



# LUND UNIVERSITY

## Multigrid Preconditioners for the Discontinuous Galerkin Spectral Element Method Construction and Analysis

Versbach, Lea Miko

2020

*Document Version:*

Publisher's PDF, also known as Version of record

[Link to publication](#)

*Citation for published version (APA):*

Versbach, L. M. (2020). *Multigrid Preconditioners for the Discontinuous Galerkin Spectral Element Method: Construction and Analysis*. [Licentiate Thesis, Centre for Mathematical Sciences]. Lund University.

*Total number of authors:*

1

*Creative Commons License:*

Unspecified

**General rights**

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

**Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00

# Multigrid Preconditioners for the Discontinuous Galerkin Spectral Element Method



# Multigrid Preconditioners for the Discontinuous Galerkin Spectral Element Method

## Construction and Analysis

by Lea Miko Versbach



**LUND**  
UNIVERSITY

Licentiate Thesis

Thesis advisors: Philipp Birken and Gregor J. Gassner

Faculty opponent: Matthias Bolten

To be presented, with the permission of the Faculty of Science of Lund University, for public criticism in the room MH:Hörmander at the Centre for Mathematical Sciences on Monday, the 14th of September

2020 at 13:00.

Organization <b>LUND UNIVERSITY</b> Centre for Mathematical Sciences Box 118 SE-221 00 LUND Sweden		Document name <b>Licentiate thesis</b>	
		Date of presentation <b>2020-09-14</b>	
		Sponsoring organization Swedish Research Council, grant number 2015-04133	
Author(s) Lea Miko Versbach			
Title and subtitle <b>Multigrid Preconditioners for the Discontinuous Galerkin Spectral Element Method: Construction and Analysis</b>			
Abstract  <p>Discontinuous Galerkin (DG) methods offer a great potential for simulations of turbulent and wall bounded flows with complex geometries since these high-order schemes offer a great potential in handling eddies. Recently, space-time DG methods have become more popular. These discretizations result in implicit schemes of high order in both spatial and temporal directions. In particular, we consider a specific DG variant, the DG Spectral Element Method (DG-SEM), which is suitable to construct entropy stable solvers for conservation laws. Since the size of the corresponding nonlinear systems is dependent on the order of the method in all dimensions, the problem arises to efficiently solve these huge nonlinear systems with regards to CPU time as well as memory consumption.</p> <p>Currently, there is a lack of good solvers for three-dimensional DG applications, which is one of the major obstacles why these high order methods are not used in e.g. industry. We suggest to use Jacobian-free Newton-Krylov (JFNK) solvers, which are advantageous in memory minimization. In order to improve the convergence speed of these solvers, an efficient preconditioner needs to be constructed for the Krylov sub-solver. However, if the preconditioner requires the storage of the DG system Jacobian, the favorable memory consumption of the JFNK approach is obsolete.</p> <p>We therefore present a multigrid based preconditioner for the Krylov sub-method which retains the low memory consumption, i.e. a Jacobian-free preconditioner. To achieve this, we make use of an auxiliary first order finite volume replacement operator. With this idea, the original DG mesh is refined but can still be implemented algebraically. As smoother, we consider the pseudo time iteration W3 with a symmetric Gauss-Seidel type approximation of the Jacobian. Numerical results are presented demonstrating the potential of the new approach.</p> <p>In order to analyze multigrid preconditioners, a common tool is the Local Fourier Analysis (LFA). For a space-time model problem we present this analysis and its benefits for calculating smoothing and two-grid convergence factors, which give more insight into the efficiency of the multigrid method.</p>			
Key words Discontinuous Galerkin Method, Finite Volume Method, Implicit Schemes, Local Fourier Analysis, Multigrid Method, Preconditioner, Space-Time Discretization, Spectral Element Method			
Classification system and/or index terms (if any)			
Supplementary bibliographical information		Language English	
ISSN and key title 1404-028X Licentiate theses in mathematical sciences		ISBN 978-91-7895-592-3 (print) 978-91-7895-593-0 (pdf)	
Recipient's notes		Number of pages 105	Price
		Security classification	

I, the undersigned, being the copyright owner of the abstract of the above-mentioned dissertation, hereby grant to all reference sources the permission to publish and disseminate the abstract of the above-mentioned dissertation.

Signature



Date July 22, 2020

# Multigrid Preconditioners for the Discontinuous Galerkin Spectral Element Method

Construction and Analysis

by Lea Miko Versbach



**LUND**  
UNIVERSITY

**Funding information:** The thesis work was financially supported by the Swedish Research Council, grant number 2015-04133.

Numerical Analysis  
Centre for Mathematical Sciences  
Lund University  
Box 118  
SE-221 00 Lund  
Sweden

[www.maths.lu.se](http://www.maths.lu.se)

Licentiate Thesis in Mathematical Sciences 2020:2  
ISSN: 1404-028X

ISBN: 978-91-7895-592-3 (print)

ISBN: 978-91-7895-593-0 (pdf)

LUNFNA-2005-2020

© Lea Miko Versbach, 2020

Printed in Sweden by Media-Tryck, Lund University, Lund 2020



Media-Tryck is a Nordic Swan Ecolabel certified provider of printed material. Read more about our environmental work at [www.mediatryck.lu.se](http://www.mediatryck.lu.se)

**MADE IN SWEDEN** 

# Abstract

Discontinuous Galerkin (DG) methods offer a great potential for simulations of turbulent and wall bounded flows with complex geometries since these high-order schemes offer a great potential in handling eddies. Recently, space-time DG methods have become more popular. These discretizations result in implicit schemes of high order in both spatial and temporal directions. In particular, we consider a specific DG variant, the DG Spectral Element Method (DG-SEM), which is suitable to construct entropy stable solvers for conservation laws. Since the size of the corresponding nonlinear systems is dependent on the order of the method in all dimensions, the problem arises to efficiently solve these huge nonlinear systems with regards to CPU time as well as memory consumption.

Currently, there is a lack of good solvers for three-dimensional DG applications, which is one of the major obstacles why these high order methods are not used in e.g. industry. We suggest to use Jacobian-free Newton-Krylov (JFNK) solvers, which are advantageous in memory minimization. In order to improve the convergence speed of these solvers, an efficient preconditioner needs to be constructed for the Krylov sub-solver. However, if the preconditioner requires the storage of the DG system Jacobian, the favorable memory consumption of the JFNK approach is obsolete.

We therefore present a multigrid based preconditioner for the Krylov sub-method which retains the low memory consumption, i.e. a Jacobian-free preconditioner. To achieve this, we make use of an auxiliary first order finite volume replacement operator. With this idea, the original DG mesh is refined but can still be implemented algebraically. As smoother, we consider the pseudo time iteration  $W_3$  with a symmetric Gauss-Seidel type approximation of the Jacobian. Numerical results are presented demonstrating the potential of the new approach.

In order to analyze multigrid preconditioners, a common tool is the Local Fourier Analysis (LFA). For a space-time model problem we present this analysis and its benefits for calculating smoothing and two-grid convergence factors, which give more insight into the efficiency of the multigrid method.





# List of Publications

Publications concerning the work of this thesis have been made as follows:

**I Finite volume based multigrid preconditioners for discontinuous Galerkin methods**

Lea M. Versbach, Philipp Birken, Gregor J. Gassner  
PAMM (Proceedings in Applied Mathematics and Mechanics), Vol 18 (2018)

**II Subcell finite volume multigrid preconditioning for high-order discontinuous Galerkin methods**

Philipp Birken, Gregor J. Gassner, Lea M. Versbach  
International Journal of Computational Fluid Dynamics (2019)

The publications can be found in the end of this thesis, sorted by publication year. The proceedings article contains the novel idea for constructing the preconditioner and first numerical results for a linear problem. In the second article, a nonlinear problem is considered and numerical tests for the constructed multigrid preconditioner are presented.

All papers are reproduced with the permission of the respective publisher.



## Acknowledgements

This licentiate thesis would not have been possible without the help of others and I want to thank all those that supported me along the way.

Most importantly, thanks to my main supervisor Philipp Birken, for your support, guidance, patience and motivating words when I needed them and everything you have taught me so far about research and science. Many thanks also to my co-supervisor Gregor Gassner for all the good ideas, help, guidance and optimism you spread.

Next I want to thank the Numerical Analysis group, both the current members as well as the ones who have left the group, for creating a nice work environment. I would like to offer my special thanks to Claus Führer, who supported me when I decided to do a PhD and still has a lot of motivating words and advices during all the ups and downs doing a PhD includes. I also wish to acknowledge the help provided by Viktor Linders, thanks for proofreading this thesis and discussing patiently a lot of things with me.

Thanks to my fellow PhD students both in the mathematical department as well as in NDR, some of you have become really good friends and it feels good to share this intense PhD time with all of you!

Finally, I want to thank my family for always believing in me, supporting me and giving me the opportunity to be where I am today.

Last but not least my biggest thanks to Niclas, who is always on my side, both in the less and more stressful times. Thanks for your endless patience, help, understanding and that you always believe in me.

## Funding

The work in this thesis is made possible due to the support of the Swedish Research Council, grant number 2015-04133.



# Contents

Abstract . . . . .	vii
List of Publications . . . . .	ix
Acknowledgements . . . . .	xi
<b>1 Introduction</b>	<b>1</b>
1 Organization of the Thesis . . . . .	4
<b>2 Discontinuous Galerkin Spectral Element Methods</b>	<b>5</b>
1 Method of Lines: DG with Implicit Time-Stepping Schemes . . . . .	6
1.1 DG-SEM Discretization . . . . .	9
1.2 Numerical Fluxes . . . . .	11
1.3 Finite Volume Discretization . . . . .	12
2 Space-Time DG-SEM . . . . .	13
3 Equivalence of DG Approximations to Lobatto Schemes . . . . .	16
<b>3 Solving Nonlinear Systems</b>	<b>19</b>
1 Newton's Method . . . . .	19
2 Krylov Subspace Methods . . . . .	21
2.1 GMRES . . . . .	22
2.2 Preconditioning . . . . .	22
3 Jacobian-free Newton-GMRES . . . . .	23
<b>4 Multigrid Methods</b>	<b>25</b>
1 Classical Iterative Schemes . . . . .	25
2 Multigrid Methods for Linear Systems . . . . .	27
2.1 Multigrid Components . . . . .	30
2.2 Smoothers . . . . .	31
2.3 Multigrid Preconditioners . . . . .	32
<b>5 Local Fourier Analysis</b>	<b>35</b>
1 Model Problem . . . . .	36
1.1 LFA Basics for the Model Problem . . . . .	39
2 Smoothing Analysis . . . . .	42

2.1	Semi-Coarsening in Time . . . . .	46
2.2	Full Space-Time Coarsening . . . . .	50
2.3	Concluding Remarks on Smoothing Factors . . . . .	51
3	Two-Grid Analysis . . . . .	51
3.1	Semi-Coarsening in Time . . . . .	57
3.2	Full Space-Time Coarsening . . . . .	59
3.3	Asymptotic Convergence Factors . . . . .	61
3.4	Numerical Results . . . . .	63
<b>6</b>	<b>Conclusions and Outlook</b>	<b>67</b>
1	Conclusion . . . . .	67
2	Future work . . . . .	68
	<b>Bibliography</b>	<b>74</b>
	<b>Scientific publications</b>	<b>75</b>
	Author contributions . . . . .	75
	Paper I: Finite volume based multigrid preconditioners for discontinuous Galerkin methods . . . . .	77
	Paper II: Subcell finite volume multigrid preconditioning for high-order dis- continuous Galerkin methods . . . . .	81

# Chapter I

## Introduction

The goal of our research are Large Eddy Simulation (LES) of turbulent and wall bounded flows with complex geometries, which are needed for example for the design of next generation jet engines, air frames and wind turbines. These complex multi-scale problems may have on the order of 100 million unknowns, thus a fast low memory parallel solver is needed. Furthermore, the problems are of such a nature that stable time integration requires efficient solvers for linear and nonlinear equation systems. In this thesis we focus on different aspects in the construction of efficient Jacobian-free multigrid preconditioners for high order Discontinuous Galerkin (DG) discretizations with implicit time integration.

DG methods are high order methods offering a great potential for LES since it has been shown that low-order schemes can contribute dramatically to the dissipation of eddies [41]. The idea of DG methods is to approximate the solution element-wise by a polynomial and, in contrast to the finite element method, allow discontinuities across element interfaces [32], [39]. The local computations on the elements are very dense and communication as well as coupling of the degrees of freedom (DOF) is done both across faces and within the elements. It has been shown that DG methods are very well suited for domain-decomposition-based parallelization due to its structure [30], [60]. In this thesis we consider a specific DG variant called DG Spectral Element Method (DG-SEM), see e.g. [40]. Here, a Lagrange type nodal basis with Gauss-Lobatto (GL) quadrature nodes is collocated with the discrete integration of the weak form. This gives DG operators that satisfy the summation-by-parts (SBP) property [23]. SBP is the discrete analogue to integration-by-parts and is key to construct discretely entropy stable and kinetic energy preserving methods [20], [63]. This ensures that the numerical scheme obeys the second law of thermodynamics since it provides a bound on the mathematical entropy at any



time according to the given initial and boundary conditions.

DG-SEM discretizations in the spatial directions result in big systems of stiff ODEs. In order to avoid time integration with very restrictive CFL conditions, we only consider implicit time integrators. This can be done either with a methods of lines (MOL) ansatz using standard implicit time stepping methods or a full discretization ansatz resulting in a space-time DG method.

Efficiency can only be achieved when the arising large non-linear systems are solved cheaply. Besides CPU time, the memory consumption should be minimized. Solvers for linear and nonlinear equation systems are severely lacking for 3D DG applications and are one of four major obstacles that need to be solved before industry adopts high order methods [61]. Possible solvers are Full-Approximation (FAS) multigrid schemes (MG) and preconditioned Jacobian-free Newton-Krylov methods (JFNK) [38]. The latter one are numerical methods to solve non-linear problems using Krylov subspace linear solvers in order to solve a linear sub-problem which includes the Jacobian of the system. For these solvers, multigrid can be used as preconditioner for the linear sub-problem [5]. MG methods are iterative methods designed for solving differential equations using a hierarchy of grids for the discretization, [58]. Moreover, these methods can be used as preconditioners, i.e. operators to transform a given problem into a form that is more suitable for the Krylov subspace solver and therefore improve convergence speed. Multigrid preconditioned JFNK methods have been analyzed for the RANS equations in [7].

The JFNK technology is interesting in the sense of memory minimization. Although DG systems have a block structure which are coupled via the faces, the blocks themselves can be very large. In particular, the number of unknowns per element increases dramatically with increasing polynomial degree and dimension, leading to large dense Jacobian blocks, see [4], [6]. For a  $d$ -dimensional problem, the block size of a finite volume method is  $2 + d$ , whereas for a DG-SEM method with  $p$ -th degree polynomials, it is  $(d + 2)(p + 1)^d$ . For degree 2 in 3D, this already increases to 135. If the preconditioner requires the storage of the DG system Jacobian the favorable memory consumption of the JFNK approach is obsolete and the method is not fully Jacobian-free any longer.

In this thesis we present a novel idea for the construction of a well-performing preconditioner for the JFNK approach, while retaining the low memory use, i.e. a Jacobian free multigrid based preconditioner [59], [8]. In Figure 1.1 we show a visualization of our idea to construct MG based implicit space(-time) preconditioners for DG solvers. We start by discretizing the PDE with an implicit DG method (either MOL or space-time DG) and solve the resulting algebraic system with a JFNK method. In order to improve convergence speed we construct a multigrid preconditioner based on a lower order replacement operator for the GMRES sub-solver.

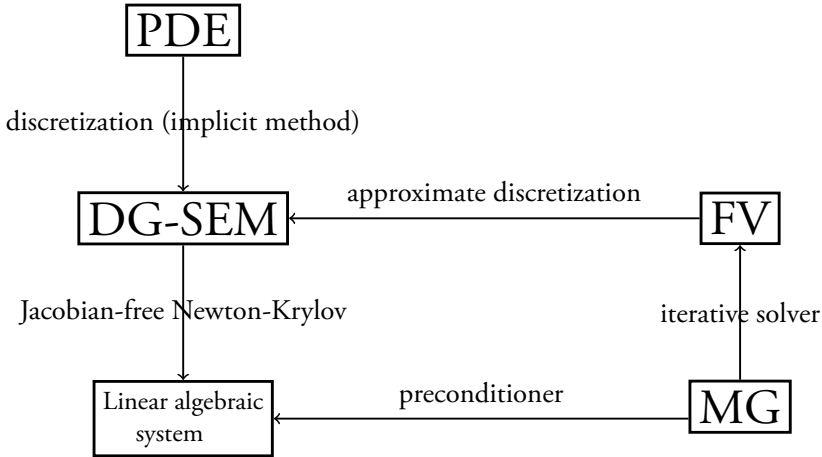


Figure 1.1: Work flow to construct a MG based preconditioner using a FV replacement operator to solve implicit DG-SEM discretizations.

Thus, the core idea is to construct a simplified replacement operator in order to avoid calculating the Jacobian of the DG operator. One could for instance choose a different polynomial order in the element to generate a replacement operator as presented in [6], [17] and [46]. However, we want to keep the number of degrees of freedom in the replacement operator the same. This can be achieved by introducing a subcell grid in each element. On the subcell-element grid, the simplest replacement operator is a first order finite volume (FV) discretization. A motivation for this choice is the equivalence between a DG-SEM discretization and a high order FV discretization proved in [19]. In the resulting approximate Jacobian, we only have  $(d+2)(p+1)(2d+1)$  entries [4]. Moreover, this replacement operator allows to use the available knowledge about fast multigrid (MG) methods for FV discretizations on block structured meshes. As a smoother for our FV discretization we use a state of the art low memory  $W_3$  smoother from [7] which has been show to work very efficient for this MG setup.

In [1] a related approach was presented for spectral difference methods with a FV replacement operator on a potentially fine grid, which is overlaid and not embedded. This makes it necessary to interpolate the solution between different grids, which we avoid with the embedded FV grid we suggest.

The efficiency of the preconditioner is directly influenced by the efficiency of the multigrid method. In order to gain more insight into MG methods, we perform a local Fourier Analysis (LFA), an important tool to analyze the quality of MG methods first presented in [10]. Usually, a two-grid LFA is applied to study the design and the structure of a MG method [58]. Moreover, the quality of the smoother can be analyzed using a so-called

smoothing factor. We perform an LFA for a space-time DG multigrid model problem. Similar analysis for different problems have been presented in [21], [27], [31], [37], [49].

## I Organization of the Thesis

In Chapter 2 we present the DG-SEM discretization with implicit time discretization. We discuss both an method of lines and a space-time ansatz. We give an overview over Newton-Krylov solvers for the resulting nonlinear equation systems in Chapter 3.

The core part of this thesis is to present an idea to construct efficient preconditioners for these solvers and to analyze them. In Chapter 4 we present MG methods and discuss how they can be used as preconditioners. We then show how to study the efficiency of these methods using Local Fourier Analysis for a model problem in Chapter 5.

Conclusions and future work are found in Chapter 6. Numerical results from previous publications are presented in the end of the thesis.

## Chapter 2

# Discontinuous Galerkin Spectral Element Methods

The goal of this work is to numerically approximate solutions of a  $d$ -dimensional conservation law

$$\mathbf{u}_t + \nabla \cdot \mathbf{f}(\mathbf{u}) = 0, \mathbf{x} \in \Omega \subset \mathbb{R}^d, t \in [0, T], \quad (2.1)$$

with initial and boundary conditions. In this chapter we present the discretization technique we use in two variants.

Discontinuous Galerkin (DG) methods were first introduced 1973 in [48]. These schemes are a combination of finite element (FE) and finite volume (FV) schemes since they are derived from the weak form of PDEs with basis functions on elements, which are connected using numerical surface fluxes. DG methods have less limitations than FE schemes in the following aspects: Standard finite element methods assume continuity on the interfaces between two neighboring elements. In consequence, global continuity on the considered domain is assumed. This causes problems when solving hyperbolic conservation laws, as their solutions are typically discontinuous. The DG method allows discontinuities at the interfaces and is therefore more suitable for these problems. Another advantage of the DG method is that the order of accuracy can be improved by simply increasing the number of nodes within each element. Good introductions to the DG methods we consider in this thesis can be found in [28] and [39].

Numerical schemes for conservation laws do not necessarily describe the correct physical behavior of a fluid. One reason for this problem is that most numerical scheme do not obey the second law of thermodynamics and are therefore not conservative. Over the

last decades, numerical schemes which are both conservative and obey the second law of thermodynamics have been developed. These schemes are called entropy stable. Tadmor was the first one to develop conservative and entropy stable schemes for low order finite volume methods [55]. Based on this, entropy-conservative and entropy-stable fluxes have been constructed [45], [56], [57].

High-order, conservative and entropy stable boundary schemes for linear conservation laws have been developed in [13] based on summation-by-parts (SBP) operators. These operators were first designed 1974 in [42] and mimic integration by parts. In the continuous entropy analysis, e.g when using Lax-Friedrich in stability proofs, integration by parts is applied several times. Thus, SPB operators allow to mimic the continuous analysis in the discrete case.

For standard DG methods applied to systems of conservation laws,  $L^2$  stability can not be guaranteed at the boundaries. It has been shown in [12] and [23] that discontinuous Galerkin spectral element methods (DG-SEM) with Legendre-Gauss-Lobatto (LGL) nodes satisfy the discrete SBP property. The boundary and interface conditions can be weakly imposed by a so called simultaneous approximation term (SAT), see [53]. The conservative and entropy stable schemes for low order methods developed by Tadmor were then extended to high order schemes with SBP operators in [23] and [18].

All schemes mentioned so far are based on semi-discrete discretizations. We will in this thesis not go into all details about constructing entropy stable DG methods and refer to [20] for the derivation of fully discrete entropy stable schemes.

We start by discussing a method of lines approach, where we first discretize the problem in space to transform it into a system of ordinary differential equations and then discretize in time. We present the choices that lead to a DG-SEM method and give a short overview over numerical fluxes. Next, we present a space-time ansatz where we discretize simultaneously in space and time using DG. We finish this chapter by showing the equivalence of DG approximations to Gauss Lobatto schemes.

## I Method of Lines: DG with Implicit Time-Stepping Schemes

In this section we present a semi-discrete method, where only the spatial direction is discretized using DG. For simplicity we derive the DG discretization for the one-dimensional case, i.e.  $d = 1$  in (2.1). It can be easily extended to the multi-dimensional case using a Kronecker product ansatz. This is discussed in the next section.

We start by dividing the computational domain  $\Omega$  into  $N$  non-overlapping elements

$e^k = [x_L^k, x_R^k]$ , then  $\Omega = \bigcup_{k=1}^N e^k$ . To derive the weak form of the problem, we multiply (2.1) by a test function  $\psi(\mathbf{x})$  and integrate over the spatial domain

$$\int_{\Omega} (\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x) \psi dx = 0. \quad (2.2)$$

We require that the test functions satisfy on each element

$$\int_{e^k} \mathbf{u}_t \psi dx + \int_{e^k} \mathbf{f}(\mathbf{u})_x \psi dx = 0, \quad k = 1, \dots, N. \quad (2.3)$$

It is of advantage to map each element to the reference element  $\hat{\Omega} = [-1, 1]$  in order to derive the DG-SEM method later. This is done using the linear mapping

$$\xi(x) = 2 \frac{x - x_L^k}{\Delta x^k} - 1, \quad \Delta x^k = x_R^k - x_L^k.$$

Then (2.3) becomes

$$\frac{\Delta x^k}{2} \int_{-1}^1 \mathbf{u}_t \psi d\xi + \int_{-1}^1 \mathbf{f}(\mathbf{u})_{\xi} \psi d\xi = 0, \quad k = 1, \dots, N. \quad (2.4)$$

We apply integration by parts to the second term in (2.4) to finally get the weak form of the problem

$$\frac{\Delta x^k}{2} \int_{-1}^1 \mathbf{u}_t \psi d\xi + \mathbf{f}(\mathbf{u}) \psi \Big|_{-1}^1 - \int_{-1}^1 \mathbf{f}(\mathbf{u}) \psi_{\xi} d\xi = 0, \quad k = 1, \dots, N. \quad (2.5)$$

The solution  $\mathbf{u}$  and the physical flux function  $\mathbf{f}$  are approximated on each element by a nodal polynomial of degree  $p_x$  written in the reference space:

$$\begin{aligned} \mathbf{u}(\xi, t)|_{e^k} &\approx \mathbf{u}^k(\xi, t) = \sum_{j=1}^{N_x} u_j^k(t) \varphi_j(\xi), \\ \mathbf{f}(\xi, t)|_{e^k} &\approx \mathbf{f}^k(\xi, t) = \sum_{j=1}^{N_x} f_j^k(t) \varphi_j(\xi), \end{aligned}$$

with time dependent coefficients  $u_j^k, f_j^k$  and a set of basis functions  $\{\varphi_j\}_{j=1}^{N_x}$  for  $N_x = p_x + 1$ . The choice of the basis functions gives rise to different DG methods. The global solution  $\mathbf{u}(\xi, t)$  is then approximated by a piecewise polynomial of degree  $p_x$ :

$$\mathbf{u}(\xi, t) \approx \mathbf{u}^{\text{num}}(\xi, t) = \bigoplus_{k=1}^N \mathbf{u}^k(\xi, t).$$

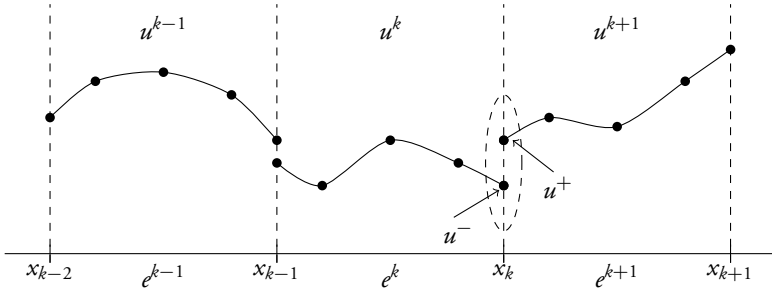


Figure 2.1: DG approximations are assumed to be continuous on elements, but not on interfaces.

