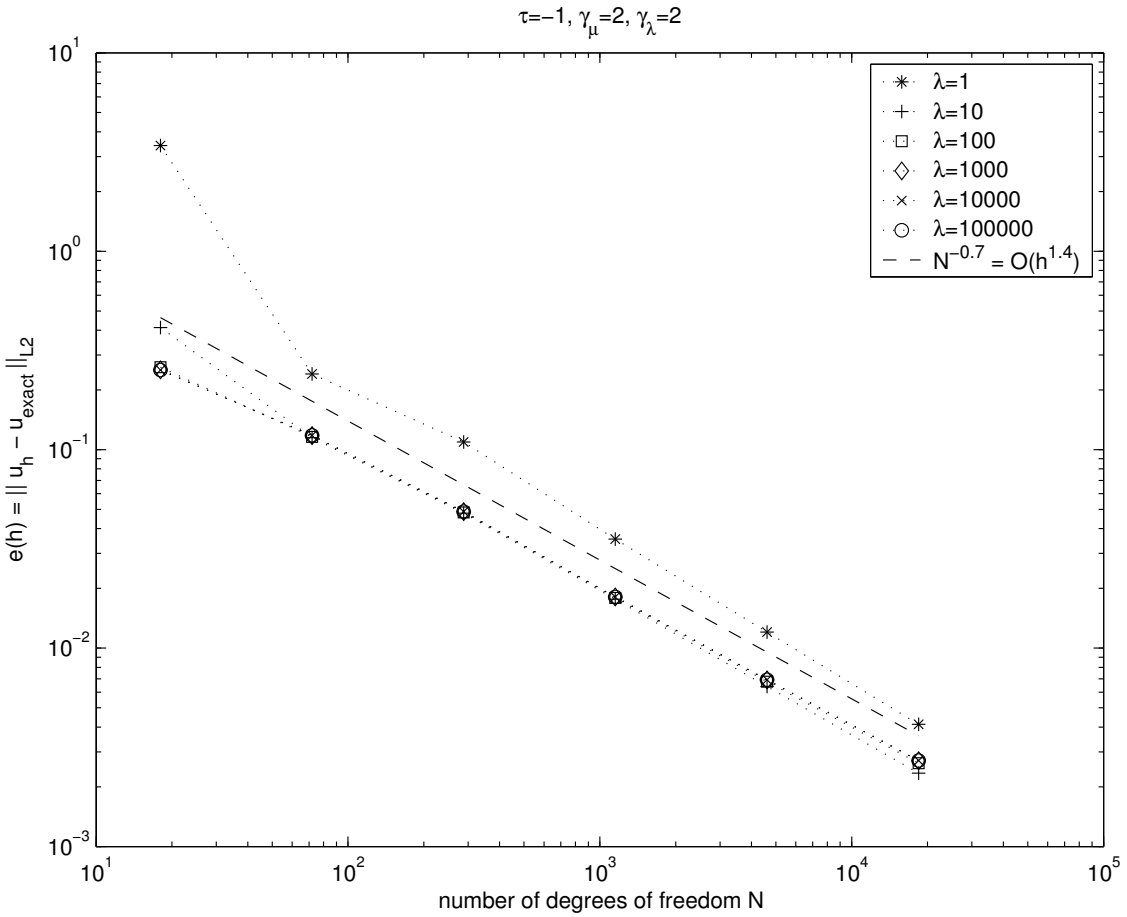

Figure 4.5: Mesh convergence for singular static problem, symmetric case.

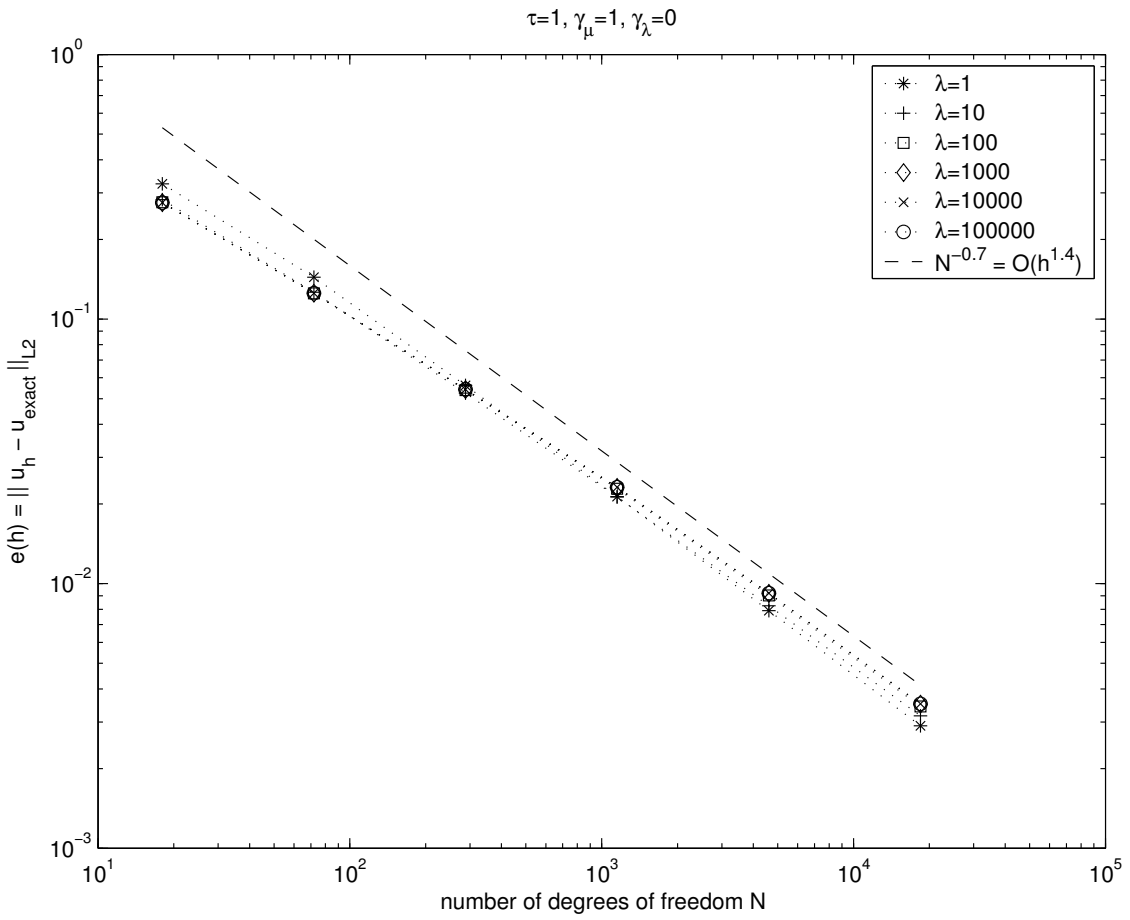

Figure 4.6: Mesh convergence for singular static problem, nonsymmetric case.

4.2 Eigenvalues and Eigenfunctions

The eigenproblem (2.6) can be discretized using the same discretization techniques as for the dynamic problem. We end up with the matrix eigenvalue problem

$$\underline{\underline{B}}\,\underline{W} \;=\; \eta\,\underline{\underline{M}}\,\underline{W}$$

where $\underline{\underline{B}}$ and $\underline{\underline{M}}$ are defined as in section 2.3.2, $\underline{W}$ denotes the FE coefficient vector of the eigenfunction $\underline{w}\left(\underline{x}\right)$ and η is the according eigenvalue.

4.2.1 Properties in Symmetric Case

With $\tau = -1$, the matrix $\underline{\underline{B}}$ is symmetric and with $\gamma_{\mu,\lambda} \geq 2$, the underlying bilinear form is coercive and the matrix is therefore positive definite. Together, it follows that the matrix $\underline{\underline{B}}$ has real positive eigenvalues. As shown in section 2.4.2, this is a sufficient condition for stability of the semidiscrete problem.

The behaviour of the generalized eigenvalues when λ increases and h decreases can also be seen from theory developed in 2.1.3. The zero eigenvalues will not occur if we have nonperiodic boundary conditions. The smallest nonzero eigenvalue η_{min} will therefore satisfy $1/\eta_{min} = \mathcal{O}\left(1/\mu\right)$, while the largest will be $\mathcal{O}\left(\left(2\mu+\lambda\right)\left|\underline{k}\right|^2\right)$. The $\underline{k}$ to be considered is the largest spacial frequency that can be resolved and is therefore $\mathcal{O}\left(h^{-1}\right)$, so the largest eigenvalue will be $\mathcal{O}\left(h^{-2}\lambda\right)$, and the condition number $\kappa\left(\underline{\underline{M}}^{-1}\underline{\underline{B}}\right)$ will be $\mathcal{O}\left(h^{-2}\lambda/\mu\right)$.

This behavior was observed in a numerical experiment on $\Omega = \left(0,1\right)^2$ with $\Gamma_D = \partial\Omega$ and $\underline{g}_D = \underline{0}$.

4.2.2 Properties in Nonsymmetric Case

With $\tau = 1$, the matrix $\underline{\underline{B}}$ is not symmetric. Although we know that the underlying bilinear form is coercive for $\gamma_\mu = 1$ and $\gamma_\lambda = 0$, this result cannot be used for a statement about the eigenvalues. They still can be complex conjugate.

Numerical experiments show that complex conjugate eigenvalues really occur. As shown in section 2.4.2, the semidiscrete problem can then be unstable.

4.3 Dynamic Tests

4.3.1 Efficiency Considerations

Let's do a rough a-priory check of the computational costs of explicit and implicit timestepping. In order to stay general in this considerations, we denote the spacial dimensionality by d, the order of accuracy of the timestepper by q and assume that the

final time T is fixed. We will use that

$$\begin{aligned} \eta_{max}\left(\underline{\underline{M}}^{-1}\underline{\underline{B}}\right) &= \mathcal{O}\left(h^{-2}\lambda\right) \text{ and} \\ N\ h^d &= \mathcal{O}(1) \end{aligned}$$

Explicit Timestepping

With an explicit timestepper, there is a stability criterion of the form

$$\Delta t \ < \ C/\sqrt{\eta_{max}\left(\underline{\underline{M}}^{-1}\underline{\underline{B}}\right)} = \mathcal{O}\left(h/\sqrt{\lambda}\right)$$

The computational cost per timestep Z_1 is $\mathcal{O}(N) = \mathcal{O}\left(h^{-d}\right)$. We need to solve a linear system with the mass matrix, but since it is diagonal, the cost can easily be kept $\mathcal{O}(N)$. So the total cost Z is

$$Z \ = \ \mathcal{O}\left(Z_1/\Delta t\right) = \mathcal{O}\left(h^{-(d+1)}\sqrt{\lambda}\right)$$

Implicit Timestepping

With an implicit timestepper, the timestep can be arbitrarily large from the point of view of stability. From the point of view of the overall error, which is $\mathcal{O}(\Delta t^q + h)$, we must require

$$\Delta t \ = \ \mathcal{O}\left(h^{1/q}\right)$$

in order the temporal discretisation error not to dominate the spacial one. Note: We observe a spacial L^2 error which is $\mathcal{O}(h^2)$, but as the results of [10] and [6] only guarantee the error in the energy norm to be $\mathcal{O}(h)$, we stick to this result for these efficiency considerations.

In every timestep, we have to solve a linear system with a matrix of the form $\underline{\underline{M}} + \alpha\Delta t^2\underline{\underline{B}}$, where α is a constant depending only on the chosen timestepping scheme. So for big systems and specially for $d = 3$, we will require an iterative solver, as a direct solver would need too much memory. The computational cost per timestep Z_1 is therefore

$$Z_1 \ = \ Z_{iter}\ N_{iter}$$

where Z_{iter} is the cost per solver iteration, and N_{iter} is the number of iterations. Typical iterative solvers have

$$\begin{aligned} Z_{iter} &= \mathcal{O}(N) = \mathcal{O}\left(h^{-d}\right) \\ N_{iter} &= \mathcal{O}\left(\sqrt{\kappa\left(\underline{\underline{M}} + \alpha\Delta t^2\underline{\underline{B}}\right)}\right) \end{aligned}$$

The spectral properties of $\underline{\underline{M}}^{-1}\underline{\underline{B}}$ are known, those of $\underline{\underline{M}}$ are:

$$\begin{aligned} 1/\eta_{min}\left(\underline{\underline{M}}\right) &= \mathcal{O}\left(h^{-2}\right) \\ \eta_{max}\left(\underline{\underline{M}}\right) &= \mathcal{O}\left(h^2\right) \end{aligned}$$

Thus, we have

$$\begin{aligned} 1/\eta_{min}\left(\underline{\underline{M}}+\alpha\Delta t^2\underline{\underline{B}}\right) &= \mathcal{O}\left(h^{-2}\right) \\ \eta_{max}\left(\underline{\underline{M}}+\alpha\Delta t^2\underline{\underline{B}}\right) &= \mathcal{O}\left(h^{2/q}\lambda\right) \\ N_{iter} &= \mathcal{O}\left(\sqrt{\kappa\left(\underline{\underline{M}}+\alpha\Delta t^2\underline{\underline{B}}\right)}\right)=\mathcal{O}\left(h^{1/q-1}\sqrt{\lambda}\right) \\ Z_1 &= Z_{iter}\ N_{iter}=\mathcal{O}\left(h^{-(d+1-1/q)}\sqrt{\lambda}\right) \end{aligned}$$

So the total cost Z is

$$Z = \mathcal{O}\left(Z_1/\Delta t\right)=\mathcal{O}\left(h^{-(d+1)}\sqrt{\lambda}\right)$$

Remarks:

- This estimate is the same as for the explicit case. In reality, the estimate of cpu time in the explicit case turns out to be sharp, while the behaviour of the implicit case is better than the worst case estimate established namely for the number of iterations.
- As soon as a robust optimal solver for $\underline{\underline{M}}+\alpha\Delta t^2\underline{\underline{B}}$ is found, Z_1 drops to $\mathcal{O}\left(N\right)$ and we have $Z=\mathcal{O}\left(h^{-(d+1/q)}\right)$. Optimal means that computational costs grow only linear with N, whereas robust means that they are independent of λ and Δt. Optimal solvers are typically preconditioned iterative solvers. A good candidate for a respective preconditioner is the one proposed in [4]. However, the question of its robustness is open.

4.3.2 Benchmark Problem

In order to carry out a benchmark, it is favorable to have an exact solution the numerical one can be compared to. The first candidate for such a solution is usually a linear combination of eigenmodes. In section 2.1.3, we have only found eigenfunctions for periodic boundary conditions. As the translation is an eigenfunction with eigenvalue zero in this case, we would not have a bounded solution.

