ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Efficient preconditioners for a shock capturing space-time discontinuous Galerkin method for systems of conservation laws

A. Hiltebrand and S. Mishra

Research Report No. 2014-04
February 2014

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Funding ERC: 306279 SPARCCLE

EFFICIENT PRECONDITIONERS FOR A SHOCK CAPTURING SPACE-TIME DISCONTINUOUS GALERKIN METHOD FOR SYSTEMS OF CONSERVATION LAWS.

A. HILTEBRAND AND S. MISHRA

Abstract. An entropy stable fully discrete shock capturing space-time Discontinuous Galerkin (DG) method was proposed in a recent paper [20] to approximate hyperbolic systems of conservation laws. This numerical scheme involves the solution of a very large nonlinear system of algebraic equations, by a Newton-Krylov method, at every time step. In this paper, we design efficient preconditioners for the large, non-symmetric linear system, that needs to be solved at every Newton step. Two sets of preconditioners, one of the block Jacobi and another of the block Gauss-Seidel type are designed. Fourier analysis of the preconditioners reveals their robustness and a large number of numerical experiments are presented to illustrate the gain in efficiency that results from preconditioning. The resulting method is employed to compute approximate solutions of the compressible Euler equations, even for very high CFL numbers.

1. Introduction

Hyperbolic systems of conservation laws are systems of nonlinear partial differential equations that model many interesting phenomena in physics and engineering. Examples include the shallow water equations of oceanography, the compressible Euler equations of aerodynamics, the magnetohydrodynamics (MHD) equations of plasma physics and the equations of nonlinear elastodynamics [6].

It is well known that solutions of systems of conservation laws can form discontinuities such as *shock waves*, even when the initial data is smooth. Hence, the solutions of systems of conservation laws is interpreted in the *weak* (distributional) sense. These weak solutions are not necessarily unique. Further admissibility criteria in the form of *entropy* conditions need to be imposed in order to guarantee uniqueness. Detailed formulation of entropy solutions is provided in Section 2 and in standard textbooks such as [6]. But in fact, recent work [14, 11] suggests that the notion of solutions has to be further weakened into the more general *entropy measure valued solutions* in order to obtain wellposedness for multi-dimensional systems of conservation laws. Measure valued solutions are space-time parametrized probability measures and are shown to be natural limits of numerical approximations [11].

1.1. Numerical schemes.

Given the nonlinear nature of systems of conservation laws, it is not possible to obtain explicit solution formulas. Consequently, numerical methods play a key role in the study of these equations. Various numerical methods of the finite difference, finite volume, finite element and spectral type are available for the approximation of systems of conservation laws. In particular, the finite volume (difference) methods, that update cell averages (point values) in terms of numerical fluxes, are heavily used [28]. The numerical fluxes are obtained by using exact or approximate Riemann solvers. Higher-order spatial accuracy results from non-oscillatory piecewise polynomial reconstruction in each cell. Reconstruction procedures such as TVD [28], ENO [17] and WENO [30] are typically employed. Higher order temporal accuracy is achieved by using strong stability preserving (SSP) Runge-Kutta (RK) time integrators. An alternative to high-order finite volume methods is the discontinuous Galerkin finite element method [4, 5]. At lowest (first) order, these methods reduce to the standard finite volume method. However, high-order accuracy is obtained by using piecewise polynomial test and trial functions in each element. Limiters are employed to damp oscillations near shocks. Temporal accuracy is again increased by using SSP-RK methods. High-order finite volume methods

Date: February 14, 2014.

1991 *Mathematics Subject Classification.* 65M60,35L65.

The research of both AH and SM was funded in part by ERC STG. NN 306279 SPARCCLE.

and RKDG methods have been very successful in carrying out realistic large scale simulations of conservation laws [31].

The key questions in the numerical analysis of systems of conservation laws are that of *stability* and *convergence* of numerical schemes [15]. These questions have been carefully investigated in the simple case of scalar conservation laws. For this class of problems, first-order monotone schemes [15] satisfy a discrete maximum principle, a discrete form of the entropy inequality as well as the TVD property. Hence, they can be shown to converge to the entropy solution. Similar results have also been derived for higher order schemes, see [12] and references therein. However, the questions of stability and convergence are much harder to tackle for systems of conservation laws. For such equations, *entropy stability*, i.e., compliance with a discrete form of the entropy inequality, seems to be a reasonable stability requirement for numerical schemes [32]. Entropy stable finite difference schemes were developed in the pioneering papers of Tadmor [32, 33], see [9, 24] for recent advances. More recently, arbitrarily high-order entropy stable finite difference schemes were proposed in [10]. Furthermore, these *TeCNO* schemes were also shown to converge to the appropriate notion of *entropy measure valued solutions* for systems of conservation laws [12, 11].

Although these entropy stable finite difference schemes have nice theoretical properties and are shown to be quite efficient computationally [10, 11], they are still deficient in the following manner:

i. The entropy stable schemes of [32, 33, 10] are semi-discrete. The time integration is performed using SSP Runge-Kutta methods and the resulting fully discrete scheme may not inherit the entropy stability and convergence properties of the semi-discrete scheme.
ii. The TeCNO schemes are restricted to Cartesian grids in several space dimensions. Some first-order entropy stable schemes are available for unstructured grids such as those in [33, 24] and references therein. However, it is unclear if these schemes can be extended to arbitrarily high-order of accuracy.

Given the above deficiencies, the existing entropy stable finite difference schemes are not suitable for the approximation of multi-dimensional systems on domains with complex geometry.

1.2. **Space-time DG methods.** The above discussion illustrates the need for a numerical method that is

(a.) (Formally) arbitrarily high-order accurate.
(b.) Entropy stable.
(c.) Convergent to entropy solutions of scalar conservation laws as well as entropy measure valued solutions of systems of conservation laws.
(c.) Fully discrete in both space and time.
(d.) Able to approximate problems with complicated domain geometry such as those discretized with unstructured grids.

It turns out that suitable space-time discontinuous Galerkin (DG) methods satisfy all the above requirements. These finite element methods, based on piecewise polynomial basis functions without any continuity conditions across elements, were considered in [26, 27, 25] and references therein, see also [23]. A more modern take on these methods with suitable shock capturing and streamline diffusion operators was provided by Barth [1] and references therein. In a recent paper [20], we considered a variant of this method and proved it to be entropy stable and convergent to entropy measure valued solutions for systems of conservation laws. Convergence to entropy solutions of scalar conservation laws is considered in a forthcoming paper [19].

1.3. **Aims and scope of the current paper.** The shock capturing space-time DG method of [20] is based on a judicious combination of the use of entropy variables as degrees of freedom, suitable entropy stable spatial numerical fluxes, streamline diffusion operators as well as a residual based shock capturing term. The method is *implicit* in time and involves the solution of a large *nonlinear* system of algebraic equations, at every time step. This nonlinear system is solved in [20] and references therein (see also Section 3), using a damped Newton method. In turn, the Newton solver requires the solution of a linear system at every Newton sub-step. This linear system is *very large, sparse and non-symmetric*. Using direct solvers is out of the question on account of the large system size. Hence, we need to use suitable iterative Krylov type linear system solvers. The design of such solvers is quite complicated as the Jacobian is not well-conditioned. Hence, we need to employ suitable *preconditioners* to keep the number of Krylov iterations, per Newton step, reasonable. In fact, *the design of*

suitable preconditioners is perhaps the single most important step in enhancing the computational efficiency of the space-time DG method. The description of such preconditioners is the main aim of the current paper.

In the current paper,

- We describe the shock capturing space-time DG method of [20] and highlight the need for preconditioning the resulting linear system at each Newton sub-step of every time step.
- We design suitable preconditioners of the block Jacobi and block Gauss-Seidel type for the resulting linear system.
- We will analyze these preconditioners in order to explain why they work for a large range of mesh sizes as well as time step sizes.
- We present a large set of numerical experiments to illustrate the efficiency of the preconditioners and to demonstrate the robustness of the space-time DG method in approximating systems of conservation laws.

To this end, the rest of the paper is organized as follows. In Section 2, we describe the continuous problem for systems of conservation laws and define the notions of entropy solutions as well as entropy measure valued solutions. The shock capturing space-time DG method of [20] is presented in Section 3 and different computational aspects of the method are highlighted in Section 4. In Section 5, we describe the block Jacobi preconditioners. The block Gauss-Seidel preconditioners are described in Section 6.

2. The continuous problem

The generic form of the Cauchy problem for a system of conservation laws is

$$\begin{aligned} \mathbf{U}_t + \sum_{k=1}^{d} \mathbf{F}^k(\mathbf{U})_{x_k} = 0, \quad (x,t) \in \Omega \times \mathbb{R}_+, \\ \mathbf{U}(x,0) = \mathbf{U}_0(x), \quad x \in \Omega. \end{aligned} \tag{2.1}$$

Here, $\mathbb{R}_+ = [0,\infty)$, $\Omega \subset \mathbb{R}^d$ $(d = 1,2,3)$ is a spatial domain and $\mathbf{U} : \Omega \mapsto \mathbb{R}^m$ is the vector of unknowns. $\mathbf{F}^k$ is the (smooth) flux vector in the k-th direction.

2.1. Entropy solutions.

The system of conservation laws (2.1) is termed *hyperbolic* if the directional Jacobians $\sum_{k=1}^{d} \partial_{\mathbf{U}}(\mathbf{F}^k \nu_k)$, have real eigenvalues and an eigenbasis for every normal direction ν. As mentioned in the introduction, the presence of discontinuities such as shock waves imply that the solutions of (2.1) are interpreted in the following *weak* sense, i.e., for every compactly supported test function $\boldsymbol{\varphi} \in (C_c^\infty(\Omega \times \mathbb{R}_+))^m$, $\mathbf{U} \in \left(L^1(\Omega \times \mathbb{R}_+)\right)^m$ is said to be a weak solution of (2.1) if the following integral identity is satisfied:

$$\int_{\mathbb{R}_+} \int_{\Omega} \left(\langle \mathbf{U}, \boldsymbol{\varphi}_t \rangle + \sum_{d=1}^{k} \langle \mathbf{F}^k(\mathbf{U}), \boldsymbol{\varphi}_{x_k} \rangle \right) dx dt + \int_{\Omega} \langle \mathbf{U}(x,0), \boldsymbol{\varphi}(x,0) \rangle dx = 0. \tag{2.2}$$

Here, we have implicitly ignored boundary conditions by considering test functions that are compactly supported in the spatial domain Ω.

Weak solutions are not necessarily unique. Additional admissibility criteria or *entropy conditions* need to be imposed in order to single out the physically relevant solutions of (2.1). We assume that there exists a strictly convex entropy function S and entropy flux functions Q^k such that the following compatibility conditions are satisfied,

$$\partial_{\mathbf{U}} Q^k = \langle \mathbf{V}, \partial_{\mathbf{U}} \mathbf{F}^k \rangle, \quad \forall k = 1, 2, \cdots, d. \tag{2.3}$$

Here, $\mathbf{V} = \partial_{\mathbf{U}} S$ is termed as the vector of entropy variables. The weak solution $\mathbf{U}$ of (2.1) is said to be an entropy solution [6] if it satisfies the entropy inequality,

$$S_t + \sum_{k=1}^{d} Q^k_{x_k} \leq 0, \tag{2.4}$$

in the sense of distributions.

Integrating the entropy inequality (2.4) in space results in the estimate,

$$\frac{d}{dt}\int_\Omega S(\mathbf{U}(x,t))dx \leq 0. \tag{2.5}$$

This bound on the total entropy, together with the strict convexity of the entropy function, yields an L^2 stability estimate for the entropy solution $\mathbf{U}$ [6]. This "energy" estimate is currently the only available generic global a priori estimate for systems of conservation laws [6].

Furthermore, as the entropy S is strictly convex, the conservative variables $\mathbf{U}$ and the entropy variables $\mathbf{V}$ are one to one [6]. Consequently, the conservation law (2.1) can be recast in terms of entropy variables as,

$$\mathbf{U}(\mathbf{V})_t + \sum_{k=1}^{d} \mathbf{F}^k(\mathbf{V})_{x_k} = 0, \quad (x,t) \in \Omega \times \mathbb{R}_+, \tag{2.6}$$

Here, we have used the change of variable $\mathbf{U} = \mathbf{U}(\mathbf{V})$ and retained the notation $\mathbf{F}^k(\mathbf{V}) = \mathbf{F}^k(\mathbf{U}(\mathbf{V}))$ for all k for notational convenience. When convenient, we will work with the equivalent representation (2.6) of the conservation law (2.1)

2.2. **Entropy measure valued solutions.** Wellposedness of entropy solutions (weak solutions that satisfy the entropy inequality (2.4)) is only available for scalar conservation laws (in several space dimensions) and for one-dimensional systems [6]. No generic global existence and uniqueness results for entropy solutions of multi-dimensional systems are currently available. In fact, non-uniqueness of entropy solutions for some specific multi-dimensional systems have been established recently in [3] and references therein. Furthermore, it was recently established in [11] (see also [14]) that state of the art numerical schemes such as the entropy stable TeCNO schemes of [10] may not even converge to an entropy solution as the mesh is refined. Thus, one needs to consider a more general concept of solutions of (2.1) in order to establish existence and stability of solutions and in order to realize the limit of stable numerical approximations. A suitable notion of solutions is provided by the framework of entropy measure valued solutions [11]. As first suggested by DiPerna in [7], we consider a probability (non-negative with unit mass) measure $\boldsymbol{\mu}$, realized as a map:

$$\boldsymbol{\mu} : (x,t) \in \Omega \times \mathbb{R}_+ \mapsto Prob(\mathbb{R}^m),$$

for each x, t and define it as a measure valued solution of the system (2.1) (based on the equivalent representation (2.6)) if it satisfies,

$$\int_\Omega \int_{\mathbb{R}_+} \left(\langle \mathbf{U}, \boldsymbol{\mu}_{x,t} \rangle \boldsymbol{\varphi}_t + \sum_{k=1}^{d} \langle \mathbf{F}^k, \boldsymbol{\mu}_{x,t} \rangle \boldsymbol{\varphi}_{x_k} \right) dxdt = 0, \tag{2.7}$$

for all test functions $\boldsymbol{\varphi} \in (C_c^\infty(\Omega \times (0,\infty)))^m$. Here,

$$\langle \mathbf{g}, \boldsymbol{\mu}_{x,t} \rangle = \int_{\mathbb{R}^m} \mathbf{g}(\lambda) d\boldsymbol{\mu}_{x,t}(\lambda).$$

Note that we realize the Young measure in terms of the entropy variables $\mathbf{V}$. The corresponding Young measure for the conservative variables $\mathbf{U}$ can be realized as $\mathbf{U}(\boldsymbol{\mu})$, using the one-one mapping $\mathbf{U}(\mathbf{V})$.

Furthermore, as the system (2.1) is equipped with an entropy function S and entropy flux functions Q^k for $k = 1, 2, \cdots, d$, then $\boldsymbol{\mu}$ is defined to be an entropy measure valued solution of (2.1) if it is a measure valued solution as well as it satisfies,

$$\int_\Omega \int_{\mathbb{R}_+} \left(\langle S, \boldsymbol{\mu}_{x,t} \rangle \varphi_t + \sum_{k=1}^{d} \langle Q^k, \boldsymbol{\mu}_{x,t} \rangle \varphi_{x_k} \right) dxdt \geq 0, \tag{2.8}$$

for all non-negative test functions $0 \leq \varphi \in C_c^\infty(\Omega \times (0,\infty))$

It is well known that measure valued solutions reduce to the standard notion of weak solutions of (2.1) if the measure-valued solution is atomic, i.e., $\boldsymbol{\mu} = \delta_{\mathbf{V}(x,t)}$ [7]. However, examples in the recent paper [11] do suggest that entropy measure valued solutions are not necessarily atomic.