Note that we only assume continuity of the polynomial approximation on the elements, but not on the interfaces, i.e. we allow for discontinuous numerical approximations. Thus the numerical solution will be of some form as visualized in Figure 2.1. In consequence, the surface term  $\mathbf{f}(\mathbf{u})\psi|_{-1}^1$  in (2.5) is not exchanging information with the neighboring elements, i.e. the elements are uncoupled. As in the finite volume method, this can be resolved by introducing a numerical surface flux  $\mathbf{f}^*(\mathbf{u}^-, \mathbf{u}^+)$ , which is a function of the interface values of the neighboring elements, see Figure 2.1. Different choices of numerical fluxes will be presented in the end of this section.

Moreover, we assume that the test function can also be written in the same basis

$$\psi(\xi) = \sum_{j=1}^{N_x} \psi_j \varphi_j(\xi).$$

Inserting the numerical solution, the numerical flux function and the numerical surface flux in (2.5) and choosing  $\psi_j \equiv 1$  we get the DG formulation in the weak form on the reference element  $\hat{\Omega} = [-1, 1]$

$$\frac{\Delta x^k}{2} \sum_{j=1}^{N_x} u_j^k(t) \int_{-1}^1 \varphi_j \varphi_i d\xi + \mathbf{f}^* \varphi_i|_{-1}^{+1} - \sum_{j=1}^{N_x} f_j^k(t) \int_{-1}^1 \varphi_j \varphi_{i\xi} d\xi = 0 \quad (2.6)$$

for  $i = 1, \dots, N_x$ . This can be rewritten in a more compact form by using inner products.

**Definition 2.1** ([28]). We define the local inner product and the corresponding  $\mathcal{L}^2$  norm on the reference element  $\hat{\Omega}$  by

$$(u, v)_{\mathcal{L}^2(\hat{\Omega})} := \int_{-1}^1 u v dx, \quad \|u\|_{\mathcal{L}^2(\hat{\Omega})} := (u, u)_{\mathcal{L}^2(\hat{\Omega})}.$$

For each element we define a mass matrix  $\tilde{\mathbf{M}} \in \mathbb{R}^{N_x \times N_x}$ , a derivative matrix  $\tilde{\mathbf{D}} \in \mathbb{R}^{N_x \times N_x}$  and a surface matrix  $\tilde{\mathbf{S}} \in \mathbb{R}^{N_x \times N_x}$  as

$$\begin{aligned}\tilde{M}_{ij} &:= (\varphi_j, \varphi_i)_{\mathcal{L}^2(\hat{\Omega})}, \\ \tilde{D}_{ij} &:= (\varphi_j, \varphi_{i\xi})_{\mathcal{L}^2(\hat{\Omega})}, \\ \tilde{S}_{ij} &:= -\delta_{1j}\varphi_i(-1) + \delta_{N_x j}\varphi_i(1),\end{aligned}\tag{2.7}$$

and

$$\begin{aligned}\mathbf{u}^k &= (u_1^k, \dots, u_{N_x}^k)^T, \\ \mathbf{f}^k &= (f_1^k, \dots, f_{N_x}^k)^T, \\ \mathbf{f}^* &= (f^*(\mathbf{u}^k(t, -1)), 0, \dots, 0, f^*(\mathbf{u}^k(t, 1)))^T.\end{aligned}\tag{2.8}$$

These matrices only depend on the basis and the geometry and not on the solution and can therefore be precomputed after the grid generation. Moreover, since all elements are mapped to the reference element, the matrices are the same on each element. Thus the weak form (2.6) can be written on each element  $e^k$  as

$$\frac{\Delta x^k}{2} \tilde{\mathbf{M}} \mathbf{u}_t^k = \tilde{\mathbf{D}}^T \tilde{\mathbf{M}} \mathbf{f}^k - \tilde{\mathbf{S}} \mathbf{f}^*, \quad k = 1, \dots, N.\tag{2.9}$$

This results in a system of ODEs which can be solved using any time-stepping method, e.g. Runge-Kutta methods.

## 1.1 DG-SEM Discretization

The DG spectral element method is based on collocating the interpolation nodes with the corresponding quadrature nodes. The integrals are approximated with Legendre-Gauss-Lobatto (LGL) quadrature on the reference interval using a Lagrange basis [29]. We can define the nodes and weights using Legendre polynomials [39].

**Definition 2.2** (Legendre-Gauss-Lobatto nodes and weights). Consider the Legendre polynomials defined recursively as

$$L_{n+1}(x) = \frac{2n+1}{n-1} x L_n(x) - \frac{n}{n+1} L_{n-1}(x),$$

with  $L_0(x) = 1$ ,  $L_1(x) = x$ . Then the LGL nodes include the endpoints of the reference interval,  $\pm 1$ , and the interior nodes are the roots of the polynomial

$$q(x) = L_{N+1}(x) - L_{N-1}(x).$$



The weights are given by

$$\omega_j = \frac{2}{N(N+1)(L_N(\xi_j))^2}.$$

As basis functions Lagrange polynomials of degree  $p_x$  based on the LGL nodes on the reference interval  $\hat{\Omega}$  are used.

**Definition 2.3** (Lagrange polynomial). For a given set of points  $\{x_1, \dots, x_N\}$  the  $j$ -th Lagrange polynomial of degree  $N-1$  is defined by

$$\ell_j(x) := \prod_{i=1, i \neq j}^N \frac{x - x_i}{x_j - x_i}, \quad j = 1, \dots, N,$$

satisfying the property

$$\ell_j(x_i) = \delta_{ji} := \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

The collocation of the interpolation nodes with the corresponding quadrature nodes results in a diagonal mass matrix  $\tilde{\mathbf{M}}$  in (2.7), since

$$M_{ji} = \int_{-1}^1 \ell_j \ell_i d\xi \approx \sum_{m=1}^{N_x} \omega_m \ell_j(\xi_m) \ell_i(\xi_m) = \omega_m \delta_{ij}. \quad (2.10)$$

LGL quadrature is only exact for polynomials of degree  $2N_x + 1$ . Exact integration of the Lagrange basis polynomials increases the cost of the integral approximation, thus we accept an integration error in (2.10). This is called mass lumping [22]. Applying the same quadrature gives the derivative matrix  $\tilde{\mathbf{D}}$  in (2.7)

$$D_{ij} = \int_{-1}^1 \ell_j \ell_{i\xi} d\xi \approx \sum_{m=1}^{N_x} \omega_m \ell_j(\xi_m) \ell_{i\xi}(\xi_m) = \omega_j \ell_{i\xi}(\xi_j). \quad (2.11)$$

This integral is calculated exactly even with the LGL quadrature because of its order. Since the mass matrix is nonsingular, we get

$$\mathbf{u}_r^k = -\frac{2}{\Delta x^k} \mathbf{M}^{-1} (\mathbf{Sf}^* - \mathbf{D}^T \mathbf{Mf}^k).$$

We can collect the left hand side in a long vector  $\mathbf{u} = [\mathbf{u}^1, \dots, \mathbf{u}^N]^T$  and the right hand side a nonlinear function  $\underline{\mathbf{G}}$ . Then we get an ODE system

$$\dot{\mathbf{u}} = \underline{\mathbf{G}}(\mathbf{u}). \quad (2.12)$$

Applying any time stepping method to (2.12), for example implicit Euler, yields

$$\underline{\mathbf{u}}^{n+1} - \underline{\mathbf{u}}^n - \Delta t \underline{\mathbf{G}}(\underline{\mathbf{u}}^{n+1}) =: \underline{\mathbf{F}}(\underline{\mathbf{u}}^{n+1}) = 0. \quad (2.13)$$

Therefore, a DG discretization in space with implicit time stepping methods results in solving a root problem in each time step. This will be discussed in Chapter 3.

## 1.2 Numerical Fluxes

The choice of the numerical flux is very important for the DG formulation in order to connect the elements. Since the numerical solution will be discontinuous at the element interfaces, a numerical flux function  $\mathbf{f}^*$  is defined on these interfaces by taking states from the left and the right side of the face and approximating the physical flux  $\mathbf{f}$  based on these states. The simplest idea to define a numerical flux function would be to use the average of the physical fluxes from the left and the right. Since this leads to an unconditionally unstable scheme, additional stabilizing terms are needed. A numerical flux function

$$\mathbf{f}^* := \mathbf{f}^*(\mathbf{u}^-, \mathbf{u}^+), \quad (2.14)$$

is called consistent if it is Lipschitz continuous and if  $\mathbf{f}(\mathbf{u}) = \mathbf{f}^*(\mathbf{u}, \mathbf{u})$ . This implies that for finer discretization, the numerical flux approximates the physical flux better. Here,  $\mathbf{u}^-$  denotes the value to the left of the considered node and  $\mathbf{u}^+$  the value to the right of the considered node. In this thesis we use the following two numerical fluxes.

### Upwind Flux

The simplest flux is the upwind flux, defined as

$$\mathbf{f}_{\text{up}}^*(\mathbf{u}^-, \mathbf{u}^+) = \frac{\mathbf{f}(\mathbf{u}^-) + \mathbf{f}(\mathbf{u}^+)}{2} + \frac{\hat{\mathbf{n}}^- \mathbf{f}(\mathbf{u}^-) + \hat{\mathbf{n}}^+ \mathbf{f}(\mathbf{u}^+)}{2}, \quad (2.15)$$

with the respective normal  $\hat{\mathbf{n}}$ .

### Rusanov Flux

The Rusanov flux, also known as local Lax-Friedrich flux, is defined as

$$\mathbf{f}_{\text{LLF}}^*(\mathbf{u}^-, \mathbf{u}^+) = \frac{\mathbf{f}(\mathbf{u}^-) + \mathbf{f}(\mathbf{u}^+)}{2} - \frac{\lambda_{\max}}{2}(\mathbf{u}^+ - \mathbf{u}^-), \quad (2.16)$$

with  $\lambda_{\max} = \max_{\mathbf{u}^-, \mathbf{u}^+} \left| \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|$  is an estimate of the maximum wave speed at the interface.

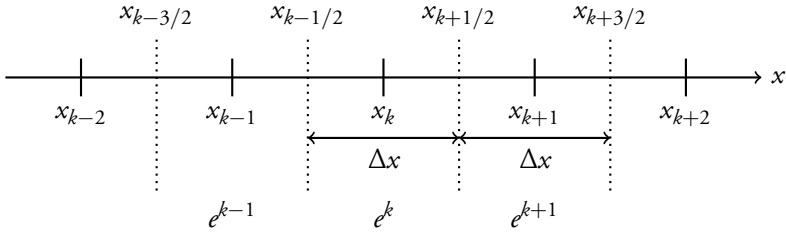


Figure 2.2: Finite volume grid in one dimension.

### 1.3 Finite Volume Discretization

We now want to present the easiest DG discretization with constant polynomials on each element. This reduces to the finite volume method, which is very common to apply to conservation laws (2.1). Due to the fact that the finite volume method is close to the physics of the flow system and since it is easy to implement, it is one of the standard methods in industry and academia and we use this discretization for constructing multigrid preconditioners. They correspond to a DG method with  $p_x = 0$  and the only node for each element is located at the center of the respective element. In order to simplify the notation, it is of advantage to denote the elements by  $e^k = [x_{k-1/2}, x_{k+1/2}]$ ,  $k = 1, \dots, N$ , and the nodes by  $x_k$ , see Figure 2.2.

A semi-discrete first order finite volume method on  $N$  elements with fixed element width  $\Delta x$  for a problem (2.1) is defined by

$$\mathbf{u}_{k,t} + \frac{1}{\Delta x} (\mathbf{f}_{k+1/2}^* - \mathbf{f}_{k-1/2}^*) = 0 \quad (2.17)$$

with  $\mathbf{u}_k := \mathbf{u}(x_k, t)$ ,  $k = 1, \dots, N$ , and the numerical flux  $\mathbf{f}_{k-1/2}^* := \mathbf{f}^*(\mathbf{u}_k, \mathbf{u}_{k-1})$ ,  $k = 2, \dots, N$ .

Let us as an example consider the simplest conservation law, namely the linear advection equation

$$\mathbf{u}_t + a\mathbf{u}_x = 0,$$

with  $a > 0$  and periodic boundary conditions. Since the information travels from left to right, it is most natural to use a first order upwind flux,  $\mathbf{f}_{k-1/2}^* = \mathbf{u}_{k-1}$ . Then an equidistant FV discretization with mesh width  $\Delta x$  gives an evolution equation for the cell average  $u_k$  in one cell  $k$

$$u_{k,t} + \frac{a}{\Delta x} (u_k - u_{k-1}).$$

With the vector  $\mathbf{u} = (u_1, \dots, u_N)^T$  and the matrix

$$\mathbf{B} = \begin{pmatrix} 1 & & & & -1 \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & & \ddots & \ddots \\ & & & & -1 & 1 \end{pmatrix}$$

we obtain the system of ODEs

$$\mathbf{u}_t + \frac{a}{\Delta x} \mathbf{B} \mathbf{u} = 0.$$

As before, this can be solved using any time-stepping method.

## 2 Space-Time DG-SEM

Instead of a semi-discrete method as presented in the previous section, we can also apply DG to the temporal direction. This results in a space-time DG method. The main focus in this section lies on the temporal DG discretization and we therefore consider again a one-dimensional conservation law

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0, \quad \mathbf{x} \in \Omega \subset \mathbb{R}, \quad t \in [0, T]. \quad (2.18)$$

An extension to the multi-dimensional case can be found in [20]. In the following we treat time like another spatial dimension and therefore start by discretizing the space-time domain  $\Omega \times [0, T]$  into non-overlapping elements. For simplicity we assume that all elements have width  $\Delta x$  and height  $\Delta t$ . An example for such a space-time grid can be seen in Figure 2.3. Let  $N_t$  denote the number of elements in time and  $N_x$  the number of elements in space. Then we have space-time elements  $e^{m,n} := [t_m, t_{m+1}] \times [x_n, x_{n+1}]$  for  $m = 1, \dots, N_t$  and  $n = 1, \dots, N_x$ .

We derive the space-time DG discretization as presented in [20] and similar to the semi-discrete case by multiplying the conservation law (2.18) by a test function  $\psi := \psi(t, x)$  and integrating in space and time over each element

$$\int_{t_m}^{t_{m+1}} \int_{x_n}^{x_{n+1}} (\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x) \psi \, dx dt = 0.$$

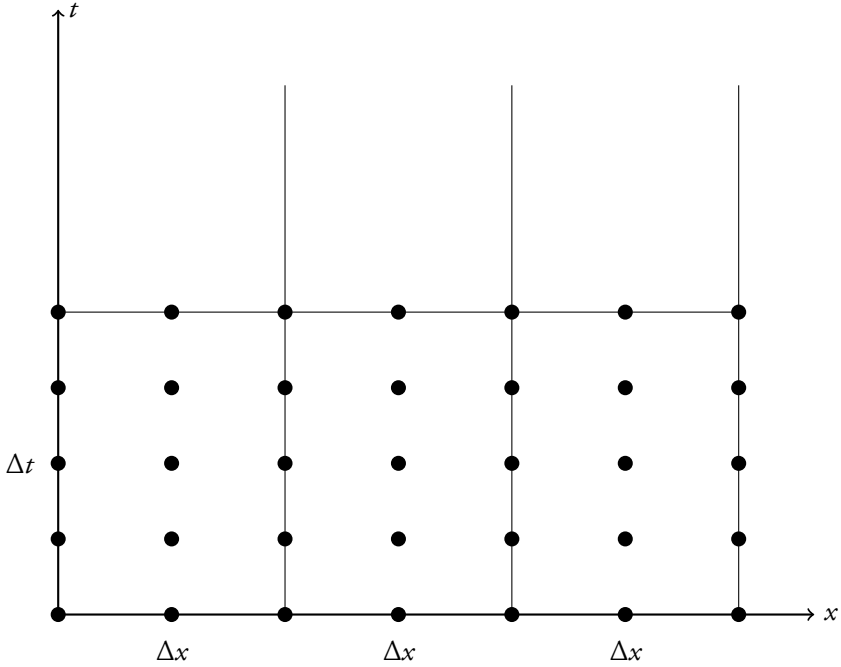


Figure 2.3: Space-time grid in 1D with three nodes in the spatial and four nodes in the temporal direction in each space-time slab.

As in the semi-discrete case, we map all elements to the reference element  $\hat{\Omega}^2 = [-1, 1]^2$  using the linear maps

$$\tau(t) = 2 \frac{t - t_m}{t_{m+1} - t_m} - 1,$$

$$\xi(x) = 2 \frac{x - x_n}{x_{n+1} - x_n} - 1.$$

The change of variables gives

$$\frac{2}{\Delta t} \int_{-1}^1 \int_{-1}^1 \mathbf{u}_\tau \psi d\tau d\xi + \frac{2}{\Delta x} \int_{-1}^1 \int_{-1}^1 \mathbf{f}(\mathbf{u})_\xi \psi d\tau d\xi = 0.$$

We now perform integration by parts in both temporal and spatial direction to obtain the weak form

$$\frac{2}{\Delta t} \int_{-1}^1 \left( \mathbf{u} \psi|_{-1}^{+1} - \int_{-1}^1 \mathbf{u} \psi_\tau d\tau \right) d\xi + \frac{2}{\Delta x} \int_{-1}^1 \left( \mathbf{f}(\mathbf{u}) \psi|_{-1}^{+1} - \int_{-1}^1 \mathbf{f}(\mathbf{u}) \psi_\xi d\xi \right) d\tau = 0.$$

For the space-time DG-SEM ansatz we approximate the solution  $\mathbf{u}$  and the physical flux  $\mathbf{f}$  on each space-time element  $e^{m,n}$  by polynomials of degree  $p_t$  in space and  $p_x$  in time:

$$\begin{aligned}\mathbf{u}(\tau, \xi) &\approx \sum_{i=1}^{N_\tau} \sum_{j=1}^{N_\xi} u_{ij} \ell_i(\tau) \ell_j(\xi), \\ \mathbf{f}(\tau, \xi) &\approx \sum_{i=1}^{N_\tau} \sum_{j=1}^{N_\xi} f_{ij} \ell_i(\tau) \ell_j(\xi).\end{aligned}\tag{2.19}$$

Here,  $\ell_i$  are Lagrange polynomials of degree  $p_\tau = N_\tau - 1$  and  $\ell_j$  Lagrange polynomials of degree  $p_\xi = N_\xi - 1$ , see Definition 2.3. As in the semi-discrete case we choose LGL nodes  $\tau_i$  and  $\xi_j$ ,  $i = 1, \dots, N_\tau$  and  $j = 1, \dots, N_\xi$ . Moreover, we assume that  $\psi$  can also be represented in the basis of Lagrange polynomials

$$\psi := \sum_{i=1}^{N_\tau} \sum_{j=1}^{N_\xi} \psi_{ij} \ell_i(\tau) \ell_j(\xi).$$

We choose the coefficients  $\psi_{ij}$  such that

$$\begin{aligned}&\frac{2}{\Delta t} \int_{-1}^1 \left( \mathbf{u} \ell_i|_{-1}^{+1} - \int_{-1}^1 \mathbf{u} \ell_i d\tau \right) \ell_j d\xi \\ &+ \frac{2}{\Delta x} \int_{-1}^1 \left( \mathbf{f}(\mathbf{u}) \ell_j|_{-1}^{+1} - \int_{-1}^1 \mathbf{f}(\mathbf{u}) \ell_j d\xi \right) \ell_i d\tau = 0\end{aligned}$$

for  $i = 1, \dots, N_\tau$  and  $j = 1, \dots, N_\xi$ . In the space-time ansatz we also get numerical surface fluxes in the solution  $\mathbf{u}^*$  in addition to  $\mathbf{f}^*$ . The solution flux  $\mathbf{u}^*$  is due to the temporal DG discretization and it is most natural to model it using upwind since this captures the travel direction of time. Inserting approximations (2.19), the surface fluxes and approximating all integrals with LGL quadrature as described before yields the implicit weak space-time discretization

$$\begin{aligned}&\frac{2\omega_j}{\Delta t} (\mathbf{u}_{N_\tau j}^* \partial_{iN_\xi} - \mathbf{u}_{0j}^* \partial_{i0} - \sum_{l=1}^{N_\tau} \omega_l D_{li} u_{lj}) \\ &+ \frac{2\omega_i}{\Delta x} (\mathbf{f}_{iN_\xi}^* \partial_{N_\tau j} - \mathbf{f}_{i0}^* \partial_{0j} - \sum_{l=1}^{N_\xi} \omega_l D_{lj} f_{li}) = 0.\end{aligned}$$

This can be written in compact form on each element  $e^{m,n}$  with the same notation as in (2.7) and (2.8) using a Kronecker product

$$(\mathbf{S}_\tau \mathbf{u}^* - \mathbf{D}_\tau^T \mathbf{M}_\tau \mathbf{u}) \otimes \frac{2}{\Delta t} \mathbf{M}_\xi + \frac{2}{\Delta x} \mathbf{M}_\tau \otimes (\mathbf{S}_\xi \mathbf{f}^* - \mathbf{D}_\xi^T \mathbf{M}_\xi \mathbf{f}) = 0,\tag{2.20}$$

where the index  $\tau$  denotes the temporal matrices and the index  $\xi$  the spatial matrices. Due to the choice of basis functions and the numerical quadrature we derived a space-time DG-SEM discretization in (2.20).

### 3 Equivalence of DG Approximations to Lobatto Schemes

We will finish this chapter by proving the equivalence of the spatial DG approximation to a subclass of implicit Runge-Kutta time integration methods called Lobatto schemes. The theorems and results of this section are needed in Chapter 5. Lets consider the DG discretization in time for the scalar evolution equation

$$\mathbf{u}_t(t) + a\mathbf{u}(t) = 0 \quad (2.21)$$

with  $a > 0$ ,  $t \in [0, T]$  and  $\mathbf{u}(0) = \mathbf{u}_0 \in \mathbb{R}$ . As in the derivation of (2.9) we have on each DG reference element  $n = 1, \dots, N$

$$-u_i^n \int_{-1}^1 \ell_i(\tau) \ell_{j_\tau}(\tau) d\tau + u_i^n \ell_i(1) \ell_j(1) + \frac{a\Delta t}{2} u_i^n \int_{-1}^1 \ell_i(\tau) \ell_j(\tau) d\tau = u_i^n \ell_i(-1) \ell_j(-1)$$

for  $i, j = 1, \dots, N_t$ . The only reasonable and entropy stable flux in the temporal direction is the upwind flux, which gives for  $n = 2, \dots, N - 1$

$$\begin{aligned} & -u_i^n \int_{-1}^1 \ell_i(\tau) \ell_{j_\tau}(\tau) d\tau + u_i^n \ell_i(1) \ell_j(1) + \frac{a\Delta t}{2} u_i^n \int_{-1}^1 \ell_i(\tau) \ell_j(\tau) d\tau \\ & = u_i^{n-1} \ell_i(-1) \ell_j(-1). \end{aligned} \quad (2.22)$$

With

$$\begin{aligned} \tilde{K}_{ij} &:= - \int_{-1}^1 \ell_i(\tau) \ell_{j_\tau}(\tau) d\tau + \ell_i(1) \ell_j(1), \\ \tilde{M}_{ij} &:= \int_{-1}^1 \ell_i(\tau) \ell_j(\tau) d\tau, \\ \tilde{C}_{ij} &:= \ell_i(-1) \ell_j(-1), \end{aligned}$$

we can write (2.22) as a system of linear equations

$$\left( \tilde{\mathbf{K}} + \frac{a\Delta t}{2} \tilde{\mathbf{M}} \right) \mathbf{u}^n = \tilde{\mathbf{C}} \mathbf{u}^{n-1}. \quad (2.23)$$

It can be shown that these schemes are equivalent to specific Runge-Kutta schemes when choosing Gauss-Lobatto quadrature:

**Theorem 2.4** ([9],[26]). If the integrals of equation (2.23) are approximated by Gauss-Lobatto quadrature of order  $2p_t + 1$ , then the discontinuous Galerkin approximation of the model problem (2.21) is equivalent to the  $(p_t + 1)$ -stage Runge-Kutta scheme Lobatto IIIC.