So we prescribe the exact solution $\underline{u}^*$ and calculate the right hand side such that the differential equation holds:

$$\begin{aligned} \Omega &= (0,1)^2 \\ \Gamma_D &= \partial\Omega \\ \underline{u}^*\left(\underline{x},t\right) &= \cos\left(\omega^* t\right)\begin{pmatrix} -\sin\left(\pi x\right)^2\sin\left(2\pi y\right) \\ \sin\left(2\pi x\right)\sin(\pi y)^2 \end{pmatrix} \\ \omega^* &= 2\sqrt{2}\,\pi \quad \mu=1 \quad \rho=1 \\ \underline{g}_D &= \underline{0} \end{aligned}$$

$$\begin{aligned}\underline{f} &= -\underline{\nabla} \cdot \underline{\underline{\sigma}}\left(\underline{u}^*\right) \\ &= \mu \cos\left(\omega^* t\right) \begin{pmatrix} 2\pi^2 \sin\left(2\pi y\right)\left(1 - 4\sin\left(\pi x\right)^2\right) \\ -2\pi^2 \sin\left(2\pi x\right)\left(1 - 4\sin\left(\pi y\right)^2\right) \end{pmatrix}\end{aligned}$$

The coarsest triangulation of the computational domain is shown in figure 4.7. This triangulation is refined uniformly for the convergence study, i. e. each triangle is split into 4 similar triangles recursively.

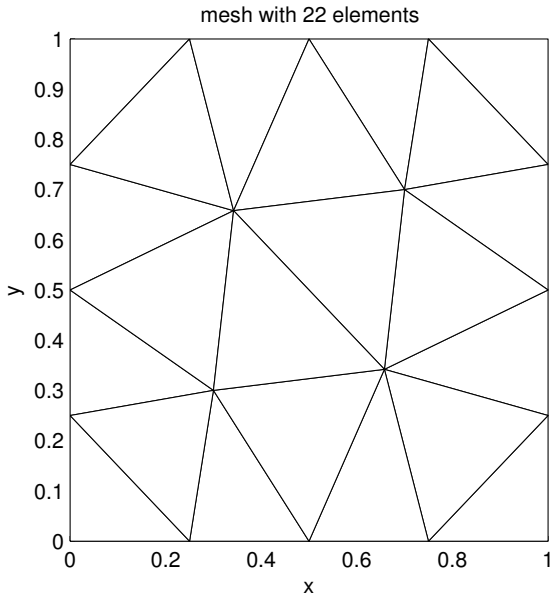

Figure 4.7: Computational domain and coarsest mesh for the dynamic problem

The suggested benchmark problem allows for a simple estimation of amplitude and frequency error. We have

$$\left\|\underline{u}^*\left(t\right)\right\|^2_{L^2(\Omega)} = A^* \cos\left(\omega^* t\right)^2 \quad \text{with } A^* = 3/8$$

so if we fit a function of the form $A\cos\left(\omega t + \varphi\right)^2$ against $\left\|\underline{u}_h\left(t\right)\right\|^2_{L^2(\Omega)}$ over one or two periods, we can use the value $\sqrt{A/A^*} - 1$ as an estimate for the relative amplitude error and $\omega/\omega^* - 1$ as an estimate for the relative frequency error. The least squares fit is done with a heuristic first guess followed by a few Newton iteration steps.

We assess the performance of the method by calculating the average L^2 norm of the error:

$$e\left(h\right) = \left\|\underline{u}_h - \underline{u}^*\right\|_{L^1(0,T;L^2(\Omega))}$$

where $\underline{u}_h$ is the finite element solution. In the optimal case, we could expect $e(h) = \mathcal{O}\left(h^2\right)$.

In the following subsections, we will test different timestepping schemes with this benchmark problem. We only show test results for the symmetric case. As we can conclude from the results in section 4.2, the unsymmetric case is not guaranteed to be stable. No stable case has been found in numerical experiments.

4.3.3 Trapezoidal Scheme

We simulate until final time $T = 10$, which is about 14 periods of the exact solution. Simulations with small meshes for $T = 360$ (>500 periods) indicate that the system does not get unstable, so shorter simulations are sufficent for assessing the method's performance.

The occuring linear systems $\underline{\underline{A}}\ \underline{x} = \underline{b}$ are solved with a conjugate gradient solver without preconditioner. The stopping criterion is chosen in terms of the relative residuum:

$$\frac{\left\|\underline{\underline{A}}\ \underline{x}_k - \underline{b}\right\|_2}{\left\|\underline{b}\right\|_2} < \text{tol}$$

where $\|\cdot\|_2$ denotes the Euclidean vector norm. We use $\text{tol} = 10^{-8}$ in our tests.

The timestep is chosen as follows:

$$\Delta t = \frac{\sqrt{2}}{40}\ \frac{h}{h_0}$$

where h_0 denotes the mesh size of the coarsest mesh. This choice satisfies two constraints:

- Global convergence rate: The time discretization error is $\mathcal{O}\left(\Delta t^2\right)$. With this choice, it is $\mathcal{O}\left(h^2\right)$ and will therefore not dominate the overall error.
- Sampling rate: In order to estimate the frequency error and the amplitude error, we need enough samples within one period of $\left\|\underline{u}^*\left(t\right)\right\|^2_{L^2(\Omega)}$, which is $\sqrt{2}/4$. With the above choice, we have 10 samples per period for the coarsest mesh and even more for the finer meshes.

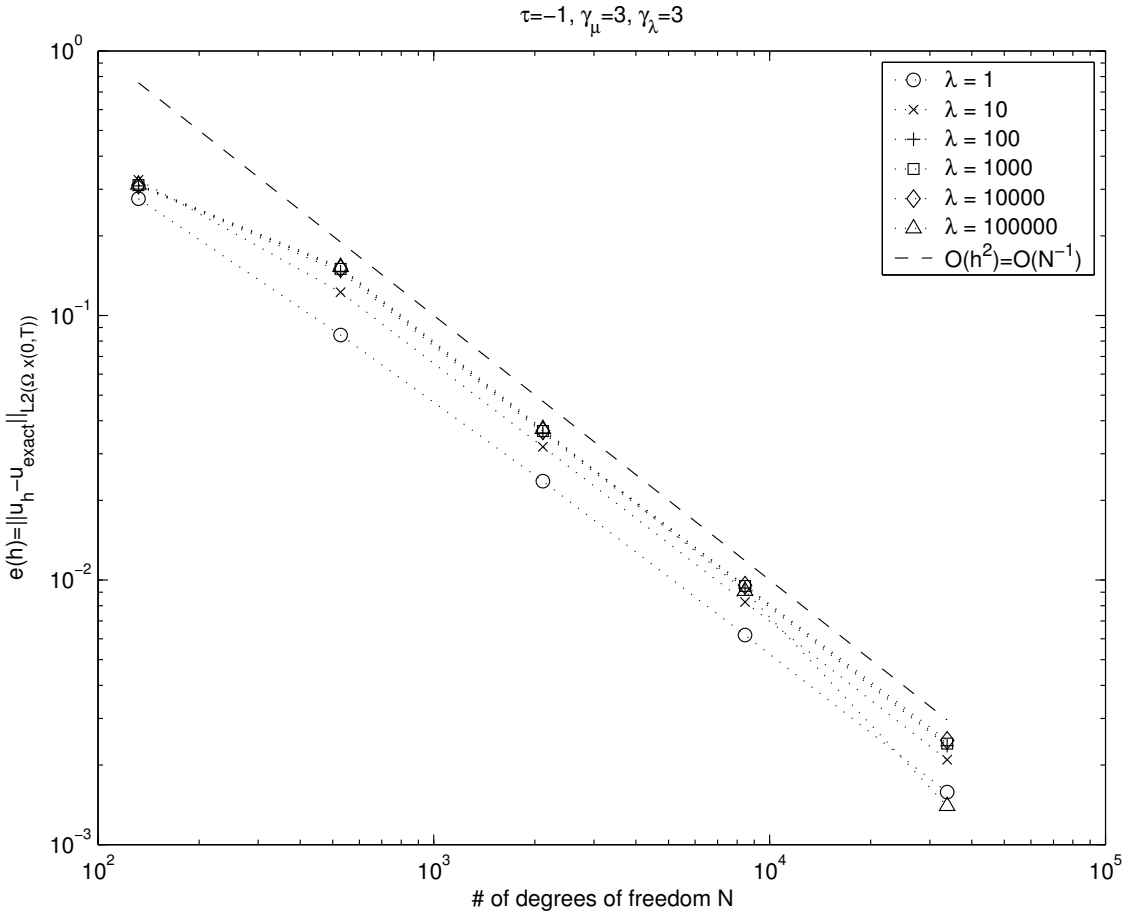

Figure 4.8: Mesh convergence of L^2 error for dynamic problem, trapezoidal scheme.

The average L^2 error is $\mathcal{O}(h^2)$, as can be seen in figure 4.8. That means, we have robust and optimal convergence also in the time dependent case.

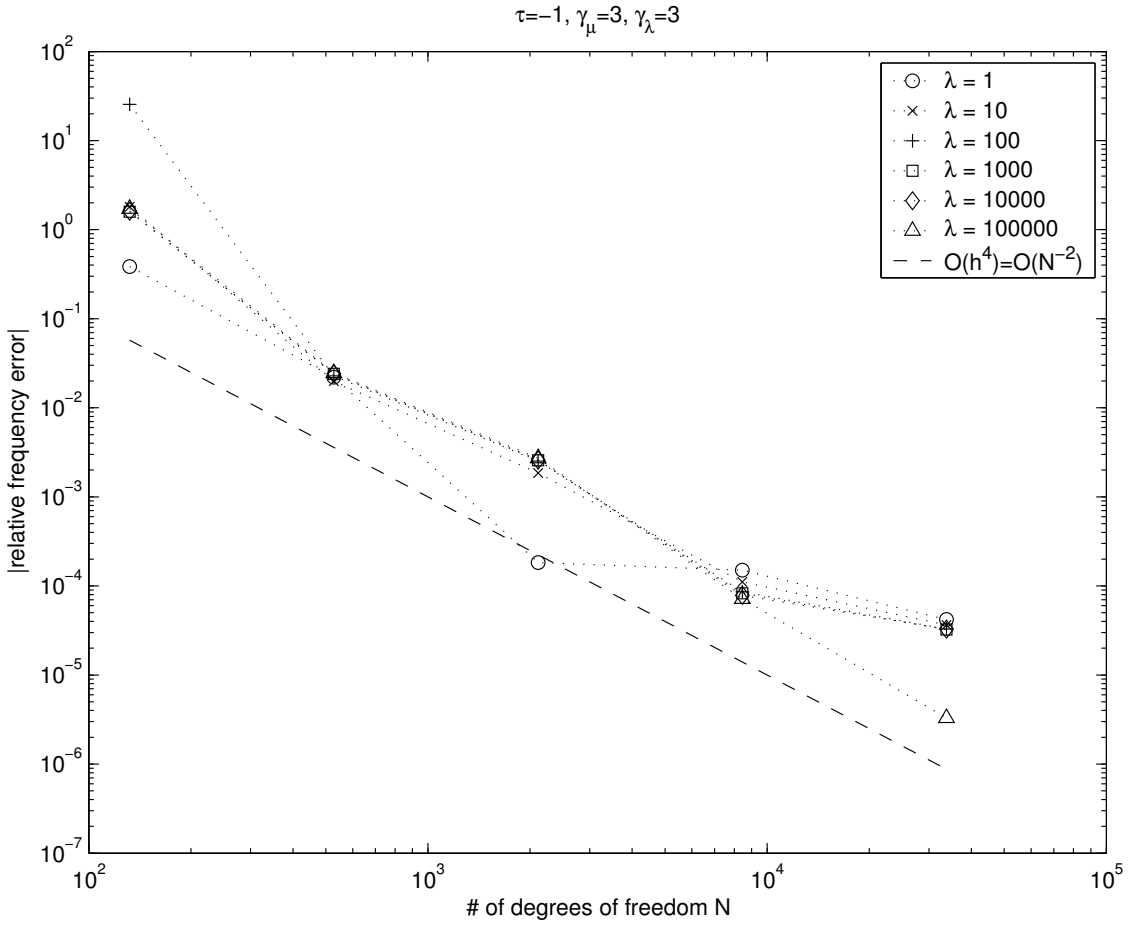

Figure 4.9: Mesh convergence of relative frequency error for dynamic problem, trapezoidal scheme.

The average relative frequency error even converges with about order 4 in h as can be seen in figure 4.9.

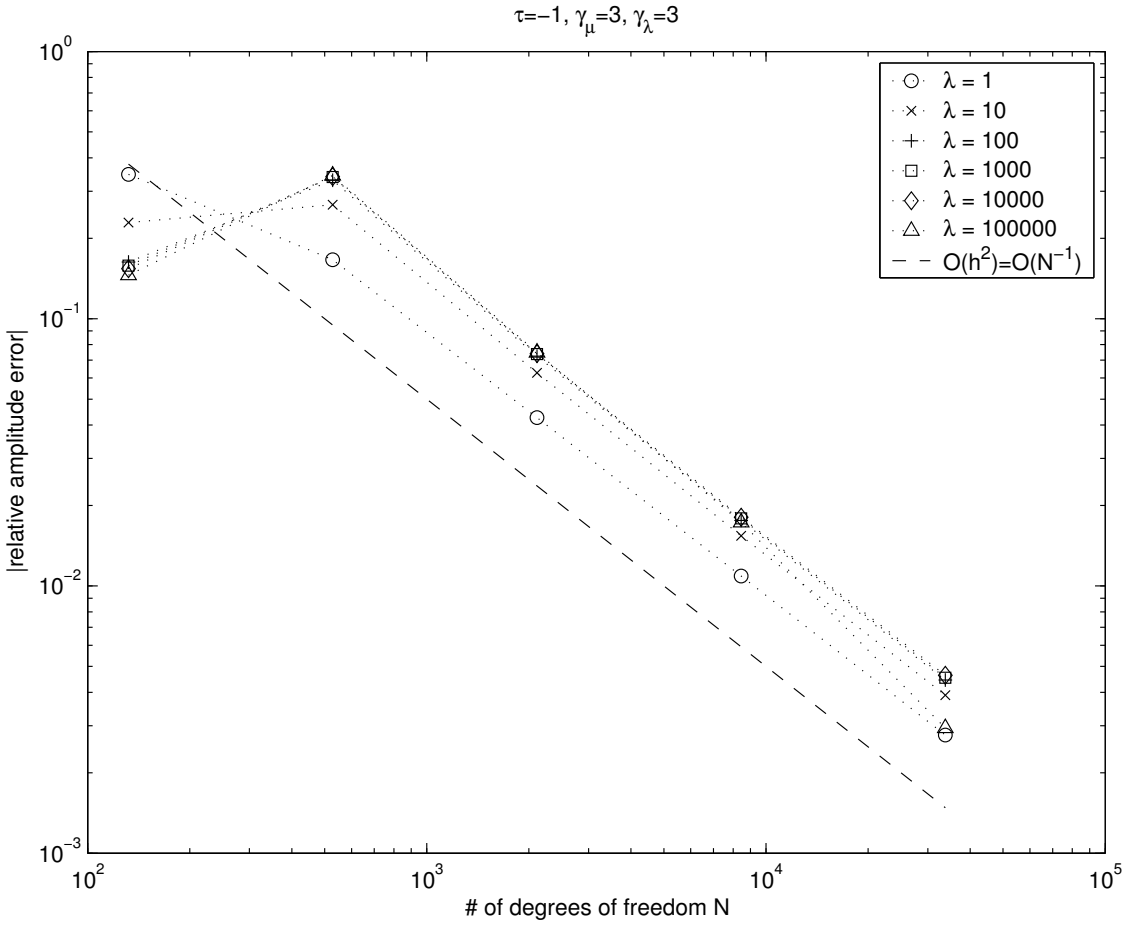

Figure 4.10: Mesh convergence of relative amplitude error for dynamic problem, trapezoidal scheme.