3. The shock capturing space-time DG method

As mentioned in the introduction, we seek a numerical method that is entropy stable, i.e., it satisfies a discrete form of the entropy inequality (2.4) and it converges to an entropy measure valued solution (as defined above) of the system (2.1) as the mesh is refined. The shock capturing space-time DG method of [20] is one such numerical method. We describe this method below.

3.1. **The mesh.** At the n-th time level t^n, we denote the time step as Δt^n and the update time interval as $I^n = [t^n, t^{n+1})$ and $t^{n+1} - t^n = \Delta t^n$. For simplicity, we assume that the spatial domain $\Omega \subset \mathbb{R}^d$ is polyhedral and divide it into a triangulation $\mathcal{T}$, i.e., a set of open convex polyhedra $K \subset \mathbb{R}^d$ with plane faces. Furthermore, we assume mesh regularity [25]. For a generic element (cell) K, we denote

$$\begin{aligned}\Delta x_K &= diam(K),\\ \mathcal{N}(K) &= \{K' \in \mathcal{T} : K' \neq K \wedge \text{meas}_{d-1}(\overline{K} \cap \overline{K'}) > 0\}.\end{aligned}$$

The *mesh width* of the triangulation is $\Delta x(\mathcal{T}) = \max_K \Delta x_K$. A generic space-time element is the prism:

$$K \times I^n.$$

We also assume that there exists a constant $C > 0$ such that $(1/C)\Delta x \leq \Delta t^n \leq C\Delta x$ for all time levels n.

3.2. **Variational formulation.** Following [33, 1, 20], we approximate the equivalent representation of the conservation law (2.6) by a DG method. On a given triangulation $\mathcal{T}$ with mesh width Δx, we seek entropy variables

$$\begin{aligned}\mathbf{V}^{\Delta x} \in \mathcal{V}_p &= (\mathbb{P}_p(\Omega \times [0,T]))^m\\ &= \{\mathbf{W} \in \left(L^1(\Omega \times [0,T])\right)^m : \mathbf{W}|_{K\times I^n} \quad \text{is a polynomial of degree } p \text{ in each component}\}\end{aligned} \tag{3.1}$$

such that the following quasilinear variational form is satisfied for each $\mathbf{W}^{\Delta x} \in \mathcal{V}_p$:

$$\mathcal{B}(\mathbf{V}^{\Delta x}, \mathbf{W}^{\Delta x}) = \mathcal{B}_{DG}(\mathbf{V}^{\Delta x}, \mathbf{W}^{\Delta x}) + \mathcal{B}_{SD}(\mathbf{V}^{\Delta x}, \mathbf{W}^{\Delta x}) + \mathcal{B}_{SC}(\mathbf{V}^{\Delta x}, \mathbf{W}^{\Delta x}) = 0. \tag{3.2}$$

We elaborate on each of the three quasilinear (nonlinear in the first argument and linear in the second) forms in the following.

3.3. **The DG quasilinear form.** The form $\mathcal{B}_{DG}$ is given by,

(3.3)

$$\begin{aligned}\mathcal{B}_{DG}(\mathbf{V}^{\Delta x}, \mathbf{W}^{\Delta x}) = &-\sum_n \sum_K \int_{I^n}\int_K \left(\left\langle \mathbf{U}(\mathbf{V}^{\Delta x}), \mathbf{W}_t^{\Delta x}\right\rangle + \sum_{k=1}^d \left\langle \mathbf{F}^k(\mathbf{V}^{\Delta x}), \mathbf{W}_{x_k}^{\Delta x}\right\rangle\right) dxdt\\ &+ \sum_n \sum_K \int_K \left\langle \mathbb{U}(\mathbf{V}_{n+1,-}^{\Delta x}, \mathbf{V}_{n+1,+}^{\Delta x}), \mathbf{W}_{n+1,-}^{\Delta x}\right\rangle dx - \sum_n \sum_K \int_K \left\langle \mathbb{U}(\mathbf{V}_{n,-}^{\Delta x}, \mathbf{V}_{n,+}^{\Delta x}), \mathbf{W}_{n,+}^{\Delta x}\right\rangle dx\\ &+ \sum_n \sum_K \sum_{K' \in \mathcal{N}(K)} \int_{I^n}\int_{\partial_{KK'}} \left(\sum_{k=1}^d \left\langle \mathbb{F}^{k,*}(\mathbf{V}_{K,-}^{\Delta x}, \mathbf{V}_{K,+}^{\Delta x}), \mathbf{W}_{K,-}^{\Delta x}\right\rangle \nu_{KK'}^k\right) d\sigma(x)dt\\ &- \frac{1}{2}\sum_n \sum_K \sum_{K' \in \mathcal{N}(K)} \int_{I^n}\int_{\partial_{KK'}} \left\langle \mathbf{W}_{K,-}^{\Delta x}, \mathbf{D}(\mathbf{V}_{K,+}^{\Delta x} - \mathbf{V}_{K,-}^{\Delta x})\right\rangle d\sigma(x)dt\end{aligned}$$

Here we have employed the notation,

$$\begin{aligned}\mathbf{W}_{n,\pm}(x) &= \mathbf{W}(x, t_\pm^n),\\ \partial_{KK'} &= \overline{K} \cap \overline{K'},\\ \nu_{KK'} &= \text{Unit normal for edge KK}' \text{ pointing outwards from element K.}\\ \mathbf{W}_{K,\pm}(x,t) &= \lim_{h\to 0} \mathbf{W}(x \pm h\nu, t), \quad \forall x \in \partial_{KK'},\\ \mathbf{D} &= \mathbf{D}(\mathbf{V}_{K,-}^{\Delta x}, \mathbf{V}_{K,+}^{\Delta x}; \nu_{KK'})\end{aligned}$$

for all $\mathbf{W} \in \mathcal{V}_p$. We remark that the boundary condition is ignored in the above variational form by considering compactly supported (in the spatial domain) solutions and test functions.

3.3.1. *Numerical fluxes.* Both the temporal and spatial numerical fluxes, need to be specified to complete the description of the DG quasilinear form. In order to obtain causality (marching) after each time step, we choose the temporal numerical flux to be the *upwind* flux:

$$\mathbb{U}(a,b) = \mathbf{U}(a). \tag{3.4}$$

This ensures that we can use the values at the previous time step in order to compute an update at the time level t^n. Any other choice of temporal numerical fluxes will imply that all the degrees of freedom (for all times) are coupled and force us to solve a very large non-linear algebraic system of equations.

The spatial numerical flux consists of the following two components,

3.3.2. *Entropy conservative flux:* The entropy conservative flux (in the k-th direction) is any flux [32] that satisfies the relation:

$$\langle b-a, \mathbb{F}^{k,*}(a,b)\rangle = \Psi^k(b) - \Psi^k(a). \tag{3.5}$$

Here, $\Psi^k = \langle \mathbf{V}, \mathbf{F}^k\rangle - Q^k$ is the entropy potential. The existence of such fluxes (for any generic conservation law with an entropy framework) was shown by Tadmor in [32]. More recently, explicit expressions of entropy conservative fluxes for specific systems of interest like the shallow water equations [9] and Euler equations [24] have been obtained.

3.3.3. *Numerical diffusion operators:* Following [33, 9, 10], we choose the numerical diffusion operator as,

$$\mathbf{D}(a,b;\nu) = \mathbf{R}_\nu \mathbf{P}(\Lambda_\nu(\cdot);a,b)\mathbf{R}_\nu^\top. \tag{3.6}$$

Here, $\Lambda_\nu, \mathbf{R}_\nu$ are the eigenvalue and eigenvector matrices of the Jacobian $\partial_{\mathbf{U}}(\langle \mathbf{F}, \nu\rangle)$ in the normal direction ν. $\mathbf{R}_\nu$ is evaluated at an averaged state, e.g. $(a+b)/2$, and scaled such that $\mathbf{R}_\nu \mathbf{R}_\nu^\top = \mathbf{U}_\mathbf{V}$. $\mathbf{P}$ is a non-negative matrix function. Examples of $\mathbf{P}$ include $\mathbf{P}(\Lambda_\nu(\cdot);a,b) = |\Lambda_\nu(\frac{a+b}{2})|$ which leads to a Roe type scheme and $\mathbf{P}(\Lambda_\nu(\cdot);a,b) = \max\{\lambda_{max}(a;\nu), \lambda_{max}(b;\nu)\}\mathbf{ID}$, which leads to a Rusanov type scheme [10], where $\lambda_{max}(\mathbf{U};\nu)$ is the maximal wave speed in direction of ν, i.e., $\lambda_{max}(\mathbf{U};\nu)$ is the spectral radius of $\Lambda_\nu(\mathbf{U})$.

3.4. **Streamline diffusion operator.** There is no numerical diffusion in the interior of the space-time element $K \times I^n$. In order to suppress the resulting unphysical oscillations near shocks, we choose the following streamline diffusion operator,

$$\mathcal{B}_{SD}(\mathbf{V}^{\Delta x}, \mathbf{W}^{\Delta x}) = \sum_n \sum_K \int_{I^n}\int_K \left\langle \left(\mathbf{U}_\mathbf{V}(\mathbf{V}^{\Delta x})\mathbf{W}_t^{\Delta x} + \sum_{k=1}^d \mathbf{F}_\mathbf{V}^k(\mathbf{V}^{\Delta x})\mathbf{W}_{x_k}^{\Delta x}\right), \mathbf{D}^{SD}\mathrm{Res}\right\rangle dxdt \tag{3.7}$$

with intra-element residual:

$$\mathrm{Res} = \mathbf{U}(\mathbf{V}^{\Delta x})_t + \sum_{k=1}^d \mathbf{F}^k(\mathbf{V}^{\Delta x})_{x_k}, \tag{3.8}$$

and the scaling matrix is chosen as

$$\mathbf{D}^{SD} = C^{SD}\Delta t^n \mathbf{U}_\mathbf{V}^{-1}(\mathbf{V}^{\Delta x}), \tag{3.9}$$

for any positive constant C^{SD}. Note that the intra-element residual is well defined as we are taking first-derivatives of a polynomial function.

3.5. **Shock capturing operator.** The streamline diffusion operator adds numerical diffusion in the direction of the streamlines. However, we need further numerical diffusion in order to reduce possible oscillations at shocks. We use the following shock capturing operator (similar to Barth [1]):

(3.10a)

$$\mathcal{B}_{SC}(\mathbf{V}^{\Delta x},\mathbf{W}^{\Delta x})=\sum_{n,K}\int_{I^n}\int_K D_{n,K}^{SC}\left(\frac{(\Delta t^n)^2}{\Delta x^2}\left\langle \mathbf{W}_t^{\Delta x},\mathbf{U}_{\mathbf{V}}(\tilde{\mathbf{V}}_{n,K})\mathbf{V}_t^{\Delta x}\right\rangle+\sum_{k=1}^{d}\left\langle \mathbf{W}_{x_k}^{\Delta x},\mathbf{U}_{\mathbf{V}}(\tilde{\mathbf{V}}_{n,K})\mathbf{V}_{x_k}^{\Delta x}\right\rangle\right)dxdt,$$

with

$$\tilde{\mathbf{V}}_{n,K}=\frac{1}{\text{meas}(I^n\times K)}\int_{I^n}\int_K \mathbf{V}^{\Delta x}(x,t)dxdt.$$

being the cell average and the scaling factor,

$$D_{n,K}^{SC}=\frac{\Delta x C^{SC}\overline{\text{Res}}_{n,K}+\Delta x(\Delta t^n)^{-1/2}\bar{C}^{SC}\overline{\text{BRes}}_{n,K}}{\sqrt{\int_{I^n}\int_K\left(\frac{(\Delta t^n)^2}{\Delta x^2}\left\langle \mathbf{V}_t^{\Delta x},\mathbf{U}_{\mathbf{V}}(\tilde{\mathbf{V}}_{n,K})\mathbf{V}_t^{\Delta x}\right\rangle+\sum_{k=1}^{d}\left\langle \mathbf{V}_{x_k}^{\Delta x},\mathbf{U}_{\mathbf{V}}(\tilde{\mathbf{V}}_{n,K})\mathbf{V}_{x_k}^{\Delta x}\right\rangle\right)dxdt+\epsilon}},\tag{3.10b}$$

with $\epsilon=\Delta x^{\frac{d}{2}-1}(\Delta t^n)^{\frac{1}{2}}\left(\frac{\Delta x}{\text{diam}(\Omega)}\right)^{\theta}$, $\theta\geq 1/2$ (chosen as 1) and

$$\overline{\text{Res}}_{n,K}=\sqrt{\int_{I^n}\int_K\left\langle \text{Res},\mathbf{U}_{\mathbf{V}}^{-1}(\mathbf{V}^{\Delta x})\text{Res}\right\rangle dxdt},\tag{3.10c}$$

(3.10d)

$$\overline{\text{BRes}}_{n,K}=\left(\int_K\|\mathbf{U}(\mathbf{V}_{n,-}^{\Delta x})-\mathbf{U}(\mathbf{V}_{n,+}^{\Delta x})\|^2_{\mathbf{U}_{\mathbf{V}}^{-1}(\mathbf{V}_{n,+}^{\Delta x})}dx\right.$$

$$\left.+\sum_{K'\in\mathcal{N}(K)}\int_{I^n}\int_{\partial_{KK'}}\frac{\Delta t^n}{\Delta x}\left\|\sum_{k=1}^{d}\left(\mathbb{F}^{k,*}(\mathbf{V}_{K,-}^{\Delta x},\mathbf{V}_{K,+}^{\Delta x})-\mathbf{F}^k(\mathbf{V}_{K,-}^{\Delta x})\right)\nu_{KK'}^k-\frac{1}{2}\mathbf{D}(\mathbf{V}_{K,+}^{\Delta x}-\mathbf{V}_{K,-}^{\Delta x})\right\|^2_{\mathbf{U}_{\mathbf{V}}^{-1}(\mathbf{V}_{K,-}^{\Delta x})}d\sigma(x)dt\right)^{\frac{1}{2}}.$$

Here, $C^{SC},\bar{C}^{SC}$ are positive constants and for a symmetric positive definite matrix $\mathbf{A}\in\mathbb{R}^{m\times m}$ and $\mathbf{a}\in\mathbb{R}^m$ we denote $\|\mathbf{a}\|_{\mathbf{A}}=\sqrt{\langle\mathbf{a},\mathbf{A}\mathbf{a}\rangle}$.

3.6. **Analysis of the space-time DG method.** The streamline diffusion shock-capturing space-time DG method (3.2) was analyzed in detail in the recent paper [20] and the following theorems were proved.