Since the products of the test functions are of degree at most  $2p_t$ , we can apply the GL quadrature of order  $2p_t + 1$  to all integrals. We get the following results about Lobatto IIIC schemes:

**Theorem 2.5** ([25],[36]). For  $s \in \mathbb{N}$  the  $s$ -stage Lobatto IIIC scheme is of nonstiff order  $2s - 1$  and their stability function  $R(z)$  is given by the  $(s - 2, s)$ -Padé approximation to the exponential function  $e^z$ . The method is  $L$ -stable and furthermore algebraically stable, thus B-stable and A-stable, i.e.,

$$|R(z)| < 1 \text{ for } z \in \mathbb{C} \text{ with } \operatorname{Re}(z) < 0.$$

Padé approximants will be of importance in Chapter 5, thus we define them here.

**Definition 2.6** (Padé approximant). For a given scalar function  $f(z)$  the rational function  $r_{km}(z)$  is a  $(k, m)$ -Padé approximant of  $f$  if  $r_{km} \in \mathcal{R}_{km}$ ,  $q_{km}(0) = 1$  and

$$f(z) - r_{km}(z) = \mathcal{O}(z^{k+m+1}).$$

If a  $(k, m)$ -Padé approximant exists then it is unique.

We can now conclude that stability function of the DG discretization is given by a Padé approximation to the exponential function.

**Corollary 2.7.** The stability function  $R(z)$  of the discontinuous Galerkin approximation with polynomial degree  $p_t \in \mathbb{N}$ ,  $p_t \geq 1$ , is given by the  $(p_t - 1, p_t + 1)$ -Padé approximation to the exponential function  $e^z$ .

*Proof.* For the Dahlquist test equation  $\dot{u} = \lambda u$ ,  $\lambda \in \mathbb{C}$ , by theorem 2.4 the DG scheme is equivalent to the Lobatto IIIC method. Thus, both methods have the same stability function  $R(z)$ . By theorem 2.5 the stability function is given as the  $(p_t - 1, p_t + 1)$ -Padé approximation to the exponential function  $e^z$  and is A-stable.  $\square$

The Padé approximant for the exponential function can be computed directly.



**Theorem 2.8** (Padé approximant of  $e^z$ ). The Padé approximant

$$r_{km}(z) = \frac{p_{km}(z)}{q_{km}(z)}$$

to the exponential function  $e^z$  is given by

$$p_{km}(z) = \sum_{j=0}^k \frac{(k+m-j)! k!}{(k+m)! (k-j)!} \cdot \frac{z^j}{j!},$$
$$q_{km}(z) = \sum_{j=0}^m \frac{(k+m-j)! m!}{(k+m)! (m-j)!} \cdot \frac{(-z)^j}{j!}.$$

## Chapter 3

# Solving Nonlinear Systems

In this chapter we present methods to solve nonlinear systems of equations which result from the implicit DG discretizations described in Chapter 2. These nonlinear systems to be solved have block structure and the block size depends on the number of unknowns in each element. For high order DG methods the blocks are very large and moreover dense. Therefore storage and computational time need to be considered when choosing a suitable solver. We suggest to use Newton-Krylov methods since there exist easy to implement matrix-free variants and the sparsity of the Jacobian can be exploited. We first present so called inexact Newton methods. We then discuss Krylov subspace methods for the solution of the linear systems within Newton methods and finish this chapter with the easy to implement modification called Jacobian-free Newton-Krylov methods.

### I Newton's Method

Newton's method is a classical method to solve nonlinear equation systems. We only explain those parts of the theory relevant to the needed methodology here and refer to [14] for more details. Newton's method is an iterative method to solve the root finding problem

$$\underline{\mathbf{F}}(\underline{\mathbf{u}}) = \underline{\mathbf{0}} \tag{3.1}$$

for a differentiable function  $\underline{\mathbf{F}}$ . The iteration reads

$$\begin{aligned} \text{solve } \frac{\partial \underline{\mathbf{F}}(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} \Big|_{\underline{\mathbf{u}}^{(k)}} \Delta \underline{\mathbf{u}} &= -\underline{\mathbf{F}}(\underline{\mathbf{u}}^{(k)}), \\ \underline{\mathbf{u}}^{(k+1)} &= \underline{\mathbf{u}}^{(k)} + \Delta \underline{\mathbf{u}}, \quad k = 0, 1, 2, \dots \end{aligned} \quad (3.2)$$

The iteration is terminated if

$$\|\underline{\mathbf{F}}(\underline{\mathbf{u}}^{(k+1)})\| \leq \text{TOL} \cdot \|\underline{\mathbf{F}}(\underline{\mathbf{u}}^{(0)})\|. \quad (3.3)$$

Newton's method as presented has two drawbacks, which are impractical for large systems to be solved, i.e. systems of high dimension: The Jacobian has to be computed in each iteration and the linear systems have to be solved exactly. The first problem will be discussed later. For the latter one, we use inexact Newton methods where the linear system is solved using an iterative scheme. This can be done for example with some Krylov subspace methods, which are the subjects of the next section. These can be terminated prematurely, based on the residual of the linear equation system. Thus an approximate solution  $\widetilde{\Delta \underline{\mathbf{u}}}$  to the Newton update  $\Delta \underline{\mathbf{u}}$  in step  $k$  is accepted if the relative residual in the linear system is below a certain tolerance for a so called forcing term  $\eta_k$ . The inexact Newton method reads

$$\begin{aligned} \left\| \frac{\partial \underline{\mathbf{F}}(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} \Big|_{\underline{\mathbf{u}}^{(k)}} \widetilde{\Delta \underline{\mathbf{u}}} + \underline{\mathbf{F}}(\underline{\mathbf{u}}^{(k)}) \right\| &\leq \eta_k \|\underline{\mathbf{F}}(\underline{\mathbf{u}}^{(k)})\|, \\ \underline{\mathbf{u}}^{(k+1)} &= \underline{\mathbf{u}}^{(k)} + \Delta \underline{\mathbf{u}}, \quad k = 0, 1, 2, \dots \end{aligned} \quad (3.4)$$

When choosing the forcing terms  $\eta_k \in \mathbb{R}$  one has to keep in mind that it is not necessary to solve the first few linear systems very accurately. While far away from the solution, we do not need the optimal search direction for Newton's method, but just a reasonable one, to get us in the generally correct direction.

Eisenstat and Walker introduced in [15] a method to choose forcing terms which fulfill the following theorem:

**Theorem 3.1.** Assume that (3.1) has a solution  $\underline{\mathbf{u}}^*$ ,  $\frac{\partial \underline{\mathbf{F}}}{\partial \underline{\mathbf{u}}} : \Omega \rightarrow \mathbb{R}^{m \times m}$  is Lipschitz continuous and  $\frac{\partial \underline{\mathbf{F}}(\underline{\mathbf{u}}^*)}{\partial \underline{\mathbf{u}}}$  is nonsingular. Then there exists a  $\delta$  such that if  $\underline{\mathbf{u}}^{(0)}$  is in a  $\delta$ -neighborhood of  $\underline{\mathbf{u}}^*$ ,  $\{\eta_k\} \subset [0, \eta]$  with  $\eta < 1$  then the inexact Newton iteration (3.4) converges linearly. Moreover, if  $\eta_k \rightarrow 0$ , convergence is superlinear and if  $\eta_k \leq K_\eta \|\underline{\mathbf{F}}(\underline{\mathbf{u}}^{(k)})\|^p$  for some  $K_\eta > 0$  and  $p \in [0, 1]$  and the convergence is superlinear with order  $p + 1$ .

For the sequence

$$\eta_k^A = \gamma \frac{\|\underline{\mathbf{F}}(\underline{\mathbf{u}}^{(k)})\|^2}{\|\underline{\mathbf{F}}(\underline{\mathbf{u}}^{(k-1)})\|^2}, \quad \gamma \in (0, 1],$$

they prove that this sequence has the convergence behavior required for the theorem. Moreover, the theorem says that we achieve quadratic convergence if this sequence is bounded away from one uniformly. Thus, we set  $\eta_0 = \eta_{max}$  for some  $\eta_{max} < 1$  and

$$\eta_k^B = \min(\eta_{max}, \eta_k^A), \quad k > 0.$$

To avoid unexpected decrease in  $\eta_k$  they recommend to refine the definition to

$$\eta_k^C = \begin{cases} \eta_{max}, & k = 0, \\ \eta_k^B, & k > 0, \quad \gamma \eta_{k-1}^2 \leq 0.1, \\ \min(\eta_{max}, \max(\eta_k^A, \gamma \eta_{k-1}^2)), & k > 0, \quad \gamma \eta_{k-1}^2 > 0.1. \end{cases}$$

In order to not oversolve the final stages, Eistenstat and Walker suggest to finally compute

$$\eta_k = \min \left( \eta_{max}, \max \left( \eta_k^C, 0.5 \frac{TOL \cdot \|\underline{\mathbf{F}}(\underline{\mathbf{u}}^{(0)})\|}{\|\underline{\mathbf{F}}(\underline{\mathbf{u}}^{(k)})\|} \right) \right),$$

for the tolerance  $TOL$  at which the Newton iteration would terminate in (3.3).

## 2 Krylov Subspace Methods

Krylov subspace methods are one of the most important iterative techniques available for solving large linear systems [50], [52]. These techniques approximate the solution of a linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

for a nonsingular  $\mathbf{A} \in \mathbb{R}^{m \times m}$  and are based on projections onto Krylov subspaces. In the inexact Newton method (3.4) the system matrix  $\mathbf{A}$  corresponds to the Jacobian in (3.2).

**Definition 3.2** (Krylov subspace, [50]). A Krylov subspace is defined as

$$\mathcal{K}_m = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{m-1}\mathbf{r}_0\} = \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0),$$

with  $\mathbf{r}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$  for an initial guess  $\mathbf{x}_0$ .

Thus, approximations obtained from Krylov subspace methods are of the form

$$\mathbf{A}^{-1}\mathbf{b} \approx \mathbf{x}_m = \mathbf{x}_0 + q_{m-1}(\mathbf{A})\mathbf{r}_0,$$

with  $q_{m-1}$  a polynomial of degree  $m - 1$ .

A Krylov subspace technique computes an orthonormal basis of the space  $\mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$  and calculates the next iterate using projection. One of the most popular Krylov subspace methods is the generalized minimal residual method (GMRES) from Saad and Schultz [52], which generalizes the minimal residual method. In the inexact Newton method (3.4), the iterations are terminated prematurely. Moreover, we know that the system matrix is the Jacobian of (3.1) and has a sparse block structure. GMRES consists of one matrix vector product and only a few scalar products per iteration and is therefore very efficient for our problems when only a few iterations are applied.

## 2.1 GMRES

The generalized minimal residual method (GMRES) is an iterative method to find numerical solutions to nonsymmetric linear systems of equations.

In the  $i$ -th GMRES iteration the minimization problem

$$\min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_i} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2,$$

has to be solved. Since the vectors in  $\mathcal{K}_i$  can be almost linearly dependent, it is of advantage to construct an orthonormal basis  $\{\mathbf{v}_1, \dots, \mathbf{v}_i\}$  of the Krylov subspace. This can be achieved using Arnoldi's method [2], where (modified) Gram-Schmidt is used to construct the basis. An orthogonal matrix  $\mathbf{V}_i \in \mathbb{R}^{i \times m}$  can be constructed with the orthonormal basis as columns. Then the upper Hessenberg matrix  $\tilde{\mathbf{H}}_i = \mathbf{V}_{i+1}^T \mathbf{A} \mathbf{V}_i \in \mathbb{R}^{i+1 \times i}$  is transformed to an upper triangular matrix, for example using Givens rotations.

## 2.2 Preconditioning

Preconditioning is the technique of transforming the original linear system into one with the same solution which is easier to solve with an iterative solver [51]. Let  $\mathbf{P}_L$  and  $\mathbf{P}_R$  be nonsingular matrices. Then

$$\mathbf{A}\mathbf{x} = \mathbf{b} \Leftrightarrow \mathbf{P}_L \mathbf{A} \mathbf{P}_R \mathbf{z} = \mathbf{P}_L \mathbf{b}, \mathbf{P}_R \mathbf{z} = \mathbf{x}. \quad (3.5)$$

$\mathbf{P}_L$  and  $\mathbf{P}_R$  are called left and right preconditioner, respectively, and we refer to right preconditioning if  $\mathbf{P}_L = \mathbf{I}$  and to left preconditioning if  $\mathbf{P}_R = \mathbf{I}$ .

We summarize the influence of both techniques on the GMRES algorithm, and refer to [51] for more details and pseudocodes.

In the case of left preconditioning the residual  $\mathbf{r}_0^P = \mathbf{P}\mathbf{A}\mathbf{x} - \mathbf{P}\mathbf{b} = \mathbf{P}\mathbf{r}_0$  changes, which influences the termination criteria of the method. The Krylov subspace is given by

$$\mathcal{K}_k(\mathbf{P}\mathbf{A}, \mathbf{r}_0^P) = \text{span}\{\mathbf{r}_0^P, \mathbf{P}\mathbf{A}\mathbf{r}_0^P, \dots, \mathbf{P}\mathbf{A}^{k-1}\mathbf{r}_0^P\}.$$

Right preconditioning leaves the residual  $\mathbf{r}_0^P = \mathbf{A}\mathbf{P}\mathbf{z}_0 - \mathbf{b} = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$  unchanged. The corresponding Krylov space is given by

$$\mathcal{K}_k(\mathbf{A}\mathbf{P}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{P}\mathbf{r}_0, \dots, \mathbf{A}\mathbf{P}^{k-1}\mathbf{r}_0\}.$$

In the GMRES algorithm,  $\mathbf{A}\mathbf{x}$  has to be replaced by  $\mathbf{A}\mathbf{P}\mathbf{x}$ . This implies that the preconditioner is applied once in the beginning and once in the end, see the pseudocode in Algorithm 1 from [50]. The advantage of right preconditioning is that the termination criteria does not need to be adapted since the right hand side is not changed by the preconditioner. Therefore we always refer to right preconditioning when we mention preconditioning in the following.

A way to construct efficient preconditioners is presented in Chapter 4.

### 3 Jacobian-free Newton-GMRES

As mentioned before, Newton's method (3.2) has two drawbacks: The Jacobian has to be computed in each iteration and the linear systems have to be solved exactly. For the latter problem we have already discussed methods to terminate the solver prematurely, i.e. Eistenstat and Walker's termination criteria [15]. The problem regarding the computation of the Jacobian remains to be solved. In [38] an extensive overview over Jacobian free Newton-Krylov methods is presented. The idea is to use a cheap approximation of the Jacobian, i.e. a difference quotient, combined with a matrix-vector product routine to save CPU time when assembling it. This is called Jacobian-free Newton-GMRES and reads

$$\frac{\partial \mathbf{F}(\tilde{\mathbf{u}})}{\partial \mathbf{u}} \mathbf{q} \approx \frac{\mathbf{F}(\tilde{\mathbf{u}} + \varepsilon \mathbf{q}) - \mathbf{F}(\tilde{\mathbf{u}})}{\varepsilon}. \quad (3.6)$$

The parameter  $\varepsilon$  has to be chosen carefully. If  $\varepsilon$  is very small, the approximation improves but cancellation errors occur. A simple choice for the parameter that is moderately small but avoids cancellation is

$$\varepsilon = \frac{\sqrt{\varepsilon ps}}{\|\mathbf{q}\|_2},$$

---

**Algorithm 1** GMRES with preconditioner

---

```
Choose  $\mathbf{x}_0$  and calculate  $\mathbf{r}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$ 
if  $\mathbf{r}_0 == 0$  then
    END
else
     $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$ 
    for  $j = 1, 2, \dots, m$  do
         $\mathbf{w}_j = \mathbf{A}\mathbf{P}\mathbf{v}_j$ 
        for  $i = 1, \dots, j$  do
             $h_{ij} = \mathbf{w}_j^T \mathbf{v}_i$ 
             $\mathbf{w}_j = \mathbf{w}_j - h_{ij}\mathbf{v}_i$ 
        end for
         $h_{j+1,j} = \|\mathbf{w}_j\|_2$ 
        if  $h_{j+1,j} == 0$  then
            Set  $m = j$  and BREAK
        else
             $\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}}$ 
        end if
    end for
     $\mathbf{V}_m := [\mathbf{v}_1, \dots, \mathbf{v}_m]$ ,  $\tilde{\mathbf{H}}_m = \{h_{ij}\}_{i=1, \dots, m+1}^{j=1, \dots, m}$ 
    Compute  $\mathbf{y}_m = \operatorname{argmin}_{\mathbf{y}} \|\mathbf{r}_0\|_2 - \tilde{\mathbf{H}}_m \mathbf{y}\|$  and  $\mathbf{x}_m = \mathbf{x}_0 + \mathbf{P}\mathbf{V}_m \mathbf{y}_m$ 
end if
```

---

with the machine accuracy *eps* [47].

The GMRES method performs better than others in the matrix free context since the vectors in matrix vector multiplications are normalized. The preconditioned matrix vector product is given by

$$\frac{\partial \mathbf{F}(\tilde{\mathbf{u}})}{\partial \underline{\mathbf{u}}} \mathbf{P} \underline{\mathbf{q}} \approx \frac{\mathbf{F}(\tilde{\mathbf{u}} + \varepsilon \mathbf{P} \underline{\mathbf{q}}) - \mathbf{F}(\tilde{\mathbf{u}})}{\varepsilon}. \quad (3.7)$$

Without calculating the Jacobian it is difficult to construct a preconditioner for the GMRES solver. In the next chapter we discuss the construction of multigrid preconditioners using a replacement operator in order to avoid calculating the Jacobian of the original system.

## Chapter 4

# Multigrid Methods

Multigrid methods (MG) are a class of iterative methods specifically designed to solve systems arising from discretized differential equations. They are linearly convergent and it has been demonstrated for large classes of partial differential equations, amongst others for the Navier-Stokes equations, that the convergence rates are independent of the mesh size. Moreover, only a few steps are needed to compute the solution. This property is called textbook multigrid efficiency. Good introductions to the basic principles of MG are given in [11], [24], [58] and [62]. In the case of differential operators with periodic boundary conditions it has been noticed for a lot of problems that the eigenvectors of the discretized problem are discrete evaluations of the eigenfunctions, which happen to be periodic. The fundamental idea of MG is to divide the error into low and high frequency parts. A so-called smoother damps the high frequency parts in a few iterations. The low frequency parts can be approximated on a coarser grid, where the transformed problem is taken care of in a space with fewer unknowns using the same approach. This leads to a recursive method on multiple grids. We first explain the concept of basic iterative methods before we describe in more details the elements of multigrid methods for linear problems.

### I Classical Iterative Schemes

Solving discretized PDEs with Jacobian-free Newton-Krylov methods as presented in (3.2) involves solving linear systems of the form

$$\mathbf{Ax} = \mathbf{b}. \tag{4.1}$$



Classical iterative schemes are fixed-point iterations to solve systems (4.1). A good overview over the history of iterative methods from Gauss to modern techniques can be found in [51].

Lets assume that we have an approximation  $\tilde{\mathbf{x}}$  to the solution  $\mathbf{x}$ . Then the algebraic error

$$\mathbf{e} = \tilde{\mathbf{x}} - \mathbf{x} \quad (4.2)$$

is not necessarily accessible since  $\mathbf{x}$  is unknown. On the other hand, the residual

$$\mathbf{r} = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b} \quad (4.3)$$

can be computed easily. These two quantities are connected via the residual equation

$$\mathbf{A}\mathbf{e} = \mathbf{r}. \quad (4.4)$$

We can solve equation (4.4) for the algebraic error  $\mathbf{e}$  and use equation (4.2) to compute the solution by

$$\begin{aligned} \mathbf{x} &= \tilde{\mathbf{x}} - \mathbf{A}^{-1}\mathbf{r} \\ &= (\mathbf{I} - \mathbf{A}^{-1}\mathbf{A})\tilde{\mathbf{x}} + \mathbf{A}^{-1}\mathbf{b}. \end{aligned} \quad (4.5)$$

However, (4.5) is problematic since the exact inverse of the system matrix  $\mathbf{A}^{-1}$  is needed. Therefore we could just solve problem (4.1) directly. The idea is to replace  $\mathbf{A}^{-1}$  by some non-singular approximation  $\mathbf{N}^{-1}$  to get  $\mathbf{x} \approx (\mathbf{I} - \mathbf{N}^{-1}\mathbf{A})\tilde{\mathbf{x}} + \mathbf{N}^{-1}\mathbf{b}$ . This has to be done iteratively to get a good approximation to the solution and yields a linear fixed point iteration

$$\mathbf{x}^{k+1} = \mathbf{M}\mathbf{x}^k + \mathbf{N}^{-1}\mathbf{b} \quad (4.6)$$

with  $\mathbf{M} := \mathbf{I} - \mathbf{N}^{-1}\mathbf{A}$ , thus  $\mathbf{A} = \mathbf{N} - \mathbf{N}\mathbf{M}$ . If (4.6) converges, then  $\mathbf{N}\mathbf{x} = \mathbf{N}\mathbf{M}\mathbf{x} + \mathbf{b}$  which is equivalent to  $\mathbf{A}\mathbf{x} = \mathbf{b}$  and therefore solves the original problem (4.1). Several choices of  $\mathbf{N}$  correspond to well known methods, see [62], where iterative methods are derived based on matrix splitting. We want to mention for example Jacobi, Gauss-Seidel and successive over-relaxation (SOR) with

$$\begin{aligned} \mathbf{N}_{JA} &= \mathbf{L} + \mathbf{D}, \\ \mathbf{N}_{GS} &= \mathbf{D}, \\ \mathbf{N}_{SOR} &= \frac{1}{\omega(2-\omega)}(\mathbf{D} + \mathbf{U})^{-1}\mathbf{D}(\mathbf{D} + \mathbf{L}), \end{aligned} \quad (4.7)$$

where  $\mathbf{L}$  is the strict lower left part of  $\mathbf{A}$ ,  $\mathbf{D}$  the diagonal and  $\mathbf{U}$  the strict upper right part and  $\omega$  a relaxation factor [50], [58], [62].

The convergence of iterative methods of the form (4.6) depends on the spectral radius of the system matrix  $\mathbf{M}$ :

**Theorem 4.1** ([62]). Iterative schemes (4.6) converge to the solution of (4.1) if and only if  $\rho(\mathbf{M}) < 1$ .

One problem with the methods presented so far is that the convergence rate decreases after a couple of iterations [11]. The reason behind this behavior is that high frequency error modes are damped efficiently in the first iterations, while these methods have problems damping the low frequency error components. However, these components can be mapped onto a coarser mesh, where parts of them seem high frequent. This method is called multigrid.