The average relative amplitude error converges with order 2 in h as can be seen in figure 4.10.

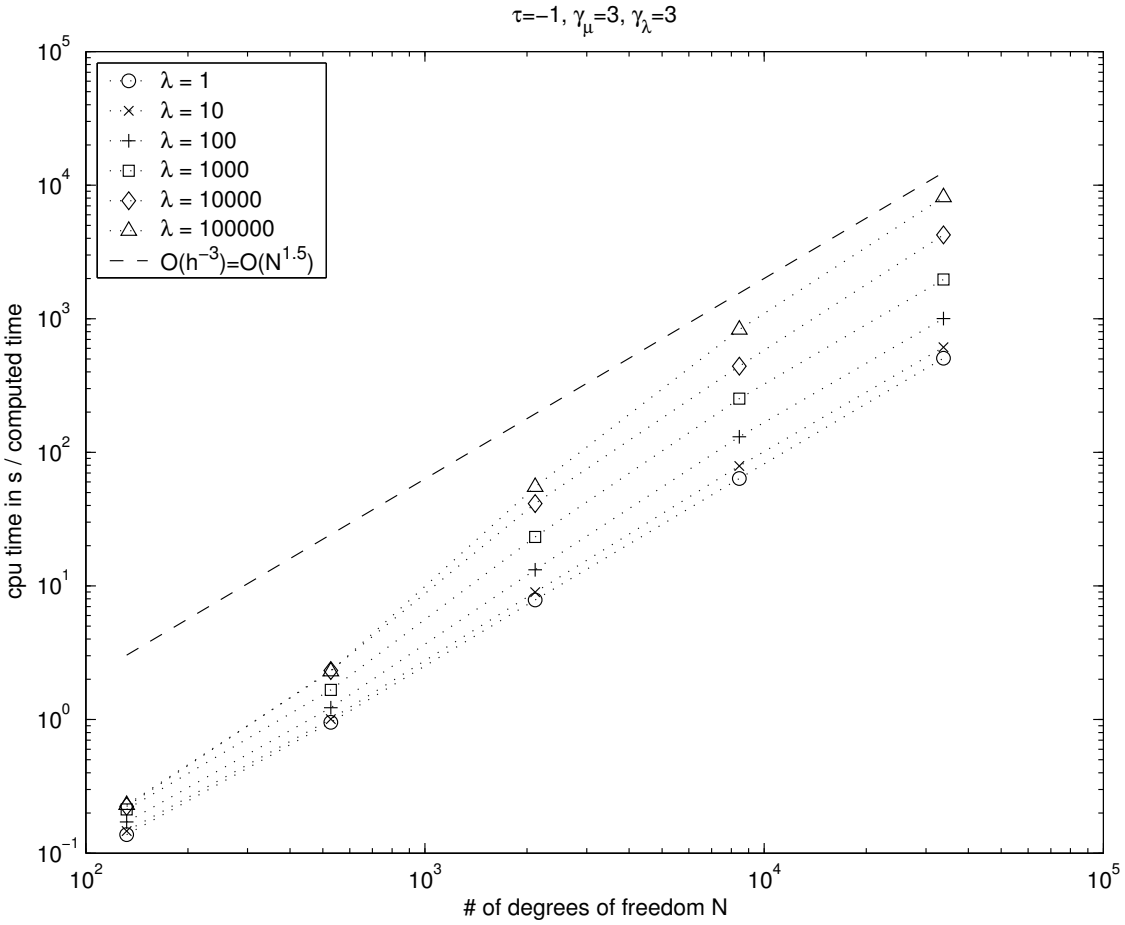

Figure 4.11: Mesh behaviour of cpu time for dynamic problem, trapezoidal scheme.

From section 4.3.1, we have to expect the computational costs to be $\mathcal{O}\left(h^{-(1+d)}\sqrt{\lambda}\right)$. With $d = 2$, we can expect the costs to be

$$\mathcal{O}\left(h^{-3}\sqrt{\lambda}\right) = \mathcal{O}\left(N^{1.5}\sqrt{\lambda}\right)$$

In the numerical experiment, the mesh behaviour seems to fit this prediction quite well, as can be seen in figure 4.11. The behaviour with λ seems to be not as bad as it would have to be expected.

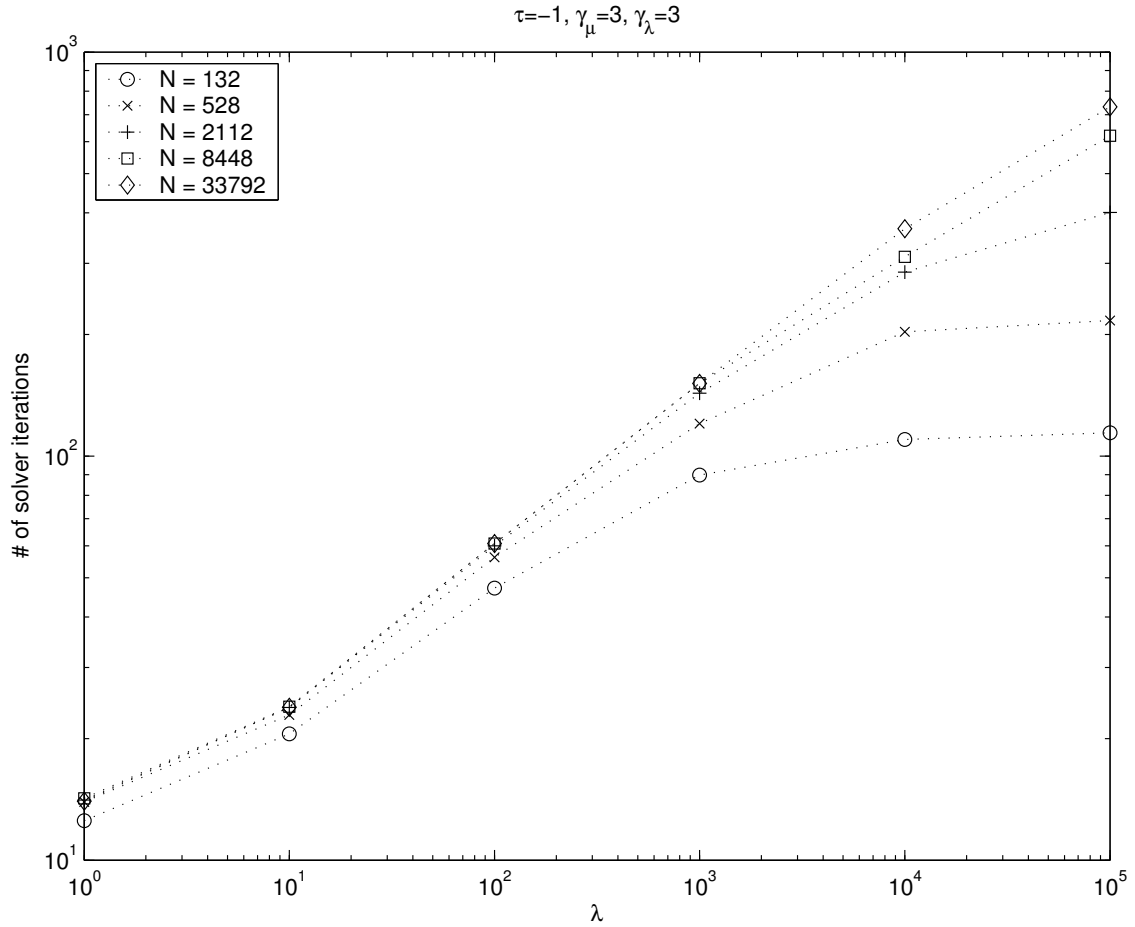

Figure 4.12: number of solver iterations for dynamic problem dependent on λ and number of degrees of freedom N, trapezoidal scheme.

In section 4.3.1, the number of iterations for the iterative solver N_{iter} was estimated to be $\mathcal{O}\left(\sqrt{\lambda N}\right)$. Figure 4.12, shows a slightly slower growth of the number of iterations with λ. Astonishingly, for small λ, the number of degrees of freedom N does not influence the number of iterations much. For large values of λ, the condition number gets so bad that the number of iterations gets close to N. As the conjugate gradient solver would provide the exact solution after N steps in exact arithmetics, N determines the number of iterations in this case.

Influence of solver tolerance

In a very similar setting, we test the influence of the solver tolerance. We simulate until $T = 140$ but let $\lambda = 100$ fixed and vary the solver tolerance tol. We measure the maximal instead of the average L^2 error. Everything else is left identic.

One could expect that solving iteratively and therefore not exact, introduces errors in each timestep, which then are propagated without damping. So the smaller the tolerance, the smaller would be the global error.

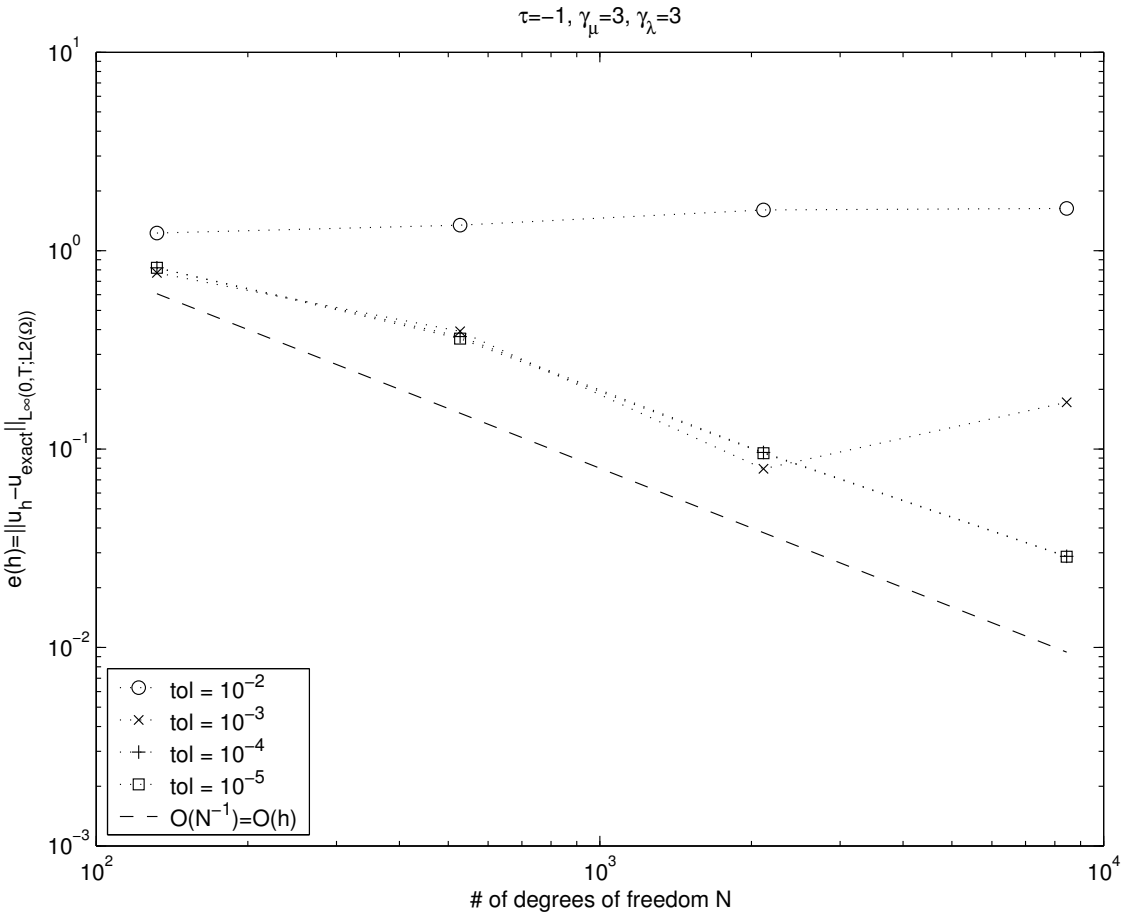

Figure 4.13: mesh convergence of maximal L^2 error for $\lambda = 100$, dependent on solver tolerance, trapezoidal scheme.

As can be seen in figure 4.13, the choice of the solver tolerance does matter, but is not sensitive. A solver tolerance of 10^{-2} is clearly too big in all cases. But whether one chooses 10^{-4} or 10^{-5} does not change the overall error visibly.

4.3.4 Hilber-Hughes-Taylor Scheme

The Hilber-Hughes-Taylor scheme allows for a dissipation of high frequency modes at the expense of not being conservative any more. We simulate until final time $T = 2$, with a dissipation parameter $\alpha = -0.05$. The results are similar in terms of errors, but the computation time is up to twice as high than without dissipation. So dissipation does not show any advantages in the symmetric case.

In the nonsymmetric case, the introduction of numerical dissipation for the high frequency modes seems attractive, as one might hope that this dissipation might outweigh the nonstable character of the eigenmodes with complex conjugate eigenvalues. However, this is only a hope: Numerical experiments show that even the maximally dissipative parameter $\alpha = -1/3$ is not sufficient to stabilize the system in the benchmark problem.

4.3.5 Nyström Scheme

We simulate until final time $T = 2$. In the occuring linear systems, only the mass matrix is involved, which is diagonal due to the orthogonal choice of the basis functions. The solving is therefore trivial. The timestep is chosen as follows:

$$\Delta t \;=\; \frac{1.9}{\sqrt{\eta_{max}\left(\underline{\underline{M}}^{-1}\underline{\underline{B}}\right)}}$$

where $\eta_{max}\left(\underline{\underline{M}}^{-1}\underline{\underline{B}}\right)$ is estimated by the power method. Like this, we are on the safe side of the stability border of the Nyström scheme.