Theorem 3.1. *[ENTROPY STABILITY]. Consider the system of conservation laws* (2.1) *with strictly convex entropy function S and entropy flux functions $Q^k_{(1\leq k\leq d)}$. For simplicity, assume that the exact and approximate solutions have compact support inside the spatial domain Ω. Let the final time be denoted by t_-^N. Then, the streamline diffusion-shock capturing-Discontinuous Galerkin scheme* (3.2) *approximating* (2.1) *has the following properties:*

(i.) The scheme (3.2) *is conservative, i.e., the approximate solutions $\mathbf{U}^{\Delta x}=\mathbf{U}(\mathbf{V}^{\Delta x})$ satisfy*

$$\int_\Omega \mathbf{U}^{\Delta x}(x,t_-^N)dx=\int_\Omega \mathbf{U}^{\Delta x}(x,t_-^0)dx.\tag{3.11}$$

(ii.) The scheme (3.2) *is entropy stable, i.e., the approximate solutions satisfy,*

$$\int_\Omega S(\mathbf{U}^*(t_-^0))dx\leq\int_\Omega S(\mathbf{U}^{\Delta x}(x,t_-^N))dx\leq\int_\Omega S(\mathbf{U}^{\Delta x}(x,t_-^0))dx,\tag{3.12}$$

with $\mathbf{U}^*$ being the domain average:

$$\mathbf{U}^*(t_-^0)=\frac{1}{\text{meas}(\Omega)}\int_\Omega\mathbf{U}(\mathbf{V}(x,t_-^0))dx.$$

Theorem 3.2. *[CONVERGENCE.] Let* $\mathbf{U}^{\Delta x} = \mathbf{U}(\mathbf{V}^{\Delta x})$ *be the approximate solutions of the system* (2.1) *generated by the streamline diffusion-shock capturing DG scheme* (3.2)*. Under the assumption that the approximation solutions satisfy the uniform* L^∞ *bound,*

$$\|\mathbf{V}^{\Delta x}\|_{L^\infty(\Omega\times\mathbb{R}_+)} \leq C, \tag{3.13}$$

the approximate solutions converge to a measure valued solution (2.7) *of the conservation law* (2.1)*.*

Furthermore, the limit measure valued solution $\boldsymbol{\mu}$ *satisfies the entropy condition* (2.8)*.*

The above theorems (proved in [20]) show that the space-time DG method (3.2) is entropy stable and it converges to an entropy measure valued solution of (2.1). The assumption of the L^∞ bound in the convergence theorem 3.2 can be relaxed as suggested in the forthcoming paper [13] by considering a generalized version of the measure valued solution that allows for concentrations.

Furthermore, the corresponding DG method was also proved to converge to the weak solution of a *linear, symmetrizable* system in [20]. A variant of this method is also shown to converge to entropy solutions of scalar conservation laws in [19].

4. The computational method

We will now proceed to describe the concrete realization (implementation) of the variational formulation (3.2) for a given system of conservation laws (2.1).

4.1. **Choice of basis functions.** As the approximate solution of (3.2) $\mathbf{V}^{\Delta x} \in \mathcal{V}_p$, we start by specifying a suitable basis for this space of piecewise polynomials. To this end, we express $\mathbf{V}^{\Delta x}$ as a linear combination of basis functions $\boldsymbol{\phi}^n_{Kki} \in \mathcal{V}_p$:

$$\mathbf{V}^{\Delta x} = \sum_{K,k,i,n} \hat{v}^n_{Kki} \boldsymbol{\phi}^n_{Kki}, \tag{4.1}$$

where $0 \leq n \leq N-1$, $K \in \mathcal{T}$, $1 \leq k \leq m$ and $1 \leq i \leq n_f$ and the coefficients $\hat{v}^n_{Kki}$ are the degrees of freedom. The indices indicate the support of the basis functions: We choose the basis functions $\boldsymbol{\phi}^n_{Kki}$ such that they are nonzero only in one space-time element, i.e., $\boldsymbol{\phi}^n_{Kki}$ is nonzero in $K \times I^n$. Furthemore, only one component, namely the k-th component for $\boldsymbol{\phi}^n_{Kki}$, is non-zero. Also, we use the same scalar basis functions ϕ^n_{Ki} for all the components, i.e.,

$$(\boldsymbol{\phi}^n_{Kki})_l = \delta_{kl}\phi^n_{Ki}, \quad 1 \leq l \leq m. \tag{4.2}$$

The scalar basis functions have to span $\mathbb{P}_p(K \times I^n)$. We use a set of shifted and scaled monomials,

$$\phi^n_{Ki}|_{K\times I^n} = \left(\frac{t-t^{n+1}}{\Delta t^n}\right)^{p^{t,i}} \prod_{k=1}^{d} \left(\frac{x_k - \bar{x}^K_k}{\Delta x_K}\right)^{p^{x,k,i}}, \tag{4.3}$$

where $\bar{x}^K$ is the centroid of cell K and $p^{t,i}$, $p^{x,k,i}$, $k = 1, \ldots, d$ is the polynomial degree of the i-th scalar basis function in t, resp. x_k-direction. We use a maximal degree of p, hence:

$$p^{t,i} + \sum_{k=1}^{d} p^{x,k,i} \leq p. \tag{4.4}$$

In general there are $n_f = \binom{1+d+p}{p}$ scalar basis functions. This leads to a total number of degrees of freedom of $N_c N m n_f = N_c N m \binom{1+d+p}{p}$, where $N_c = |\mathcal{T}|$ is the number of cells.

4.2. **Quadrature rules.** There are three types of numerical integration that have to be performed in evaluating (3.2) over basis elements, namely, integration over whole space-time elements $K \times I^n$, integration over spatial elements K, and integration over the edges of the elements and in time $\partial_{KK'} \times I^n$. We use tensor product numerical integration formulas for the tensor product domains $K \times I^n$ and $\partial_{KK'} \times I^n$. Hence, we only need to specify the quadrature rules for the spatial element K, the edges $\partial_{KK'}$ and the time interval I^n. For the integration over triangles K, Dunavant quadrature rules of order $2p+2$ are applied [8], while for the integration over $\partial_{KK'}$ and I^n, Gaussian quadrature rules of order $2p+2$ are employed.

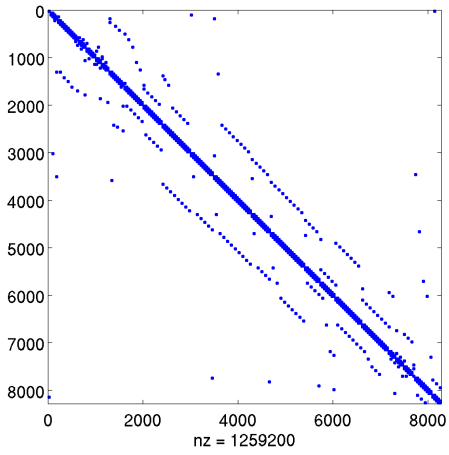


FIGURE 1. Nonzero structure of the Jacobian J for a 2-D Euler flow with 207 cells and $p = 2$.

4.3. **Choice of parameters.** We specify the following parameter values, streamline diffusion coefficient $C^{SD} = 10$ and shock capturing coefficients $C^{SC} = 1$ and $\bar{C}^{SC} = 0.1$, in order to complete the description of the numerical method (3.2). As mentioned in [20], the quality of numerical results was rather insensitive to the choice of parameters.

4.4. **Nonlinear and linear solvers.** Given the fact that the quasilinear form $\mathcal{B}$ is linear in the test function $\mathbf{W}^{\Delta x}$, it is enough to consider all the basis functions $\boldsymbol{\phi}^n_{Kki}$ as test functions in (3.2). Furthermore, the choice of the upwind temporal flux (3.4) allows us to march forward in time, i.e., given the approximate solution at time level t^n, we can determine the approximate solution at time level t^{n+1}. Combining the above facts and defining

$$F^n_{K'lj} = \mathcal{B}(\mathbf{V}^{\Delta x}, \boldsymbol{\phi}^n_{K'lj}),$$

we need to solve the following nonlinear algebraic system,

$$F^n_{K'lj}(\hat{v}^n) = 0, \quad K' \in \mathcal{T}, 1 \le l \le m, 1 \le j \le n_f, \tag{4.5}$$

to determine the vector of all the degrees of freedom $\hat{v}^n$ in the n-th time slab.

We use a damped Newton method [18] to solve this system: Starting from an initial guess $\hat{v}^n_0$, the approximation is iteratively improved by specifying,

$$\hat{v}^n_{i+1} = \hat{v}^n_i + \lambda_i \delta\hat{v}^n_i, \quad i = 0, 1, 2, \ldots \tag{4.6}$$

using the Newton correction $\delta\hat{v}^n_i$. The damping parameter $\lambda_i \le 1$ is chosen using a line search.

The Newton correction $\delta\hat{v}^n$ at the current state $\hat{v}^n$ (for ease of notation we drop the iteration index i) is computed from the linear algebraic system:

$$J^n(\hat{v}^n)\delta\hat{v}^n = -F^n(\hat{v}^n), \tag{4.7}$$

with the Jacobian being given by

$$J^n_{K'lj,Kki} = (F_{K'lj})_{\hat{v}^n_{Kki}}, \quad K, K' \in \mathcal{T}, 1 \le k, l \le m, 1 \le i, j \le n_f \tag{4.8}$$

The Jacobian is formed analytically and the resulting matrix is assembled. This matrix has a block structure with blocks of size $mn_f \times mn_f$, we will denote them by $J^n_{K',K}$. The block $J^n_{K',K}$ corresponds to the coupling of the degrees of freedom of cell K' to the degrees of freedom of the cell K. As only degrees of freedom of neighbouring cells are (directly) coupled, the Jacobian is sparse: Only the diagonal blocks $J^n_{K,K}$ and a few offdiagonal blocks are nonzero. The number of nonzero offdiagonal blocks for cell K' is given by the number of neighbouring cells of cell K' and is therefore small. Figure 1 illustrates the nonzero structure of J for a two-dimensional Euler case.

Summarizing, the linear system that need to be solved in every Newton step (4.7), consists of a matrix that is large and sparse. Furthermore, it is not symmetric. Direct solvers, such as a sparse LU decomposition can be used in one space dimension but the sheer magnitude of problem size prohibits the use of direct solvers in several space dimensions. In particular, we need a scalable iterative solver.

To this end, we use restarted GMRES (Generalized Minimal RESidual method) [29] as the iterative solver (restart every 30 iterations). However, as shown in the next section, our Jacobian matrix in (4.7) is not well-conditioned. It is well known [29] that iterative Krylov solvers are not efficient when the underlying matrix is ill-conditioned. In particular, the number of iterations can become prohibitively large. Hence, it is absolutely imperative to design suitable preconditioners for this problem.

5. Block Jacobi Preconditioner

5.1. **Description.** Given the block diagonal structure of the Jacobian matrix in (4.7), it is natural to consider block Jacobi preconditioners. The consequent preconditioner D is a block diagonal matrix, specified as

$$D_{K',K} = \delta_{K',K} J^n_{K',K},$$

so each block corresponds to an element K. Here, δ is the Kronecker delta symbol.

We need to invert the above matrix D in order to apply the preconditioner. But this corresponds only to inverting each block $J^n_{K,K}$ separately. This "local" inversion can be performed by a standard LU decomposition with reasonable effort as the blocks are quite small (of size $mn_f \times mn_f$). Furthermore, the work per block does not grow with the number of cells N_c. To illustrate typical block sizes that arise here, we consider the Euler equations in two space dimensions with piecewise quadratic trial functions ($p = 2$). These equations consist of $m = 4$ components and the polynomial degree $p = 2$ leads in two dimensions to $n_f = \binom{1+2+2}{2} = 10$ degrees of freedom per component and cell. Therefore, in this case the block size is 40×40. This (and even smaller) sizes are representative of typical block sizes in defining the preconditioner and are amenable to a direct solve.

5.2. **Fourier analysis.** Given any diagonalisable matrix $A = XDX^{-1}$, the fundamental estimate for the performance of GMRES was provided by Saad and Schultz [29]:

$$\|r_{m+1}\| \leq \kappa(X) \min_{p \in P_m, p(0)=1} \max_{\lambda \in \sigma} |p(\lambda)| \, \|r_0\| \,. \tag{5.1}$$

Here $\kappa(A) = \|X\| \left\|X^{-1}\right\|$ denotes the condition number, P_m the space of polynomials of degree at most m, σ denotes the spectrum of A and r_m is the residual in the m-th step of GMRES.

Given this basic estimate, we aim to compute the eigenvalues of the unpreconditoned matrix J in (4.7) as well as the preconditioned matrix $D^{-1}J$ in order to ascertain if the preconditioner did improve the conditioning. We use the following simplifications in order to make this problem tractable,

- We require periodic boundary conditions as we will use a Fourier transformation of the system. Uniform grids are also assumed for simplification.
- As it is not possible (algebraically) to perform the analysis for a generic linearized system, we will analyze the linear advection equation as a model problem.

5.2.1. *Linear advection in one space dimension.* We consider the simplest example of a conservation law, the linear advection equation:

$$u_t + a u_x = 0 \tag{5.2}$$

in the one-dimensional domain $[-1, 1]$ with periodic boundary conditions. For simplicity we assume $a > 0$ and discretize (5.2) on a regular grid; numbered (here) $0, \ldots, N_c - 1$ and choose a Rusanov diffusion operator in (3.6) resulting in an upwind spatial numerical flux. Neglecting the shock-capturing operator, the resulting Jacobian

in (4.7) for this problem has the structure,

$$\frac{1}{\Delta x}J = \begin{pmatrix} A & & & & B \\ B & A & & & \\ & B & A & & \\ & & \ddots & \ddots & \\ & & & B & A \end{pmatrix} = I \otimes A + S \otimes B \tag{5.3}$$

with fixed matrices $A, B \in \mathbb{R}^{n_f m \times n_f m}$ (depending on p and the time step). Here, $\otimes$ denotes the Kronecker product and the matrix S is defined as,

$$(S)_{j,k} = \begin{cases} 1, & j = k+1 \mod N_c \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

Let F be the Fourier matrix, i.e.,

$$F = \frac{1}{\sqrt{N_c}}(\omega^{jk})_{j,k=0}^{N_c-1}, \quad \omega = e^{-2\pi i/N_c} \tag{5.5}$$

Since the matrix S is a circulant matrix, it can diagonalised by the Fourier matrix as

$$S = F\,\mathrm{diag}(d)F^*. \tag{5.6}$$

With F^* denoting the Hermitian of F and

$$\begin{aligned} d &= \sqrt{N_c}F((S)_{0,0}, \ldots, (S)_{0,N_c-1})^T \\ &= \sqrt{N_c}F(0, \ldots, 0, 1)^T \\ &= (\omega^{0(N_c-1)}, \ldots, \omega^{(N_c-1)(N_c-1)})^T \end{aligned} \tag{5.7}$$

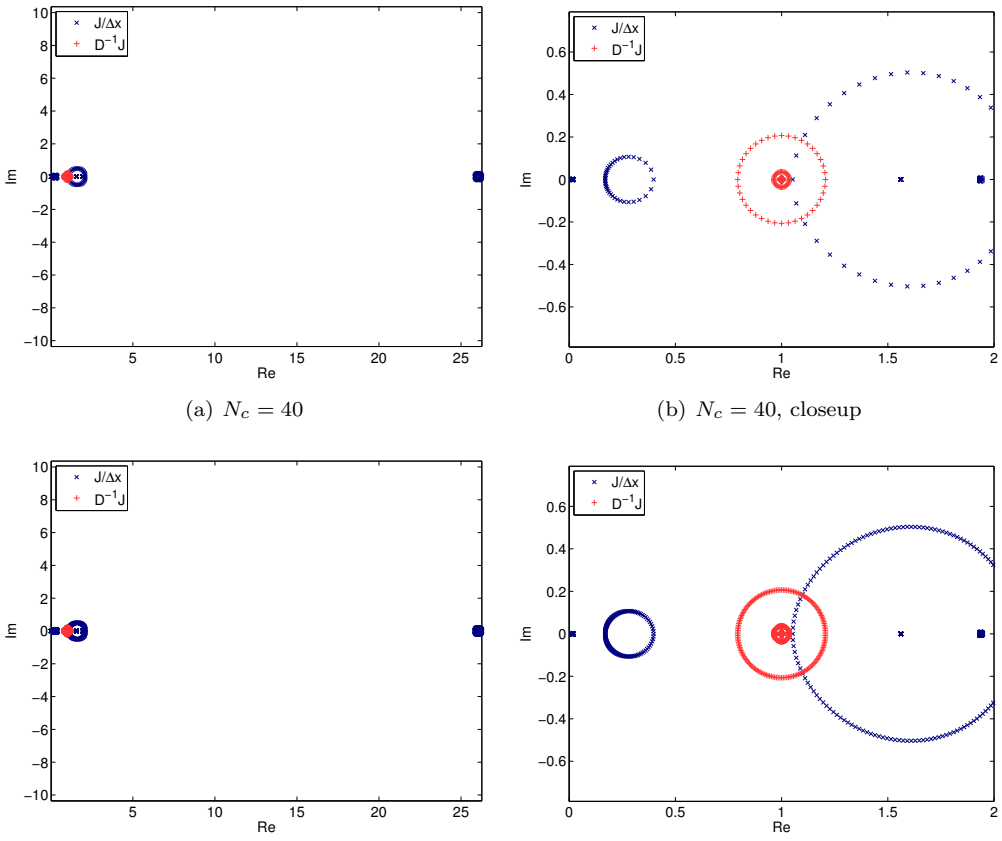
Therefore

$$\begin{aligned} \frac{1}{\Delta x}J &\sim (F \otimes I)^*(I \otimes A + S \otimes B)(F \otimes I) \\ &= (F^*IF) \otimes (IAI) + (F^*SF) \otimes (IBI) \\ &= I \otimes A + \mathrm{diag}(d) \otimes B \\ &= \begin{pmatrix} A + d_0 B & & \\ & A + d_1 B & \\ & & \ddots \end{pmatrix} \end{aligned} \tag{5.8}$$

where $\sim$ denotes similarity.