## 2 Multigrid Methods for Linear Systems

The first multigrid method was designed in 1964 for the Poisson equation [16], which reads

$$\begin{aligned} -u_{xx} &= f(x), \quad x \in (0, 1), \quad f \in \mathcal{C}((0, 1), \mathbb{R}), \\ u(0) &= u(1) = 0. \end{aligned}$$

Later it was generalized to other problems, see [62]. Second order finite difference discretization with fixed mesh width  $\Delta x$  results in the system

$$\mathbf{A} \mathbf{u} = \mathbf{b}$$

with

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & & & & & & & & & \\ -1 & 2 & -1 & & & & & & & & \\ & & \ddots & \ddots & \ddots & & & & & & \\ & & & & & -1 & 2 & -1 & & & \\ & & & & & & -1 & 2 & & & \end{pmatrix} \in \mathbb{R}^{n \times n}, \quad \mathbf{b} = \Delta x^2 \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{n-1}) \end{pmatrix}.$$

The eigenvalues of  $\mathbf{A}$  are given by

$$\lambda_k = 4 \sin^2 \left( \frac{\theta_k}{2} \right), \quad \theta_k = \frac{k\pi}{n+1}, \quad k = 1, \dots, n,$$

and the corresponding eigenvectors by

$$\omega_k = (\sin(\theta_k), \sin(2\theta_k), \dots, \sin(n\theta_k))^T, \quad k = 1, \dots, n.$$

These are the discrete evaluations of the eigenfunctions

$$\phi(x) = \sin(k\pi x), \quad k \in \mathbb{N},$$

of  $-u_{xx}$  satisfying the boundary conditions. In Figure 4.1 we see that the eigenvectors can be divided into low and high frequency vectors. If we define a coarse grid by dropping every other grid point, only the smooth eigenvectors can be represented there, see Figure 4.2. Moreover, on the coarser grid some of the low frequencies become high frequencies. This gives the basic idea for multigrid methods: Classical iterative methods damp high frequency error modes in a few iterations, while working poorly on low frequency modes [24]. These modes can be presented without loss of information on a coarser grid, where some of them become high frequent and can be damped using the same iterative method as before.

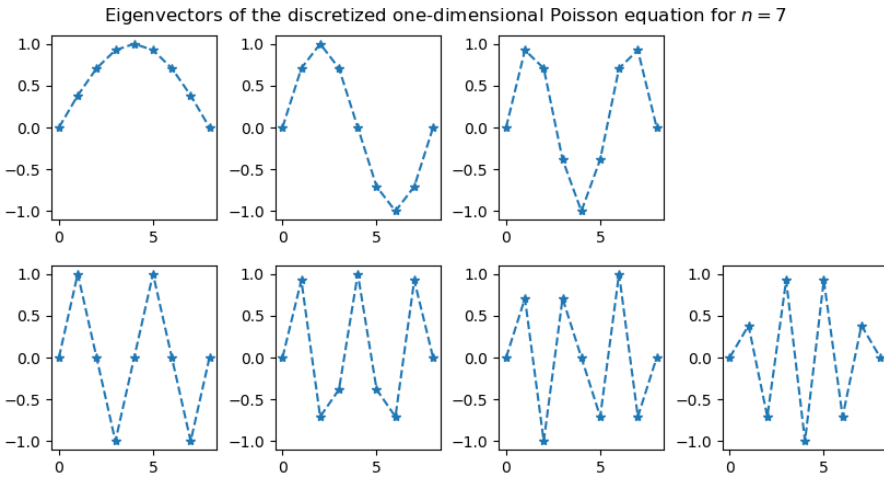


Figure 4.1: Eigenvectors of discretized Poisson equation: figures on top are low frequencies, figures on bottom are high frequencies.

This observation leads to the simple idea of multigrid methods: On the fine grid, the high frequency error parts can be taken care of by a smoother. On the coarse level we solve (4.4) for the error and then correct the fine level solution by the prolonged error on the coarse level. This can be done iteratively on a hierarchy of grids  $\Omega_\ell$ , where each grid is denoted by their level  $\ell$ , and a smaller index corresponds to a coarser grid. A restriction operator  $\mathbf{R}_{\ell-1}^\ell$  is used to transfer a grid function from a finer level  $\ell$  to the next coarser level  $\ell - 1$  and a prolongation operator  $\mathbf{P}_\ell^{\ell-1}$  for the reverse operation. Let us denote the smoothing operators on level  $\ell$  by  $\mathbf{M}_{S,\ell}$  and  $\mathbf{N}_{S,\ell}$  and the system matrix and right hand side by  $\mathbf{A}_\ell$  and  $\mathbf{b}_\ell$ . More details about all these operators will be given in the next section. The pseudo code for a MG scheme to solve a linear system  $\mathbf{Ax} = \mathbf{b}$  is given by Algorithm 2.

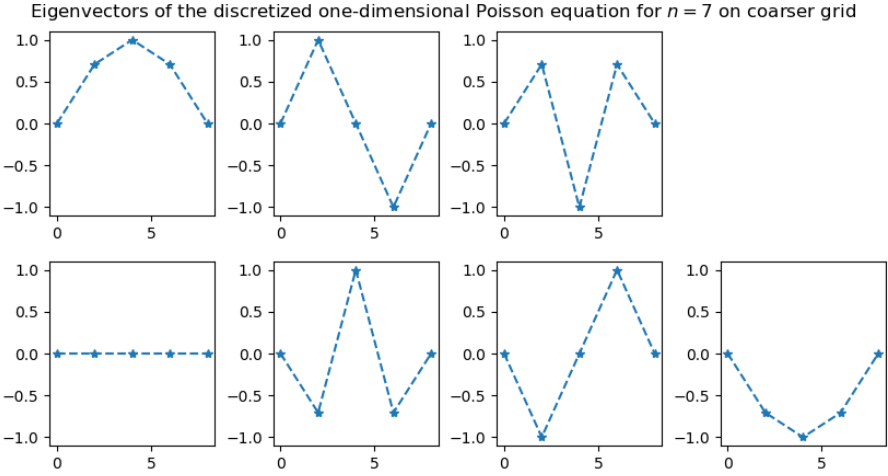


Figure 4.2: Eigenvectors of discretized Poisson equation displayed on every other grid point.

This gives rise to an iterative method of the form  $\mathbf{x}_\ell^{k+1} = \mathbf{M}_{MG}\mathbf{x}_\ell^k + \mathbf{N}_{MG}^{-1}\mathbf{b}_\ell$ . In the case of an  $\ell_{max}$ -level multigrid cycle with  $\gamma = 1$  and pre-smoothing on the coarsest level we can define recursively

$$\begin{aligned} \mathbf{M}_{0,MG} &= 0, \text{ and for } \ell = 1, \dots, \ell_{max} : \\ \mathbf{M}_{\ell,MG} &= \mathbf{M}_{S,\ell}(\mathbf{I} - \mathbf{P}_\ell^{\ell-1}\mathbf{N}_{\ell-1,MG}^{-1}\mathbf{R}_{\ell-1}^\ell\mathbf{A}_\ell)\mathbf{M}_{S,\ell}^{-1} \end{aligned} \quad (4.8)$$

and

$$\begin{aligned} \mathbf{N}_{0,MG} &= \mathbf{N}_{S,0}^{-1}, \text{ and for } \ell = 1, \dots, \ell_{max} : \\ \mathbf{N}_{\ell,MG}^{-1} &= \mathbf{M}_{S,\ell}(\mathbf{N}_{S,\ell}^{-1} - \mathbf{P}_\ell^{\ell-1}\mathbf{N}_{\ell-1,MG}^{-1}\mathbf{R}_{\ell-1}^\ell\mathbf{A}_\ell\mathbf{N}_{S,\ell}^{-1} + \mathbf{P}_\ell^{\ell-1}\mathbf{N}_{\ell-1,MG}^{-1}\mathbf{R}_{\ell-1}^\ell) + \mathbf{N}_{S,\ell}^{-1}. \end{aligned} \quad (4.9)$$

In Figure 4.3 we see the visualization of a 4-grid iteration with  $\gamma = 1$ , called V-cycle due to its shape. Figure 4.4 shows a 4-grid iteration with  $\gamma = 2$ , called W-cycle due to its shape.

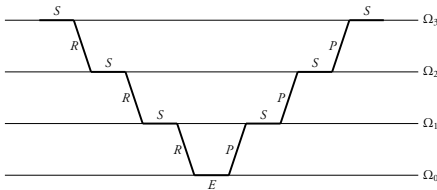


Figure 4.3: Visualization of V cycle,  $\gamma = 1$

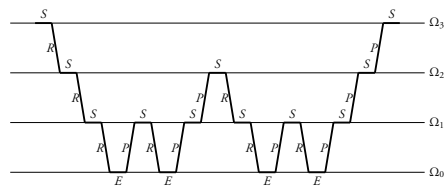


Figure 4.4: Visualization of W cycle,  $\gamma = 2$

---

**Algorithm 2**  $\text{MG}(\mathbf{x}_\ell, \mathbf{b}_\ell, \ell)$ 

---

```
 $\mathbf{x}_\ell = \mathbf{M}_{S,\ell} \mathbf{x}_\ell + \mathbf{N}_{S,\ell}^{-1} \mathbf{b}_\ell$  (pre-smoothing)
if  $\ell > 0$  then
   $\mathbf{r}_{\ell-1} = \mathbf{R}_{\ell-1}^\ell (\mathbf{A}_\ell \mathbf{x}_\ell - \mathbf{b}_\ell)$  (restriction)
   $\mathbf{v}_{\ell-1} = \mathbf{0}$ 
  for  $j = 1, \dots, \gamma$  do
     $\mathbf{v}_{\ell-1} = \text{MG}(\mathbf{v}_{\ell-1}, \mathbf{r}_{\ell-1}, \ell - 1)$ 
  end for
   $\mathbf{x}_\ell = \mathbf{x}_\ell - \mathbf{P}_\ell^{\ell-1} \mathbf{v}_{\ell-1}$  (fine-grid correction)
   $\mathbf{x}_\ell = \mathbf{M}_{S,\ell} \mathbf{x}_\ell + \mathbf{N}_{S,\ell}^{-1} \mathbf{b}_\ell$  (post-smoothing)
end if
```

---

## 2.1 Multigrid Components

The first choice to make in a MG algorithm is the coarsening strategy. The most frequently used and simplest strategy is standard coarsening, where the mesh size is doubled in each direction. In the case of several dimensions, semi-coarsening refers to doubling the mesh size in some directions only. There exist further coarsening strategies which we do not use in this thesis and therefore refer to [11] and [58] for more details.

The next question is how to choose the coarse grid operator  $\mathbf{A}_{\ell-1}$ . One possibility is to discretize the problem on each grid. Another option is to use a so-called Galerkin coarse grid approximation [58], [62]

$$\mathbf{A}_{\ell-1} = \mathbf{R}_{\ell-1}^\ell \mathbf{A}_\ell \mathbf{P}_\ell^{\ell-1}.$$

The choice of transfer operators  $\mathbf{R}_{\ell-1}^\ell$  and  $\mathbf{P}_\ell^{\ell-1}$  is connected to the coarsening strategy. For standard coarsening, typical restriction operators are given by injection, full weighting and half weighting. Typical prolongation operators are linear or bilinear interpolation and projection [58], [62].

In the case of conservation and balance laws the intergrid operators need to be conservative. The corresponding method is called agglomeration multigrid since a coarse grid is obtained by agglomerating a number of neighboring cells. The fine grid values are summed up, weighted by the volumes of the respective cells and divided by the total volume to obtain the restricted value. For an equidistant grid in one dimension, the

corresponding restriction operator reads

$$\mathbf{R}_{\ell-1}^{\ell} = \frac{1}{2} \begin{pmatrix} 1 & 1 & & & & \\ & & 1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & & 1 & 1 \end{pmatrix}.$$

The corresponding prolongation is injection, where the value in the coarse cell is given by the value on all the corresponding fine cells, which corresponds to  $\mathbf{P}_{\ell}^{\ell-1} = 2 (\mathbf{R}_{\ell-1}^{\ell})^T$ .

## 2.2 Smoothers

Another key aspect of efficient multigrid algorithms is a good smoother. Classical smoothers are given in (4.7). These smoothers do not perform well for all problems, e.g. for convection dominated flows [44]. Very efficient MG schemes to solve the steady Euler equations around an airfoil were developed in [35], where the smoothers are based on the SGS method. Unfortunately, these methods do not perform well for the Navier-Stokes equations on high aspect grids. For these equations,  $W_3$  smoothers, a special choice of Rosenbrock methods where the Jacobian is replaced by an approximation, are very efficient [7]. These smoothers can be constructed using a dual time stepping approach developed in [33], where a pseudo time step is added to the steady state of the equation:

$$\frac{\partial \mathbf{x}}{\partial t^*} + \mathbf{A} \mathbf{x} - \mathbf{b} = 0.$$

Then each time stepping scheme becomes an iterative method. An  $s$ -stage scheme is given by

$$\begin{aligned} \mathbf{x}^{(0)} &= \mathbf{x}^{(n)}, \\ \mathbf{x}^{(j)} &= \mathbf{x}^{(n)} - \alpha_j \Delta t^* \mathbf{W}^{-1} (\mathbf{A} \mathbf{x}^{(j-1)} - \mathbf{b}), \quad j = 1, \dots, s \\ \mathbf{x}^{(n+1)} &= \mathbf{x}^{(s)}, \end{aligned}$$

with  $\mathbf{W} \approx \mathbf{I} + \eta \Delta t^* \mathbf{A}$ , pseudo time step width  $\Delta t^*$  and problem dependent parameters  $\alpha_j$  and  $\eta$ . The specific approximation to define  $\mathbf{W}$  is based on a symmetric Gauss-Seidel (SGS) approach as suggested in [54] and modified in [34]. The first step is to approximate the Jacobian by using a different first order discretization. It is based on a splitting  $\mathbf{A} = \mathbf{A}^+ + \mathbf{A}^-$  of the flux Jacobian. This is evaluated in the average of the values on both sides of the interface. The split Jacobians correspond to positive and negative eigenvalues and can be written in terms of the matrix of right eigenvectors  $\mathbf{Q}$  as  $\mathbf{A}^+ = \mathbf{Q} \mathbf{\Lambda}^+ \mathbf{Q}^{-1}$ ,  $\mathbf{A}^- = \mathbf{Q} \mathbf{\Lambda}^- \mathbf{Q}^{-1}$ , where  $\mathbf{\Lambda}^{\pm}$  are diagonal matrices containing

the positive and negative eigenvalues, respectively. These are then bounded away from zero using a parabolic function which takes care when the modulus of the eigenvalue  $\lambda$  is smaller or equal to a fraction  $ad$  of the speed of sound  $a$  with free parameter  $d \in [0, 1]$ :

$$|\lambda| = \frac{1}{2} \left( ad + \frac{|\lambda|^2}{ad} \right), \quad |\lambda| \leq ad.$$

With this, an upwind discretization of the split Jacobian in cell  $i$  is given by

$$\mathbf{x}_{i,*} + \mathbf{x}_i + \frac{\Delta t}{\Delta x_i} ((\mathbf{A}_{ii}^+ \mathbf{x}_i + \mathbf{A}_{i,i+1}^- \mathbf{x}_{i+1}) - (\mathbf{A}_{i-1,i}^+ \mathbf{x}_{i-1} + \mathbf{A}_{ii}^- \mathbf{x}_i)) = 0.$$

The corresponding approximation of the Jacobian is then used to construct a preconditioner. Specifically, in [7] a block SGS preconditioner

$$\mathbf{Q}^{-1} = (\mathbf{D} + \mathbf{L})^{-1} \mathbf{D} (\mathbf{D} + \mathbf{U})^{-1},$$

is considered where  $\mathbf{L}$ ,  $\mathbf{D}$  and  $\mathbf{U}$  are block matrices with  $3 \times 3$  blocks.

With  $\mathbf{L} + \mathbf{D} + \mathbf{U} = \mathbf{I} + \eta \Delta t^* \mathbf{J}$  one obtains

$$\begin{aligned} \mathbf{L}_{i-1,i} &= -\frac{\eta \Delta t \Delta t_i^*}{\Delta x_i} \mathbf{A}_{i-1,i}^+, & \mathbf{U}_{i,i+1} &= \frac{\eta \Delta t \Delta t_i^*}{\Delta x_i} \mathbf{A}_{i,i+1}^-, \\ \mathbf{D}_{ii} &= \mathbf{I} + \eta \Delta t^* \mathbf{I} + \frac{\eta \Delta t \Delta t_i^*}{\Delta x_i} (\mathbf{A}_{ii}^+ - \mathbf{A}_{ii}^-). \end{aligned}$$

Applying this preconditioner requires solving  $3 \times 3$  systems coming from the diagonal, which can be done directly.

To improve the efficiency of the smoother, an optimization process can be applied. Either the parameters in the smoother itself are optimized in order to damp high frequency error components efficiently or the spectral radius of the MG iteration matrix is minimized. Both methods are discussed in [3] for Runge-Kutta smoothers.

### 2.3 Multigrid Preconditioners

Multigrid methods can not only be used as a solver but also to construct preconditioners. We recall that they are iterative methods of the form

$$\mathbf{x}^{k+1} = \mathbf{M}_{MG} \mathbf{x}^k + \mathbf{N}_{MG}^{-1} \mathbf{b}$$

where the matrices are defined recursively, see (4.8) and (4.9). We notice that we would construct the most efficient scheme if  $\mathbf{N}_{MG}^{-1} = \mathbf{A}^{-1}$ . Since for a well-designed multigrid method  $\mathbf{N}_{MG}^{-1}$  approximates the inverse of  $\mathbf{A}$ , this can be used as an efficient preconditioner for an external iterative solver. We need preconditioners in the Jacobian-free context for the Newton-Krylov solver in Algorithm 1. The MG preconditioner can be applied by replacing the matrix vector products in the algorithm with the finite difference approximation (3.7) and due to the definition of the MG method, the preconditioner can be implemented matrix free as well. Unfortunately, more advanced smoothers as presented before can not be implemented fully matrix free. This is still an open problem in the construction of these preconditioners. Therefore MG can be used to construct efficient low storage preconditioners. For DG methods this has been tried in [43], [46].

In the attached publications we demonstrate the potential of such multigrid preconditioners for implicit DG solvers. Using a  $\mathbb{W}_3$  smoother as presented before we construct a preconditioner for matrix-free Newton Krylov solvers, see the work flow chart in Figure 4.5. The core idea is to replace the DG Jacobian by a simplified replacement operator. We choose a first order finite volume operator, since we want to keep the number of degrees of freedom in the replacement operator the same. This can be achieved by introducing a subcell grid in each element. Moreover, this choice is motivated by the equivalence between a DG-SEM discretization and a high order FV discretization [19] and the fact that this replacement operator allows to use the available knowledge about fast multigrid (MG) methods for FV discretizations on block structured meshes. For a linear problem the results are presented in [59], and for a nonlinear problem in [8], see part 2 of this thesis.

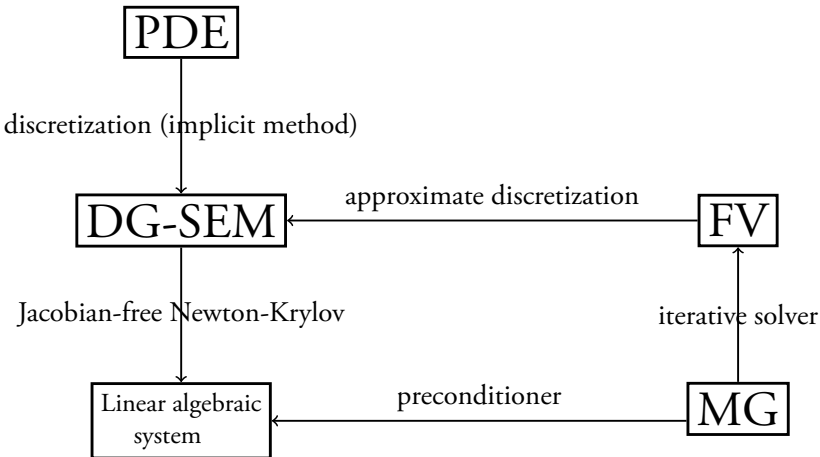


Figure 4.5: Work flow to construct a MG based preconditioner using a FV replacement operator to solve implicit DG-SEM discretizations.





## Chapter 5

# Local Fourier Analysis

The Local Fourier Analysis (LFA) was first presented in [10] as a tool to analyze the quality of multigrid methods. In this chapter we present the basic concept of the LFA as described in [24] and [58] and apply it to a space-time model problem, similar to [21].

The LFA can be used both to analyze the efficiency of a smoother and of a two-grid algorithm. The smoothing factor and the asymptotic convergence factor are quantities to measure the efficiency of a multigrid method. They are both related to the eigenvalues of the operators for the smoother and the two-grid algorithm. Since these operators are very large for space-time discretizations, a discrete Fourier transform is performed. This is of advantage since the operators are of block diagonal form in the Fourier space, which allows to work with operators of much smaller dimension, implying that all calculations become feasible.

The LFA is based on transforming the given problem into the frequency domain using a discrete Fourier transform and so-called grid functions of the form

$$\varphi_k(\theta) = e^{ik\theta}, \quad (5.1)$$

with frequencies  $\theta$  on an infinite grid. One of the core ideas of LFA is to study everything locally. General discrete operators with non-constant coefficients can be linearized locally and locally replaced by an operator with constant coefficients and are therefore formally defined on an infinite grid. This implies that boundary conditions are not taken into account in the analysis. From the definition of the grid functions it follows that it suffices to consider frequencies  $\theta \in (-\pi, \pi]$ .

The core idea of the LFA is presented the following Lemma:

**Lemma 5.1** (Fourier Symbol of Operators, [58]). For  $\theta \in (-\pi, \pi]$ , all grid functions  $\varphi(\theta)$  are eigenfunctions of any discrete operator  $\mathbf{L}$  corresponding to a difference stencil  $\mathbf{L}w(x) = \sum_j u_j w(x + j\Delta x)$  with constant coefficients  $u_j$  and can be written

$$\mathbf{L}\varphi(\theta) = \mathcal{L}(\theta)\varphi(\theta),$$

with  $\mathcal{L}(\theta) = \sum_j u_j e^{ij\theta}$  called the Fourier symbol of  $\mathbf{L}$ .

The idea is to write all discrete operators of an MG algorithm, i.e. the system, smoother, prolongation and restriction, in terms of the Fourier modes and replace the operators by their Fourier symbols in order to calculate the smoothing and asymptotic convergence factors.

In the next sections we present the LFA as a formal tool to analyze a space-time multigrid solver for a model problem. We follow [21], where the heat equation is discretized using a standard finite element method in space and a discontinuous Galerkin approximation in time, and an LFA for this problem is presented. Instead, we consider in this thesis the linear advection equation which we discretize with a finite volume method in space and DG-SEM in time and perform an LFA in order to compute smoothing and two-grid convergence factors. As in [21] we compare two different coarsening strategies: coarsening in both temporal and spatial direction as well as coarsening in the temporal direction only.

## I Model Problem

We analyze a space-time multigrid method for a model problem, the one-dimensional linear advection equation

$$\mathbf{u}_t + a\mathbf{u}_x = 0, \tag{5.2}$$

with  $a > 0$  and  $(x, t) \in [x_L, x_R] \times [0, T] =: \Omega \subset \mathbb{R}^2$  and periodic boundary conditions in space and time.

Following Chapter 2 we discretize the advection problem (5.2) using a space-time DG ansatz. We choose the spatial polynomial degree to be 0, thus the spatial discretization simplifies to a first order finite volume discretization, see Section 1.3 in Chapter 2 for more details. In the temporal direction we allow for a variable polynomial degree  $p_t$ . Moreover, we use an upwind flux in both directions. The spatial discretization matrices are in the following denoted by the index  $\xi$  and the temporal discretization matrices by the index  $\tau$ .