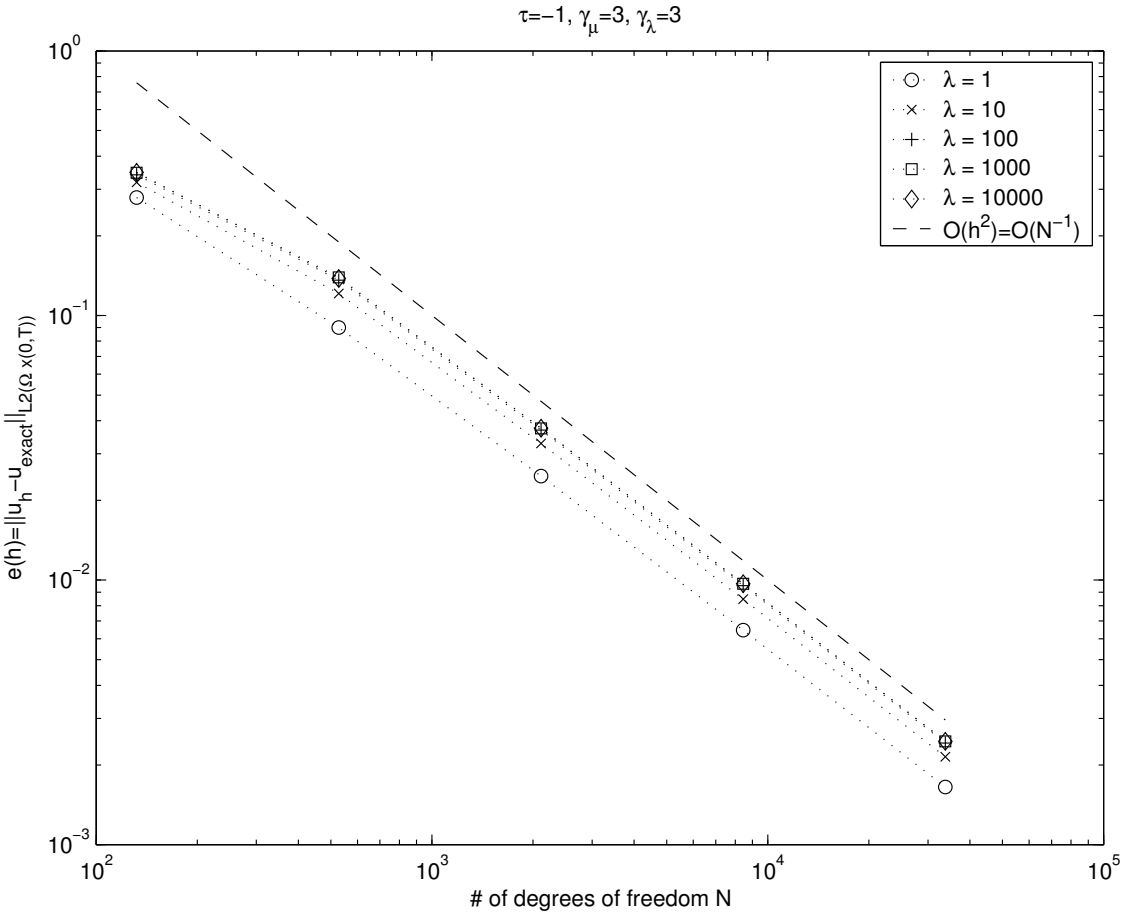

Figure 4.14: Mesh convergence of L^2 error for dynamic problem, Nyström scheme.

The average L^2 error is $\mathcal{O}(h^2)$, as can be seen in figure 4.14. That means, we have robust and optimal convergence also for the Nyström scheme.

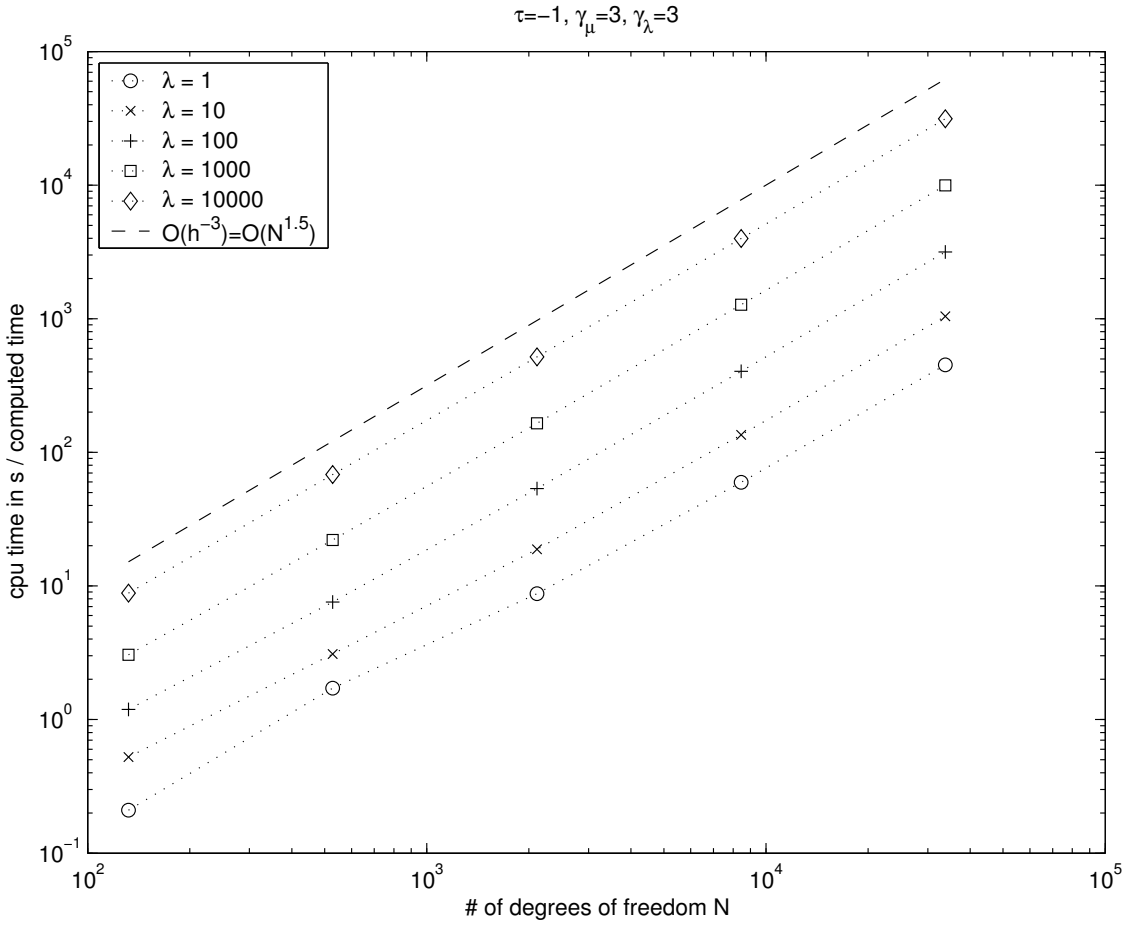

Figure 4.15: Mesh behaviour of cpu time for dynamic problem, Nyström scheme.

From section 4.3.1, we have to expect the computational costs to be $\mathcal{O}\left(h^{-(1+d)}\sqrt{\lambda}\right)$. With $d = 2$ we can expect the costs to be

$$\mathcal{O}\left(h^{-3}\sqrt{\lambda}\right) = \mathcal{O}\left(N^{1.5}\sqrt{\lambda}\right)$$

In the numerical experiment, the behaviour seems to fit this prediction quite well, as can be seen in figure 4.15. The estimate of computational cost seems to be sharp.

5 A Mixed FEM Approach

In order to overcome the growth in computation costs for growing λ, an equivalent alternative formulation is considered thereafter. The main idea is to introduce a new variable p and to couple it to $\underline{u}$ by the equation $p = -\lambda \underline{\nabla} \cdot \underline{u}$. Discretization of the mixed system yields a saddle point problem. For the static case, the resulting system reads

$$\begin{pmatrix} \underline{\underline{A}} & \underline{\underline{D}} \\ \underline{\underline{D}}^\top & -\underline{\underline{C}} \end{pmatrix} \begin{pmatrix} \underline{U} \\ \underline{P} \end{pmatrix} = \begin{pmatrix} \underline{L_v} \\ \underline{L_p} \end{pmatrix}$$

where $\underline{\underline{A}}$ and $\underline{\underline{D}}$ are independent of λ and $\underline{\underline{C}}$ scales with $1/\lambda$. Efficient solvers are known for problems of this type even if $\underline{\underline{C}} = 0$. So robustness with respect to λ can be expected.

5.1 Formulation

In [6], Hansbo and Larson formulate and analyze mixed discontinuous Galerkin method for linear elasticity which is equivalent to the non-mixed system presented in chapter 2. We again generalize the scheme to the time dependent problem in the following sections.

5.1.1 Continuous Equations

$$\begin{aligned}
\rho\, \partial_t^2 \underline{u}(\underline{x},t) - 2\mu\, \underline{\nabla} \cdot \underline{\underline{\varepsilon}}(\underline{u}(\underline{x},t)) - \underline{\nabla} p(\underline{x},t) &= \underline{f}(\underline{x},t) && (\underline{x},t) \in \Omega \times (0,T) \\
p(\underline{x},t)/\lambda &= -\underline{\nabla} \cdot \underline{u}(\underline{x},t) && (\underline{x},t) \in \Omega \times (0,T) \\
\partial_t \underline{u}(\underline{x},0) &= \underline{u}_1(\underline{x}) && \underline{x} \in \Omega \\
\underline{u}(\underline{x},0) &= \underline{u}_0(\underline{x}) && \underline{x} \in \Omega \\
\partial_t p(\underline{x},0)/\lambda &= -\underline{\nabla} \cdot \underline{u}_1(\underline{x}) && \underline{x} \in \Omega \\
p(\underline{x},0)/\lambda &= -\underline{\nabla} \cdot \underline{u}_0(\underline{x}) && \underline{x} \in \Omega \\
\underline{u}(\underline{x},t) &= \underline{g}_D(\underline{x},t) && (\underline{x},t) \in \Gamma_D \times (0,T) \\
2\mu\, \underline{\underline{\varepsilon}}(\underline{u}(\underline{x},t)) \cdot \underline{n} + p(\underline{x},t)\ \underline{n} &= \underline{g}_N(\underline{x},t) && (\underline{x},t) \in \Gamma_N \times (0,T)
\end{aligned}$$

In order the pressure parameter p to be unique, we have to require either $\lambda < \infty$, $|\Gamma_N| > 0$ or $\int_\Omega p \, \mathrm{d}\underline{x} = 0$. Note that the pressure parameter p is *not* the hydrostatic pressure unless $\lambda \to \infty$.

5.1.2 Finite Element Space

In addition to $\mathcal{S}_1$, we define the space of piecewise constant functions $\mathcal{S}_0$ analogously:

$$\begin{aligned}
\mathcal{S}_0 = \mathcal{S}^{0,0}(\Omega, \mathcal{T}) &:= \left\{ u \in L^2(\Omega) \ : \ u|_K \in \mathcal{P}_0(K), \ K \in \mathcal{T} \right\} \\
\mathcal{P}_0(K) &:= \{ u(\underline{x}) = a, \ a \in I\!R \} \\
\mathcal{S}_0 &= \operatorname{span} \{\psi_n\}_{n \in I_L}
\end{aligned}$$

The trial and test functions in this space are written as

$$p = \sum_{n \in I_L} (\underline{P})_n \psi_n \qquad q = \sum_{n \in I_L} (\underline{Q})_n \psi_n$$

5.1.3 Discontinuous Variational Formulation

Find $\underline{u} \in C^2([0,T], \underline{\mathcal{S}}_1)$ and $p \in C^2([0,T], \mathcal{S}_0)$ such that

$$\begin{aligned}
\rho \, \partial_t^2(\underline{u}, \underline{v}) + a(\underline{u}, \underline{v}) + d(p, \underline{v}) &= L_{\underline{v}}(\underline{v}) \ (t) \quad \forall \underline{v} \in \underline{\mathcal{S}}_1, \ t \in [0,T] \\
d(q, \underline{u}) - c(p, q) &= L_q(q) \ (t) \quad \forall \underline{q} \in \mathcal{S}_0, \ t \in [0,T] \\
\partial_t(\underline{u}, \underline{v}) &= (\underline{u}_1, \underline{v}) \quad \forall \underline{v} \in \underline{\mathcal{S}}_1, \ t = 0 \\
(\underline{u}, \underline{v}) &= (\underline{u}_0, \underline{v}) \quad \forall \underline{v} \in \underline{\mathcal{S}}_1, \ t = 0 \\
\partial_t(p/\lambda, q) &= (-\underline{\nabla} \cdot \underline{u}_1, q) \quad \forall q \in \mathcal{S}_0, \ t = 0 \\
(p/\lambda, q) &= (-\underline{\nabla} \cdot \underline{u}_0, q) \quad \forall q \in \mathcal{S}_0, \ t = 0
\end{aligned}$$

with:

$$\begin{aligned}
a(\underline{u}, \underline{v}) &:= \sum_{l \in I_L} \left(2\mu \, \underline{\underline{\varepsilon}}(\underline{u}), \underline{\underline{\varepsilon}}(\underline{v}) \right)_{K_l} \\
&\quad - \sum_{(l,l') \in F_D} \left(\left\langle 2\mu \, \underline{\underline{\varepsilon}}(\underline{u}) \cdot \underline{n}_{ll'} \right\rangle, [\underline{v}] \right)_{\Gamma_{ll'}} + \left([\underline{u}], \left\langle 2\mu \, \underline{\underline{\varepsilon}}(\underline{v}) \cdot \underline{n}_{ll'} \right\rangle \right)_{\Gamma_{ll'}} \\
&\quad + \mu \sum_{(l,l') \in F_D} \frac{\gamma_\mu}{h_{ll'}} \left([\underline{u}], [\underline{v}] \right)_{\Gamma_{ll'}} + \lambda \sum_{(l,l') \in F_D} \frac{\gamma_\lambda}{h_{ll'}} \left([\underline{u} \cdot \underline{n}_{ll'}], [\underline{v} \cdot \underline{n}_{ll'}] \right)_{\Gamma_{ll'}} \\
d(p, \underline{v}) &:= -\sum_{l \in I_L} (p, \underline{\nabla} \cdot \underline{v})_{K_l} + \sum_{(l,l') \in F_D} (\langle p \rangle, [\underline{v}] \cdot \underline{n}_{ll'})_{\Gamma_{ll'}} \\
c(p, q) &:= \sum_{l \in I_L} \left(\frac{1}{\lambda} p, q \right)_{K_l}
\end{aligned}$$

$$
\begin{aligned}
L_{\underline{v}}\left(\underline{v}\right)\ (t) \ :=\ & \sum_{l\in I_L}\left(\underline{f}\left(t\right),\underline{v}\right)_{K_l} \\
& + \sum_{(l,l')\in G_N}\left(\underline{g}_N\left(t\right),\underline{v}\right)_{\Gamma_{ll'}} - \sum_{(l,l')\in G_D}\left(\underline{g}_D\left(t\right),2\mu\ \underline{\underline{\varepsilon}}\left(\underline{v}\right)\cdot\underline{n}_{ll'}\right)_{\Gamma_{ll'}} \\
& + \mu\sum_{(l,l')\in G_D}\frac{\gamma_\mu}{h_{ll'}}\left(\underline{g}_D\left(t\right),\underline{v}\right)_{\Gamma_{ll'}} + \lambda\sum_{(l,l')\in F_D}\frac{\gamma_\lambda}{h_{ll'}}\left(\underline{g}_D\cdot\underline{n}_{ll'},\underline{v}\cdot\underline{n}_{ll'}\right)_{\Gamma_{ll'}} \\
L_q\left(q\right)\ (t) \ :=\ & \sum_{(l,l')\in G_D}\left(q,\underline{g}_D\left(t\right)\cdot\underline{n}_{ll'}\right)_{\Gamma_{ll'}}
\end{aligned}
$$

where

$\gamma_{\mu,\lambda}$: stabilization coefficients
$h_{ll'}$: suitable measure of mesh size on $\Gamma_{ll'}$

Remarks:

- The parameter γ_λ may be set to 0 in the mixed formulation. This has the essential advantage that $a(\cdot,\cdot)$ does not depend on λ any more and the condition number of the respective matrix must therefore be independent of λ.
- Only the symmetric variant is shown here, as the nonsymmetric one is neither analyzed in [6], nor promising to be stable in the time dependent case.
- The projections for the initial conditions $(\underline{u},\underline{v}) = (\underline{u}_0,\underline{v})$ and $(p/\lambda,q) = (-\underline{\nabla}\cdot\underline{u}_0,q)$ do not guarantee $p/\lambda = -\underline{\nabla}\cdot\underline{u}$. Numerical displacement and pressure parameter approximate exact displacement and pressure parameters best in L^2 sense. The latter are consistent, while the former do not have to be. This might be overcome by choosing a different projection for the initial conditions.