From the above calculation, we have established that $\frac{1}{\Delta x}J$ is similar to a blockdiagonal matrix with blocks $A + d_i B$. Furthermore, observe that the number of cells N_c only enters the expression through the factors d_i. However, as $|d_i| = 1, \forall i$, the eigenvalues can only lie on curves specified by the eigenvalues of $A + e^{i\phi}B$, parametrized by ϕ. Hence, refining the mesh (increasing N_c) only leads to increase in the denseness of the eigenvalues on these curves. Furthermore, assuming that none of these level curves cross the origin, the closeness of the eigenvalues to zero is almost independent of the mesh size. These findings are validated in figures 2 and 3. Furthermore as demonstrated in Figure 3, the increase of polynomial degree p leads to an increase in the number of degrees of freedom per element (block size) and thus increases the number of curves on which the eigenvalues lie. Next, we consider the role of the preconditioner,

$$\frac{1}{\Delta x}D = I \otimes A. \tag{5.9}$$



(a) $N_c = 40$ (b) $N_c = 40$, closeup

(c) $N_c = 160$ (d) $N_c = 160$, closeup

FIGURE 2. Eigenvalues of $\frac{1}{\Delta x} J$ and $D^{-1}J$ for the 1-D linear advection equation with different number of cells for polynomial degree $p = 2$.

The resulting preconditioned system is

$$
\begin{aligned}
D^{-1}J &= (I \otimes A)^{-1}(I \otimes A + S \otimes B) \\
&= I \otimes I + S \otimes A^{-1}B \\
&\sim (F \otimes I)^H (I \otimes I + S \otimes A^{-1}B)(F \otimes I) \\
&= (F^H I F) \otimes (III) + (F^H S F) \otimes (I A^{-1} B I) \\
&= I \otimes I + \mathrm{diag}(d) \otimes A^{-1}B \\
&= \begin{pmatrix} I + d_0 A^{-1}B & & \\ & I + d_1 A^{-1}B & \\ & & \ddots \end{pmatrix}
\end{aligned} \tag{5.10}
$$

Let $\tilde{d}$ denote the vector of eigenvalues of $A^{-1}B$. Then the eigenvalues of $I + d_i A^{-1}B$ (and consequently of $D^{-1}J$) are given by $1 + d_i\tilde{d}$. Since $|d_i| = 1$, the eigenvalues lie on circles $1 + e^{i\phi}\tilde{d}_j$, parametrized by ϕ (see Figure 2 and 3). Hence, if the eigenvalues $\tilde{d}_j$ of $A^{-1}B$ are small, then the eigenvalues of the preconditioned system are clustered around one (therefore bounded away from the origin). This will clearly improve the spectral contribution to the performance estimate (5.1) of GMRES.

However, it is clear from the estimate (5.1) that it is not enough to consider the distribution of eigenvalues. We also need to study the conditioning of the eigenvector matrix X in order to ascertain the performance of GMRES. To this end, we denote V as the matrix corresponding to an eigenbasis of $A^{-1}B$. (5.10) implies that $X = F \otimes V$ diagonalises $D^{-1}J$. Hence, its condition number is given by,

$$
\begin{aligned}
\kappa(X) &= \kappa(F \otimes V) \\
&= \kappa((F \otimes I)(I \otimes V)) \\
&\leq \|F \otimes I\| \, \|I \otimes V\| \, \left\|(F \otimes I)^{-1}\right\| \, \left\|(I \otimes V)^{-1}\right\| \\
&= \kappa(V),
\end{aligned} \tag{5.11}
$$

as the Fourier matrix F is unitary. Therefore, the condition number of X is bounded independently of the number of cells N_c.

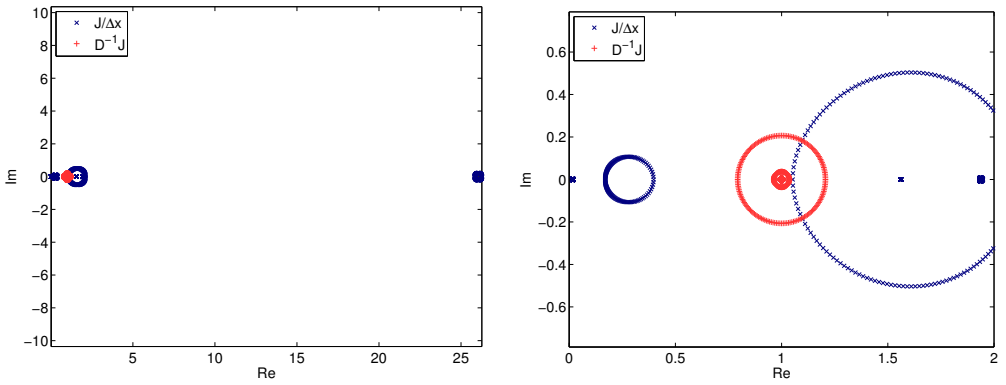
Combining the above results with (5.1), we establish that the number of iterations of the Jacobi preconditioned GMRES is independent of mesh size. Furthermore, the smallness of the eigenvalues $\tilde{d}$ will lead to a clustering of the eigenvalues of the preconditioned matrix, away from the origin (as seen in figures 2 and 3). Hence, this analysis demonstrates that the Jacobi preconditioner enhances the performance of GMRES, at least for the linear advection equation.

5.2.2. *Role of the shock capturing term.* The inherent non-linearity of the shock capturing term in (3.10) implies that the variational formulation (3.2) is nonlinear, even when the underlying PDE is a linear conservation law such as the advection equation. Consequently, the Jacobian matrix J in (4.7) is solution dependent. In order to analyze the role played by the shock capturing term, we consider the linear advection equation (5.2) with periodic initial data (5.16) and display the eigenstructure of the Jacobian at a typical time step in Figure 4. The figure compares the eigenstructure of the Jacobian as well as the preconditioned Jacobian, in the absence and in the presence of the shock capturing term. As seen in the figure, the shock capturing term has a very minor effect on the eigenvalue distribution in both the non-preconditioned as well as preconditioned matrices. Its main effect is a distortion in the shape of some level curves.

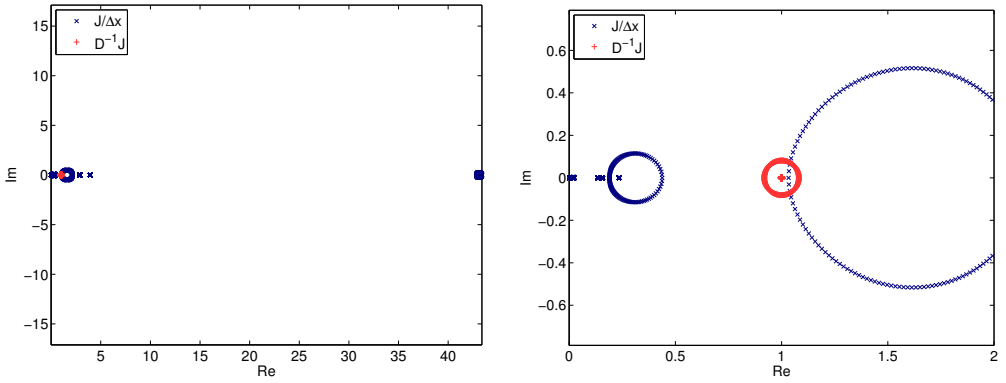
5.2.3. *Linear advection in two space dimensions.* As prototype for multi-dimensional problems, we consider the two-dimensional version of the linear advection equation:

$$
u_t + a^x u_x + a^y u_y = 0 \tag{5.12}
$$

in the domain $[-1, 1]^2$ with periodic boundary conditions. We discretize this equation with the space-time DG method (3.2) (neglecting the shock capturing term) on a uniform rectangular grid with mesh size Δx. Following

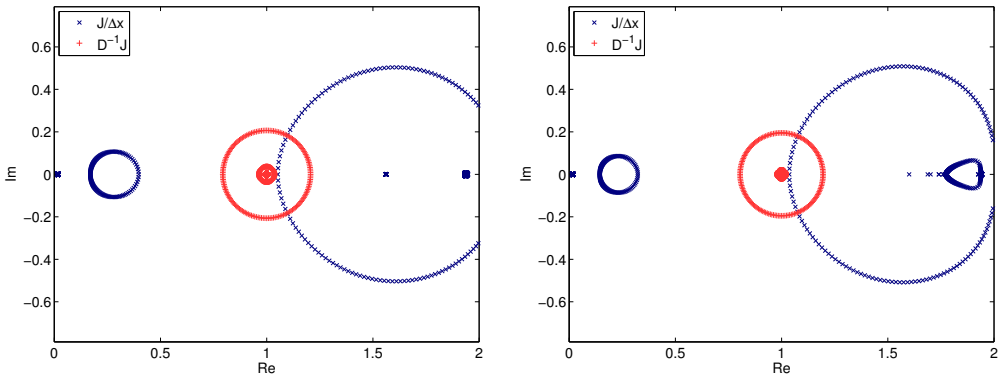


(a) $p = 2$ (b) $p = 2$, closeup



(c) $p = 3$ (d) $p = 3$, closeup

FIGURE 3. Eigenvalues of $\frac{1}{\Delta x}J$ and $D^{-1}J$ for the 1-D linear advection for different degrees and for $N_c = 160$.



(a) $N_c = 160$, no SC, closeup (b) $N_c = 160$, SC, closeup

FIGURE 4. Eigenvalues of $\frac{1}{\Delta x}J$ and $D^{-1}J$ with or without the shock capturing term (SC).

the previous section, we can write the Jacobian as,

$$\frac{1}{\Delta x^2} J = I \otimes I \otimes A + S^x \otimes I \otimes B + I \otimes S^y \otimes C \tag{5.13}$$

with fixed matrices $A, B, C \in \mathbb{R}^{n_f m \times n_f m}$ (depending on polynomial degree p and the time step) and the matrices S^x, S^y, analogously defined as before. Fourier transforming in both x and y directions, one can show that $\frac{1}{\Delta x^2} J$ is blockdiagonal with blocks of the form

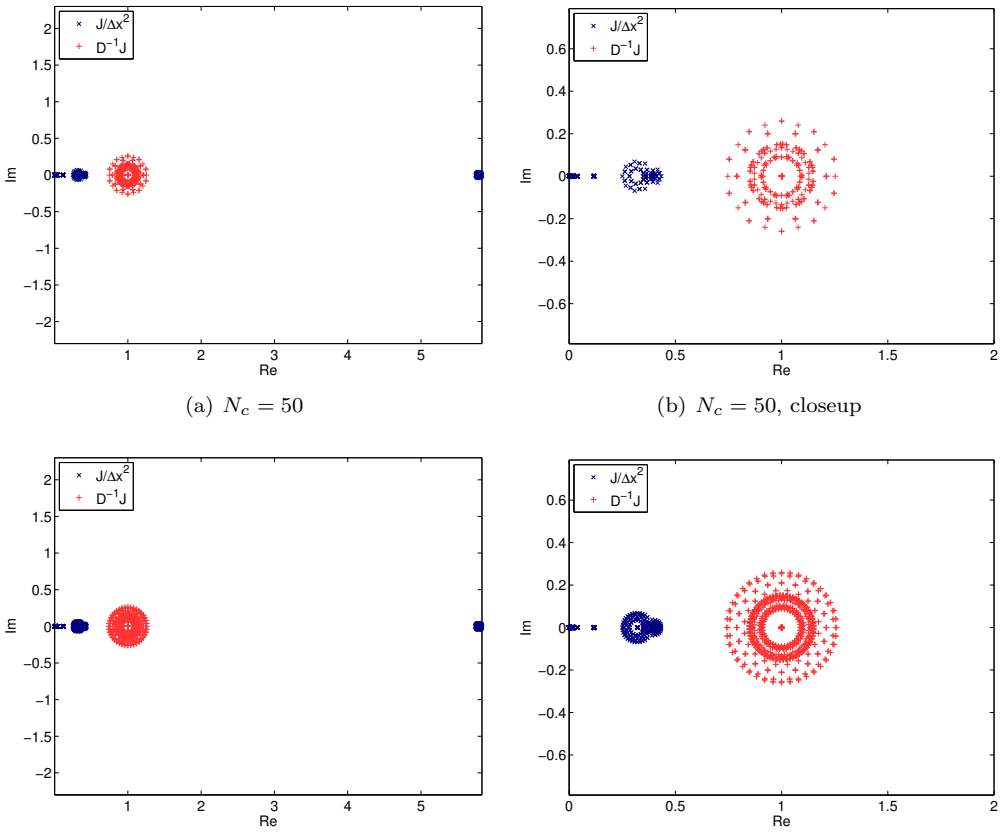
$$A + e^{i\phi^x} B + e^{i\phi^y} C \tag{5.14}$$

parametrized by ϕ^x and ϕ^y. The Jacobi preconditioned matrix is similar to a blockdiagonal matrix with blocks of the form

$$I + e^{i\phi^x} A^{-1} B + e^{i\phi^y} A^{-1} C. \tag{5.15}$$

The eigenstructure in 2-D is more complicated than in 1-D as, in general, $A^{-1}B$ and $A^{-1}C$ are not simultaneously diagonalisable.