On each space-time slab  $n = 1, \dots, N$  we denote the vector of unknowns by  $\mathbf{u}^n \in \mathbb{R}^{N_x \cdot N_t}$ , with  $N_x$  the degrees of freedom in space and  $N_t$  the degrees of freedom in one time element. The components of  $\mathbf{u}^n$  are given by  $\mathbf{u}_{j,k}^n \in \mathbb{R}$ , where  $n$  denotes the time step,  $j$  is the index w.r.t. the unknowns in space and  $k$  w.r.t. the unknowns in one time element, i.e.  $\mathbf{u}_j^n \in \mathbb{R}^{N_t}$ . We have this index notation throughout the whole chapter for vectors in the space  $\mathbb{R}^{N_x \cdot N_t}$ . Combining discretization in space and time yields a linear system to be solved on each space-time slab:

$$(\mathbf{I}_\xi \otimes \mathbf{K}_\tau + \mathbf{K}_\xi \otimes \mathbf{M}_\tau) \mathbf{u}^{n+1} = \mathbf{I}_\xi \otimes \mathbf{C}_\tau \mathbf{u}^n, \quad (5.3)$$

with  $\mathbf{I}_\xi \in \mathbb{R}^{N_x \times N_x}$ ,  $\mathbf{M}_\tau \in \mathbb{R}^{N_t \times N_t}$  as in (2.10),  $\mathbf{K}_\tau = \mathbf{E}_{N_t} - \mathbf{D}^T \mathbf{M} \in \mathbb{R}^{N_t \times N_t}$  with  $\mathbf{E}_{N_t} = \text{diag}(0, \dots, 0, 1) \in \mathbb{R}^{N_t \times N_t}$  and  $\mathbf{D}$  as in (2.11), and

$$\mathbf{K}_\xi = \frac{a}{\Delta x} \begin{pmatrix} 1 & & & & -1 \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{N_x \times N_x}, \quad \mathbf{C}_\tau = \begin{pmatrix} 0 & & 1 \\ & \ddots & \\ & & 0 \end{pmatrix} \in \mathbb{R}^{N_t \times N_t}.$$

The full space-time system for time steps  $n = 1, \dots, N$  is given by a block triangular system

$$\begin{pmatrix} \mathbf{A}_{\tau,\xi} & & & & \mathbf{B}_{\tau,\xi} \\ \mathbf{B}_{\tau,\xi} & \mathbf{A}_{\tau,\xi} & & & \\ & \ddots & \ddots & & \\ & & & \mathbf{B}_{\tau,\xi} & \mathbf{A}_{\tau,\xi} \end{pmatrix} \begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \vdots \\ \mathbf{u}^N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (5.4)$$

with  $\mathbf{A}_{\tau,\xi} := \mathbf{I}_\xi \otimes \mathbf{K}_\tau + \mathbf{K}_\xi \otimes \mathbf{M}_\tau \in \mathbb{R}^{N_x \cdot N_t \times N_x \cdot N_t}$ ,  $\mathbf{B}_{\tau,\xi} := -\mathbf{I}_\xi \otimes \mathbf{C}_\tau \in \mathbb{R}^{N_x \cdot N_t \times N_x \cdot N_t}$ . We can rewrite (5.4) in compact form as

$$\underline{\mathbf{L}}_{\tau,\xi} \underline{\mathbf{u}} = \underline{\mathbf{b}}, \quad (5.5)$$

with  $\underline{\mathbf{L}}_{\tau,\xi} \in \mathbb{R}^{N \cdot N_t \cdot N_x \times N \cdot N_t \cdot N_x}$  and  $\underline{\mathbf{u}}, \underline{\mathbf{b}} \in \mathbb{R}^{N \cdot N_x \cdot N_t}$ . Here,  $\underline{\mathbf{u}} = (\mathbf{u}^1, \dots, \mathbf{u}^N)^T$  is a long vector containing vectors of unknowns  $\mathbf{u}^n \in \mathbb{R}^{N_x \cdot N_t}$  in each space-time slab  $n$ .

The linear system (5.5) can be solved using a multigrid method. We start by constructing a sequence of space-time grids. Let  $\Omega_\ell \subset \mathbb{R}^2$  denote the grid on level  $\ell = 0, \dots, M$ , with  $\ell = 0$  the coarsest and  $\ell = M$  the finest level. On each space-time grid  $\Omega_\ell$  the system matrix  $\underline{\mathbf{L}}_{\tau_\ell, \xi_\ell}$  has to be set up and the original system (5.5) is obtained on level  $M$ .

The next step is to choose an appropriate smoother. We use a damped block Jacobi smoother of the form

$$\underline{\mathbf{u}}^{(k+1)} = \omega_t (\underline{\mathbf{D}}_{\tau_\ell, \xi_\ell})^{-1} \underline{\mathbf{b}} + (\mathbf{I} - \omega_t (\underline{\mathbf{D}}_{\tau_\ell, \xi_\ell})^{-1} \underline{\mathbf{L}}_{\tau_\ell, \xi_\ell}) \underline{\mathbf{u}}^{(k)}, \quad (5.6)$$

on each grid level  $\ell$  with the block diagonal matrix  $\underline{\mathbf{D}}_{\tau_\ell, \xi_\ell} := \text{diag}\{\mathbf{A}_{\tau_\ell, \xi_\ell}\}_{n=1}^{N_{\ell_t}}$  and the system matrix

$$\underline{\mathbf{S}}_{\tau_\ell, \xi_\ell} = \underline{\mathbf{I}} - \omega_t (\underline{\mathbf{D}}_{\tau_\ell, \xi_\ell})^{-1} \underline{\mathbf{L}}_{\tau_\ell, \xi_\ell}. \quad (5.7)$$

Here, the blocks correspond to a space-time slab on the given grid level. The damping factor  $\omega_t$  depends on the order of the DG method in time and has therefore the index  $t$ .

Next, we need to construct the space-time restriction and prolongation operators  $\underline{\mathbf{R}}_{\ell-1}^\ell$  and  $\underline{\mathbf{P}}_\ell^{\ell-1}$ . For multigrid level  $\ell \in \mathbb{N}_0$  we denote the number of time steps by  $N_{\ell_t} \in \mathbb{N}$  and the degrees of freedom in space by  $N_{\ell_x} \in \mathbb{N}$ . The order of the DG time discretization is not changed and thus the degree of freedom in time for one time step is given by  $N_t$  on each grid.

We will study two different coarsening strategies: both in space and time, referred to as full-coarsening and in the following denoted by index  $f$ ; and coarsening only in the temporal direction, which we will refer to as semi-coarsening with index  $s$ . Since we want to keep the polynomial degree constant on all levels in the temporal direction, we interpolate from coarse to fine grids by combining two time steps to one coarse time step. The restriction and prolongation matrices in time are given by an  $L_2$  projection and are of the form

$$\mathbf{R}_{\ell_t-1}^{\ell_t} := \begin{pmatrix} \mathbf{R}_1 & \mathbf{R}_2 & & & \\ & \mathbf{R}_1 & \mathbf{R}_2 & & \\ & & \ddots & \ddots & \\ & & & \mathbf{R}_1 & \mathbf{R}_2 \end{pmatrix} \in \mathbb{R}^{N_t \cdot N_{\ell_t-1} \times N_t \cdot N_{\ell_t}}, \quad (5.8)$$

$$\mathbf{P}_{\ell_t-1}^{\ell_t} := (\mathbf{R}_{\ell_t-1}^{\ell_t})^T,$$

with the local prolongation matrices  $\mathbf{R}_1^T := \mathbf{M}_{\tau_\ell}^{-1} \widetilde{\mathbf{M}}_{\tau_\ell}^1$  and  $\mathbf{R}_2^T := \mathbf{M}_{\tau_\ell}^{-1} \widetilde{\mathbf{M}}_{\tau_\ell}^2$ , see [21] and [37]. For the basis functions  $\{\ell_k\}_{k=1}^{N_t} \subset \mathbb{P}^{p_t}(0, \tau_\ell)$  on the fine and  $\{\tilde{\ell}_k\}_{k=1}^{N_t} \subset \mathbb{P}^{p_t}(0, 2\tau_\ell)$  on the coarse grid the local projection matrices from coarse to fine grid are defined for  $k, l = 1, \dots, N_t$  by

$$\widetilde{\mathbf{M}}_{\tau_\ell}^1(k, l) := \int_0^{\tau_\ell} \tilde{\ell}_l(t) \ell_k(t) dt \quad \text{and} \quad \widetilde{\mathbf{M}}_{\tau_\ell}^2(k, l) := \int_{\tau_\ell}^{2\tau_\ell} \tilde{\ell}_l(t) \ell_k(t - \tau) dt.$$

The restriction and prolongation matrices in space are given by agglomeration, i.e.

$$\mathbf{R}_{\ell_x-1}^{\ell_x} := \frac{1}{2} \begin{pmatrix} 1 & 1 & & & \\ & & 1 & 1 & \\ & & & \ddots & \ddots \\ & & & & 1 & 1 \end{pmatrix} \in \mathbb{R}^{N_{\ell_x-1} \times N_{\ell_x}}, \quad (5.9)$$

$$\mathbf{P}_{\ell_x-1}^{\ell_x} := (2\mathbf{R}_{\ell_x-1}^{\ell_x})^T \in \mathbb{R}^{N_{\ell_x} \times N_{\ell_x-1}}.$$

Then the restriction and prolongation matrices for the space-time system are defined via a Kronecker product

$$(\mathbf{R}_{\ell-1}^\ell)^s := \mathbf{I}_{N_{\ell_x}} \otimes \mathbf{R}_{\ell_t-1}^{\ell_t}, \quad (\mathbf{R}_{\ell-1}^\ell)^f := \mathbf{R}_{\ell_x-1}^{\ell_x} \otimes \mathbf{R}_{\ell_t-1}^{\ell_t}, \quad (5.10)$$

$$(\mathbf{P}_\ell^{\ell-1})^s := \mathbf{I}_{N_{\ell_x}} \otimes \mathbf{P}_{\ell_t}^{\ell_t-1}, \quad (\mathbf{P}_\ell^{\ell-1})^f := \mathbf{P}_{\ell_x}^{\ell_x-1} \otimes \mathbf{P}_{\ell_t}^{\ell_t-1}. \quad (5.11)$$

With this, the iteration matrices for the two-grid cycles, with pre- and post-smoothing on the fine grid and solving the system on the coarse grid, are given by

$$\underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^s := \underline{\mathbf{S}}_{\tau_\ell, \xi_\ell}^{\nu_2} \left[ \mathbf{I} - (\mathbf{P}_\ell^{\ell-1})^s (\mathbf{L}_{2\tau_\ell, \xi_\ell})^{-1} (\mathbf{R}_{\ell-1}^\ell)^s \underline{\mathbf{L}}_{\tau_\ell, \xi_\ell} \right] \underline{\mathbf{S}}_{\tau_\ell, \xi_\ell}^{\nu_1}, \quad (5.12)$$

$$\underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^f := \underline{\mathbf{S}}_{\tau_\ell, \xi_\ell}^{\nu_2} \left[ \mathbf{I} - (\mathbf{P}_\ell^{\ell-1})^f (\mathbf{L}_{2\tau_\ell, 2\xi_\ell})^{-1} (\mathbf{R}_{\ell-1}^\ell)^f \underline{\mathbf{L}}_{\tau_\ell, \xi_\ell} \right] \underline{\mathbf{S}}_{\tau_\ell, \xi_\ell}^{\nu_1}, \quad (5.13)$$

for semi-coarsening and full-coarsening.

We start by deriving some basics for the LFA with regards to the given model problem.

### 1.1 LFA Basics for the Model Problem

In the following, the discrete Fourier transform is formulated for the given space-time problem and the notation introduced before.

**Definition 5.2** (Fourier modes and frequencies). The function  $\varphi(\theta_k) := [\varphi_1(\theta_k), \dots, \varphi_N(\theta_k)]^T$  with  $\varphi_j(\theta_k) := e^{ij\theta_k}$ ,  $j = 1, \dots, N$ ,  $N \in \mathbb{N}$ , is called Fourier mode with frequency

$$\theta_k \in \Theta := \left\{ \frac{2k\pi}{N} : k = 1 - \frac{N}{2}, \dots, \frac{N}{2} \right\} \subset (-\pi, \pi].$$

The frequencies  $\Theta$  can be separated into high and low frequencies

$$\begin{aligned} \Theta^{low} &:= \Theta \cap \left( -\frac{\pi}{2}, \frac{\pi}{2} \right], \\ \Theta^{high} &:= \Theta \cap \left( \left( -\pi, -\frac{\pi}{2} \right] \cup \left( \frac{\pi}{2}, \pi \right] \right). \end{aligned}$$

**Theorem 5.3** (Discrete Fourier transform [62]). Let  $\mathbf{u} \in \mathbb{R}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t}}$  for  $N_t, N_{\ell_x}, N_{\ell_t} \in \mathbb{N}$ , and assume that  $N_{\ell_x}$  and  $N_{\ell_t}$  are even. Then the vector  $\mathbf{u}$  can be represented as

$$\mathbf{u} = \sum_{\theta_x \in \Theta_{\ell_x}} \sum_{\theta_t \in \Theta_{\ell_t}} \underline{\psi}(\theta_x, \theta_t),$$

and  $\underline{\psi}(\theta_x, \theta_t) \in \mathbb{R}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t}}$  consists of the vectors

$$\psi_j^n(\theta_x, \theta_t) := \mathbf{U}(\theta_x, \theta_t) \Phi_j^n(\theta_x, \theta_t) \in \mathbb{C}^{N_t}, \quad n = 1, \dots, N_{\ell_t}, \quad j = 1, \dots, N_{\ell_x},$$

where the vector  $\Phi_j^n(\theta_x, \theta_t) \in \mathbb{C}^{N_t}$  has elements

$$\Phi_{j,l}^n(\theta_x, \theta_t) := \varphi_n(\theta_t) \varphi_j(\theta_x), \quad l = 1, \dots, N_t.$$

Moreover, we define the coefficient matrix as

$$\mathbf{U}(\theta_x, \theta_t) := \text{diag}(\hat{u}_1, \dots, \hat{u}_{N_t}) \in \mathbb{C}^{N_t \times N_t},$$

with coefficients

$$\hat{u}_l := \frac{1}{N_{\ell_x}} \frac{1}{N_{\ell_t}} \sum_{j=1}^{N_{\ell_x}} \sum_{n=1}^{N_{\ell_t}} u_{j,l}^n \varphi_j(-\theta_x) \varphi_n(-\theta_t), \quad l = 1, \dots, N_t.$$

**Definition 5.4** (Fourier space). For the space level  $\ell_x$  and the time level  $\ell_t$  consider the frequencies  $\theta_x \in \Theta_{\ell_x}$  and  $\theta_t \in \Theta_{\ell_t}$  and the vector  $\Phi_j^n(\theta_x, \theta_t)$  as in Theorem 5.3. Then we define the linear space of Fourier modes with frequencies  $(\theta_x, \theta_t)$  as

$$\begin{aligned} \Psi_{\ell_x, \ell_t}(\theta_x, \theta_t) &:= \text{span}\{\underline{\Phi}(\theta_x, \theta_t)\} \\ &:= \{\underline{\psi}(\theta_x, \theta_t) \in \mathbb{C}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t}} : \psi_j^n(\theta_x, \theta_t) := \mathbf{U} \Phi_j^n(\theta_x, \theta_t), \\ &\quad \text{for } n = 1, \dots, N_{\ell_t}, \quad j = 1, \dots, N_{\ell_x} \text{ and } \mathbf{U} \in \mathbb{C}^{N_t \times N_t}\}. \end{aligned}$$

One of the key properties of the LFA which we will use frequently is the shifting equality.

**Lemma 5.5** (Shifting equality). For the space and time level  $\ell_x$  and  $\ell_t$  let  $\theta_x \in \Theta_{\ell_x}$ ,  $\theta_t \in \Theta_{\ell_t}$  and  $\underline{\psi}(\theta_x, \theta_t) \in \Psi_{\ell_x, \ell_t}(\theta_x, \theta_t)$ . Then the shifting equalities

$$\begin{aligned} \psi_j^{n-1}(\theta_x, \theta_t) &= e^{-i\theta_t} \psi_j^n(\theta_x, \theta_t), \\ \psi_{j-1}^n(\theta_x, \theta_t) &= e^{-i\theta_x} \psi_j^n(\theta_x, \theta_t), \end{aligned}$$

hold for  $n = 2, \dots, N_{\ell_t}$  and  $j = 2, \dots, N_{\ell_x}$ .

*Proof.* It holds

$$\varphi_{n-1}(\theta) = e^{i(n-1)\theta} = e^{-i\theta} e^{in\theta} = e^{-i\theta} \varphi_n(\theta),$$

and therefore with Theorem 5.3

$$\begin{aligned}\Phi_{j,l}^{n-1}(\theta_x, \theta_t) &= \varphi_{n-1}(\theta_t)\varphi_j(\theta_x) = e^{-i\theta_t}\varphi_n(\theta_t)\varphi_j(\theta_x) = e^{-i\theta_t}\Phi_{j,l}^n(\theta_x, \theta_t), \\ \Phi_{j-1,l}^n(\theta_x, \theta_t) &= \varphi_n(\theta_t)\varphi_{j-1}(\theta_x) = e^{-i\theta_x}\varphi_n(\theta_t)\varphi_j(\theta_x) = e^{-i\theta_x}\Phi_{j,l}^n(\theta_x, \theta_t),\end{aligned}$$

for  $l = 1, \dots, N_t$ . This implies

$$\begin{aligned}\psi_j^{n-1}(\theta_x, \theta_t) &= \mathbf{U}(\theta_x, \theta_t)\Phi_j^{n-1}(\theta_x, \theta_t) = e^{-i\theta_t}\mathbf{U}(\theta_x, \theta_t)\Phi_j^n(\theta_x, \theta_t) \\ &= e^{-i\theta_t}\psi_j^n(\theta_x, \theta_t), \\ \psi_{j-1}^n(\theta_x, \theta_t) &= \mathbf{U}(\theta_x, \theta_t)\Phi_{j-1}^n(\theta_x, \theta_t) = e^{-i\theta_x}\mathbf{U}(\theta_x, \theta_t)\Phi_j^n(\theta_x, \theta_t) \\ &= e^{-i\theta_x}\psi_j^n(\theta_x, \theta_t),\end{aligned}$$

for  $n = 2, \dots, N_{\ell_t}$  and  $j = 2, \dots, N_{\ell_x}$ .  $\square$

We can now derive the Fourier symbol of the system matrix  $\underline{\mathbf{L}}_{\tau_\ell, \xi_\ell}$  (5.5) for the model problem (5.2).

**Lemma 5.6** (Fourier symbol of  $\underline{\mathbf{L}}_{\tau_\ell, \xi_\ell}$ ). For the frequencies  $\theta_x \in \Theta_{\ell_x}$  and  $\theta_t \in \Theta_{\ell_t}$  we consider the vector  $\underline{\psi}(\theta_x, \theta_t) \in \Psi_{\ell_x, \ell_t}(\theta_x, \theta_t)$ . Then for

$$\mathcal{L}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) := -e^{-i\theta_t}\mathbf{C}_{\tau_\ell} + \mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x}(-e^{-i\theta_x} + 1)\mathbf{M}_{\tau_\ell} \in \mathbb{C}^{N_t \times N_t}$$

it holds

$$(\underline{\mathbf{L}}_{\tau_\ell, \xi_\ell} \underline{\psi}(\theta_x, \theta_t))_j^n = \mathcal{L}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)\psi_j^n(\theta_x, \theta_t),$$

for  $n = 2, \dots, N_{\ell_t}$  and  $j = 2, \dots, N_{\ell_x} - 1$  and we call  $\mathcal{L}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)$  the Fourier symbol of  $\underline{\mathbf{L}}_{\tau_\ell, \xi_\ell}$ .

*Proof.* With the shifting equality from Lemma 5.5 and for  $\underline{\psi}(\theta_x, \theta_t) \in \Psi_{\ell_x, \ell_t}(\theta_x, \theta_t)$  we get

$$\begin{aligned}(\underline{\mathbf{L}}_{\tau_\ell, \xi_\ell} \underline{\psi}(\theta_x, \theta_t))^n &= \mathbf{B}_{\tau_\ell, \xi_\ell}\psi^{n-1}(\theta_x, \theta_t) + \mathbf{A}_{\tau_\ell, \xi_\ell}\psi^n(\theta_x, \theta_t) \\ &= (e^{-i\theta_t}\mathbf{B}_{\tau_\ell, \xi_\ell} + \mathbf{A}_{\tau_\ell, \xi_\ell})\psi^n(\theta_x, \theta_t)\end{aligned}$$

for  $n = 2, \dots, N_{\ell_t}$ . Thus, we have to study the product of  $\mathbf{A}_{\tau_\ell, \xi_\ell} = \mathbf{I}_{\xi_\ell} \otimes \mathbf{K}_{\tau_\ell} + \mathbf{K}_{\xi_\ell} \otimes \mathbf{M}_{\tau_\ell}$  and  $\mathbf{B}_{\tau_\ell, \xi_\ell} = -\mathbf{I}_{\xi_\ell} \otimes \mathbf{C}_{\tau_\ell}$  with the vector  $\psi^n(\theta_x, \theta_t)$ . We get

$$\begin{aligned}(\mathbf{B}_{\tau_\ell, \xi_\ell}\psi^n(\theta_x, \theta_t))_{j,l} &= -\sum_{i=1}^{N_{\ell_x}} \sum_{k=1}^{N_t} I_{\xi_\ell}(j, i) C_{\tau_\ell}(l, k)\psi_{i,k}^n(\theta_x, \theta_t) \\ &= -\sum_{k=1}^{N_t} C_{\tau_\ell}(l, k)\psi_{j,k}^n(\theta_x, \theta_t) = -(\mathbf{C}_{\tau_\ell}\psi_j^n(\theta_x, \theta_t))_l\end{aligned}$$



and

$$\begin{aligned}
(\mathbf{A}_{\tau_\ell, \xi_\ell} \boldsymbol{\psi}^n(\theta_x, \theta_t))_{j,l} &= \sum_{i=1}^{N_{\ell_x}} \sum_{k=1}^{N_t} I_{\xi_\ell}(j, i) K_{\tau_\ell}(l, k) \psi_{i,k}^n(\theta_x, \theta_t) \\
&+ \sum_{i=1}^{N_{b_\ell}} \sum_{k=1}^{N_t} K_{\xi_\ell}(j, i) M_{\tau_\ell}(l, k) \psi_{i,k}^n(\theta_x, \theta_t) \\
&= (\mathbf{K}_{\tau_\ell} \boldsymbol{\psi}_j^n(\theta_x, \theta_t))_l \\
&+ \frac{a}{\Delta x} (-e^{-i\theta_x} + 1) (\mathbf{M}_{\tau_\ell} \boldsymbol{\psi}_j^n(\theta_x, \theta_t))_l \\
&= (\mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x} (-e^{-i\theta_x} + 1) \mathbf{M}_{\tau_\ell}) \boldsymbol{\psi}_j^n(\theta_x, \theta_t)_l
\end{aligned}$$

for  $j = 2, \dots, N_{\ell_x} - 1$  and  $l = 1, \dots, N_t$ . It follows that

$$(\underline{\mathbf{L}}_{\tau_\ell, \xi_\ell} \underline{\boldsymbol{\psi}}(\theta_x, \theta_t))_j^n = (-e^{-i\theta_t} \mathbf{C}_{\tau_\ell} + \mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x} (-e^{-i\theta_x} + 1) \mathbf{M}_{\tau_\ell}) \boldsymbol{\psi}_j^n(\theta_x, \theta_t)$$

and thus

$$\mathcal{L}_{\tau_\ell, \xi_\ell} = -e^{-i\theta_t} \mathbf{C}_{\tau_\ell} + \mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x} (-e^{-i\theta_x} + 1) \mathbf{M}_{\tau_\ell} \in \mathbb{C}^{N_t \times N_t}.$$

□

## 2 Smoothing Analysis

In this section we study the performance of the Jacobi method (5.6). We start by constructing the Fourier symbol of the smoother.

**Lemma 5.7** (Fourier symbol of  $\underline{\mathbf{S}}_{\tau_\ell, \xi_\ell}$ ). For the frequencies  $\theta_x \in \Theta_{\ell_x}$  and  $\theta_t \in \Theta_{\ell_t}$  we consider the vector  $\underline{\boldsymbol{\psi}}(\theta_x, \theta_t) \in \Psi_{\ell_x, \ell_t}(\theta_x, \theta_t)$ . Then for

$$\mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) := (1 - \omega_t) \mathbf{I}_{N_t} + \omega_t e^{-i\theta_t} (\mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x} (-e^{-i\theta_x} + 1) \mathbf{M}_{\tau_\ell})^{-1} \mathbf{C}_{\tau_\ell} \in \mathbb{C}^{N_t \times N_t}$$

it holds

$$(\underline{\mathbf{S}}_{\tau_\ell, \xi_\ell} \underline{\boldsymbol{\psi}}(\theta_x, \theta_t))_j^n = \mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) \boldsymbol{\psi}_j^n(\theta_x, \theta_t)$$

for  $n = 1, \dots, N_{\ell_x}$ ,  $j = 1, \dots, N_{\ell_x}$  and we call  $\mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)$  the Fourier symbol of  $\underline{\mathbf{S}}_{\tau_\ell, \xi_\ell}$ .