5.1.4 Element Computations

This section gives a broad overview of the element computations. A more detailed presentation is provided in appendix B.

Stiffness Matrix

$$
\begin{aligned}
a\left(\underline{u},\underline{v}\right) &= \underline{V}^\top\underline{\underline{A}}\,\underline{U} \\
\left(\underline{\underline{A}}\right)_{nn'} &= c\left(\underline{\varphi}_{n'},\underline{\varphi}_n\right)
\end{aligned}
$$

Divergence Matrix

$$
\begin{aligned}
d\left(p,\underline{v}\right) &= \underline{V}^\top\underline{\underline{D}}\,\underline{P} \\
\left(\underline{\underline{D}}\right)_{nn'} &= d(\psi_{n'},\underline{\varphi}_n)
\end{aligned}
$$

Compression Matrix

$$\begin{aligned} c\left(p,q\right) &= \underline{Q}^{\top}\underline{\underline{C}}\,\underline{P} \\ \left(\underline{\underline{C}}\right)_{nn'} &= c\left(\psi_{n'},\psi_n\right) \end{aligned}$$

Note that the compression matrix will be diagonal as the mass matrix.

Load Vector

$$\begin{aligned} L_{\underline{v}}\left(\underline{v}\right)\ \left(t\right) &= \underline{V}^{\top}\underline{L}_{\underline{v}}\left(t\right) \\ \left(\underline{L}_{\underline{v}}\left(t\right)\right)_n &= L_{\underline{v}}\left(\underline{\varphi}_n\right)\ \left(t\right) \\ L_q\left(q\right)\ \left(t\right) &= \underline{Q}^{\top}\underline{L}_q\left(t\right) \\ \left(\underline{L}_q\left(t\right)\right)_n &= L_q\left(\psi_n\right)\ \left(t\right) \end{aligned}$$

Initial Conditions

The initial conditions for $\underline{u}$ are the same as in the non-mixed formulation. Those for the pressure parameter read:

$$\begin{aligned} \left(p/\lambda,q\right) &= c\left(p,q\right) = \underline{Q}^{\top}\underline{\underline{C}}\,\underline{P} \\ \left(-\underline{\nabla}\cdot\underline{u}_i,q\right) &= \underline{Q}^{\top}\underline{P}_i \\ \left(\underline{P}_i\right)_n &= \left(-\underline{\nabla}\cdot\underline{u}_i,\psi_n\right) \end{aligned}$$

5.1.5 Differential Algebraic System of Equations

By inserting the expressions found in the element computation section into the variational formulation, we get a differential algebraic system of equations:
Find $\underline{U}\in C^2\left(\left[0,T\right],I\!R^N\right)$ and $\underline{P}\in C^2\left(\left[0,T\right],I\!R^L\right)$ such that

$$\underbrace{\begin{pmatrix} \rho\underline{\underline{M}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{pmatrix}}_{\tilde{\underline{\underline{M}}}} \begin{pmatrix} \ddot{\underline{U}} \\ \ddot{\underline{P}} \end{pmatrix} + \underbrace{\begin{pmatrix} \underline{\underline{A}} & \underline{\underline{D}} \\ \underline{\underline{D}}^{\top} & -\underline{\underline{C}} \end{pmatrix}}_{\tilde{\underline{\underline{B}}}} \begin{pmatrix} \underline{U} \\ \underline{P} \end{pmatrix} = \begin{pmatrix} \underline{L}_{\underline{v}}\left(t\right) \\ \underline{L}_q\left(t\right) \end{pmatrix}$$

$$\begin{aligned} \underline{\underline{M}}\,\dot{\underline{U}}\left(0\right) &= \underline{U}_1 \\ \underline{\underline{M}}\,\underline{U}\left(0\right) &= \underline{U}_0 \\ \underline{\underline{C}}\,\dot{\underline{P}}\left(0\right) &= \underline{P}_1 \\ \underline{\underline{C}}\,\underline{P}\left(0\right) &= \underline{P}_0 \end{aligned}$$

Note that $\tilde{\underline{\underline{M}}}$ and $\tilde{\underline{\underline{B}}}$ are indefinite, as $\underline{\underline{C}}$ is positive definite for $\lambda<\infty$.

If we would eliminate $\underline{P}$ from the system, we would obtain an equation similar to the non-mixed one:

$$\rho \underline{\underline{M}}\, \underline{\ddot{U}} + \underbrace{\left(\underline{\underline{A}} + \underline{\underline{D}}\, \underline{\underline{C}}^{-1} \underline{\underline{D}}^{\top}\right)}_{\approx \underline{\underline{B}}} \underline{U} \;=\; \underbrace{\underline{\underline{D}}\, \underline{\underline{C}}^{-1} \underline{L_q} + \underline{L_v}}_{\approx \underline{L}}$$

This system has similar properties as the non-mixed one, so we should avoid this.

The equations for the pressure parameter represent constraints rather than evolution equations. As constraints, they are intrinsically implicit and a fully explicit timestepping scheme seems to be out of reach. Thus we will solve the system with the second order implicit timestepping schemes applied also for the non-mixed system. So, in each timestep, a system with the matrix

$$\begin{pmatrix} \rho \underline{\underline{M}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{pmatrix} + \alpha \Delta t^2 \begin{pmatrix} \underline{\underline{A}} & \underline{\underline{D}} \\ \underline{\underline{D}}^{\top} & -\underline{\underline{C}} \end{pmatrix}$$

has to be solved, where α is a constant that only depends on the chosen timestepping scheme. This system has the same type as the system arising from the stationary problem

$$\begin{pmatrix} \underline{\underline{A}} & \underline{\underline{D}} \\ \underline{\underline{D}}^{\top} & -\underline{\underline{C}} \end{pmatrix} \begin{pmatrix} \underline{U} \\ \underline{P} \end{pmatrix} \;=\; \begin{pmatrix} \underline{L_v} \\ \underline{L_p} \end{pmatrix}$$

For this type of system, we consider two algorithms:

Uzawa Algorithm in Conjugate Directions

For this algorithm, presented in [1] for $\underline{\underline{C}} = 0$, as for all Uzawa algorithms, $\underline{U}$ is eliminated from the equation resulting in a system for $\underline{P}$:

$$\left(\underline{\underline{D}}^{\top} \underline{\underline{A}}^{-1} \underline{\underline{D}} + \underline{\underline{C}}\right) \underline{P} \;=\; \underline{\underline{D}}^{\top} \underline{\underline{A}}^{-1} \underline{L_v} - \underline{L_p}$$

This system is symmetric and positive definite, but only implicitly given. Beyond that, it is well conditioned. This can be seen heuristically by the fact that the two discrete first order operators $\underline{\underline{D}}$ and $\underline{\underline{D}}^{\top}$ compensate the inverse of the discrete second order operator $\underline{\underline{A}}$. Thus solving the system using conjugate gradients (CG) requires a number of iterations bounded by a constant independent of N and λ. This behaviour was observed in the experiments. In each iteration, the action of $\underline{\underline{A}}^{-1}$ is required. This may be accomplished through an inner CG iteration or any other convenient solver. Note that there are efficient solvers for $\underline{\underline{A}}$, see e. g. [4]. Once $\underline{P}$ is known, $\underline{U}$ is calculated from

$$\underline{\underline{A}}\, \underline{U} \;=\; \underline{L_v} - \underline{\underline{D}}\, \underline{P}$$

Bramble-Pasciak Algorithm

In [2], Bramble and Pasciak suggest a transformation using a preconditioner $\underline{\underline{W}}$ approximating $\underline{\underline{A}}^{-1}$:

$$\begin{pmatrix} \underline{\underline{W}}\,\underline{\underline{A}} & \underline{\underline{W}}\,\underline{\underline{D}} \\ \underline{\underline{D}}^\top(\underline{\underline{W}}\,\underline{\underline{A}} - \underline{\underline{1}}) & \underline{\underline{C}} + \underline{\underline{D}}^\top \underline{\underline{W}}\,\underline{\underline{D}} \end{pmatrix} \begin{pmatrix} \underline{U} \\ \underline{P} \end{pmatrix} = \begin{pmatrix} \underline{\underline{W}}\,\underline{L_v} \\ \underline{\underline{D}}^\top \underline{\underline{W}}\,\underline{L_v} - \underline{L_p} \end{pmatrix}$$

The transformed system is symmetric positive definite in an inner product introduced also in [2]. This allows the resulting system to be solved with classical iterative algorithms like conjugate gradients in this inner product. The advantage is that no inner (exact) solve is required. Only a preconditioner for $\underline{\underline{A}}$ satisfying weak requirements has to be provided.

5.2 Numerical Results

All numerical tests are carried out with $\gamma_\lambda = 0$, as this is allowed and has substantial advantages. γ_μ is set to 3 for better comparability.

5.2.1 Static Tests

Setting $\partial_t \underline{u} = 0$, we obtain the static problem

$$\begin{aligned} -2\mu\, \underline{\nabla} \cdot \underline{\underline{\varepsilon}}(\underline{u}(\underline{x})) - \underline{\nabla} p(\underline{x}) &= \underline{f}(\underline{x}) && \underline{x} \in \Omega \\ p(\underline{x})/\lambda &= -\underline{\nabla} \cdot \underline{u}(\underline{x}) && \underline{x} \in \Omega \\ \underline{u}(\underline{x}) &= \underline{g}_D(\underline{x}) && \underline{x} \in \Gamma_D \\ 2\mu\, \underline{\underline{\varepsilon}}(\underline{u}(\underline{x})) \cdot \underline{n} + p(\underline{x})\ \underline{n} &= \underline{g}_N(\underline{x}) && \underline{x} \in \Gamma_N \end{aligned}$$

Applying the same discretization as for the time dependent case, we obtain the following system of linear equations:

$$\begin{pmatrix} \underline{\underline{A}} & \underline{\underline{D}} \\ \underline{\underline{D}}^\top & -\underline{\underline{C}} \end{pmatrix} \begin{pmatrix} \underline{U} \\ \underline{P} \end{pmatrix} = \begin{pmatrix} \underline{L_v} \\ \underline{L_p} \end{pmatrix}$$

We consider the same regular static problem as in section 4.1.2 with the same coarsest triangulation shown in figure 4.1. This triangulation is refined uniformly for the convergence study, i. e. each triangle is split into 4 similar triangles recursively.

For better comparability, we assess the performance of the method again by calculating the L^2 norm of the error of the displacement:

$$e(h) = \|\underline{u}_h - \underline{u}^*\|_{L^2(\Omega)}$$

where $\underline{u}_h$ is the finite element solution. In the optimal case, we can expect $e(h) = \mathcal{O}(h^2)$. The observed convergence is in fact robust and optimal for both the Bramble-Pasciak and the Uzawa algorithm.

For both algorithms, cpu time grows slightly stronger than $N^{1.5}$ and independent of λ. This is shown in figure 5.1 for the Uzawa algorithm with CG inner solver. Clearly this is an advantage over the non-mixed standard approach, where cpu time grows with λ and almost like N^2, as can be seen in figure 5.2.

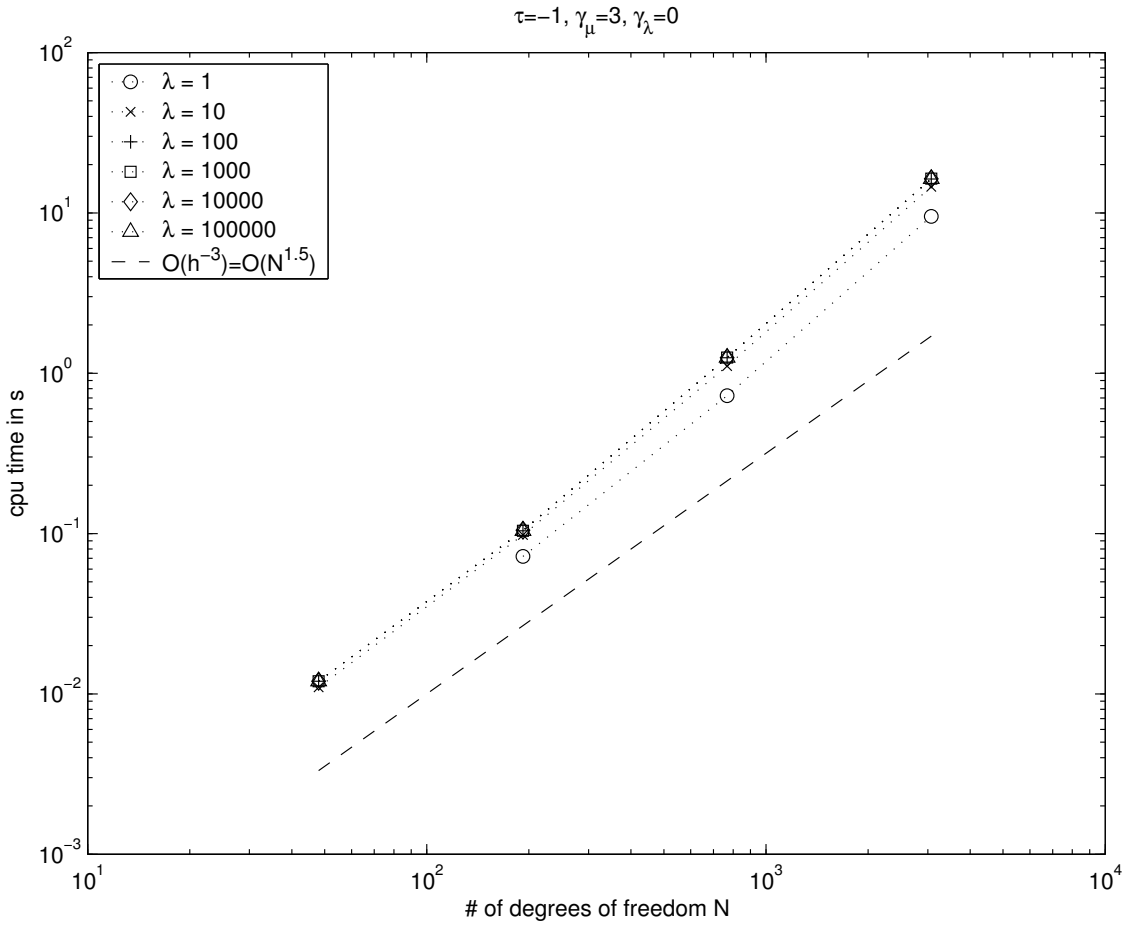

Figure 5.1: Mesh behaviour of cpu time for static problem, Uzawa algorithm with CG inner solver.