Hence, we will limit ourselves to numerical results in this case. In Figure 5 the eigenvalues of $\frac{1}{\Delta x^2} J$ and $D^{-1}J$ are shown for a regular triangular mesh. As in the 1-D case the general structures remain the same, under mesh refinement. The distribution of eigenvalues is denser but is confined to a bounded region of the complex plane. Furthermore, the preconditioner clusters the eigenvalues around 1 indicating that the preconditioner does enhance the performance of GMRES in this case.



(a) $N_c = 50$ (b) $N_c = 50$, closeup

(c) $N_c = 200$ (d) $N_c = 200$, closeup

FIGURE 5. Eigenvalues of $\frac{1}{\Delta x^2} J$ and $D^{-1}J$ for the linear advection equation in 2-D for $p = 2$ for different number of cells.

5.3. **Numerical results.** The Fourier analysis of the previous section indicates that

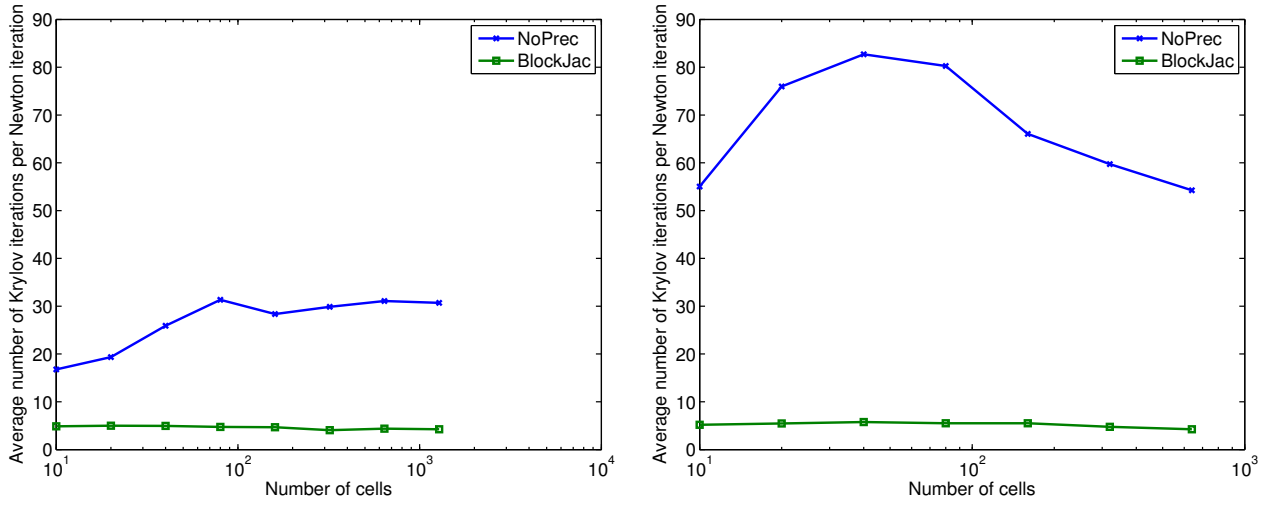
- The performance of the Jacobi preconditioned GMRES will be independent of the mesh size.
- The performance of the preconditioned GMRES will be significantly better than its unpreconditioned version as the eigenvalues of the preconditioned Jacobian are clustered around 1. Hence, they are bounded away from zero, unlike that of the unpreconditioned version, see Figure 2.

The analysis was restricted to the model case of the one-dimensional and two-dimensional advection equations. We will evaluate the performance of the Jacobi preconditioner for this model problem as well as the more complicated Euler equations of gas dynamics.

5.3.1. *Linear advection equation in one space dimension.* We consider the linear advection equation (5.2) in the domain $[-1, 1]$ with periodic boundary conditions. The initial condition is

$$u(x, 0) = \sin(2\pi x). \tag{5.16}$$

The quadratic entropy is used. Hence, the conservative and entropy variables coincide. Furthermore, a Rusanov type numerical diffusion operator in (3.6) is used. In each Newton step the linear system (4.7) is solved up to a relative tolerance of 10^{-4}. The average number of Krylov iterations per Newton iteration is depicted in Figure 6. We compare the unpreconditioned system with the system preconditioned by block Jacobi. As predicted from the analysis, the number of iterations for both the unpreconditioned as well as preconditioned systems is independent of mesh size. However and consistent with the analysis of the previous section, the number of iterations is significantly reduced from around 30 ($p = 1$) or 70 ($p = 2$) in the unpreconditioned case to approximately 5 for the block Jacobi preconditioned system. This demonstrates a significant gain in efficiency due to the block Jacobi preconditioner. The approximate solution computed with piecewise linear and piecewise



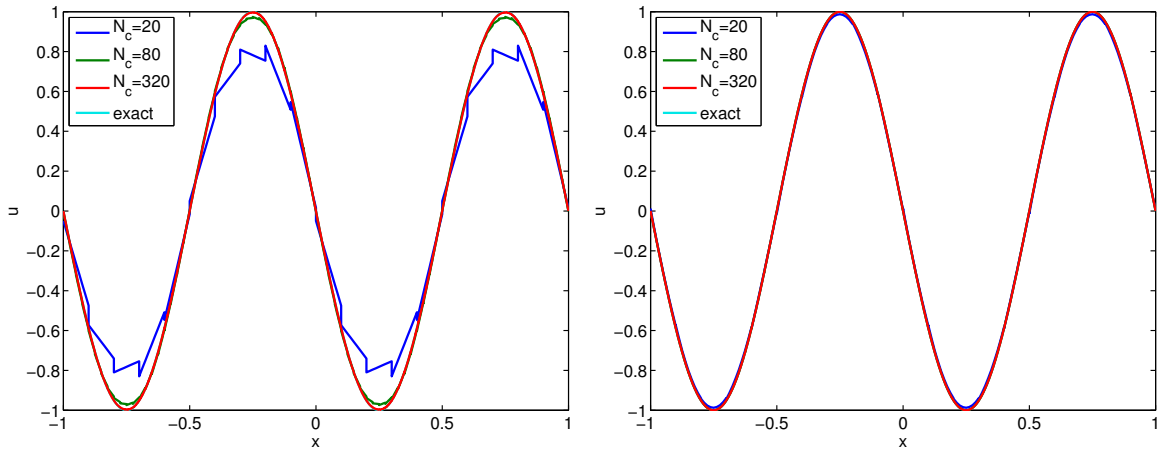
(a) $p = 1$ (b) $p = 2$

FIGURE 6. Number of Krylov iteration per Newton iteration for the one-dimensional linear advection equation.

quadratic basis functions (and with the preconditioned system) is plotted in Figure 7. The figure shows that the exact smooth solution is approximated very well with the shock capturing space-time DG method. The quality of approximation improves with increasing mesh size or increasing polynomial degree.

5.3.2. *Linear advection in two space dimensions.* We consider the linear advection equation (5.12) in the domain $[-1, 1]^2$ with constant velocities $a^x = 1$ and $a^y = 1/2$. The initial condition is a bump at $(-0.2, -0.2)$:

$$u(x, 0) = \frac{1}{2} e^{-16((x+0.2)^2+(y+0.2)^2)} (1 - e^{-16(x+1)^2})(1 - e^{-16(x-1)^2})(1 - e^{-16(y+1)^2})(1 - e^{-16(y-1)^2}). \tag{5.17}$$

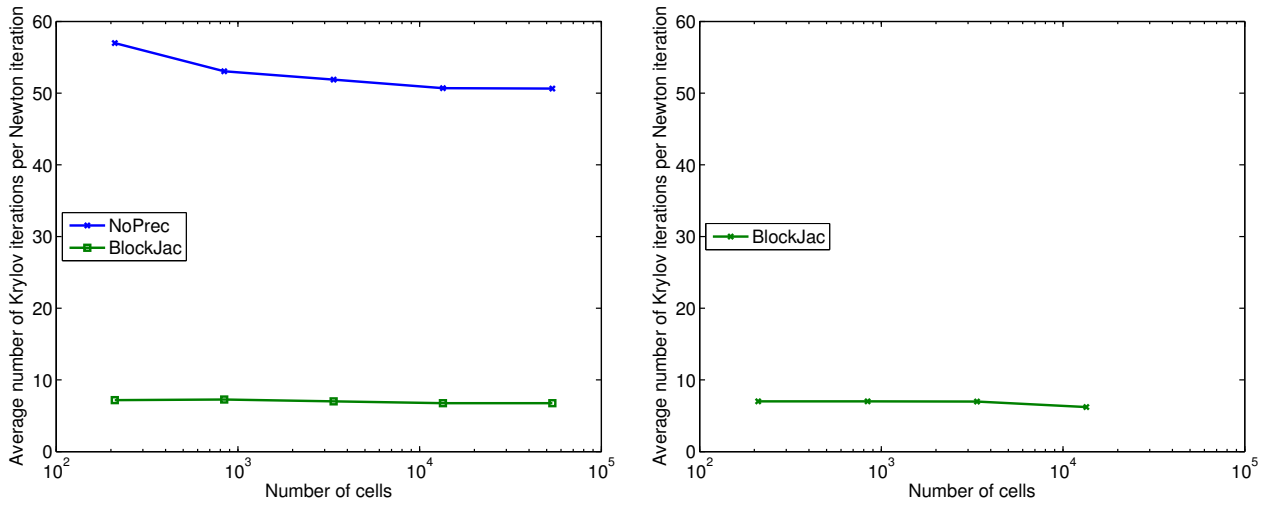


(a) $p = 1$ (b) $p = 2$

FIGURE 7. 1-D Linear advection, different number of cells.

At the inflow boundaries homogeneous Dirichlet boundary conditions are used.

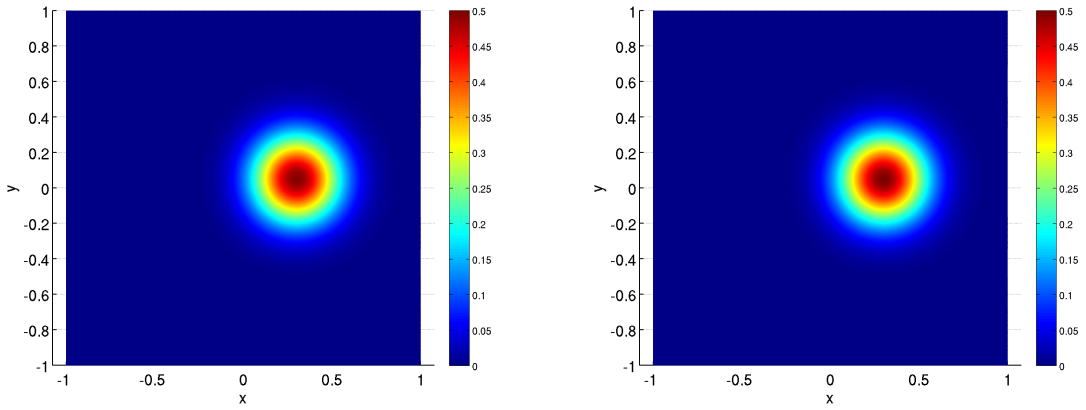
As in the previous case, we consider the quadratic entropy and a Rusanov type numerical diffusion operator. The average number of Krylov iterations per Newton iteration is displayed in Figure 8. The unpreconditioned version works only for $p = 1$, where slightly more than 50 steps are needed. For $p = 2$ the linear solver does not always converge within the maximum number of 2000 iterations. With the block Jacobi preconditioner, one needs only about 7 iterations to converge (both for $p = 1$ and $p = 2$). Furthermore, this number is independent of the mesh size. Thus, this example clearly illustrates the enhancement in efficiency on account of the preconditioner. The approximate solution, computed with preconditioned space-time DG method (3.2)



(a) $p = 1$ (b) $p = 2$

FIGURE 8. Number of Krylov iteration per Newton iteration for the linear advection equation in two dimensions.

is shown in Figure 9 and shows that the method is able to approximate the advected solution quite well, both for piecewise linear as well as piecewise quadratic basis functions.



(a) $p = 1$ (b) $p = 2$

FIGURE 9. Advection equation, smooth initial data, $N_c = 13440$.

5.3.3. *Euler equations of gas dynamics.* Next, we consider the nonlinear Euler equations of gas dynamics. In two space dimensions, they are of the form,

$$\begin{aligned} \mathbf{U}_t + \mathbf{F}^1(\mathbf{U})_x + \mathbf{F}^2(\mathbf{U})_y &= 0, \\ \mathbf{U} &= (\rho, \rho u, \rho v, \rho E), \\ \mathbf{F}^1(\mathbf{U}) &= \big(\rho u, \rho u^2 + p, \rho uv, \rho uH\big) \\ \mathbf{F}^2(\mathbf{U}) &= \big(\rho v, \rho uv, \rho v^2 + p, \rho vH\big) \end{aligned} \tag{5.18}$$

Here, ρ is the density, u, v are the velocity fields and ρE is the total energy. The auxillary quantities are the pressure p, sound speed c and the enthalpy H given by

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho(u^2 + v^2)), \quad c = \sqrt{\gamma \frac{p}{\rho}}, \quad H = \frac{c^2}{\gamma - 1} + \frac{1}{2}(u^2 + v^2)$$

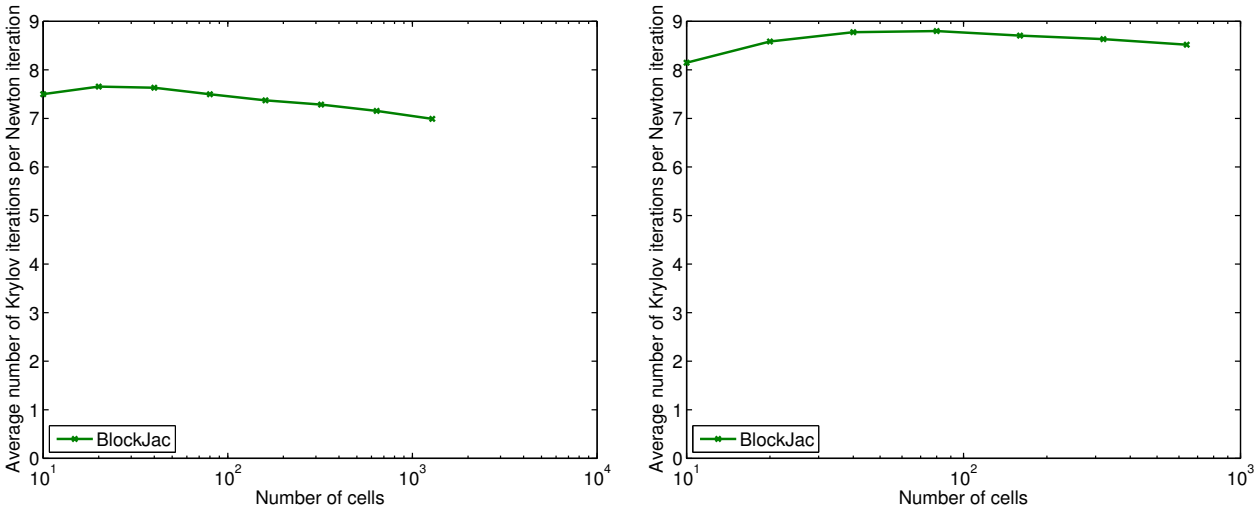
and γ is adiabatic exponent, which is set to 1.4 in all experiments.
Furthermore, the specific entropy is $s = \log p - \gamma \log \rho$, and the total entropy for the Euler equation is given by,

$$S = \frac{-\rho s}{\gamma - 1}$$

The corresponding entropy variables as well as details of the numerical fluxes are provided in [20].