*Proof.* Let  $\underline{\psi}(\theta_x, \theta_t) \in \Psi_{\ell_x, \ell_t}(\theta_x, \theta_t)$ , then we have with (5.7) for fixed  $n = 1, \dots, N_{\ell_t}$  and  $j = 1, \dots, N_{\ell_x}$

$$\begin{aligned} (\underline{\mathbf{S}}_{\tau_\ell, \xi_\ell} \underline{\psi}(\theta_x, \theta_t))_j^n &= ((\mathbf{I}_{N_t N_{\ell_x} N_{\ell_t}} - \omega_t (\underline{\mathbf{D}}_{\tau_\ell, \xi_\ell})^{-1} \underline{\mathbf{L}}_{\tau_\ell, \xi_\ell}) \underline{\psi}(\theta_x, \theta_t))_j^n \\ &= (\mathbf{I}_{N_t} - \omega_t (\hat{\mathbf{A}}_{\tau_\ell, \xi_\ell}(\theta_x))^{-1} \mathcal{L}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)) \psi_j^n(\theta_x, \theta_t) \\ &=: \mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) \psi_j^n(\theta_x, \theta_t) \end{aligned}$$

with  $\hat{\mathbf{A}}_{\tau_\ell, \xi_\ell}(\theta_x) := \mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x}(-e^{-i\theta_x} + 1)\mathbf{M}_{\tau_\ell}$  derived as in the previous proof. Moreover,

$$\begin{aligned} (\hat{\mathbf{A}}_{\tau_\ell, \xi_\ell}(\theta_x))^{-1} \mathcal{L}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) &= (\mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x}(-e^{-i\theta_x} + 1)\mathbf{M}_{\tau_\ell})^{-1} \\ &\quad (-e^{-i\theta_t} \mathbf{C}_{\tau_\ell} + \mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x}(-e^{-i\theta_x} + 1)\mathbf{M}_{\tau_\ell}) \\ &= \mathbf{I}_{N_t} - e^{-i\theta_t} (\mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x}(-e^{-i\theta_x} + 1)\mathbf{M}_{\tau_\ell})^{-1} \mathbf{C}_{\tau_\ell}. \end{aligned}$$

This implies

$$\begin{aligned} \mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) &= \mathbf{I}_{N_t} - \omega_t (\mathbf{I}_{N_t} - e^{-i\theta_t} (\mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x}(-e^{-i\theta_x} + 1)\mathbf{M}_{\tau_\ell})^{-1} \mathbf{C}_{\tau_\ell}) \\ &= (1 - \omega_t) \mathbf{I}_{N_t} + \omega_t e^{-i\theta_t} (\mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x}(-e^{-i\theta_x} + 1)\mathbf{M}_{\tau_\ell})^{-1} \mathbf{C}_{\tau_\ell}. \end{aligned}$$

□

The smoothing factor of the damped block Jacobi method can be measured using the spectral radius. In order to do so, we need the following result.

**Lemma 5.8.** For  $\lambda \in \mathbb{C}$  the eigenvalues of the matrix  $(\mathbf{K} + \lambda \mathbf{M})^{-1} \mathbf{C} \in \mathbb{C}^{N_t \times N_t}$  are given by

$$\sigma((\mathbf{K} + \lambda \mathbf{M})^{-1} \mathbf{C}) = \{0, R(-\lambda \tau)\}$$

where  $R(z)$  is the stability function of the given DG time stepping scheme, see Corollary 2.7.

*Proof.* See [21].

□

With these results we can calculate the smoothing factor using the Fourier symbol of the smoother, which is of smaller dimension.

**Lemma 5.9.** The spectral radius of the Fourier symbol  $\mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)$  is given by

$$\rho(\mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)) = \max\{|1 - \omega_t|, S(\omega_t, \theta_x, \theta_t)\}$$

with

$$S(\omega_t, \theta_x, \theta_t) := |1 - \omega_t + e^{-i\theta_t} \omega_t R(-\mu\beta(\theta_x))| \quad (5.14)$$

for the stability function  $R$  of the given DG time stepping scheme,  $\beta(\theta_x) := 1 - e^{-i\theta_x}$  and the CFL number  $\mu := \frac{a\Delta\tau_\ell}{\Delta x_\ell}$ .

*Proof.* The eigenvalues of the Fourier symbol  $\mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)$  from Lemma 5.7 are given by

$$\sigma(\mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)) = 1 - \omega_t + e^{-i\theta_t} \omega_t \sigma(\mathbf{K}_{\tau_\ell} + \frac{a}{\Delta x} (-e^{-i\theta_x} + 1) \mathbf{M}_{\tau_\ell})^{-1} \mathbf{C}_{\tau_\ell}.$$

With Lemma 5.8 we can compute the spectrum as

$$\sigma(\mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)) = \{1 - \omega_t, 1 - \omega_t + e^{-i\theta_t} \omega_t R(-\mu\beta(\theta_x))\}$$

and therefore

$$\rho(\mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)) = \max\{|1 - \omega_t|, |1 - \omega_t + e^{-i\theta_t} \omega_t R(-\mu\beta(\theta_x))|\}.$$

□

We are now able to analyze the optimal smoothing behavior of the Fourier symbol  $\mathcal{S}_{\tau_\ell, \xi_\ell}$  of the smoother. The first step is to find the optimal damping parameter  $\omega_t \in (0, 1]$  in (5.7), i.e. for the worst case frequencies  $(\theta_x, \theta_t) \in (-\pi, \pi]^2$ . We consider two different coarsening strategies for the given space-time problem: either full-coarsening in both spatial and temporal direction, or semi-coarsening, where we consider coarsening in the temporal direction only. We apply both strategies in order to smoothen the set of high frequencies from the coarser grid. For the set of space-time frequencies

$$\Theta_{\ell_x, \ell_t} := \{(\theta_x, \theta_t) : \theta_x \in \Theta_{\ell_x}, \theta_t \in \Theta_{\ell_t}\} \subset (-\pi, \pi]^2$$

they are given by

$$\begin{aligned} \Theta_{\ell_x, \ell_t}^{high,s} &:= \Theta_{\ell_x, \ell_t} \setminus \Theta_{\ell_x, \ell_t}^{low,s} \quad \text{for} \quad \Theta_{\ell_x, \ell_t}^{low,s} := \Theta_{\ell_x, \ell_t} \cap (-\pi, \pi] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right], \\ \Theta_{\ell_x, \ell_t}^{high,f} &:= \Theta_{\ell_x, \ell_t} \setminus \Theta_{\ell_x, \ell_t}^{low,f} \quad \text{for} \quad \Theta_{\ell_x, \ell_t}^{low,f} := \Theta_{\ell_x, \ell_t} \cap \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]^2, \end{aligned}$$

where the index  $s$  denotes semi-coarsening in time and the index  $f$  full space-time coarsening. In Figure 5.1 the low and high frequency modes are visualized for both coarsening strategies. First we need to determine the frequencies which are damped less efficient.

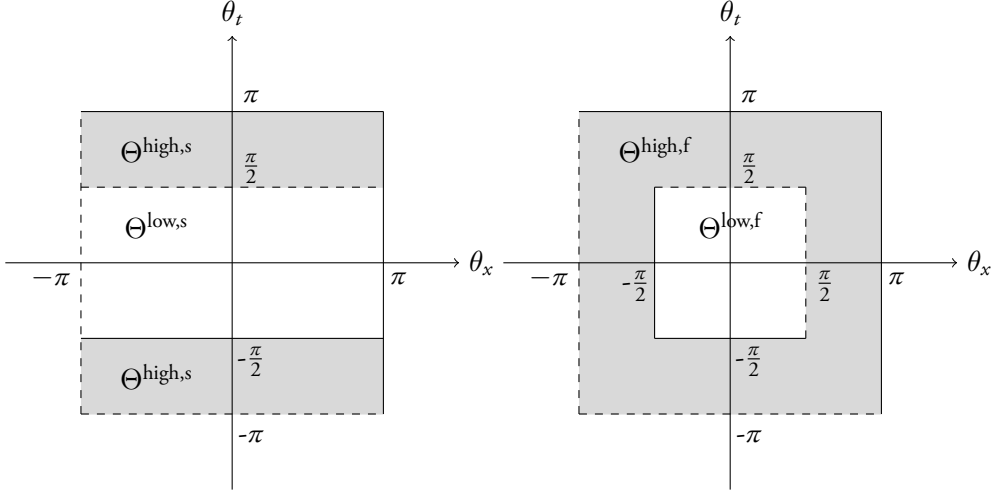


Figure 5.1: Low and high frequencies for semi coarsening (left) and full coarsening (right)

**Definition 5.10** (Worst case frequencies). The worst case frequencies for the function  $S$  in (5.14) are defined as

$$(\theta_x^*(\omega_t, \mu), \theta_t^*(\omega_t, \mu)) := \arg \sup_{(\theta_x, \theta_t) \in \Theta^{high}} S(\omega_t, \theta_x, \theta_t). \quad (5.15)$$

Since it is not possible to find an analytical expression for (5.14), we compute the worst case frequencies numerically, using that the stability function  $R$  of the DG scheme is given by a Padé approximant, see Chapter 2. However, for the  $(p_t - 1, p_t + 1)$ -Padé approximant it holds  $R(-z) \xrightarrow{p_t \rightarrow \infty} e^{-z}$  and therefore some analytical computations can be performed for the case  $p_t \rightarrow \infty$ . In this case, (5.14) becomes

$$S(\omega_t, \theta_x, \theta_t) := |1 - \omega_t + e^{-i\theta_t} \omega_t e^{-\mu\beta(\theta_x)}|,$$

and

$$e^{-\mu\beta(\theta_x)} = e^{\mu \cos(\theta_x) - \mu (\cos(\mu \sin(\theta_x)) - i \sin(\mu \sin(\theta_x)))}.$$

Moreover, for a big CFL number  $\mu \gg 0$ , which is of interest when considering implicit solvers, we get

$$e^{\mu \cos(\theta_x) - \mu} \xrightarrow{\mu \rightarrow \infty} \begin{cases} 1, & \theta_x = 0, \\ 0, & \theta_x \in (-\pi, \pi] \setminus \{0\}, \end{cases}$$

since  $\cos(\theta_x) - 1 < 0$  for  $\theta_x \in (-\pi, \pi] \setminus \{0\}$  and  $\cos(\theta_x) = 0$  for  $\theta_x = 0$ , and  $\cos(\mu \sin(\theta_x)), \sin(\mu \sin(\theta_x)) \in [-1, 1]$  for  $\theta_x \in (-\pi, \pi]$  with increasing oscillation for  $\mu \rightarrow \infty$ . This implies

$$e^{-\mu\beta(\theta_x)} \xrightarrow{\mu \rightarrow \infty} \begin{cases} 1, & \theta_x = 0, \\ 0, & \theta_x \in (-\pi, \pi] \setminus \{0\}, \end{cases}$$

and therefore

$$S(\omega_t, \theta_x, \theta_t) \xrightarrow{\mu \rightarrow \infty} \begin{cases} |1 - \omega_t + \omega_t e^{-i\theta_t}|, & \theta_x = 0, \\ |1 - \omega_t|, & \theta_x \in (-\pi, \pi] \setminus \{0\}. \end{cases}$$

We can compute the worst case frequency in space by

$$\theta_x^* = \arg \sup_{\theta_x \in (-\pi, \pi]} S(\omega_t, \theta_x, \theta_t) = 0,$$

which lies in the set of high frequencies both for semi and full coarsening, and thus

$$S(\omega_t, \theta_x^*, \theta_t) = |1 - \omega_t + \omega_t e^{-i\theta_t}| = \sqrt{(1 - \omega_t)^2 + \omega_t^2 + 2\omega_t \cos(\theta_t)(1 - \omega_t)}.$$

The worst case frequencies w.r.t. the temporal direction can then be calculated

$$\begin{aligned} \theta_t^* &= \arg \sup_{\theta_t \in [\pi/2, \pi]} S(\omega_t, \theta_x^*, \theta_t) = \frac{\pi}{2}, \\ \theta_t^* &= \arg \sup_{\theta_t \in [-\pi, -\pi/2]} S(\omega_t, \theta_x^*, \theta_t) = -\frac{\pi}{2} \end{aligned}$$

We have therefore found worst case frequencies for both semi and full coarsening. Then the optimal damping parameter can be calculated by

$$\omega_t^* = \arg \inf_{\omega_t \in (0, 1]} S(\omega_t, \theta_x^*, \theta_t^*) = 0.5.$$

These results can be seen as a comparison for the following numerical tests with Padé approximations for the case  $p_t < \infty$ .

## 2.1 Semi-Coarsening in Time

We start by analyzing semi-coarsening in time and determine numerically the worst case frequencies

$$(\theta_x^*(\omega_t, \mu), \theta_t^*(\omega_t, \mu)) \in [-\pi, \pi] \times \left[ \frac{\pi}{2}, \pi \right] \cup [-\pi, \pi] \times \left[ -\pi, -\frac{\pi}{2} \right]$$

for  $\mu > 0$  and  $\omega_t \in (0, 1]$ .

By computing the maximum over  $100 \times 100$  equidistant grid points in  $[-\pi, \pi] \times [\pi/2, \pi]$  and  $[-\pi, \pi] \times [-\pi, -\pi/2]$  we can find  $S(\omega_t, \theta_x^*(\omega_t, \mu), \theta_t^*(\omega_t, \mu))$  for fixed  $\omega_t$  and  $\mu$ . Numerical tests show that  $S(\omega_t, \theta_x^*(\omega_t, \mu), \theta_t^*(\omega_t, \mu))$  behaves very similar in its supremum over  $\theta_x$  and  $\theta_t$  for different degrees  $p_t$  of the  $(p_t - 1, p_t + 1)$ -Padé approximants, since they are all approximating  $e^z$ .

In Figure 5.2 the results for the example case  $p_t = 3$  can be seen. The four subfigures show the worst case frequency  $\theta_x^*(\omega_t, \mu)$ ,  $\theta_t^*(\omega_t, \mu)$  and  $S(\omega_t, \theta_x^*, \theta_t^*)$  for different smoothing factors  $\omega_t = 0.25, 0.5, 0.75$  respectively and for 100 equidistantly distributed  $\mu \in [0, 1000]$ . The worst case frequencies converge for all  $\omega_t$  and all  $p_t$  to  $\theta_x^*(\omega_t, \mu) = 0$  and  $\theta_t^*(\omega_t, \mu) = \frac{\pi}{2}$  for large enough CFL number  $\mu$ , while they oscillate for small  $\mu$ .

As comparison we also look at the case  $p_t \rightarrow \infty$ . Again, the worst case frequencies converge to  $\theta_x^*(\omega_t, \mu) = 0$  and  $\theta_t^*(\omega_t, \mu) = \frac{\pi}{2}$ , this time for even larger CFL number  $\mu$ . By this example we are able to detect a pattern in the oscillation: The worst case frequency oscillate with decreasing amplitude when increasing  $\mu$ . Thus we can conclude that the numerical results confirm the analytical results for the limit case  $p_t \rightarrow \infty$ .

With these worst case frequencies it is possible to calculate the optimal damping parameter analytically by taking the infimum of

$$S(\omega_t, \theta_x^*, \theta_t^*) = \sqrt{(1 - \omega_t)^2 + \omega_t^2}.$$

We get  $\omega_t^* = 0.5$  and therefore

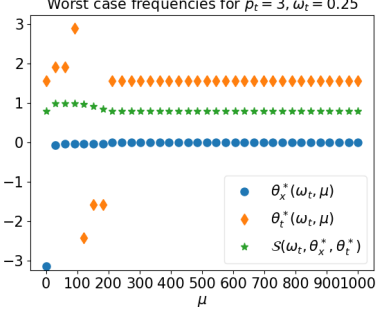
$$S(\omega_t^*, \theta_x^*, \theta_t^*) = \frac{1}{\sqrt{2}}.$$

We want to confirm the optimal damping factor  $\omega_t^*$  numerically. With the limits of the worst case frequencies we obtained before we can calculate the minimum of  $S$  for 100 equidistant distributed  $\mu \in [0, 1000]$  and  $\omega_t \in (0, 1]$  respectively.

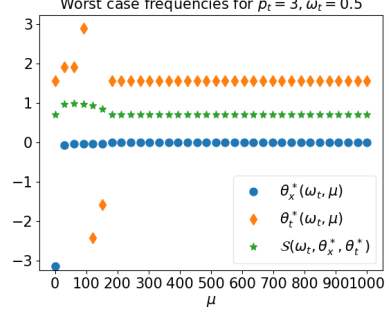
The results for  $p_t = 1, 3, 5$  as well as for the reference case  $p_t \rightarrow \infty$  can be seen in Figure 5.3. The optimal damping parameter is given by  $\omega_t^* = 0.5$  for large  $\mu$ , where with increasing  $p_t$  the critical CFL number  $\mu$  to get convergence increases. These numerical calculations coincide with the analytical result for  $p_t \rightarrow \infty$ .

For  $\mu$  large enough we see that

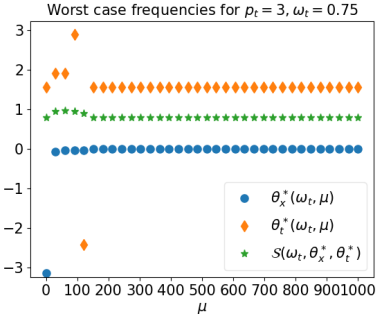
$$S(\omega_t^*, \theta_x^*, \theta_t^*) \approx \frac{1}{\sqrt{2}}.$$



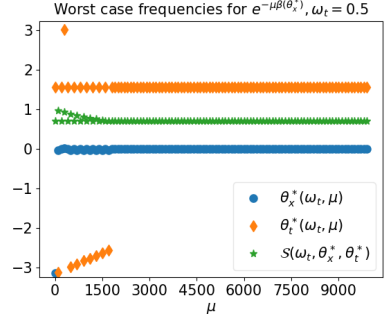
(a)  $\omega_t = 0.25$



(b)  $\omega_t = 0.5$



(c)  $\omega_t = 0.75$



(d)  $\omega_t = 0.5$  for  $p_t \rightarrow \infty$

Figure 5.2: (a)-(c): Worst case frequencies for  $(\theta_x^*(\omega_t, \mu), \theta_t^*(\omega_t, \mu)) \in [-\pi, \pi] \times [\pi/2, \pi] \cup [-\pi, \pi] \times [-\pi, -\pi/2]$  for 100 equidistant distributed  $\mu \in [0, 1000]$  and varying  $\omega_t$  for the  $(2, 4)$ -Padé approximant, (d): worst case frequencies for limit of the Padé approximants for  $p_t \rightarrow \infty$ .

and therefore

$$\inf_{\omega_t \in (0, 1]} \sup_{\substack{\theta_t \in [-\pi, -\pi/2] \cup [\pi/2, \pi] \\ \theta_x \in [0, \pi]}} S(\omega_t, \theta_x, \theta_t) \approx \frac{1}{\sqrt{2}}.$$

This shows that for the Fourier symbol  $\mathcal{S}_{\tau_\ell, \xi_\ell}$  of the smoother from Lemma 5.9 and the damping parameter  $\omega_t^* = 0.5$  the following approximation holds for any polynomial degree  $p_t$ :

$$\sup_{\substack{\theta_t \in [-\pi, -\pi/2] \cup [\pi/2, \pi] \\ \theta_x \in [-\pi, \pi]}} S(\omega_t^*, \theta_x, \theta_t) \approx \frac{1}{\sqrt{2}}.$$

Thus, applying the damped block Jacobi smoother with optimal damping parameter

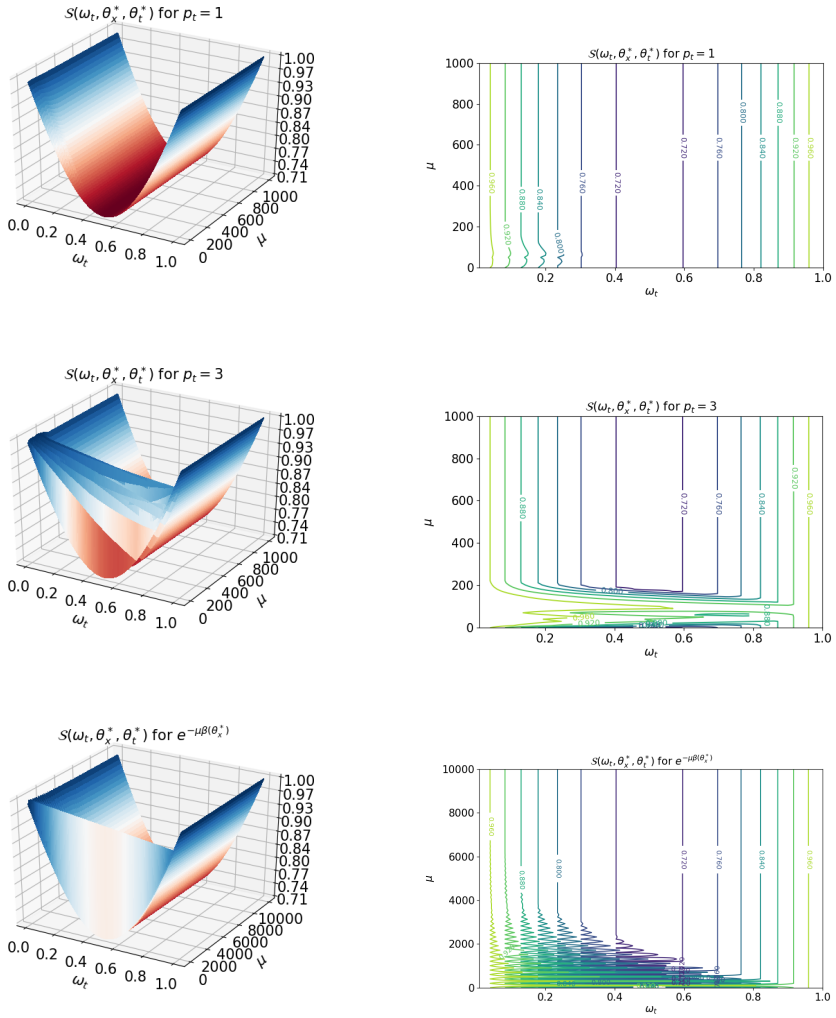


Figure 5.3: Finding the optimal damping parameter  $\omega_t$  when varying  $\mu$  for worst case frequencies  $\theta_x^*(\omega_t, \mu)$  and  $\theta_t^*(\omega_t, \mu)$  for different Padé approximations.

$\omega_t^* = 0.5$  results in asymptotically damping the error components in the high frequencies  $\Theta_{\ell_x, \ell_t}^{high, s}$  by a factor of approximately  $\frac{1}{\sqrt{2}}$ .



## 2.2 Full Space-Time Coarsening

We now study the asymptotic smoothing factor for full space-time coarsening. Using the damping parameter  $\omega_t^* = 0.5$  obtained before, we can calculate the worst case frequencies for given  $\mu$  and  $p_t$ .

In Figure 5.4 the worst case frequencies for 100 equidistant distributed values  $\mu \in [0, 1000]$  for  $p_t = 1, 2, 3$  as well as for the case  $p_t \rightarrow \infty$  can be seen, calculated with the same setup as in the previous section. The worst case frequencies are given by  $\theta_x^* = 0$  and  $\theta_t^* = -\pi/2$  for all  $p_t$  and large CFL number  $\mu$ .

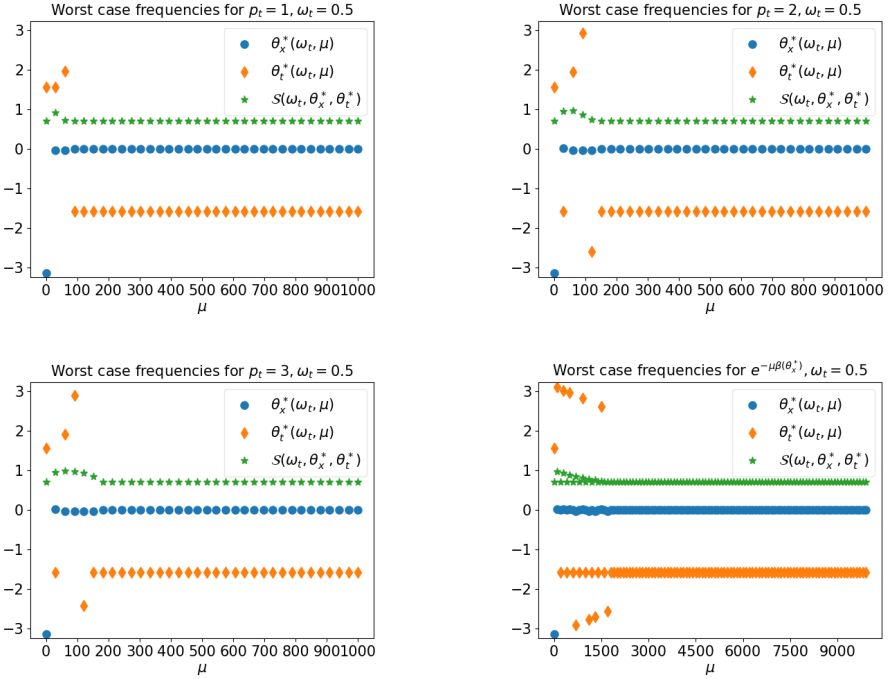


Figure 5.4: (a)-(c): Worst case frequencies for  $(\theta_x^*(\omega_t, \mu), \theta_t^*(\omega_t, \mu)) \in [-\pi, \pi]^2 \setminus [-\pi/2, \pi/2]^2$  for 100 equidistant distributed  $\mu \in [0, 1000]$  and varying Padé approximant, (d): worst case frequencies for limit of the Padé approximants for  $p_t \rightarrow \infty$ .