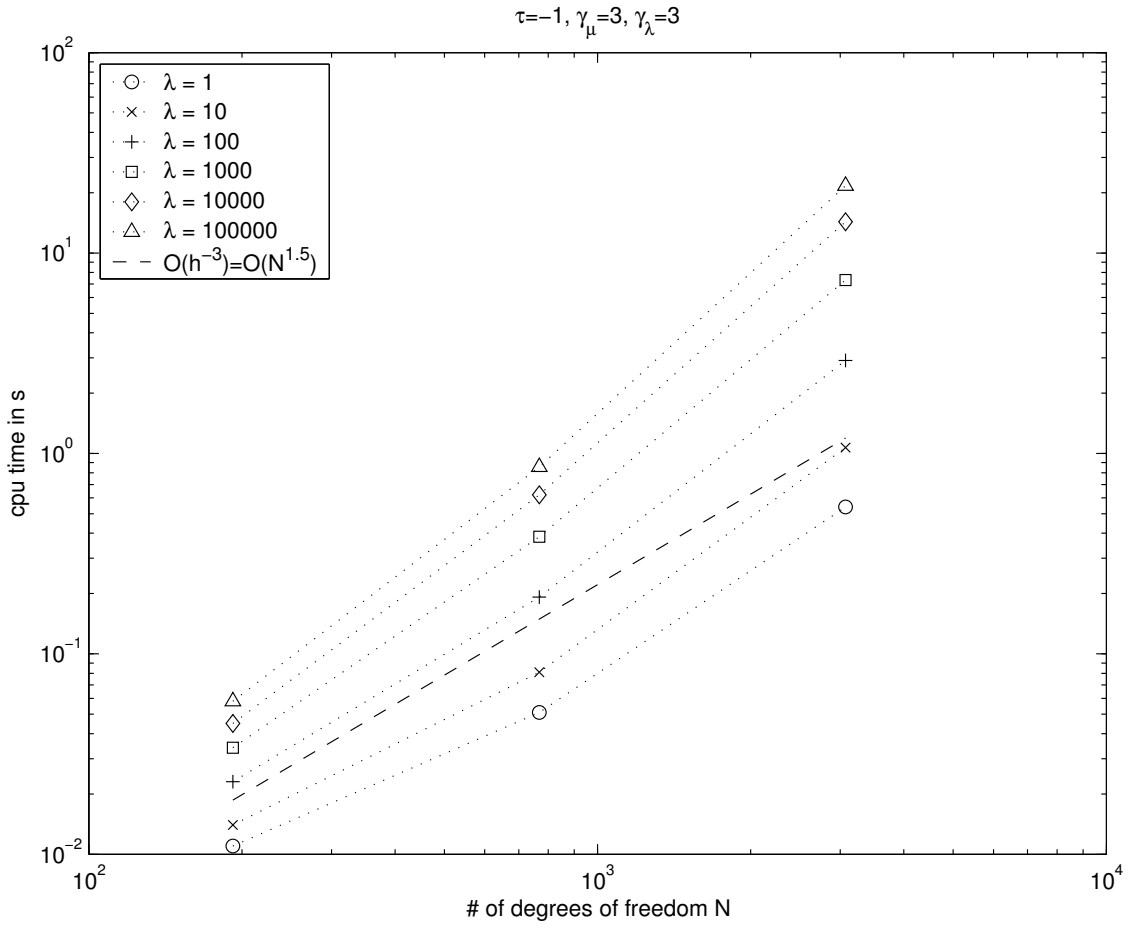

Figure 5.2: Mesh behaviour of cpu time for static problem, standard DGFEM with CG solver.

5.2.2 Dynamic Tests

For the dynamic tests, we consider the benchmark problem from section 4.3.2 with the same coarsest triangulation shown in figure 4.7, and use the same error measure.

The observed convergence is robust but suboptimal. It can be seen in figure 5.3 that the convergence order with respect to h is only about 1.5.

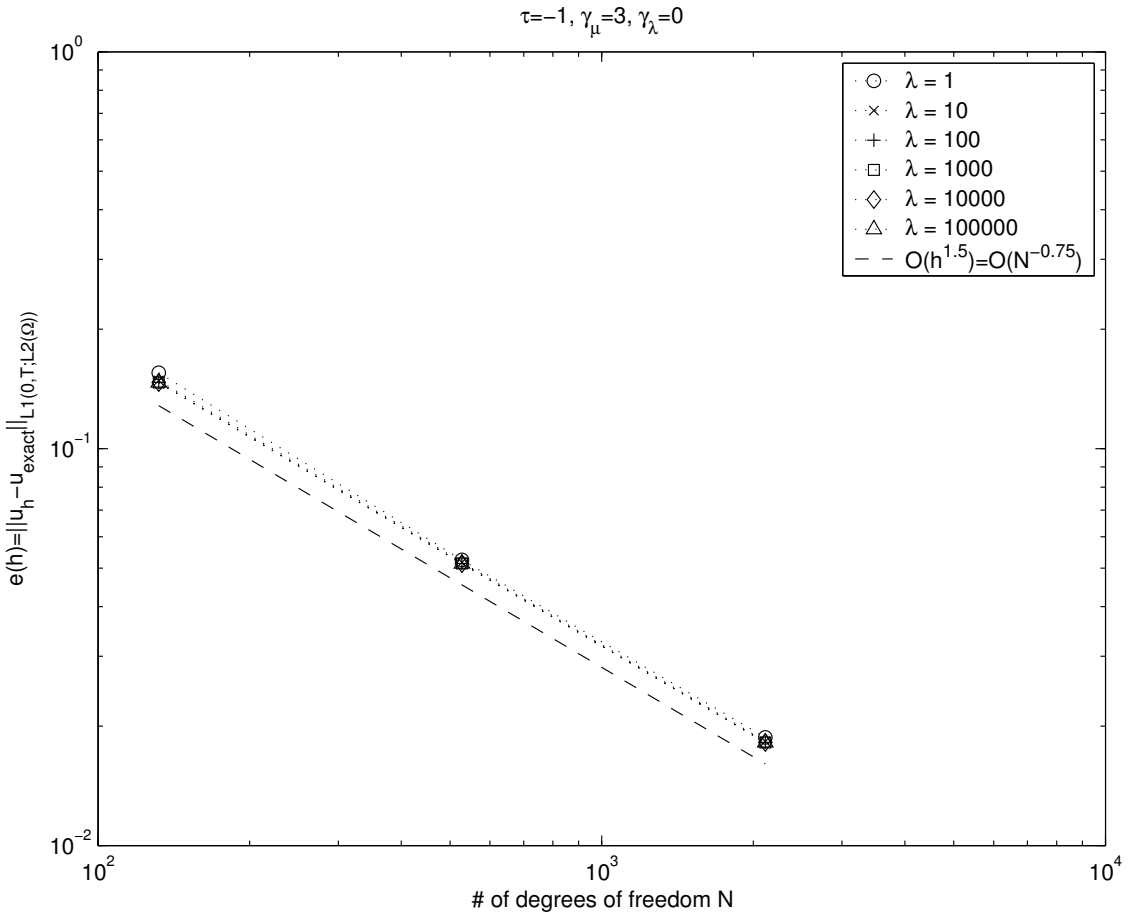

Figure 5.3: Mesh convergence of average L^2 error for dynamic problem, Uzawa algorithm with direct inner solver.

Beside that, cpu time does not depend on λ asymptotically, but it grows like N^3. This disappointing conclusion has to be drawn from figure 5.4. The reason is clear from a heuristic point of view: While the matrix of the reduced system for the static problem

$$\underline{\underline{D}}^\top \underline{\underline{A}}^{-1} \underline{\underline{D}} + \underline{\underline{C}}$$

is well conditioned because the order of operators compensate, this is not any more the case for the reduced system for the dynamic problem:

$$\alpha \Delta t^2 \left(\alpha \Delta t^2 \underline{\underline{D}}^\top \left(\underline{\underline{M}} + \alpha \Delta t^2 \underline{\underline{A}} \right)^{-1} \underline{\underline{D}} + \underline{\underline{C}} \right)$$

So the condition number of this reduced system is *not* independent of N any more.

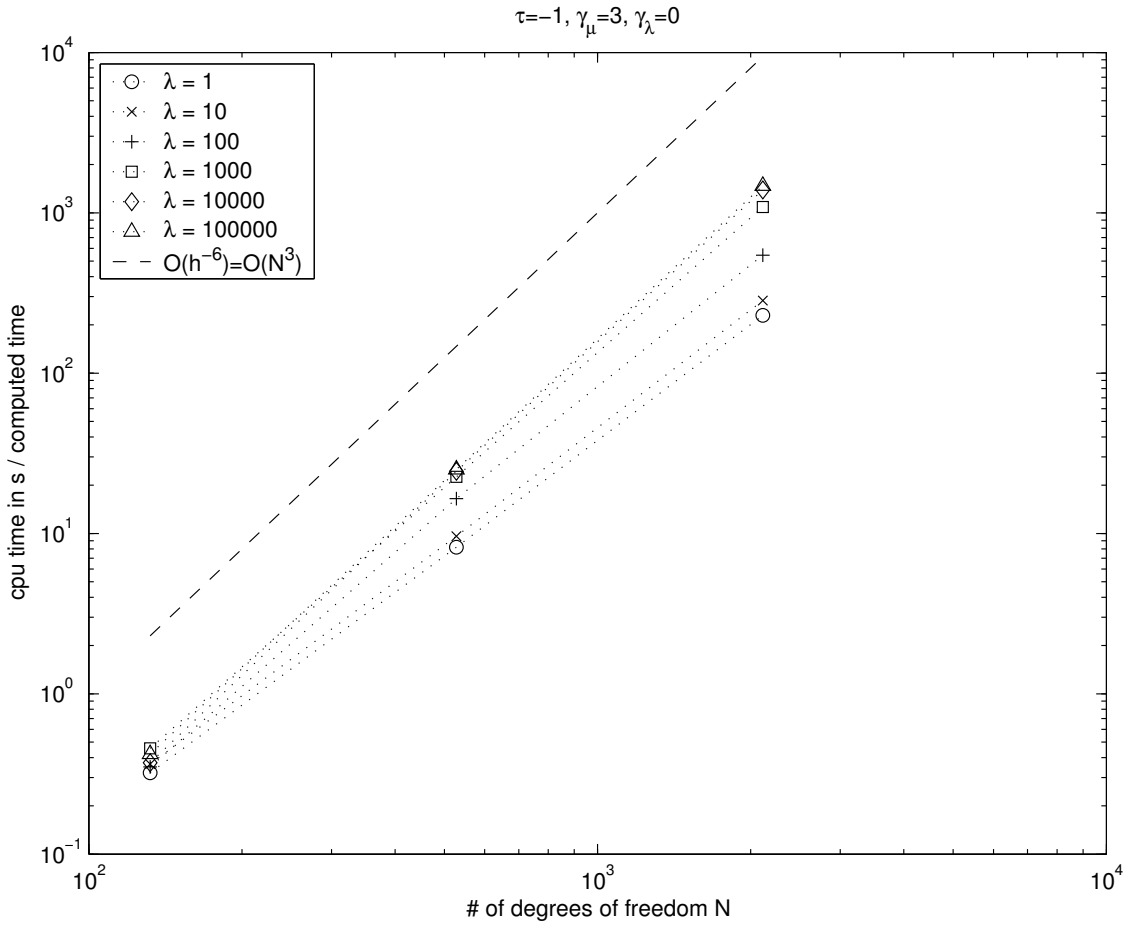

Figure 5.4: Mesh behaviour of cpu time for dynamic problem, Uzawa algorithm with direct inner solver.

6 Conclusions

The symmetric variant of the non-mixed discontinuous scheme formulated in this thesis has turned out to be locking free and converge optimally in all numerical tests carried out. Computational costs are however growing with increasing value of λ.

The nonsymmetric variant showed a divergent behaviour which can be well explained by the occurence of complex conjugate eigenvalues in the discrete spacial operator.

The mixed scheme formulated is locking free and has optimal convergence for the stationary problem. Furthermore, the computational costs do not depend on λ. In the time dependent case however, convergence is still robust but suboptimal, and computational costs are not robust in Δt.

In order to achieve optimal efficiency, a Δt-robust preconditioner for the matrix of the reduced system for the dynamic problem is pending.

Acknowledgements

I would like to thank: Prof. Christoph Schwab and Prof. Ralf Hiptmair for their guidance, Philipp Frauenfelder for his advice in software related questions, Dr. Thomas Wihler for sharing his thoughts about discontinuous Galerkin methods and Gisela Widmer, Stefan Benkler and Adrian Burri for many fruitful discussions.

A Details of Standard Approach

A.1 Componentwise Representation

General Ideas

As $\underline{\underline{\sigma}}$ and $\underline{\underline{\varepsilon}}$ are symmetric, they have $q = d\,(d+1)\,/2$ independent components which we group into vectors in $\mathbb{R}^q$:

$$\begin{aligned}&\underline{\sigma}, \underline{\varepsilon} \in \mathbb{R}^q\\ &\underline{\sigma} = \underline{\underline{E}}\ \underline{\varepsilon}, \quad \underline{\underline{E}} \in \mathbb{R}^{q\times q}\end{aligned}$$

Note: Due to the uniqueness of the deformation energy $^1/_2\underline{\varepsilon}^\top \underline{\underline{E}}\ \underline{\varepsilon}$, $\underline{\underline{E}}$ is again symmetric and has $q\,(q+1)\,/2$ independent components.