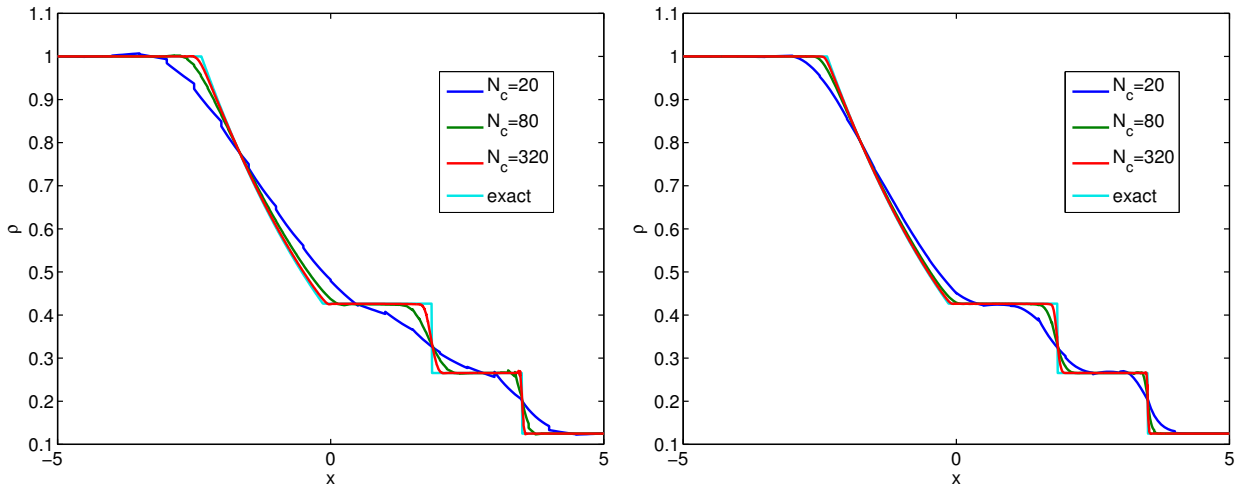
5.3.4. *Sod shock tube.* To begin with, we consider the Euler equations in one space dimension on the domain $[-5, 5]$. The Sod shock tube is a Riemann problem, centered at the origin, with initial left state $\rho = 1$, $u = 0$, $p = 1$ and initial right state $\rho = 0.125$, $u = 0$, $p = 0.1$. We use the entropy conservative flux for the Euler equations derived in [24], see also [10], together with a Rusanov type of diffusion. Again, we are interested in the average number of Krylov iterations per Newton iteration, see Figure 10. The case without preconditioner is not included as it needs several hundred iterations or fails to converge to the desired tolerance within 2000 iterations. However, with the block Jacobi preconditioner, the specified relative tolerance of 10^{-4} is reached within only about 7 ($p = 1$) or 8 ($p = 2$) iterations. Furthermore, the number of iterations is totally independent of the number of mesh points.

The computed density for different mesh resolutions and different polynomial orders is displayed in Figure 11. The space-time DG method approximates the solution, consisting of a shock wave, a contact discontinuity and a rarefaction wave, quite well. There is a significant gain in accuracy when the number of mesh points is increased. The subsequent gain in accuracy on increasing the polynomial degree from one to two is rather moderate.



(a) $p = 1$ (b) $p = 2$

FIGURE 10. Number of Krylov iteration per Newton iteration for the Sod shock tube problem.



(a) $p = 1$ (b) $p = 2$

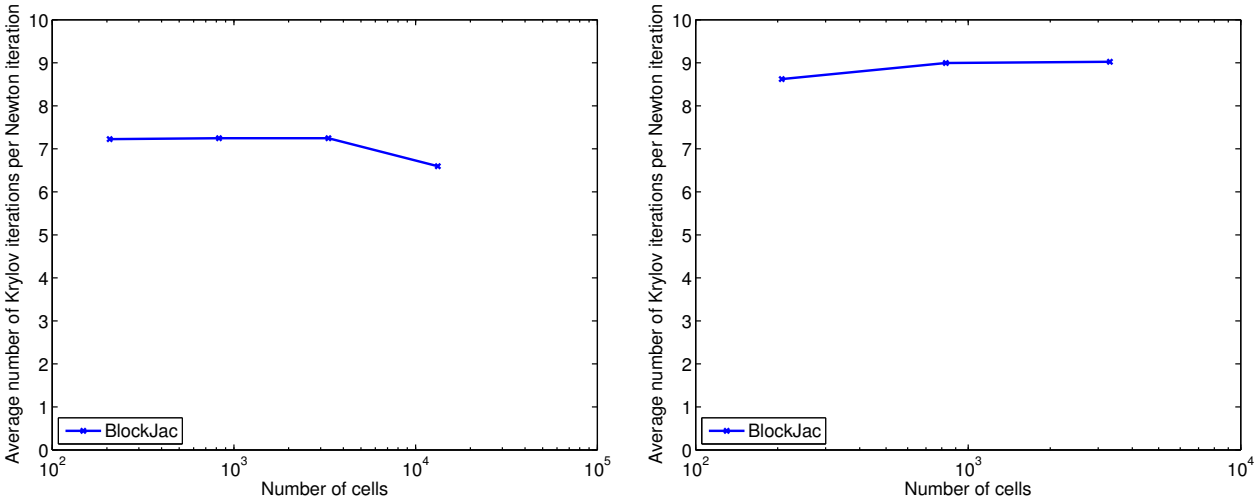
FIGURE 11. Sod shock tube, different number of cells.

5.3.5. *Two dimensional Euler vortex.* Next, we consider the advection of an Euler vortex (see [10] for the setup) in the domain $[0,10]^2$, with Dirichlet boundary conditions and a vortex centred at $x_c = 5, y_c = 5$ with $r_c = 1$ as initial condition:

$$u = 1 - (y - y_c)\phi(r), \quad v = 1 + (x - x_c)\phi(r), \quad \theta = 1 - \frac{\gamma - 1}{2\gamma}\phi(r)^2, \quad s = 0,$$

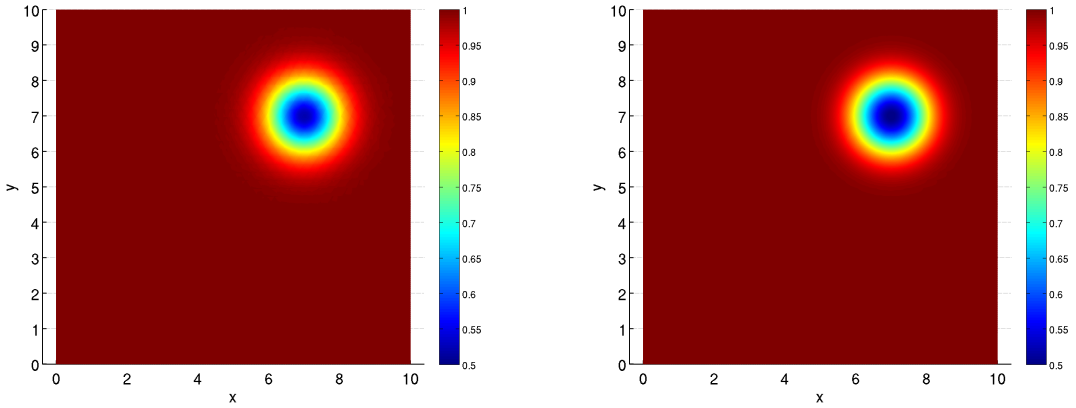
where $\theta = \frac{p}{\rho}$, $s = \log p - \gamma \log \rho$, $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$, $\phi(r) = \epsilon e^{\alpha\left(1-\left(\frac{r}{r_c}\right)^2\right)}$, $\epsilon = \frac{5}{2\pi}$ and $\alpha = 12$.

The iteration numbers are shown in Figure 12. Also in this case, the number of Krylov iterations is quite small for the block Jacobi preconditioned system: about 7 ($p = 1$) or 9 ($p = 2$) iterations per Newton iteration. The number of iterations is also independent of grid size. The computed solutions are shown in Figure 13 and



(a) $p = 1$ (b) $p = 2$

FIGURE 12. Number of Krylov iteration per Newton iteration for the vortex advection problem.



(a) $p = 1$ (b) $p = 2$

FIGURE 13. Vortex advection for the Euler equations, density, $N_c = 3312$, $t = 2$.

demonstrate that the shock capturing DG method approximates the underlying exact solution (vortex advected

along the diagonal) quite well. Summarizing the results of the above numerical experiments, we observe that the block Jacobi preconditioner does significantly reduce the number of Krylov iterations per Newton step, compared to the unpreconditioned version. Combined with the observed independence of the number of the iterations with respect to mesh size, the preconditioner increases the efficiency of the shock capturing space-time DG methods by orders of magnitude.

5.4. **CFL dependence.** The time step for the space-time DG method (3.2) is determined using the CFL number by the formulas,

$$\Delta t^{n+1} \leq C^{CFL} \min_{K\in\mathcal{T}, x\in K} \frac{\Delta x_K}{\lambda_{max}(\mathbf{U}^{\Delta x}(x,t^n))}, \tag{5.19}$$

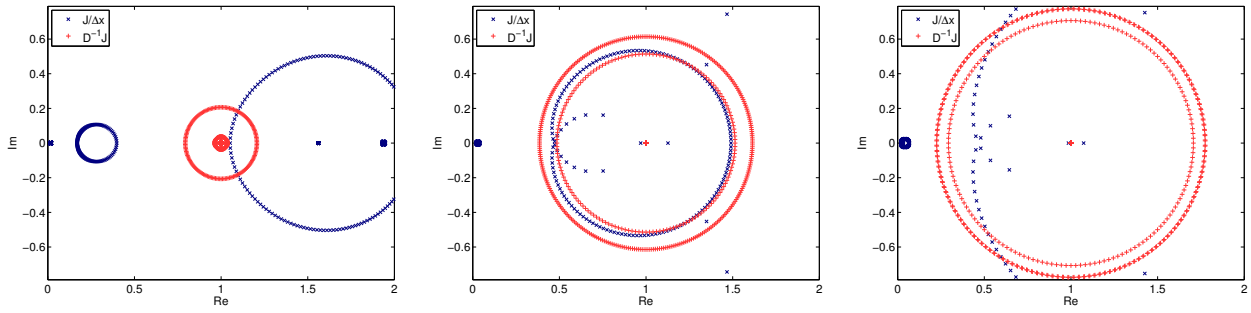
in one space dimension and

$$\Delta t^{n+1} \leq C^{CFL} \min_{K\in\mathcal{T}, x\in K} \frac{\frac{|K|}{\Delta x_K}}{\lambda_{max}(\mathbf{U}^{\Delta x}(x,t^n))}, \tag{5.20}$$

in two space dimensions. Here $\lambda_{max}(\mathbf{U}) = \max_\nu \lambda_{max}(\mathbf{U};\nu)$ is the maximal wave speed in all directions and the constant C^{CFL} is typically chosen to be $1/2$.

One of the attractive features of the shock capturing space-time DG method is the fact that the method remains entropy stable as well as convergent to the entropy measure valued solutions for any finite value of the CFL constant C^{CFL}. In particular, large CFL constants (large time steps) are allowed. This possibility of setting large time steps enables the space-time DG method to be efficient in resolving problems involving convergence to steady state (such as in aerodynamic calculations) or problems with multiple time scales (such as in radiation hydrodynamics). See the forthcoming paper [21] for further examples. Hence, it would be useful to design preconditioners that can work in the large time step (CFL number) regime.

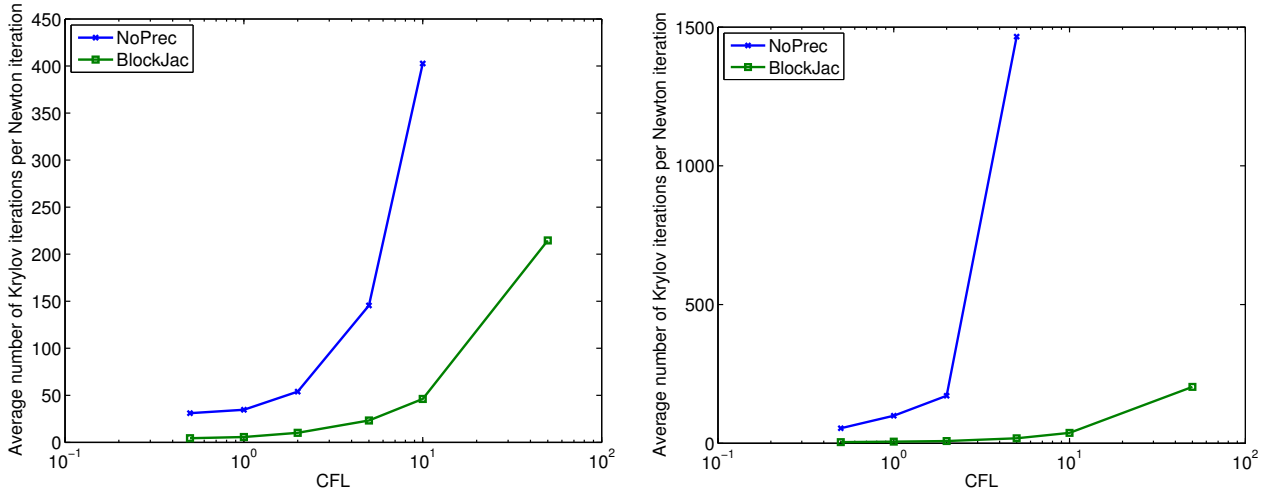
5.4.1. *Fourier analysis.* We perform the Fourier analysis for the linear advection case, as in the previous section but allow for a dependence on the CFL number. The eigenvalues of the unpreconditioned as well as block Jacobi preconditioned matrices are shown in Figure 14). The results show that some of the eigenvalues of $A^{-1}B$ get larger when the time step is increased (for the same mesh size). Consequently, the eigenvalues of the preconditioned matrix are no longer clustered around one and come close to zero. This will lead to a deterioration in the performance of the preconditioner and will increase the number of iterations per Newton step.



(a) $CFL = 0.5$, closeup (b) $CFL = 5$, closeup (c) $CFL = 10$, closeup

FIGURE 14. Eigenvalues of $\frac{1}{\Delta x}J$ and $D^{-1}J$ for the one-dimensional linear advection equation with different CFL values for $N_c = 160$ and $p = 2$.

5.4.2. *Numerical results.* This lack of robustness of the block Jacobi preconditioner to increasing the CFL number is examined through a numerical experiment for the one-dimensional linear advection equation. The same set-up as in the previous subsection is used and the average number of iterations per Newton step, for both the unpreconditioned and preconditioned systems is shown in Figure 15. As seen from the figure and consistent with the Fourier analysis, the number of iterations increases significantly with increasing CFL number. In particular, the block Jacobi preconditioner is quite well-behaved till a moderate CFL number of around 10, but the number of iterations increases significantly with further increase in the CFL number.



(a) $p = 1$ (b) $p = 2$

FIGURE 15. Number of Krylov iteration per Newton iteration for the linear advection equation in dependence of the CFL number ($N_c = 640$).

The following heuristic explains this observed dependence of the block Jacobi preconditioner with the CFL number. The exact solution of the conservation law has a finite speed of propagation, due to the hyperbolicity of the system. Even though the implicit formulation of the space-time DG method couples all the degrees of freedom (and therefore allows an infinite speed of propagation), one can still expect that most effects are local as they eventually approximate the exact solution. Block Jacobi is acting on the degrees of freedom of each cell separately, i.e., it is a local solver. So for small CFL numbers one expects it to be a good preconditioner as the time step is too small for information to propagate significant distances in the domain. However, the higher the CFL number, the further is the information allowed to travel, i.e., it will cross more and more cells in just one time step and the effects become increasingly non-local. Therefore block Jacobi, as it is a local solver, can not resolve these waves and will fail.