We have found the worst case frequencies  $\theta_x^*(\omega_t, \mu) = 0$ ,  $\theta_t^*(\omega_t, \mu) = -\pi/2$  and as in the semi-coarsening case, we get the asymptotic smoothing factor

$$\sup_{\substack{\theta_t \in [-\pi, \pi] \setminus [-\pi/2, \pi/2] \\ \theta_x \in [-\pi, \pi] \setminus [-\pi/2, \pi/2]}} S(\omega_t^*, \theta_x, \theta_t) \approx \frac{1}{\sqrt{2}}$$

for  $\mu$  large enough.

### 2.3 Concluding Remarks on Smoothing Factors

In the previous sections we obtained good smoothing behavior for high frequencies in space  $\theta_x \in \Theta_{\ell_x}^{high}$ , i.e.  $\rho(\mathcal{S}) \approx \frac{1}{\sqrt{2}}$ , if the CFL number  $\mu$  is big enough.

In general we have for all frequencies  $(\theta_x, \theta_t) \in \Theta$  the bound

$$S(\omega_t^*, \theta_x, \theta_t) \leq \left( \frac{1}{4} + \frac{1}{2} \operatorname{Re}(R(-\mu\beta(\theta_x))) + \frac{1}{4} |R(-\mu\beta(\theta_x))|^2 \right)^{1/2} \leq 1.$$

Moreover, we get  $\beta(\theta_x) = 0$  only for  $\theta_x = 0$ , implying  $R(-\mu\beta(\theta_x)) = 1$  due to the construction of Padé approximants. By the L-stability we also have

$$|R(-\mu\beta(\theta_x))| \approx 0$$

for almost all frequencies  $\theta_x \in \Theta_{L_x}$ , giving a good smoothing behavior for almost all frequencies. This can be seen in Figures 5.5 and 5.6, where  $S(\omega_t^*, \theta_x, \theta_t)$  is plotted with  $\omega_t^*$  for  $100 \times 100$  equidistant distributed nodes  $(\theta_x, \theta_t) \in [-\pi, \pi]^2$  for  $\mu = 0.01, 1, 100, 1000$  for  $p_t = 1$  in Figure 5.5 and for  $p_t \rightarrow \infty$  in Figure 5.6. We notice for both cases and  $\mu \geq 1$  that only frequencies  $\theta_x$  and  $\theta_t$  close to zero imply a smoothing factor of approximately 1.

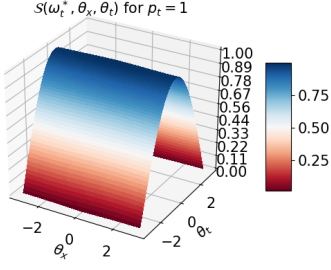
## 3 Two-Grid Analysis

In this section we analyze the two-grid procedure for both full- and the semi-coarsening strategies by studying the corresponding iteration matrices  $\underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^f$  and  $\underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^s$  from the two-grid algorithm, see (5.12) and (5.13).

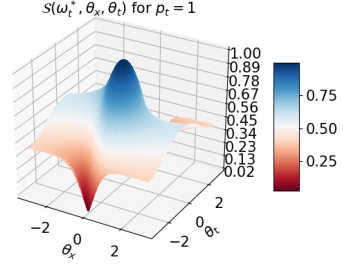
With the following lemma it suffices to only consider low frequencies in the calculations.

**Lemma 5.11.** Let  $\underline{\mathbf{u}} = (\mathbf{u}^1, \dots, \mathbf{u}^{N_{\ell_t}})^T \in \mathbb{R}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t}}$  and assume that  $N_{\ell_x}$  and  $N_{\ell_t}$  are even numbers. Then the vector  $\underline{\mathbf{u}}$  can be written as

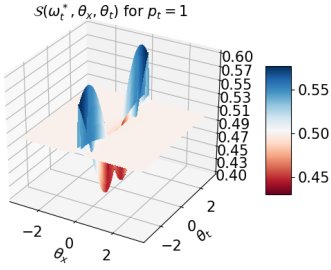
$$\underline{\mathbf{u}} = \sum_{(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low.f}} (\underline{\boldsymbol{\psi}}(\theta_x, \theta_t) + \underline{\boldsymbol{\psi}}(\gamma(\theta_x), \theta_t) + \underline{\boldsymbol{\psi}}(\theta_x, \gamma(\theta_t)) + \underline{\boldsymbol{\psi}}(\gamma(\theta_x), \gamma(\theta_t)))$$



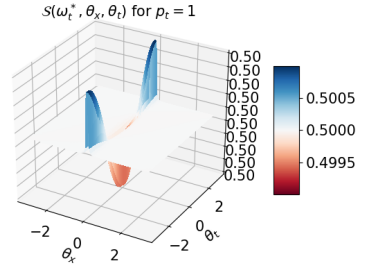
(a)  $\mu = 0.01$



(b)  $\mu = 1$



(c)  $\mu = 100$



(d)  $\mu = 1000$

Figure 5.5: Smoothing factor  $S(\omega_t^*, \theta_x, \theta_t)$  for  $(\theta_x, \theta_t) \in [-\pi, \pi]^2$  with  $\omega_t^* = 0.5$  for  $p_t = 1$  and different  $\mu$ .

with the shifting operator

$$\gamma(\theta) := \begin{cases} \theta + \pi, & \theta < 0, \\ \theta - \pi, & \theta \geq 0, \end{cases}$$

and  $\underline{\psi}(\theta_x, \theta_t) \in \mathbb{C}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t}}$  as in Lemma 5.3.

*Proof.* With Theorem 5.3 every  $\mathbf{u} \in \mathbb{R}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t}}$  can be written as

$$\mathbf{u} = \sum_{\theta_x \in \Theta_{\ell_x}} \sum_{\theta_t \in \Theta_{\ell_t}} \underline{\psi}(\theta_x, \theta_t)$$

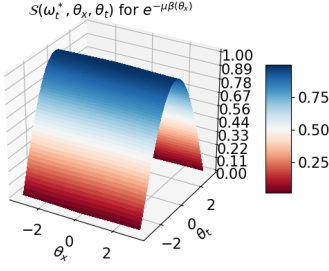
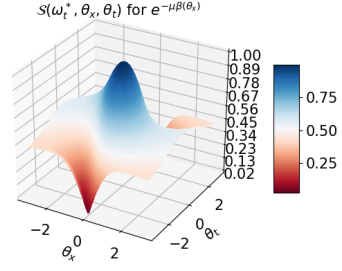
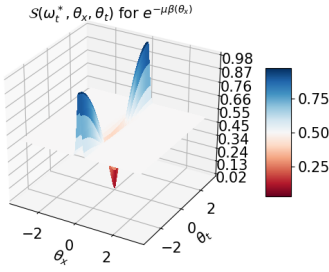
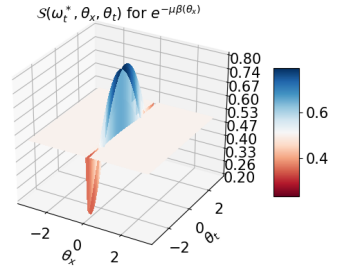
(a)  $\mu = 0.01$ (b)  $\mu = 1$ (c)  $\mu = 100$ (d)  $\mu = 1000$ 

Figure 5.6: Smoothing factor  $S(\omega_t^*, \theta_x, \theta_t)$  for  $(\theta_x, \theta_t) \in [-\pi, \pi]^2$  with  $\omega_t^* = 0.5$  for  $p_t \rightarrow \infty$  and different  $\mu$ .

with the frequencies

$$\theta \in \Theta_\ell = \left\{ \frac{(2-N)\pi}{N}, \frac{(4-N)\pi}{N}, \dots, \frac{((N-2)-N)\pi}{N}, 0, \frac{(N-(N-2))\pi}{N}, \frac{(N-(N-4))\pi}{N}, \dots, \frac{(N-2)\pi}{N}, \frac{N\pi}{N} \right\}.$$

All low frequency modes lie in the interval  $(-\frac{\pi}{2}, \frac{\pi}{2}]$ . The positive frequency modes are given for  $0 \leq b \leq N$  and can be classified as low frequency modes if

$$\frac{(N-b)\pi}{N} \leq \frac{\pi}{2} \Leftrightarrow \frac{N-b}{N} \leq \frac{1}{2} \Leftrightarrow \frac{1}{2} \leq \frac{b}{N} \Leftrightarrow b \geq \frac{N}{2}.$$

The negative frequency modes are given for  $2 \leq b \leq N-2$  and can be classified as low frequency modes if

$$\frac{(b-N)\pi}{N} > -\frac{\pi}{2} \Leftrightarrow \frac{b-N}{N} > -\frac{1}{2} \Leftrightarrow \frac{b}{N} > \frac{1}{2} \Leftrightarrow b > \frac{N}{2}.$$

If we consider for  $\theta \in \Theta^{low}$  the mapping  $\gamma(\theta)$  as defined in the lemma, then

$$\begin{aligned}\gamma(\theta) &= \theta - \pi = \frac{(N-b)\pi}{N} - \pi = -\frac{b\pi}{N} \leq -\frac{\frac{N}{2}\pi}{N} = -\frac{\pi}{2} \in \Theta^{high} \quad \text{for } \theta \geq 0, \\ \gamma(\theta) &= \theta + \pi = \frac{(b-N)\pi}{N} + \pi = \frac{b\pi}{N} > \frac{\frac{N}{2}\pi}{N} = \frac{\pi}{2} \in \Theta^{high} \quad \text{for } \theta < 0.\end{aligned}$$

This shows that the shifting operator maps low to high frequencies, i.e.  $\gamma : \Theta_\ell^{low} \rightarrow \Theta_\ell^{high}$ .

Using this shifting operator and Lemma 5.3 we can rewrite the sum as a sum over low frequency modes only:

$$\begin{aligned}\underline{\mathbf{u}} &= \sum_{(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^f} \underline{\boldsymbol{\psi}}(\theta_x, \theta_t) = \\ &\sum_{(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f}} \left( \underline{\boldsymbol{\psi}}(\theta_x, \theta_t) + \underline{\boldsymbol{\psi}}(\gamma(\theta_x), \theta_t) + \underline{\boldsymbol{\psi}}(\theta_x, \gamma(\theta_t)) + \underline{\boldsymbol{\psi}}(\gamma(\theta_x), \gamma(\theta_t)) \right).\end{aligned}$$

□

Since  $\underline{\boldsymbol{\psi}}(\theta_x, \theta_t)$  consists of the vectors  $\boldsymbol{\psi}_j^n(\theta_x, \theta_t) = \mathbf{U}\boldsymbol{\Phi}_j^n(\theta_x, \theta_t)$  which build the vector  $\boldsymbol{\Phi}(\theta_x, \theta_t)$ , the previous theorem implies that each vector  $\underline{\mathbf{u}} = (\mathbf{u}^1, \dots, \mathbf{u}^{N_{\ell_t}})^T$  can be written as a linear combination of the low frequency vectors

$$\{\underline{\boldsymbol{\Phi}}(\theta_x, \theta_t), \underline{\boldsymbol{\Phi}}(\gamma(\theta_x), \theta_t), \underline{\boldsymbol{\Phi}}(\theta_x, \gamma(\theta_t)), \underline{\boldsymbol{\Phi}}(\gamma(\theta_x), \gamma(\theta_t))\},$$

Thus, four fine grid modes get aliased to one coarse grid mode. In the following it suffices therefore to only consider low frequencies, using the shifting operator  $\gamma : \Theta_\ell^{low} \rightarrow \Theta_\ell^{high}$ .

**Definition 5.12** (Space of low frequency harmonics). For  $N_t, N_{\ell_x}, N_{\ell_t}$  consider the vector  $\underline{\boldsymbol{\psi}}(\theta_x, \theta_t) \in \mathbb{C}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t}}$  for low frequencies  $(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f}$  as in Lemma 5.3. We define the linear space of low frequency harmonics as

$$\begin{aligned}\mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t) &:= \text{span}\{\underline{\boldsymbol{\Phi}}(\theta_x, \theta_t), \underline{\boldsymbol{\Phi}}(\gamma(\theta_x), \theta_t), \underline{\boldsymbol{\Phi}}(\theta_x, \gamma(\theta_t)), \underline{\boldsymbol{\Phi}}(\gamma(\theta_x), \gamma(\theta_t))\} \\ &= \{\underline{\boldsymbol{\psi}}(\theta_x, \theta_t) \in \mathbb{C}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t}} : \boldsymbol{\psi}_j^n(\theta_x, \theta_t) = \mathbf{U}_1 \boldsymbol{\Phi}_j^n(\theta_x, \theta_t) \\ &\quad + \mathbf{U}_2 \boldsymbol{\Phi}_j^n(\gamma(\theta_x), \theta_t) + \mathbf{U}_3 \boldsymbol{\Phi}_j^n(\theta_x, \gamma(\theta_t)) + \mathbf{U}_4 \boldsymbol{\Phi}_j^n(\gamma(\theta_x), \gamma(\theta_t)), \\ &\quad n = 1, \dots, N_{\ell_t}, j = 1, \dots, N_{\ell_x} \text{ and } \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_4 \in \mathbb{C}^{N_t \times N_{\ell_t}}\}.\end{aligned}$$

With Theorem 5.3 and Lemma 5.6 we get for the system matrix  $\underline{\mathbf{L}}_{\tau_\ell, \xi_\ell}$  and low frequencies  $(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f}$  the following mapping property:

$$\underline{\mathbf{L}}_{\tau_\ell, \xi_\ell} : \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t) \rightarrow \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t),$$

$$\begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} \rightarrow \begin{pmatrix} \mathcal{L}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) \mathbf{U}_1 \\ \mathcal{L}_{\tau_\ell, \xi_\ell}(\gamma(\theta_x), \theta_t) \mathbf{U}_2 \\ \mathcal{L}_{\tau_\ell, \xi_\ell}(\theta_x, \gamma(\theta_t)) \mathbf{U}_3 \\ \mathcal{L}_{\tau_\ell, \xi_\ell}(\gamma(\theta_x), \gamma(\theta_t)) \mathbf{U}_4 \end{pmatrix} =: \tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix}, \quad (5.16)$$

with a block diagonal matrix  $\tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) \in \mathbb{C}^{4N_t \times 4N_t}$  and  $\mathcal{L}_{\tau_\ell, \xi_\ell} \in \mathbb{C}^{N_t \times N_t}$  as defined in Lemma 5.6. Accordingly, we obtain with Lemma 5.7 for the smoother  $\underline{\mathbf{S}}_{\tau_\ell, \xi_\ell}$  and low frequencies  $(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f}$

$$\underline{\mathbf{S}}_{\tau_\ell, \xi_\ell} : \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t) \rightarrow \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t),$$

$$\begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} \rightarrow \begin{pmatrix} \mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) \mathbf{U}_1 \\ \mathcal{S}_{\tau_\ell, \xi_\ell}(\gamma(\theta_x), \theta_t) \mathbf{U}_2 \\ \mathcal{S}_{\tau_\ell, \xi_\ell}(\theta_x, \gamma(\theta_t)) \mathbf{U}_3 \\ \mathcal{S}_{\tau_\ell, \xi_\ell}(\gamma(\theta_x), \gamma(\theta_t)) \mathbf{U}_4 \end{pmatrix} =: \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix}, \quad (5.17)$$

with a block diagonal matrix  $\tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) \in \mathbb{C}^{4N_t \times 4N_t}$  and  $\mathcal{S}_{\tau_\ell, \xi_\ell} \in \mathbb{C}^{N_t \times N_t}$  as defined in Lemma 5.7.

Next, we want to analyze the two-grid cycle on the space of low frequency harmonics  $\mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t)$ . In order to do so, we need to study the mapping properties of the restriction and prolongation operators for full- and semi-coarsening.

**Lemma 5.13** (Fourier symbols for spatial prolongation and restriction). Consider the spatial restriction and prolongation operators  $\mathbf{R}_{\ell_x-1}^{\ell_x}$  and  $\mathbf{P}_{\ell_x-1}^{\ell_x-1}$  as defined in (5.9). Let  $\varphi^{\ell_x}(\theta_x) \in \mathbb{C}^{N_{\ell_x}}$  be a fine Fourier mode and  $\varphi^{\ell_x-1}(2\theta_x) \in \mathbb{C}^{N_{\ell_x-1}}$  a coarse Fourier mode for low frequencies  $\theta_x \in \Theta_{\ell_x}^{low}$ . Then for  $\mathcal{R}_{\ell_x-1}^{\ell_x}(\theta_x) := \frac{1}{2}(e^{-i\theta_x} + 1)$  it holds

$$(\mathbf{R}_{\ell_x-1}^{\ell_x} \varphi^{\ell_x}(\theta_x))_j = \mathcal{R}_{\ell_x-1}^{\ell_x}(\theta_x) \varphi_j^{\ell_x-1}(2\theta_x)$$

for  $j = 1, \dots, N_{\ell_x-1}$  and we call  $\mathcal{R}_{\ell_x-1}^{\ell_x}(\theta_x)$  the Fourier symbol for the restriction operator in space.

For  $\mathcal{P}_{\ell_x-1}^{\ell_x-1}(\theta_x) := \frac{1}{2}(e^{i\theta_x} + 1)$  it holds

$$(\mathbf{P}_{\ell_x-1}^{\ell_x-1} \varphi^{\ell_x-1}(2\theta_x))_i = \mathcal{P}_{\ell_x-1}^{\ell_x-1}(\theta_x) \varphi_i^{\ell_x}(\theta_x) + \mathcal{P}_{\ell_x-1}^{\ell_x-1}(\gamma(\theta_x)) \varphi_i^{\ell_x}(\gamma(\theta_x))$$

for  $i = 1, \dots, N_{\ell_x}$  and we call  $\mathcal{P}_{\ell_x-1}^{\ell_x-1}(\theta_x)$  the Fourier symbol for the prolongation operator in space.

*Proof.* For the spatial restriction operator we have

$$\begin{aligned}
(\mathbf{R}_{\ell_x-1}^{\ell_x} \varphi^{\ell_x}(\theta_x))_j &= \frac{1}{2}(\varphi_{2j-1}^{\ell_x}(\theta_x) + \varphi_{2j}^{\ell_x}(\theta_x)) \\
&= \frac{1}{2}(e^{-i\theta_x} + 1)\varphi_{2j}^{\ell_x}(\theta_x) \\
&= \frac{1}{2}(e^{-i\theta_x} + 1)\varphi_j^{\ell_x-1}(2\theta_x) \\
&= \mathcal{R}_{\ell_x-1}^{\ell_x}(\theta_x)\varphi_j^{\ell_x-1}(2\theta_x),
\end{aligned}$$

for  $j = 1, \dots, N_{\ell_x-1}$  using the shifting Lemma 5.5 and  $\varphi_{2j}^{\ell_x}(\theta_x) = \varphi_j^{\ell_x-1}(2\theta_x)$ .

For the spatial prolongation operator we have

$$(\mathbf{P}_{\ell_x}^{\ell_x-1} \varphi^{\ell_x-1}(2\theta_x))_{2j-1} = \varphi_j^{\ell_x-1}(2\theta_x) = \varphi_{2j}^{\ell_x}(\theta_x) = e^{i\theta_x} \varphi_{2j-1}^{\ell_x}(\theta_x)$$

and

$$(\mathbf{P}_{\ell_x}^{\ell_x-1} \varphi^{\ell_x-1}(2\theta_x))_{2j} = \varphi_j^{\ell_x-1}(2\theta_x) = \varphi_{2j}^{\ell_x}(\theta_x)$$

for  $j = 1, \dots, N_{\ell_x-1}$ , using the same arguments as before. Then

$$(\mathbf{P}_{\ell_x}^{\ell_x-1} \varphi^{\ell_x-1}(2\theta_x))_j = \begin{cases} e^{i\theta_x} \varphi_j^{\ell_x}(\theta_x), & j \text{ odd}, \\ \varphi_j^{\ell_x}(\theta_x), & j \text{ even}, \end{cases} \quad j = 1, \dots, N_{\ell_x}.$$

Moreover,

$$\varphi_j^{\ell_x}(\gamma(\theta_x)) = e^{ij\gamma(\theta_x)} = \begin{cases} e^{ij\pi} e^{ij\theta_x}, & \theta_x < 0, \\ e^{-ij\pi} e^{ij\theta_x}, & \theta_x \geq 0, \end{cases} = \begin{cases} -\varphi_j^{\ell_x}(\theta_x), & j \text{ odd}, \\ \varphi_j^{\ell_x}(\theta_x), & j \text{ even}, \end{cases}$$

and

$$\mathcal{P}_{\ell_x}^{\ell_x-1}(\gamma(\theta_x)) = \frac{1}{2}(e^{i\gamma(\theta_x)} + 1) = \frac{1}{2}(-e^{i\theta_x} + 1)$$

for  $j = 1, \dots, N_{\ell_x}$ . This implies

$$\begin{aligned}
&\mathcal{P}_{\ell_x}^{\ell_x-1}(\theta_x)\varphi_j^{\ell_x}(\theta_x) + \mathcal{P}_{\ell_x}^{\ell_x-1}(\gamma(\theta_x))\varphi_j^{\ell_x}(\gamma(\theta_x)) \\
&= \frac{1}{2}(e^{i\theta_x} + 1)\varphi_j^{\ell_x}(\theta_x) + \frac{1}{2}(-e^{i\theta_x} + 1) \begin{cases} -\varphi_j^{\ell_x}(\theta_x), & j \text{ odd}, \\ \varphi_j^{\ell_x}(\theta_x), & j \text{ even} \end{cases} \\
&= \begin{cases} e^{i\theta_x} \varphi_j^{\ell_x}(\theta_x), & j \text{ odd}, \\ \varphi_j^{\ell_x}(\theta_x), & j \text{ even}, \end{cases} = (\mathbf{P}_{\ell_x}^{\ell_x-1} \varphi^{\ell_x-1}(2\theta_x))_j
\end{aligned}$$

for  $j = 1, \dots, N_{\ell_x}$ . □

**Lemma 5.14** (Fourier symbols for temporal prolongation and restriction). Consider the temporal restriction and prolongation operators  $\mathbf{R}_{\ell_t-1}^{\ell_t}$  and  $\mathbf{P}_{\ell_t}^{\ell_t-1}$  as defined in (5.8). Let  $\Phi^{\ell_t}(\theta_t) \in \mathbb{C}^{N_t \cdot N_{\ell_t}}$  be a fine Fourier mode and  $\Phi^{\ell_t-1}(\theta_t) \in \mathbb{C}^{N_t \cdot N_{\ell_t-1}}$  a coarse Fourier mode for low frequencies  $\theta_t \in \Theta_{\ell_t}^{low}$  with elements

$$\begin{aligned}\Phi_l^{n,\ell_t}(\theta_t) &:= \varphi_n(\theta_t), \quad l = 1, \dots, N_t, \quad n = 1, \dots, N_{\ell_t}, \\ \Phi_l^{n,\ell_t-1}(\theta_t) &:= \varphi_n(\theta_t), \quad l = 1, \dots, N_t, \quad n = 1, \dots, N_{\ell_t-1}.\end{aligned}$$

Then for  $\mathcal{R}_{\ell_t-1}^{\ell_t}(\theta_t) := e^{-i\theta_t} \mathbf{R}_1 + \mathbf{R}_2 \in \mathbb{R}^{N_t \times N_t}$ , with  $\mathbf{R}_1$  and  $\mathbf{R}_2$  as defined in (5.8), it holds

$$(\mathbf{R}_{\ell_t-1}^{\ell_t} \Phi^{\ell_t}(\theta_t))^n = \mathcal{R}_{\ell_t-1}^{\ell_t}(\theta_t) \Phi^{n,\ell_t-1}(2\theta_t)$$

for  $n = 1, \dots, N_{\ell_t-1}$  and we call  $\mathcal{R}_{\ell_t-1}^{\ell_t}(\theta_t)$  the Fourier symbol for the restriction operator in time.