$\underline{\varepsilon}$ can now be obtained from $\underline{u}$ with the differential operator matrix $\underline{\underline{D}}$:

$$\underline{\varepsilon}\,(\underline{u}) \;=\; \underline{\underline{D}}\ \underline{u} = \sum_{j\in I_d} \underline{\underline{C}}_j \partial_j \underline{u}$$

Two Dimensions

$$\begin{aligned}\underline{\sigma} &= \begin{pmatrix} \sigma_{00} & \sigma_{11} & \sigma_{01}\end{pmatrix}^\top\\ \underline{\varepsilon} &= \begin{pmatrix} \varepsilon_{00} & \varepsilon_{11} & 2\varepsilon_{01}\end{pmatrix}^\top\\ \underline{\underline{E}} &= \begin{pmatrix} A_{0000} & A_{0011} & A_{0001}\\ A_{0011} & A_{1111} & A_{1101}\\ A_{0001} & A_{1101} & A_{0101}\end{pmatrix} = \begin{pmatrix} \lambda+2\mu & \lambda & 0\\ \lambda & \lambda+2\mu & 0\\ 0 & 0 & \mu\end{pmatrix}\\ \underline{\underline{D}} &:= \begin{pmatrix} \partial_0 & 0\\ 0 & \partial_1\\ \partial_1 & \partial_0\end{pmatrix}, \quad \underline{\underline{C}}_0 = \begin{pmatrix} 1 & 0\\ 0 & 0\\ 0 & 1\end{pmatrix}, \quad \underline{\underline{C}}_1 = \begin{pmatrix} 0 & 0\\ 0 & 1\\ 1 & 0\end{pmatrix}\end{aligned}$$

Three Dimensions

$$\begin{aligned}\underline{\sigma} &= \begin{pmatrix} \sigma_{00} & \sigma_{11} & \sigma_{22} & \sigma_{12} & \sigma_{20} & \sigma_{01}\end{pmatrix}^\top\\ \underline{\varepsilon} &= \begin{pmatrix} \varepsilon_{00} & \varepsilon_{11} & \varepsilon_{22} & 2\varepsilon_{12} & 2\varepsilon_{20} & 2\varepsilon_{01}\end{pmatrix}^\top\end{aligned}$$

$$
\begin{aligned}
\underline{\underline{E}} &= \begin{pmatrix}
A_{0000} & A_{0011} & A_{0022} & A_{0012} & A_{0020} & A_{0001} \\
A_{0011} & A_{1111} & A_{1122} & A_{1112} & A_{1120} & A_{1101} \\
A_{0022} & A_{1122} & A_{2222} & A_{2212} & A_{2220} & A_{2201} \\
A_{0012} & A_{1112} & A_{2212} & A_{1212} & A_{1220} & A_{1201} \\
A_{0020} & A_{1120} & A_{2220} & A_{1220} & A_{2020} & A_{2001} \\
A_{0001} & A_{1101} & A_{2201} & A_{1201} & A_{2001} & A_{0101}
\end{pmatrix} \\
\underline{\underline{E}} &= \begin{pmatrix}
\lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\
\lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\
\lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\
0 & 0 & 0 & \mu & 0 & 0 \\
0 & 0 & 0 & 0 & \mu & 0 \\
0 & 0 & 0 & 0 & 0 & \mu
\end{pmatrix} \\
\underline{\underline{D}} &:= \begin{pmatrix}
\partial_0 & 0 & 0 \\
0 & \partial_1 & 0 \\
0 & 0 & \partial_2 \\
0 & \partial_2 & \partial_1 \\
\partial_2 & 0 & \partial_0 \\
\partial_1 & \partial_0 & 0
\end{pmatrix} \\
\underline{\underline{C}}_0 &= \begin{pmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix}, \quad \underline{\underline{C}}_1 = \begin{pmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 0
\end{pmatrix}, \quad \underline{\underline{C}}_2 = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}
\end{aligned}
$$

Consequences

For these choices of $\underline{\sigma}$, $\underline{\varepsilon}$ and $\underline{\underline{C}}_j$, we have:

$$
\begin{aligned}
\underline{\sigma}\left(u\underline{e}_j\right) : \underline{\varepsilon}\left(v\underline{e}_{j'}\right) = \underline{\sigma}\left(u\underline{e}_j\right)^\top \underline{\varepsilon}\left(v\underline{e}_{j'}\right) &= \left(\underline{D}\ v\underline{e}_{j'}\right)^\top \underline{\underline{E}}\ \underline{D}\ u\underline{e}_j \\
&= \left(\underline{\nabla} v\right)^\top \underline{\underline{C}}_{j'}^\top \underline{\underline{E}}\ \underline{\underline{C}}_j\ \underline{\nabla} u
\end{aligned}
$$

where u and v are any scalar functions and $\underline{e}_j$ is the canonic unit vector with $\left(\underline{e}_j\right)_{j'} = \delta_{jj'}$.

We introduce the notation

$$
\begin{aligned}
\underline{\underline{E}}^{jj'} &:= \underline{\underline{C}}_j^\top\ \underline{\underline{E}}\ \underline{\underline{C}}_{j'} \quad j, j' \in I_d \\
\underline{\underline{C}}\left(\underline{n}\right) &:= \sum_{j \in I_d} \left(\underline{n}\right)_j \underline{\underline{C}}_j
\end{aligned}
$$

Note that with the above definitions, we have

$$\begin{aligned}
\underline{\underline{E}}^{jj'} &= \left(\underline{\underline{E}}^{j'j}\right)^{\top} \\
\left(\underline{\underline{E}}^{jj'}\right)_{ss'} &= A_{jsj's'} \\
\underline{\underline{C}}\left(\underline{n}\right)\underline{e}_j &= \underline{\underline{C}}_j\underline{n} \quad \forall j \in I_d
\end{aligned}$$

Modified Variational Formulation

$$\begin{aligned}
B\left(\underline{u},\underline{v}\right) &:= \sum_{l\in I_L}\left(\underline{\underline{D}}\ \underline{v},\underline{\underline{E}}\ \underline{\underline{D}}\ \underline{u}\right)_{K_l} \\
&\quad - \sum_{(l,l')\in F_D}\left(\underline{\underline{C}}\left(\underline{n}_{ll'}\right)^{\top}\underline{\underline{E}}\ \underline{\underline{D}}\left\langle\underline{u}\right\rangle,\left[\underline{v}\right]\right)_{\Gamma_{ll'}} + \tau\sum_{(l,l')\in F_D}\left(\left[\underline{u}\right],\underline{\underline{C}}\left(\underline{n}_{ll'}\right)^{\top}\underline{\underline{E}}\ \underline{\underline{D}}\left\langle\underline{v}\right\rangle\right)_{\Gamma_{ll'}} \\
&\quad + \mu\sum_{(l,l')\in F_D}\frac{\gamma_\mu}{h_{ll'}}\left(\left[\underline{u}\right],\left[\underline{v}\right]\right)_{\Gamma_{ll'}} + \lambda\sum_{(l,l')\in F_D}\frac{\gamma_\lambda}{h_{ll'}}\left(\left[\underline{u}\cdot\underline{n}_{ll'}\right],\left[\underline{v}\cdot\underline{n}_{ll'}\right]\right)_{\Gamma_{ll'}} \\
L\left(\underline{v}\right)(t) &:= \sum_{l\in I_L}\left(\underline{f}\left(t\right),\underline{v}\right)_{K_l} \\
&\quad + \sum_{(l,l')\in G_N}\left(\underline{g}_N\left(t\right),\underline{v}\right)_{\Gamma_{ll'}} + \tau\sum_{(l,l')\in G_D}\left(\underline{g}_D\left(t\right),\underline{\underline{C}}\left(\underline{n}_{ll'}\right)^{\top}\underline{\underline{E}}\ \underline{\underline{D}}\ \underline{v}\right)_{\Gamma_{ll'}} \\
&\quad + \mu\sum_{(l,l')\in G_D}\frac{\gamma_\mu}{h_{ll'}}\left(\underline{g}_D\left(t\right),\underline{v}\right)_{\Gamma_{ll'}} + \lambda\sum_{(l,l')\in G_D}\frac{\gamma_\lambda}{h_{ll'}}\left(\underline{g}_D\cdot\underline{n}_{ll'},\underline{v}\cdot\underline{n}_{ll'}\right)_{\Gamma_{ll'}}
\end{aligned}$$

Find $\underline{u}\in C^2\left(\left[0,T\right],\mathcal{S}_1^d\right)$ such that

$$\begin{aligned}
\rho\ \partial_t^2(\underline{u},\underline{v}) + B\left(\underline{u},\underline{v}\right) &= L\left(\underline{v}\right)(t) \qquad \forall\underline{v}\in\mathcal{S}_1^d,\ t\in[0,T] \\
\partial_t(\underline{u},\underline{v}) &= (\underline{u}_1,\underline{v}) \qquad \forall\underline{v}\in\mathcal{S}_1^d,\ t=0 \\
(\underline{u},\underline{v}) &= (\underline{u}_0,\underline{v}) \qquad \forall\underline{v}\in\mathcal{S}_1^d,\ t=0
\end{aligned}$$

A.2 Vectorial FE Basis

The (scalar) shape functions N are defined in terms of reference element shape functions $\hat{N}$:

$$N_i^l\left(\underline{x}\right) := \hat{N}_i\left(\underline{F}_{K_l}^{-1}\left(\underline{x}\right)\right)$$

For the enumeration of the vectorial shape functions, we introduce the following index conversion formulae:

$$\begin{aligned}
\mathcal{K}\left(i,j\right) &= di+j \quad \forall\left(i,j\right)\in I_{d+1}\times I_d \\
\mathcal{I}\left(k\right) &= \left(k-\left(k \bmod d\right)\right)/d \quad \forall k\in I_r,\ r=d\left(d+1\right) \\
\mathcal{J}\left(k\right) &= k \bmod d \quad \forall k\in I_r \\
\mathcal{K}^{-1}\left(k\right) &= \left(\mathcal{I}\left(k\right),\mathcal{J}\left(k\right)\right)
\end{aligned}$$

Like this, we can define the vectorial shape functions $\underline{N}$ and the global basis functions $\underline{\varphi}$:

$$\begin{aligned}
\underline{N}_k^l &:= N_{\mathcal{I}(k)}^l \underline{e}_{\mathcal{J}(k)} \\
\underline{\varphi}_n\Big|_{K_l} &= \sum_{k \in I_r} \left(\underline{\underline{T}}_l\right)_{kn} \underline{N}_k^l \\
\mathcal{S}_1^d &= \operatorname{span}\left\{\underline{\varphi}_n\right\}_{n \in I_N}, \quad N = d\,(d+1)\,L
\end{aligned}$$

As we do not have to enforce continuity, the matrices $\underline{\underline{T}}_l$ are just permutation matrices representing the bijection between the local index k and the global index n.

A.3 Element Computations

We break all expressions down to scalar shape functions but not to reference element shape functions. To calculate the remaining integrals by quadrature is the safer and the more efficient approach, as many costly and error-prone transformations can be avoided. This specially holds for $d = 3$.

Mass Matrix

$$\begin{aligned}
(\underline{u}, \underline{v}) &= \underline{V}^\top \underline{\underline{M}}\, \underline{U} \\
\underline{\underline{M}} &= \sum_{l \in I_L} \underline{\underline{T}}_l^\top \underline{\underline{M}}_l \underline{\underline{T}}_l \\
\left(\underline{\underline{M}}_l\right)_{kk'} &= \int_{K_l} N_{\mathcal{I}(k)}^l\; N_{\mathcal{I}(k')}^l \mathrm{d}\underline{x}\; \delta_{\mathcal{J}(k)\mathcal{J}(k')}
\end{aligned}$$

Stiffness Matrix

The stiffness matrix is split into a volume contribution and an interfacial contribution:

$$\begin{aligned}
B\,(\underline{u}, \underline{v}) &= \underline{V}^\top \underline{\underline{B}}\, \underline{U} \\
\underline{\underline{B}} &= \underline{\underline{B}}^{vol} + \underline{\underline{B}}^{int} \\
\left(\underline{\underline{B}}^{vol}\right)_{nn'} &= B^{vol}\left(\underline{\varphi}_{n'}, \underline{\varphi}_n\right) \\
\left(\underline{\underline{B}}^{int}\right)_{nn'} &= B^{int}\left(\underline{\varphi}_{n'}, \underline{\varphi}_n\right) \\
B^{vol}\,(\underline{u}, \underline{v}) &:= \sum_{l \in I_L} \left(\underline{\underline{D}}\; \underline{v}, \underline{\underline{E}}\; \underline{\underline{D}}\; \underline{u}\right)_{K_l}
\end{aligned}$$

$$
\begin{aligned}
B^{int}\left(\underline{u},\underline{v}\right) &:= -\sum_{(l,l')\in F_D}\left(\underline{\underline{C}}\left(\underline{n}_{ll'}\right)^{\top}\underline{\underline{E}}\,\underline{\underline{D}}\left\langle\underline{u}\right\rangle,\left[\underline{v}\right]\right)_{\Gamma_{ll'}}\\
&\quad+\tau\sum_{(l,l')\in F_D}\left(\left[\underline{u}\right],\underline{\underline{C}}\left(\underline{n}_{ll'}\right)^{\top}\underline{\underline{E}}\,\underline{\underline{D}}\left\langle\underline{v}\right\rangle\right)_{\Gamma_{ll'}}\\
&\quad+\mu\sum_{(l,l')\in F_D}\frac{\gamma_\mu}{h_{ll'}}\left(\left[\underline{u}\right],\left[\underline{v}\right]\right)_{\Gamma_{ll'}}\\
&\quad+\lambda\sum_{(l,l')\in F_D}\frac{\gamma_\lambda}{h_{ll'}}\left(\left[\underline{u}\cdot\underline{n}_{ll'}\right],\left[\underline{v}\cdot\underline{n}_{ll'}\right]\right)_{\Gamma_{ll'}}
\end{aligned}
$$

Stiffness Matrix: Volume Contribution

$$
\begin{aligned}
\underline{\underline{B}}^{vol} &= \sum_{l\in I_L}\underline{\underline{T}}_l^{\top}\underline{\underline{B}}_l^{vol}\underline{\underline{T}}_l\\
\left(\underline{\underline{B}}_l^{vol}\right)_{kk'} &= \int_{K_l}\left(\underline{\underline{D}}\,\underline{N}_k^l\right)^{\top}\underline{\underline{E}}\,\underline{\underline{D}}\,\underline{N}_{k'}^l\mathrm{d}\underline{x}\\
&= \int_{K_l}\left(\underline{\nabla}N_{\mathcal{I}(k)}^l\right)^{\top}\underline{\underline{C}}_{\mathcal{J}(k)}^{\top}\underline{\underline{E}}\,\underline{\underline{C}}_{\mathcal{J}(k')}\,\underline{\nabla}N_{\mathcal{I}(k')}^l\mathrm{d}\underline{x}\\
&= \int_{K_l}\left(\underline{\nabla}N_{\mathcal{I}(k)}^l\right)^{\top}\underline{\underline{E}}^{\mathcal{J}(k)\mathcal{J}(k')}\underline{\nabla}N_{\mathcal{I}(k')}^l\mathrm{d}\underline{x}
\end{aligned}
$$