6. Block Gauss-Seidel Preconditioners

The above discussion suggests that a *non-local* preconditioner might be more robust with respect to increasing the CFL number. In this context, we consider preconditioners of the Gauss-Seidel type. To this end, we introduce an ordering of the cells and thus of the Jacobian J in (4.7) and define the block lower triangular part as

$$L_{K',K} = \mathbb{1}_{K'>K} J^n_{K',K}$$

and the upper triangular part as

$$U_{K',K} = \mathbb{1}_{K'<K} J^n_{K',K}.$$

A simple block Gauss-Siedel preconditioner results from a forward sweep, i.e., using $D+L$ as a preconditioner. Applying the preconditioner to the system $Jx = b$ results for the forward version in

$$x = (D + L)^{-1} b \tag{6.1}$$

We can also consider the symmetric version, where an additional backwards sweep using $D+U$ is performed, resulting in,

$$\begin{aligned} \tilde{x} &= (D+L)^{-1}b \\ x &= \tilde{x} + (D+U)^{-1}(b - J\tilde{x}) \end{aligned} \tag{6.2}$$

The inversion of $D+L$ resp. $D+U$ is performed cell by cell in forward resp. backward direction. For each cell this corresponds to solving an $mn_f \times mn_f$ system as in the block Jacobi case, but the important difference is that this inversion is performed sequentially.

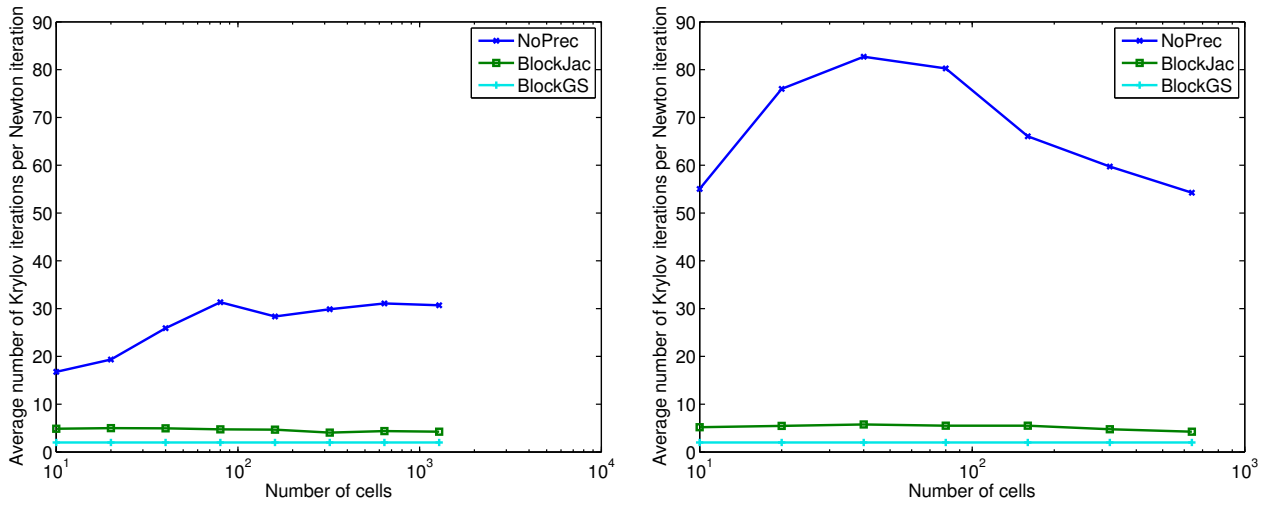
6.1. **Analysis of the spectrum.** As before, we consider the one-dimensional linear advection equation with periodic boundary conditions. The block GS preconditioned marix is given by

$$(D+L)^{-1}J = (D+L)^{-1}(D+L+U) = I + (D+L)^{-1}U. \tag{6.3}$$

The matrix $(D+L)^{-1}U$ is stricly block upper triangular (there are only two nonzero blocks) given the structure of J in (5.3). Hence, all eigenvalues are 1 and the block GS is a perfect preconditioner in this case. Furthermore, this fact is independent of the CFL condition. However, this is limited to the advection equation; for more general conservation laws the performance of block GS might be less striking.

6.2. **Numerical experiments.**

6.2.1. *One-dimensional linear advection.* We consider the one-dimensional linear advection equation (5.2) with initial data (5.16). The numerical set-up is exactly as before. The cells are ordered in the advection direction and we use the forward version of the block GS preconditioner. The iteration numbers are shown in Figure 16. Only two iterations are required for the block Gauss-Seidel preconditioned system to converge to desired tolerance. This number is independent of the grid size.

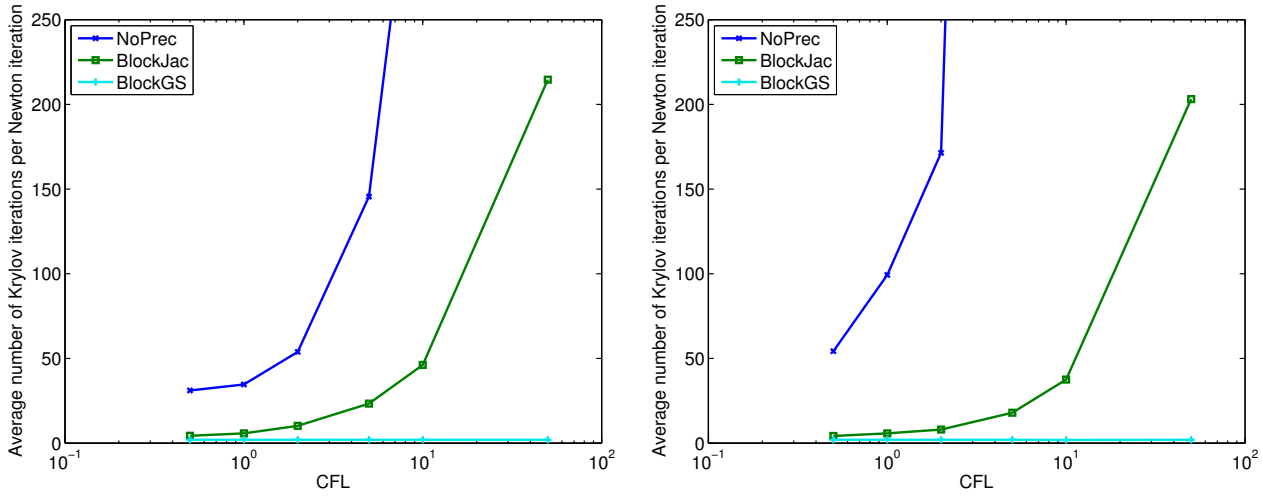


(a) $p = 1$ (b) $p = 2$

FIGURE 16. Number of Krylov iteration per Newton iteration for the linear advection equation.

However, the real test of the block Gauss-Seidel preconditioner lies in increasing the CFL number. The corresponding iteration numbers for different preconditioners are shown in Figure 17. The number of iterations with respect to the block Gauss-Seidel preconditioner remains at 2 for any value of the CFL number. This is in marked contrast to the significant increase in the number of the iterations for the unpreconditioned and the block Jacobi preconditioned systems. The excellent preconditioning properties of block GS for the advection equation was also observed earlier on in a slightly different context, see [16]. However, it depends on the ordering

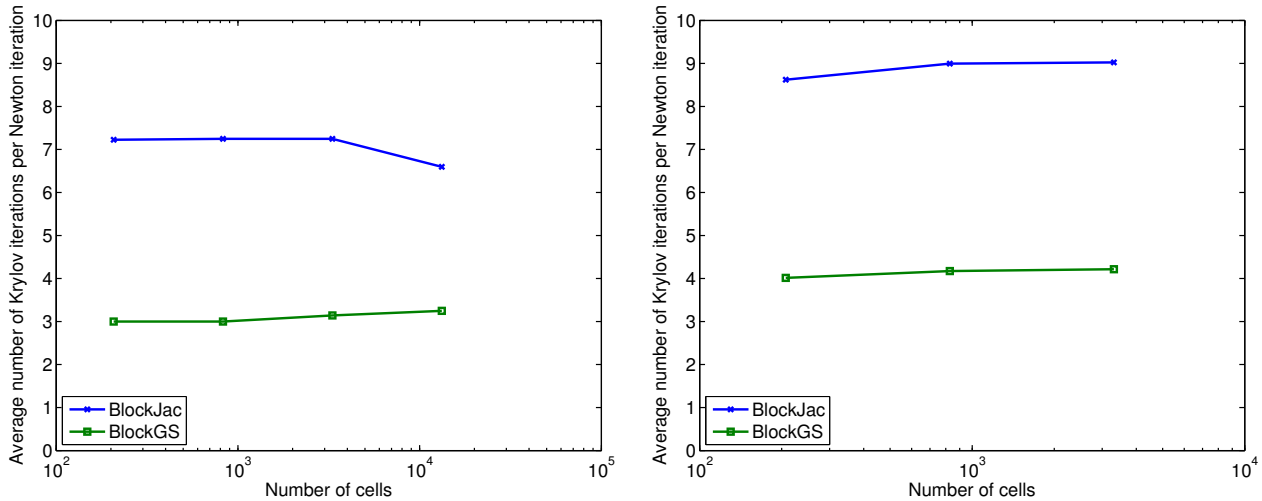
of the degrees of freedom. There is a single advection direction (a single flow of information) and the cells must be ordered in this downwind direction for the block GS preconditioner to fully unleash its power.



(a) $p = 1$ (b) $p = 2$

FIGURE 17. Number of Krylov iteration per Newton iteration for the linear advection equation in dependence of the CFL number ($N_c = 640$).

6.2.2. *2-D Euler: Vortex advection.* We consider the advection of a vortex for the two-dimensional Euler equations. The numerical set-up is exactly the same as in Section 5.3.5. The cells are ordered in the direction of the vortex advection and we use the symmetric version of block Gauss-Seidel (GS). Figure 18 shows a comparison of the Krylov iteration numbers. Using block GS the iteration numbers drop to about 3 ($p = 1$) or 4 ($p = 2$),



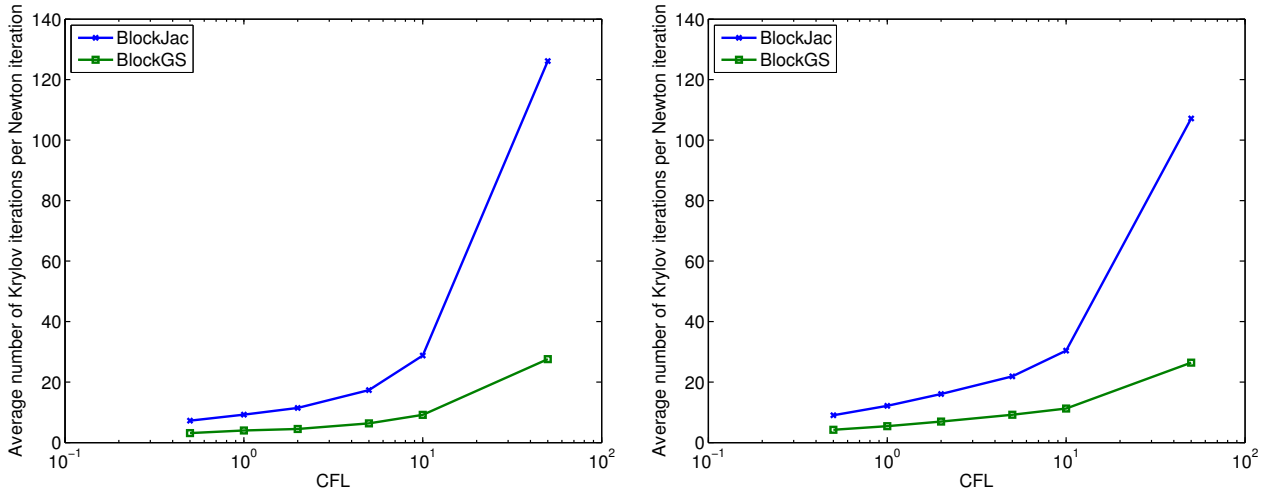
(a) $p = 1$ (b) $p = 2$

FIGURE 18. Number of Krylov iteration per Newton iteration for the vortex advection problem.

which is about one half of the numbers for the block Jacobi preconditioned system. However, as the symmetric

block GS is twice as expensive (per step) as the block Jacobi. Hence, there is no gain in performance with the block GS over the block Jacobi at this low CFL ($C^{CFL} = 0.5$).

The number of iterations with respect to increasing CFL number are shown in Figure 19. Here, both for block Jacobi and block GS, the iteration numbers grow with higher CFL. The increase for block Jacobi is bigger than for block GS. Recalling that the symmetric block GS iteration is about twice as costly as a block Jacobi iteration, the block GS is still better by a factor of about 2 for $CFL = 50$ as it needs only about 28 instead of 126 ($p = 1$) or only about 26 instead of 107 Krylov iterations ($p = 2$). Thus, even if information is no longer flowing in just one direction (the advection direction) but in all directions, the performance gain with block GS over block Jacobi is less striking but still observable.



(a) $p = 1$ (b) $p = 2$

FIGURE 19. Number of Krylov iteration per Newton iteration for the vortex advection problem in dependence of the CFL number ($N_c = 3312$).

6.3. **NACA 0012 aerofoil.** As the final numerical experiment, we consider an Euler flow around a NACA 0012 aerofoil. The aerofoil is placed along the x axis, ranging from $x = 0$ (head) to $x = 1$ (tail). Slip boundary conditions are used on the aerofoil. An artificial outer boundary is placed on a circle around $(2, 0)$ with radius 4, where the following freestream values are prescribed: Mach number $\mathrm{Ma}_\infty = 0.75$, pressure $p_\infty = 8.5419$, density $\rho_\infty = 11.4452$ and an angle of attack of 4°. We will compute and display the pressure coefficient $c_p = (p - p_\infty)/((1/2)\rho_\infty \|\mathbf{u}_\infty\|^2)$, where $\mathbf{u}_\infty$ is the freestream flow velocity. At $t = 0$, the flow is initialized by freestream values. The equations are then solved up to $t = 3.5$; the time by which the steady state is approximately reached.

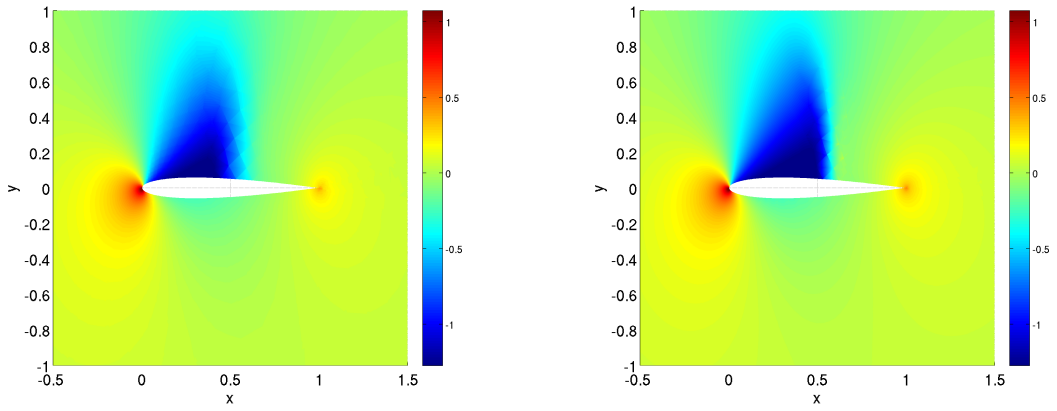
An unstructured mesh (consisting of triangles) is generated around the aerofoil. This mesh is finer near the head of the aerofoil than near the tail. As a further modification, we replace the shock capturing operator with a pressure scaled variant suggested in [20], i.e., (3.10b) is replaced by

$$D^{SC}_{n,K} = \frac{D^p_{n,K}\left(\Delta x C^{SC}\overline{\mathrm{Res}}_{n,K} + \Delta x (\Delta t^n)^{-1/2}\bar{C}^{SC}\overline{\mathrm{BRes}}_{n,K}\right)}{\sqrt{\int_{I^n}\int_K \left(\frac{(\Delta t^n)^2}{\Delta x^2}\left\langle \mathbf{V}_t^{\Delta x}, \mathbf{U}_{\mathbf{V}}(\tilde{\mathbf{V}}_{n,K})\mathbf{V}_t^{\Delta x}\right\rangle + \sum_{k=1}^{d}\left\langle \mathbf{V}_{x_k}^{\Delta x}, \mathbf{U}_{\mathbf{V}}(\tilde{\mathbf{V}}_{n,K})\mathbf{V}_{x_k}^{\Delta x}\right\rangle\right) dxdt + \epsilon}} \tag{6.4}$$

with

$$D^p_{n,K} = \Delta x_K^2 \frac{\frac{1}{\Delta t^n}\frac{1}{|K|}\int_{I^n}\int_K \sqrt{\sum_{k=1}^d p_{x_k x_k}^2}\, dxdt}{\frac{1}{\Delta t^n}\frac{1}{|K|}\int_{I^n}\int_K p dxdt}. \tag{6.5}$$

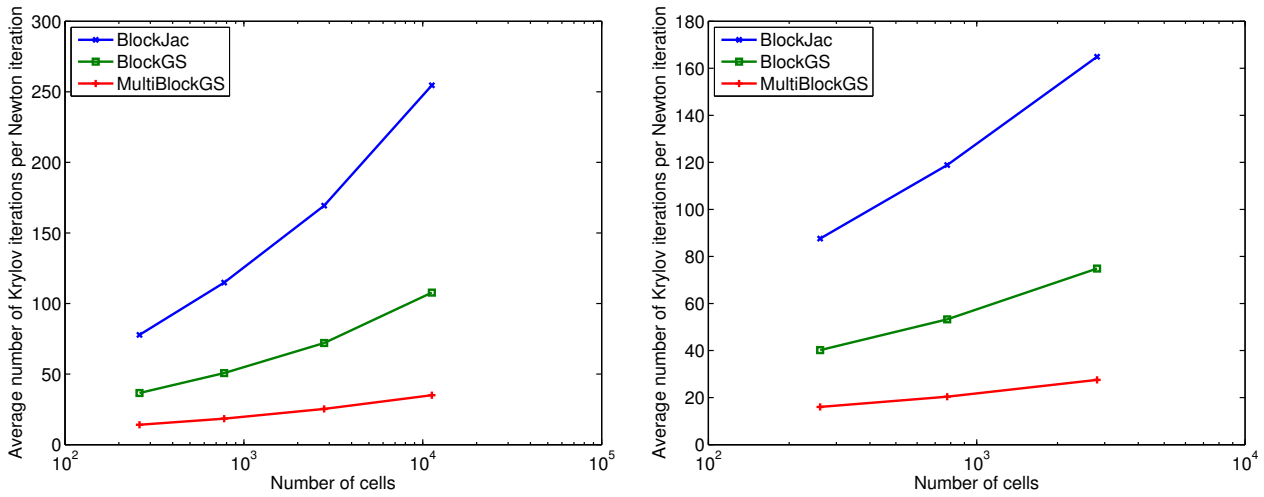
Snaphsots of the computed pressure coefficient c_p are shown in Figure 20. As the steady state is the object of interest in the current computation, both results are obtained with a very large time step resulting from $C^{CFL} = 500$, in order to accelerate convergence to the steady state. The results show that the transonic flow is resolved quite well with the shock capturing space time DG method. The shocks are resolved sharply and without strong oscillations. Furthermore, the smooth regions are also approximated well. The resolution is noticeably better with piecewise quadratic basis functions.



(a) $p = 1$ (b) $p = 2$

FIGURE 20. NACA 0012 aerofoil Euler equations, pressure coefficient c_p, $N_c = 2809$.

As we are also interested in the performance of the preconditioners developed herein, we plot the average number of Krylov iterations per Newton step in this problem in Figure 21. In addition to the block Jacobi and



(a) $p = 1$ (b) $p = 2$

FIGURE 21. Number of Krylov iteration per Newton iteration for the Euler NACA0012 aerofoil flow ($CFL = 500$).

the block GS (forward sweep), a further variant of the block GS, the multi-block GS, is also considered. This preconditioner consists of four sweeps; two forward sweeps, one based on the standard ordering and another

on an alternate ordering, coupled with two backward sweeps, performed with the directions reversed. The cell are ordered in the free stream direction and the alternative ordering is in the orthogonal direction. The results from Figure 21 show that the number of iterations is quite high. This is not unexpected, given the very high CFL number that is employed. Nevertheless, all the preconditioners lead to a performance that is very mildly dependent on the mesh size. The block Jacobi preconditioner has the worst performance, in terms of number of iterations as well as their growth with mesh size whereas the block GS and the multi-block GS lead to only a moderate growth. Given the fact that the multi-block GS is more expensive per step, both the block GS preconditioners show a similar performance in this realistic test case.

7. Conclusion

The shock capturing streamline diffusion DG methods, designed and analyzed in the recent paper [20] are particularly attractive for computing approximations of systems of hyperbolic conservation laws as they are

- Arbitrarily high-order accurate
- Fully discrete.
- Well-defined and robust on unstructured grids for approximating multi-dimensional domains with complicated geometry.
- Entropy stable.
- Convergent to entropy solutions of scalar conservation laws as well as entropy measure valued solutions of systems of conservation laws.
- CFL number and time steps can be very large.

However, this class of methods are implicit and require a large nonlinear algebraic system of equations to be solved at every time step. The nonlinear algebraic system is solved using a Newton method. The resulting linear systems (at every Newton step) are large, sparse but non-symmetric. Iterative solvers such as GMRES need a well-conditioned linear system in order to converge within a reasonable number of iterations. This necessitates the design of efficient preconditioners for the underlying Jacobians.

In this paper, we describe two sets of preconditioners to efficiently solve the linear system. The first set is of a block Jacobi type. We employ Fourier analysis to demonstrate that the resulting preconditioned system is well-conditioned. Numerical experiments illustrate that the block Jacobi preconditioner performs in an mesh independent manner and is far superior (in terms of the number of iterations) to the unpreconditioned system.

However, the block Jacobi method is not as robust to increasing the time step (CFL number). Since, one of the attractive features of the space-time DG method is the ability to take very large time steps, we propose non-local preconditioners of the block Gauss-Seidel type for this case. These preconditioners are found to perform quite well, even for very large CFL numbers.

A large number of numerical experiments, including examples of flows past aerofoils are presented to demonstrate the performance of the preconditioners. Comparing the two sets of preconditoners, the block Jacobi preconditioner is clearly well suited for low to moderate CFL numbers. However, the block Gauss-Seidel preconditioner is suitable for problem requiring very large time steps (high CFL numbers). One outstanding issue with the block GS preconditioner is that of parallelization as it requires blocks to inverted sequentially. This issue will be addressed in the future. Furthermore, we employ the preconditioned shock capturing space-time DG method to compute all speed flows in [21] and realistic flows on space-time adapted grids in [22].

References

[1] T. J. Barth. Numerical methods for gas-dynamics systems on unstructured meshes. In *An Introduction to Recent Developments in Theory and Numerics of Conservation Laws* pp 195–285. Lecture Notes in Computational Science and Engineering volume 5, Springer, Berlin. Eds: D. Kroner, M. Ohlberger, and Rohde, C., 1999

[2] T. J. Barth. An introduction to upwind finite volume and finite element methods: some unifying and contrasting themes. In *VKI Lecture series 2006-01, 34th CFD-higher order discretization methods (EUA4X)*, 2006.

[3] E. Chiodaroli, C. De Lellis, O. Kreml. Global ill-posedness of the isentropic system of gas dynamics. Preprint, 2013.

[4] B. Cockburn and C-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Math. Comput.*, 52, 1989, 411–435.

[5] B. Cockburn, S-y. Lin; C-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. III. One-dimensional systems. *J. Comput. Phys.*, 84, 1989, 90–113.

[6] C. Dafermos. *Hyperbolic conservation laws in continuum physics.* Springer, Berlin, 2000.
[7] R. J. DiPerna. Measure valued solutions to conservation laws. *Arch. Rational Mech. Anal.,* 88(3), 223-270, 1985.
[8] D. A. Dunavant. High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle. *International Journal for Numerical Methods in Engineering*, 21, 1985, 1129–1148.
[9] U. S. Fjordholm, S. Mishra and E. Tadmor. Energy preserving and energy stable schemes for the shallow water equations. *"Foundations of Computational Mathematics"*, Proc. FoCM held in Hong Kong 2008 (F. Cucker, A. Pinkus and M. Todd, eds), London Math. Soc. Lecture Notes Ser. 363, pp. 93-139, 2009.
[10] U. S. Fjordholm, S. Mishra and E. Tadmor. Arbitrary order accurate essentially non-oscillatory entropy stable schemes for systems of conservation laws. *SIAM J. Num. Anal.,* 50 (2), 544-573, 2012.
[11] U. S. Fjordholm, R. Käppeli, S. Mishra and E. Tadmor. Construction of approximate entropy measure valued solutions for hyperbolic systems of conservation laws. *Preprint*, arXiv:1402.0909, 2014.
[12] U. S. Fjordholm. *High-order accurate entropy stable numerical schemes for hyperbolic conservation laws.* ETH Zürich dissertation Nr. 21025, 2013.
[13] U. S. Fjordholm and S. Mishra. Convergence of entropy stable finite difference schemes to measure valued solutions of hyperbolic systems of conservation laws. *In preparation*, 2014.
[14] H. Lim, Y. Yu, J. Glimm, X. L. Li and D. H. Sharp. Chaos, transport and mesh convergence for fluid mixing. *Act. Math. Appl. Sin.,* 24 (3), 355-368, 2008.
[15] Edwige Godlewski and Pierre A. Raviart. *Hyperbolic Systems of Conservation Laws.* Mathematiques et Applications, Ellipses Publ., Paris (1991).
[16] W. Hackbusch and T. Probst. Downwind Gauß-Seidel smoothing for convection dominated problems. *Numerical linear algebra with applications*, 4(2):85–102, 1997.
[17] A. Harten, B. Engquist, S. Osher and S. R. Chakravarty. Uniformly high order accurate essentially non-oscillatory schemes, III. *J. Comput. Phys.*, 1987, 231-303.
[18] R. Hiptmair, R. Jeltsch and D. Kressner. Numerische Mathematik für Studiengang Rechnergestützte Wissenschaften. Lecture Notes, ETH Zurich, 2007.
[19] A. Hiltebrand, U. Koley and S. Mishra. An arbitrarily high order accurate convergent DG method for scalar conservation laws. *In preparation,* 2014.
[20] A. Hiltebrand and S. Mishra. Entropy stable shock capturing space-time discontinuous Galerkin schemes for systems of conservation laws. *Numerische Mathematik*, 126(1), 103–151, 2014.
[21] A. Hiltebrand and S. Mishra. Computation of all speed flows using a shock capturing space-time discontinuous Galerkin method. *In preparation*, 2014.
[22] A. Hiltebrand. Numerical approximation of hyperbolic systems of conservation laws with a shock capturing space-time discontinuous Galerkin method. *In preparation*, 2014.
[23] T. J. R Hughes, L. P. Franca and M. Mallet. A new finite element formulation for CFD I: Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Comp. Meth. Appl. Mech. Eng.*, 54, 1986, 223 - 234.
[24] F. Ismail and P. L. Roe. Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks *Journal of Computational Physics* 228(15), volume 228, 54105436, 2009
[25] J. Jaffre, C. Johnson and A. Szepessy. Convergence of the discontinuous galerkin finite element method for hyperbolic conservation laws. *Math. Model. Meth. Appl. Sci.,* 5(3), 367-386, 1995.
[26] C. Johnson and A. Szepessy. On the convergence of a finite element method for a nonlinear hyperbolic conservation law. *Math. Comput.*, 49 (180), 1987, 427-444.
[27] C. Johnson, P. Hansbo and A. Szepessy. On the convergence of shock capturing streamline diffusion methods for hyperbolic conservation laws. *Math. Comp.,* 54 (189), 107-129, 1990.
[28] R. J. LeVeque. Finite volume methods for hyperbolic problems. *Cambridge university press,* Cambridge, 2002.
[29] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, July 1986.
[30] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory schemes - II, *J. Comput. Phys.*, 83, 1989, 32 - 78.
[31] C. W. Shu. High-order ENO and WENO schemes for Computational fluid dynamics. *In High-order methods for computational physics, T. J. Barth and H. Deconinck eds.,* Lecture notes in computational science and engineering 9, Springer Verlag, 1999, 439-582.
[32] E. Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws, I. *Math. Comp.,* 49, 91-103, 1987.
[33] E. Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Act. Numerica,* 451-512, 2003

(Andreas Hiltebrand)
Seminar for Applied Mathematics (SAM)
Department of Mathematics, ETH Zürich,
HG J 49, Zürich -8092, Switzerland
E-mail address: andreas.hiltebrand@sam.math.ethz.ch

(Siddhartha Mishra)
Seminar for Applied Mathematics (SAM)
Department of Mathematics, ETH Zürich,
HG G 57.2, Zürich -8092, Switzerland
E-mail address: smishra@sam.math.ethz.ch

Recent Research Reports

Nr.	Authors/Title

2013-44 G. Coclite and S. Mishra and N. Risebro and F. Weber
Analysis and Numerical approximation of Brinkman regularization of two-phase flows in porous media

2013-45 M. Hutzenthaler and A. Jentzen
Numerical approximations of stochastic differential equations with non-globally Lipschitz continuous coefficients

2013-46 G. Da Prato and A. Jentzen and M. Röckner
A mild Itô formula for SPDEs

2013-47 P. Grohs and S. Keiper and G. Kutyniok and M. Schaefer
Parabolic Molecules: Curvelets, Shearlets, and Beyond

2013-48 R. Hiptmair and C. Jerez-Hanckes and C. Urzua
Optimal Operator Preconditioning for Boundary Elements on Open Curves

2013-49 P. Grohs and S. Hosseini
ε-Subgradient Algorithms for Locally Lipschitz Functions on Riemannian Manifolds

2013-50 A. Andersson and R. Kruse and S. Larsson
Duality in refined Watanabe-Sobolev spaces and weak approximations of SPDE

2014-01 M. Eigel and C.J. Gittelson and Ch. Schwab and E. Zander
A convergent adaptive stochastic Galerkin finite element method with quasi-optimal spatial meshes

2014-02 R. Kaeppeli and S. Mishra
Structure preserving schemes

2014-03 K. Grella
Sparse tensor phase space Galerkin approximation for radiative transport