Stiffness Matrix: Interfacial Contribution

For the assembly of the interfacial contribution, we do not sum over the set of elements, but over an extended set of element pairs $(l,l')\in\tilde{F}\cup F^*$:

$$
\begin{aligned}
\tilde{F} &:= \left\{(l,l'):(l,l')\in F\text{ or }(l',l)\in F\right\}\\
F^* &:= \left\{(l,l):K_l\in\mathcal{T}\right\}\\
\tilde{F}^l &:= \left\{(m,m')\in\tilde{F}:m=l\right\}\\
G_D^l &:= \left\{(m,m')\in G_D:m=l\right\}\\
G_N^l &:= \left\{(m,m')\in G_N:\ m=l\right\}\\
\underline{\underline{B}}^{int} &= \sum_{(l,l')\in\tilde{F}\cup F^*}\underline{\underline{T}}_l^{\top}\underline{\underline{B}}_{ll'}^{int}\underline{\underline{T}}_{l'}
\end{aligned}
$$

For $(l,l')\in\tilde{F}$:

$$
\left(\underline{\underline{B}}_{ll'}^{int}\right)_{kk'}=b\left(l,l',k,k';l,l',{}^1\!/_2,{}^1\!/_2,-1,-1\right)
$$

We introduce $b\left(\ldots\right)$ in order not to rewrite the similar integrals three times:

$$\begin{aligned}
& b\left(l,l',k,k';m,m',c_1,c_2,c_3,c_4\right) := \\
& \quad c_1 \int_{\Gamma_{ll'}} N^m_{\mathcal{I}(k)} \left(\underline{n}_{ll'}\right)^\top \underline{\underline{E}}^{\mathcal{J}(k)\mathcal{J}(k')} \underline{\nabla} N^{m'}_{\mathcal{I}(k')} \mathrm{d}\underline{s} \\
+ \quad & c_2\ \tau \int_{\Gamma_{ll'}} N^{m'}_{\mathcal{I}(k')} \left(\underline{n}_{ll'}\right)^\top \underline{\underline{E}}^{\mathcal{J}(k')\mathcal{J}(k)} \underline{\nabla} N^{m}_{\mathcal{I}(k)} \mathrm{d}\underline{s} \\
+ \quad & c_3\ \delta_{\mathcal{J}(k)\mathcal{J}(k')} \mu\ \frac{\gamma_\mu}{h_{ll'}} \int_{\Gamma_{ll'}} N^m_{\mathcal{I}(k)}\ N^{m'}_{\mathcal{I}(k')} \mathrm{d}\underline{s} \\
+ \quad & c_4\ \left(\underline{n}_{ll'}\right)_{\mathcal{J}(k)} \left(\underline{n}_{ll'}\right)_{\mathcal{J}(k')} \lambda \frac{\gamma_\lambda}{h_{ll'}} \int_{\Gamma_{ll'}} N^m_{\mathcal{I}(k)}\ N^{m'}_{\mathcal{I}(k')} \mathrm{d}\underline{s}
\end{aligned}$$

For $(l,l) \in F^*$:

$$\underline{\underline{B}}^{int}_{ll} = \sum_{(m,m')\in \tilde{F}^l} \underline{\underline{B}}^{inn}_{mm'} + \sum_{(m,m')\in G^l_D} \underline{\underline{B}}^{bnd}_{mm'}$$

where

$$\left(\underline{\underline{B}}^{inn}_{ll'}\right)_{kk'} = b\left(l,l'k,k';l,l,-{}^1\!/_2,{}^1\!/_2,1,1\right)$$

and

$$\left(\underline{\underline{B}}^{bnd}_{ll'}\right)_{kk'} = b\left(l,l',k,k';l,l,-1,1,1,1\right)$$

Load Vector

$$\begin{aligned}
L\left(\underline{v}\right)(t) &= \underline{V}^\top \underline{L}\left(t\right) \\
\underline{L}\left(t\right) &= \sum_{l\in I_L} \underline{\underline{T}}_l^\top \underline{L}_l\left(t\right) \\
\left(\underline{L}_l\left(t\right)\right)_k &= \int_{K_l} \underline{f}^\top\left(t\right)\ \underline{e}_{\mathcal{J}(k)} N^l_{\mathcal{I}(k)}\ \mathrm{d}\underline{x} \\
&\quad + \sum_{(m,m')\in G^l_N} \int_{\Gamma_{mm'}} \underline{g}_N^\top\left(t\right)\ \underline{e}_{\mathcal{J}(k)} N^m_{\mathcal{I}(k)}\ \mathrm{d}\underline{s} \\
&\quad + \tau \sum_{(m,m')\in G^l_D} \sum_{j\in I_d} \int_{\Gamma_{mm'}} \underline{g}_D\left(t\right)^\top \underline{e}_j \left(\underline{n}_{mm'}\right)^\top \underline{\underline{E}}^{j\mathcal{J}(k)} \underline{\nabla} N^m_{\mathcal{I}(k)} \mathrm{d}\underline{s} \\
&\quad + \mu \sum_{(m,m')\in G^l_D} \frac{\gamma_\mu}{h_{mm'}} \int_{\Gamma_{mm'}} \underline{g}_D\left(t\right)^\top \underline{e}_{\mathcal{J}(k)} N^m_{\mathcal{I}(k)} \mathrm{d}\underline{s} \\
&\quad + \lambda \sum_{(m,m')\in G^l_D} \frac{\gamma_\lambda}{h_{mm'}} \int_{\Gamma_{mm'}} \left(\underline{g}_D\left(t\right)^\top \underline{n}_{mm'}\right) N^m_{\mathcal{I}(k)} \left(\underline{n}^\top_{mm'} \underline{e}_{\mathcal{J}(k)}\right)\ \mathrm{d}\underline{s}
\end{aligned}$$

Initial Conditions

$$
\begin{aligned}
(\underline{u}_i, \underline{v}) &= \underline{V}^\top \underline{U}_i, \quad i \in \{0, 1\} \\
\underline{U}_i &= \sum_{l \in I_L} \underline{\underline{T}}_l^\top \underline{U}_{i,l} \\
\left(\underline{U}_{i,l}\right)_k &= \int_{K_l} \underline{u}_i^\top \underline{e}_{\mathcal{J}(k)} N^l_{\mathcal{I}(k)} \mathrm{d}\underline{x}
\end{aligned}
$$

B Details of Mixed Approach

The FE basis for the space $\mathcal{S}_0$ is

$$\psi_n|_{K_l} = \left(\underline{T_l}\right)_n := \delta_{ln}$$

B.1 Element Computations

Stiffness Matrix

$$a\left(\underline{u},\underline{v}\right) = \underline{V}^\top \underline{\underline{A}}\, \underline{U}$$

The bilinear form $a\left(\underline{u},\underline{v}\right)$ is equivalent to $B\left(\underline{u},\underline{v}\right)$ when setting $\lambda = 0$ and $\gamma_\lambda = 0$. So the respective matrix $\underline{\underline{A}}$ can be assembled the same way as $\underline{\underline{B}}$.

Divergence Matrix

The divergence matrix is split into a volume contribution and an interfacial contribution, like the stiffness matrix:

$$\begin{aligned}
d\left(p,\underline{v}\right) &= \underline{V}^\top \underline{\underline{D}}\, \underline{P} \\
\underline{\underline{D}} &= \underline{\underline{D}}^{vol} + \underline{\underline{D}}^{int} \\
\left(\underline{\underline{D}}^{vol}\right)_{nn'} &= d^{vol}(\psi_{n'}, \underline{\varphi}_n) \\
\left(\underline{\underline{D}}^{int}\right)_{nn'} &= d^{int}(\psi_{n'}, \underline{\varphi}_n) \\
d^{vol}\left(p,\underline{v}\right) &:= -\sum_{l\in I_L} \left(p, \underline{\nabla}\cdot\underline{v}\right)_{K_l} \\
d^{int}\left(p,\underline{v}\right) &:= \sum_{(l,l')\in F_D} \left(\langle p\rangle, [\underline{v}]\cdot\underline{n}_{ll'}\right)_{\Gamma_{ll'}}
\end{aligned}$$

Divergence Matrix: Volume Contribution

$$\begin{aligned}
\underline{\underline{D}}^{vol} &= \sum_{l\in I_L} \underline{\underline{T}}_l^\top \underline{D}_l^{vol} \underline{T}_l^\top \\
\left(\underline{D}_l^{vol}\right)_k &= -\int_{K_l} \underline{e}_{\mathcal{J}(k)} \cdot \underline{\nabla} N^l_{\mathcal{I}(k)} \mathrm{d}\underline{x}
\end{aligned}$$

Divergence Matrix: Interfacial Contribution

For the assembly of the interfacial contribution, we sum over extended sets of element pairs defined in section 2.3.2.

$$\underline{\underline{D}}^{int} = \sum_{(l,l')\in\tilde{F}\cup F^*} \underline{\underline{T}}_l^\top \underline{\underline{D}}_{ll'}^{int} \underline{\underline{T}}_{l'}^\top$$

For $(l,l') \in \tilde{F}$:

$$\left(\underline{\underline{D}}_{ll'}^{int}\right)_k = \frac{1}{2}\int_{\Gamma_{ll'}} \left(\underline{n}_{ll'}\right)_{\mathcal{J}(k)} N^l_{\mathcal{I}(k)} \mathrm{d}\underline{s}$$

For $(l,l) \in F^*$:

$$\underline{\underline{D}}_{ll}^{int} = \sum_{(m,m')\in\tilde{F}^l} \underline{\underline{D}}_{mm'}^{inn} + \sum_{(m,m')\in G_D^l} \underline{\underline{D}}_{mm'}^{bnd}$$

where

$$\left(\underline{\underline{D}}_{ll'}^{inn}\right)_k = \frac{1}{2}\int_{\Gamma_{ll'}} \left(\underline{n}_{ll'}\right)_{\mathcal{J}(k)} N^l_{\mathcal{I}(k)} \mathrm{d}\underline{s}$$

and

$$\left(\underline{\underline{D}}_{ll'}^{bnd}\right)_k = \int_{\Gamma_{ll'}} \left(\underline{n}_{ll'}\right)_{\mathcal{J}(k)} N^l_{\mathcal{I}(k)} \mathrm{d}\underline{s}$$

Compression Matrix

$$\begin{aligned} c\left(p,q\right) &= \underline{Q}^\top \underline{\underline{C}}\, \underline{P} \\ \underline{\underline{C}} &= \sum_{l\in I_L} \underline{\underline{T}}_l\, C_l\, \underline{\underline{T}}_l^\top \\ C_l &= \int_{K_l} \frac{1}{\lambda}\, \mathrm{d}\underline{x} = \frac{|K_l|}{\lambda} \end{aligned}$$

Note that the compression matrix is diagonal as the mass matrix.

Load Vector: Velocity Part

$$L_{\underline{v}}\left(\underline{v}\right)\ (t) = \underline{V}^\top \underline{L}_{\underline{v}}(t)$$

The linear form $L_{\underline{v}}\left(\underline{v}\right)$ is equivalent to $L\left(\underline{v}\right)$ when setting $\lambda = 0$ and $\gamma_\lambda = 0$. So the velocity part of the load vector $\underline{L}_{\underline{v}}$ can be assembled the same way as $\underline{L}$.

Load Vector: Pressure Parameter Part

$$
\begin{aligned}
L_q\left(q\right)\ \left(t\right) &= \underline{\underline{Q}}^{\top}\underline{L}_q\left(t\right) \\
\underline{L}_q\left(t\right) &= \sum_{l\in I_L}\underline{\underline{T}}_l L_q^l\left(t\right) \\
L_q^l\left(t\right) &= \sum_{(m,m')\in G_D^l}\int_{\Gamma_{mm'}}\underline{g}_D\left(t\right)\cdot\underline{n}_{mm'}\mathrm{d}\underline{s}
\end{aligned}
$$

Initial Conditions

The initial conditions for $\underline{u}$ are the same as in the non-mixed formulation. Those for the pressure parameter read:

$$
\begin{aligned}
\left(p/\lambda, q\right) &= c\left(p,q\right) = \underline{\underline{Q}}^{\top}\underline{\underline{C}}\ \underline{P} \\
\left(-\underline{\nabla}\cdot\underline{u}_i, q\right) &= \underline{\underline{Q}}^{\top}\underline{P}_i \\
\underline{P}_i &= \sum_{l\in I_L}\underline{\underline{T}}_l P_i^l \\
P_i^l &= -\int_{K_l}\underline{\nabla}\cdot\underline{u}_i\ \mathrm{d}\underline{x}
\end{aligned}
$$

Bibliography

[1] D. Braess: Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastitzitätstheorie. Springer, Berlin, 1997

[2] J. H. Bramble, J. E. Pasciak: A Preconditioning Technique for Indefinite Systems Resulting from Mixed Approximations of Elliptic Problems. Mathematics of Computation, Vol. 50, pp. 1-18, 1998

[3] P. Frauenfelder, C. Lage: Concepts – An Object-Oriented Software Package for Partial Differential Equations. Research Report No. 2002-09, Seminar für Angewandte Mathematik ETH Zürich, 2002.

[4] J. Gopalakrishnan, G. Kanschat: A Multilevel Discontinuous Galerkin Method. Numer. Math. Vol. 95 no. 3, pp. 527-550, 2003

[5] E. Hairer, S. P. Nørsett, G. Wanner: Solving Ordinary Differential Equations I - Nonstiff Problems. Springer, Berlin, 1992

[6] P. Hansbo, M. G. Larson: Discontinuous Galerkin methods for incompressible and nearly incompressible elasticity by Nitsche's method. Comput. Methods Appl.. Mech. Engrg., 191(17-18), pp 1895-1908, 2002

[7] T. J. R. Hughes: The Finite Element Method - Linear Static and Dynamic Finite Element Analysis. Dover, Mineola, 2000

[8] P.-A. Raviart, J.-M. Thomas: Introduction à l' analyse numérique des équations aux dérivées partielles. Dunod, Paris, 1998

[9] J. Schöberl: NETGEN, `http://www.hpfem.jku.at/netgen/`

[10] T. P. Wihler: Discontinuous Galerkin FEM for Elliptic Problems in Polygonal Domains. Ph.D. Thesis, ETH Zürich, No. 14973, 2003
`http://e-collection.ethbib.ethz.ch/show?type=diss&nr=14973`

[11] T. P. Wihler: Locking-free DGFEM for elasticity problems in polygons. IMA Journal of Numerical Analysis, Vol. 14(1), pp. 45-75, 2004