Moreover, for  $\mathcal{P}_{\ell_t}^{\ell_t-1}(\theta_t) := \frac{1}{2}(e^{i\theta_t} \mathbf{R}_1^T + \mathbf{R}_2^T) \in \mathbb{R}^{N_t \times N_t}$  it holds

$$(\mathbf{P}_{\ell_t}^{\ell_t-1} \Phi^{\ell_t-1}(2\theta_t))^n = \mathcal{P}_{\ell_t}^{\ell_t-1}(\theta_t) \Phi^{n,\ell_t}(\theta_t) + \mathcal{P}_{\ell_t}^{\ell_t-1}(\gamma(\theta_t)) \Phi^{n,\ell_t}(\gamma(\theta_t))$$

for  $n = 1, \dots, N_{\ell_t}$  and we call  $\mathcal{P}_{\ell_t}^{\ell_t-1}(\theta_t)$  the Fourier symbol for the prolongation in time.

*Proof.* See [21]. □

### 3.1 Semi-Coarsening in Time

We start by analyzing the mapping properties of the prolongation and restriction operators in the case of semi-coarsening in time.

**Definition 5.15** (Fourier space for semi-coarsening). For  $N_t, N_{\ell_x}, N_{\ell_t-1} \in \mathbb{N}$  and the frequencies  $(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low,f}$  define the vector  $\underline{\Phi}(\theta_x, \theta_t) \in \mathbb{C}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t-1}}$  as in Lemma 5.3. We define the linear space with frequencies  $(\theta_x, 2\theta_t)$  as

$$\begin{aligned}\Psi_{\ell_x, \ell_t-1}(\theta_x, 2\theta_t) &:= \text{span}\{\underline{\Phi}^{\ell_x, \ell_t-1}(\theta_x, 2\theta_t), \underline{\Phi}^{\ell_x, \ell_t-1}(\gamma(\theta_x), 2\theta_t)\} \\ &= \{\underline{\psi}^{\ell_x, \ell_t-1}(\theta_x, 2\theta_t) \in \mathbb{C}^{N_t \cdot N_{\ell_x} \cdot N_{\ell_t-1}} : \\ &\quad \psi_j^{n, \ell_x, \ell_t-1}(\theta_x, 2\theta_t) = \mathbf{U}_1 \Phi_j^{n, \ell_x, \ell_t-1}(\theta_x, 2\theta_t) \\ &\quad + \mathbf{U}_2 \Phi_j^{n, \ell_x, \ell_t-1}(\gamma(\theta_x), 2\theta_t) \text{ for } n = 1, \dots, N_{\ell_t} - 1, \\ &\quad j = 1, \dots, N_{\ell_x}, \mathbf{U}_1, \mathbf{U}_2 \in \mathbb{C}^{N_t \times N_t}\}.\end{aligned}$$



With the next lemmas we get the mapping properties for the semi-restriction and semi-prolongation operators.

**Lemma 5.16** (Fourier symbol for restriction for semi-coarsening). The restriction operator  $(\underline{\mathbf{R}}_{\ell-1}^\ell)^s$  satisfies the mapping property

$$\begin{aligned} (\underline{\mathbf{R}}_{\ell-1}^\ell)^s : \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t) &\rightarrow \Psi_{\ell_x, \ell_t-1}(\theta_x, 2\theta_t) \\ \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} &\mapsto (\tilde{\mathcal{R}}_{\ell-1}^\ell)^s(\theta_t) \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} \end{aligned}$$

with the matrix

$$(\tilde{\mathcal{R}}_{\ell-1}^\ell)^s(\theta_t) := \begin{pmatrix} \mathcal{R}_{\ell_t-1}^{\ell_t}(\theta_t) & 0 & \mathcal{R}_{\ell_t-1}^{\ell_t}(\gamma(\theta_t)) & 0 \\ 0 & \mathcal{R}_{\ell_t-1}^{\ell_t}(\theta_t) & 0 & \mathcal{R}_{\ell_t-1}^{\ell_t}(\gamma(\theta_t)) \end{pmatrix} \in \mathbb{C}^{2N_t \times 2N_t}$$

and the Fourier symbol  $\mathcal{R}_{\ell_t-1}^{\ell_t}(\theta_t) \in \mathbb{C}^{N_t \times N_t}$  as defined in Lemma 5.14.

*Proof.* See [21]. □

**Lemma 5.17** (Fourier symbol for prolongation for semi-coarsening). For  $(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^f$  the prolongation operator  $(\underline{\mathbf{P}}_\ell^{\ell-1})^s$  satisfies the mapping property

$$\begin{aligned} (\underline{\mathbf{P}}_\ell^{\ell-1})^s : \Psi_{\ell_x, \ell_t-1}(\theta_x, 2\theta_t) &\rightarrow \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t) \\ \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix} &\mapsto (\tilde{\mathcal{P}}_\ell^{\ell-1})^s(\theta_t) \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix} \end{aligned}$$

with the matrix

$$(\tilde{\mathcal{P}}_\ell^{\ell-1})^s(\theta_t) := \begin{pmatrix} \mathcal{P}_{\ell_t}^{\ell_t-1}(\theta_t) & 0 \\ 0 & \mathcal{P}_{\ell_t}^{\ell_t-1}(\theta_t) \\ \mathcal{P}_{\ell_t}^{\ell_t-1}(\gamma(\theta_t)) & 0 \\ 0 & \mathcal{P}_{\ell_t}^{\ell_t-1}(\gamma(\theta_t)) \end{pmatrix} \in \mathbb{C}^{4N_t \times 2N_t}$$

and the Fourier symbol  $\mathcal{P}_{\ell_t}^{\ell_t-1}(\theta_t) \in \mathbb{C}^{N_t \times N_t}$  as defined in Lemma 5.14.

*Proof.* See [21]. □

With Lemma 5.6 we obtain the mapping property for the coarse grid correction when semi-coarsening in time is applied:

$$\begin{aligned} (\underline{\mathbf{L}}_{2\tau_\ell, \xi_\ell})^{-1} : \Psi_{\ell_x, \ell_t-1}(\theta_x, 2\theta_t) &\rightarrow \Psi_{\ell_x, \ell_t-1}(\theta_x, 2\theta_t), \\ \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix} &\mapsto (\hat{\mathcal{L}}_{2\tau_\ell, \xi_\ell}^s(\theta_x, 2\theta_t))^{-1} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix}, \end{aligned} \quad (5.18)$$

with the matrix

$$(\hat{\mathcal{L}}_{2\tau_\ell, \xi_\ell}^s(\theta_x, 2\theta_t))^{-1} := \begin{pmatrix} (\mathcal{L}_{2\tau_\ell, \xi_\ell}(\theta_x, 2\theta_t))^{-1} & 0 \\ 0 & (\mathcal{L}_{2\tau_\ell, \xi_\ell}(\gamma(\theta_x), 2\theta_t))^{-1} \end{pmatrix} \in \mathbb{C}^{2N_x \times 2N_t}.$$

Here, a complication arises for frequencies  $(\theta_x, \theta_t)$  such that  $\mathcal{L}_{2\tau_\ell, \xi_\ell}(\theta_x, 2\theta_t) = 0$ . For some more discussion of the reasons for this formal complication we refer to [58]. In order to make sure that  $\hat{\mathcal{L}}^s$  exists, we exclude the set

$$\Lambda_s := \left\{ (\theta_x, \theta_t) \in (-\pi, \pi] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right] : \mathcal{L}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) = 0 \text{ or } \mathcal{L}_{2\tau_\ell, \xi_\ell}(\theta_x, 2\theta_t) = 0 \right\}.$$

### 3.2 Full Space-Time Coarsening

Next, we analyze the mapping properties of the prolongation and restriction operators in the case of full space-time coarsening.

**Lemma 5.18** (Fourier symbol for restriction for full-coarsening). The following mapping property holds for the full restriction operator:

$$\begin{aligned} (\underline{\mathbf{R}}_{\ell-1}^\ell)^f : \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t) &\rightarrow \Psi_{\ell_x-1, \ell_t-1}(2\theta_x, 2\theta_t), \\ \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} &\mapsto (\tilde{\mathcal{R}}_{\ell-1}^\ell)^f(\theta_x, \theta_t) \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix}, \end{aligned}$$

with the matrix

$$\begin{aligned} (\tilde{\mathcal{R}}_{\ell-1}^\ell)^f(\theta_x, \theta_t) &:= \\ \left( \hat{\mathcal{R}}_{\ell-1}^\ell(\theta_x, \theta_t) \quad \hat{\mathcal{R}}_{\ell-1}^\ell(\gamma(\theta_x), \theta_t) \quad \hat{\mathcal{R}}_{\ell_t}^\ell(\theta_x, \gamma(\theta_t)) \quad \hat{\mathcal{R}}_{\ell-1}^\ell(\gamma(\theta_x), \gamma(\theta_t)) \right) &\in \mathbb{C}^{N_x \times 4N_t} \end{aligned}$$

and the Fourier symbol

$$\hat{\mathcal{R}}_{\ell-1}^\ell(\theta_x, \theta_t) := \mathcal{R}_{\ell_x-1}^{\ell_x}(\theta_x) \mathcal{R}_{\ell_t-1}^{\ell_t}(\theta_t) \in \mathbb{C}^{N_x \times N_t},$$

with the Fourier symbols  $\mathcal{R}_{\ell_x-1}^{\ell_x}(\theta_x) \in \mathbb{C}$  from Lemma 5.13 and  $\mathcal{R}_{\ell_t-1}^{\ell_t}(\theta_t) \in \mathbb{C}^{N_t \times N_t}$  from Lemma 5.14.

*Proof.* See [21]. □

**Lemma 5.19** (Fourier symbol for prolongation for full-coarsening). The following mapping property holds for the full prolongation operator:

$$\begin{aligned} (\underline{\mathbf{P}}_\ell^{\ell-1})^f : \Psi_{\ell_x-1, \ell_t-1}(2\theta_x, 2\theta_t) &\rightarrow \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t), \\ \mathbf{U} &\mapsto (\tilde{\mathcal{P}}_\ell^{\ell-1})^f(\theta_t, \theta_x)\mathbf{U}, \end{aligned}$$

with the matrix

$$(\tilde{\mathcal{P}}_\ell^{\ell-1})^f(\theta_t, \theta_x) := \begin{pmatrix} \hat{\mathcal{P}}_\ell^{\ell-1}(\theta_x, \theta_t) \\ \hat{\mathcal{P}}_\ell^{\ell-1}(\gamma(\theta_x), \theta_t) \\ \hat{\mathcal{P}}_\ell^{\ell-1}(\theta_x, \gamma(\theta_t)) \\ \hat{\mathcal{P}}_\ell^{\ell-1}(\gamma(\theta_x), \gamma(\theta_t)) \end{pmatrix} \in \mathbb{C}^{N_t \times 4N_t},$$

and the Fourier symbol

$$\hat{\mathcal{P}}_\ell^{\ell-1}(\theta_x, \theta_t) := \mathcal{P}_{\ell_x}^{\ell_x-1}(\theta_x)\mathcal{P}_{\ell_t}^{\ell_t-1}(\theta_t) \in \mathbb{C}^{N_t \times N_t}$$

with the Fourier symbols  $\mathcal{P}_{\ell_x}^{\ell_x-1} \in \mathbb{C}$  from Lemma 5.13 and  $\mathcal{P}_{\ell_t}^{\ell_t-1} \in \mathbb{C}^{N_t \times N_t}$  from Lemma 5.14.

*Proof.* See [21]. □

Moreover, we obtain the mapping property

$$\begin{aligned} (\underline{\mathbf{L}}_{2\tau_\ell, 2\xi_\ell})^{-1} : \Psi_{\ell_x-1, \ell_t-1}(2\theta_x, 2\theta_t) &\rightarrow \Psi_{\ell_x-1, \ell_t-1}(2\theta_x, 2\theta_t), \\ \mathbf{U} &\mapsto (\hat{\mathcal{L}}_{2\tau_\ell, 2\xi_\ell}^f(2\theta_x, 2\theta_t))^{-1}\mathbf{U} \end{aligned} \tag{5.19}$$

with the matrix

$$(\hat{\mathcal{L}}_{2\tau_\ell, 2\xi_\ell}^f(2\theta_x, 2\theta_t))^{-1} := (\mathcal{L}_{2\tau_\ell, 2\xi_\ell}(2\theta_x, 2\theta_t))^{-1} \in \mathbb{C}^{N_t \times N_t}.$$

As before, a complication arises for frequencies  $(\theta_x, \theta_t)$  such that  $\mathcal{L}_{2\tau_\ell, 2\xi_\ell}(2\theta_x, 2\theta_t) = 0$ . In order to make sure that  $\hat{\mathcal{L}}^f$  exists, we exclude the set

$$\Lambda_f := \left\{ (\theta_x, \theta_t) \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]^2 : \mathcal{L}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t) = 0 \text{ or } \mathcal{L}_{2\tau_\ell, 2\xi_\ell}(2\theta_x, 2\theta_t) = 0 \right\}.$$

### 3.3 Asymptotic Convergence Factors

We are now able to write down the Fourier symbol of the two-grid operators and calculate the asymptotic convergence factors.

**Theorem 5.20** (Fourier symbol for the two-grid operator for semi-coarsening). For low frequency modes  $(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f}$  the following mapping property holds for the two-grid operator  $\underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^s$  in (5.13) with semi-coarsening in time:

$$\begin{aligned} \underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^s : \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t) &\rightarrow \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t), \\ \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} &\mapsto \mathcal{M}^s(\theta_x, \theta_t) \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix}, \end{aligned}$$

with the iteration matrix

$$\begin{aligned} \mathcal{M}^s(\theta_x, \theta_t) &:= \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_2}(\theta_x, \theta_t) (\mathbf{I}_{4N_t} - (\tilde{\mathcal{P}}_\ell^{\ell-1})^s(\theta_t) (\hat{\mathcal{L}}_{2\tau_\ell, \xi_\ell}^s(\theta_x, 2\theta_t))^{-1} \\ &\quad (\tilde{\mathcal{R}}_{\ell-1}^\ell)^s(\theta_t) \tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)) \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_1}(\theta_x, \theta_t) \in \mathbb{C}^{4N_t \times 4N_t}. \end{aligned}$$

*Proof.* The two-grid operator for semi-coarsening is given by

$$\underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^s = \underline{\mathbf{S}}_{\tau_\ell, \xi_\ell}^{\nu_2} (\mathbf{I} - (\underline{\mathbf{P}}_\ell^{\ell-1})_s (\underline{\mathbf{L}}_{2\tau_\ell, \xi_\ell})^{-1} (\underline{\mathbf{R}}_{\ell-1}^\ell)_s \underline{\mathbf{L}}_{\tau_\ell, \xi_\ell}) \underline{\mathbf{S}}_{\tau_\ell, \xi_\ell}^{\nu_1}.$$

By previous results we obtain

$$\begin{aligned} \underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^s : \mathcal{E}_{\ell_x, \ell_t} &\xrightarrow{(5.17)} \mathcal{E}_{\ell_x, \ell_t} \xrightarrow{(5.16)} \mathcal{E}_{\ell_x, \ell_t} \xrightarrow{5.16} \Psi_{\ell_x, \ell_t-1}(\theta_x, 2\theta_t) \\ &\xrightarrow{(5.18)} \Psi_{\ell_x, \ell_t-1}(\theta_x, 2\theta_t) \xrightarrow{5.17} \mathcal{E}_{\ell_x, \ell_t} \xrightarrow{(5.17)} \mathcal{E}_{\ell_x, \ell_t} \end{aligned}$$

with the mapping

$$\begin{aligned} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} &\mapsto \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_1} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} \\ &\mapsto \tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell} \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_1} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
&\mapsto (\tilde{\mathcal{R}}_{\ell-1}^\ell)^s \tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell} \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_1} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} \\
&\mapsto (\hat{\mathcal{L}}_{2\tau_\ell, \xi_\ell}^s)^{-1} (\tilde{\mathcal{R}}_{\ell-1}^\ell)^s \tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell} \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_1} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} \\
&\mapsto (\tilde{\mathcal{P}}_\ell^{\ell-1})^s (\hat{\mathcal{L}}_{2\tau_\ell, \xi_\ell}^s)^{-1} (\tilde{\mathcal{R}}_{\ell-1}^\ell)^s \tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell} \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_1} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} \\
&\mapsto (\mathbf{I}_{4N_t} - (\tilde{\mathcal{P}}_\ell^{\ell-1})^s (\hat{\mathcal{L}}_{2\tau_\ell, \xi_\ell}^s)^{-1} (\tilde{\mathcal{R}}_{\ell-1}^\ell)^s \tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell}) \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_1} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} \\
&\mapsto \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_2} (\mathbf{I}_{4N_t} - (\tilde{\mathcal{P}}_\ell^{\ell-1})^s (\hat{\mathcal{L}}_{2\tau_\ell, \xi_\ell}^s)^{-1} (\tilde{\mathcal{R}}_{\ell-1}^\ell)^s \tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell}) \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_1} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix}
\end{aligned}$$

□

**Theorem 5.21** (Fourier symbol for the two-grid operator for full-coarsening). For low frequency modes  $(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f}$  the following mapping property holds for the two-grid operator  $\underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^f$  in (5.12) with space-time coarsening:

$$\begin{aligned}
\underline{\mathbf{M}}_{\tau_\ell, \xi_\ell}^f : \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t) &\rightarrow \mathcal{E}_{\ell_x, \ell_t}(\theta_x, \theta_t), \\
\begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix} &\mapsto \mathcal{M}^f(\theta_x, \theta_t) \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \end{pmatrix},
\end{aligned}$$

with the iteration matrix

$$\begin{aligned}
\mathcal{M}^f(\theta_x, \theta_t) &:= \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_2}(\theta_x, \theta_t) (\mathbf{I}_{4N_t} - (\tilde{\mathcal{P}}_\ell^{\ell-1})^f(\theta_x, \theta_t) (\hat{\mathcal{L}}_{2\tau_\ell, 2\xi_\ell}^f(2\theta_x, 2\theta_t))^{-1} \\
&\quad (\tilde{\mathcal{R}}_{\ell-1}^\ell)^f(\theta_x, \theta_t) \tilde{\mathcal{L}}_{\tau_\ell, \xi_\ell}(\theta_x, \theta_t)) \tilde{\mathcal{S}}_{\tau_\ell, \xi_\ell}^{\nu_1}(\theta_x, \theta_t) \in \mathbb{C}^{4N_t \times 4N_t}.
\end{aligned}$$

*Proof.* The proof follows as for the previous theorem. □

Using Theorems 5.20 and 5.21 we can analyze the asymptotic convergence behavior of the two-grid cycle by simply computing the maximal spectral radius of  $\mathcal{M}_\mu^s(\theta_x, \theta_t)$  and  $\mathcal{M}_\mu^f(\theta_x, \theta_t)$  w.r.t. the low frequencies  $(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f}$ . This motivates the following definition:

**Definition 5.22** (asymptotic two-grid convergence factors). For the two-grid iteration matrices  $\mathcal{M}^s(\theta_x, \theta_t)$  and  $\mathcal{M}^f(\theta_x, \theta_t)$  as in (5.13) and (5.12) we define the asymptotic convergence factors

$$\begin{aligned} \varrho(\mathcal{M}^s) &:= \max\{\varrho(\mathcal{M}^s(\theta_x, \theta_t)) : (\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f} \setminus \Lambda_s\}, \\ \varrho(\mathcal{M}^f) &:= \max\{\varrho(\mathcal{M}^f(\theta_x, \theta_t)) : (\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f} \setminus \Lambda_f\}. \end{aligned}$$

To derive the asymptotic convergence factors  $\varrho(\mathcal{M}^s)$  and  $\varrho(\mathcal{M}^f)$  for a given discretization parameter  $\mu \in \mathbb{R}_+$  and a given polynomial degree  $p_t \in \mathbb{N}$  it is necessary to compute the eigenvalues of

$$\mathcal{M}^s(\theta_x, \theta_t) \in \mathbb{C}^{4N_t \times 4N_t} \quad \text{and} \quad \mathcal{M}^f(\theta_x, \theta_t) \in \mathbb{C}^{4N_t \times 4N_t}$$

with  $N_t = p_t + 1$  for each low frequency  $(\theta_x, \theta_t) \in \Theta_{\ell_x, \ell_t}^{low, f}$ . This shows that the local Fourier analysis provides a feasible option to calculate the asymptotic convergence factors for a huge space-time system of the dimension  $N_t \cdot N_{\ell_t} \cdot N_{\ell_x} \times N_t \cdot N_{\ell_t} \cdot N_{\ell_x}$ .

### 3.4 Numerical Results

Since we can not find analytical expressions for the eigenvalues of the two-grid operators  $\mathcal{M}^f(\theta_x, \theta_t)$  and  $\mathcal{M}^s(\theta_x, \theta_t)$  in order to calculate the asymptotic convergence factors, we compute the eigenvalues numerically. We consider a space-time discretization with  $N_x = 2^8$  spatial elements,  $N = 2^8$  temporal elements with polynomial degree  $p_t \in \{0, 1\}$  and varying CFL number  $\mu$ . When varying the CFL number and at the same time keeping the number of temporal elements on the fine grid the same, we need to adjust the time interval  $[0, T]$  for each CFL number. This setup results in very large space-time systems, making direct calculations of convergence factors not feasible, while we are left with calculating the eigenvalues of a matrix of dimension  $4N_t \times 4N_t$ , with  $N_t = p_t + 1$  with the local Fourier analysis.

For 100 equidistant CFL numbers  $\mu$  between 1 and 1000 we calculate for low frequencies  $\Theta_{\ell_x, \ell_t}^{low, f}$  the asymptotic convergence factors as given in Definition 5.22. The results obtained from the Fourier analysis can be seen in Figure 5.7 for  $p_t = 0$  and in Figure 5.8 for  $p_t = 1$ .

In both cases the asymptotic convergence factors converge to the same value for semi- and full-coarsening case, respectively.

In the case  $p_t = 0$  the asymptotic convergence factor converges to approximately 0.5 for  $\mu \approx 35$  for semi-coarsening and slightly bigger critical CFL number for the full-coarsening case. Moreover, we notice that both for semi- and full-coarsening the two-grid cycle converges for any discretization parameter  $\mu \in [1, 1000]$ . In the semi-coarsening case, the asymptotic convergence factors are in the range  $[0.5, 0.5225]$  and oscillate for CFL numbers smaller than 35. In the full-coarsening case, the convergence factors are in the range  $[0.5, 0.66]$  and oscillate as well slightly for CFL numbers smaller than 35.

When increasing the polynomial order to  $p_t = 1$ , the asymptotic convergence rate becomes smaller, approximately 0.37, when choosing the CFL number big enough. In this case, the critical CLF number is  $\mu \approx 66$  for semi-coarsening and  $\mu \approx 40$  for full-coarsening. As before, both for semi- and full-coarsening the two-grid cycle converges for any discretization parameter  $\mu \in [1, 1000]$ . In the semi-coarsening case, the asymptotic convergence factors are in the range  $[0.37, 0.65]$  and oscillate for CFL numbers smaller than 66. In the full-coarsening case, the convergence factors are in the range  $[0.37, 0.57]$  and oscillate as well slightly for CFL numbers smaller than 40.

We conclude that the two-grid cycle converges to the exact solution both for semi- and full-coarsening for polynomial degrees  $p_t = 0$  and  $p_t = 1$  and the asymptotic convergence rates are always bounded by 0.66. Moreover, the asymptotic convergence rates decrease when increasing  $p_t$ . However, in the case of full-coarsening the problem on the coarse grid becomes smaller than in the semi-coarsening case and is therefore cheaper to solve directly.

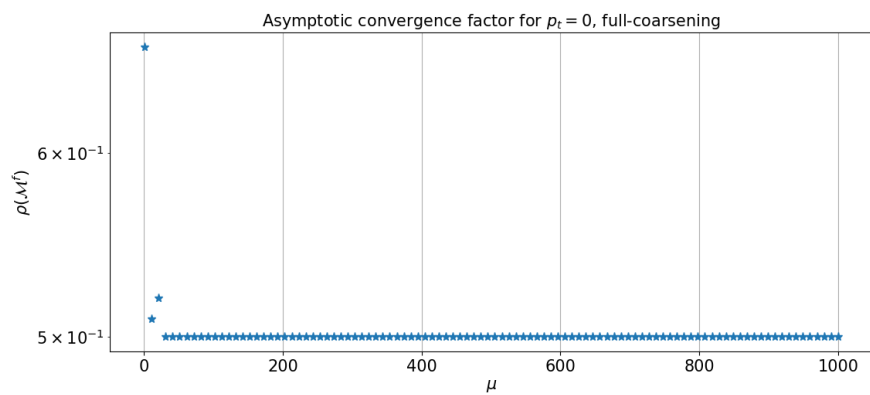
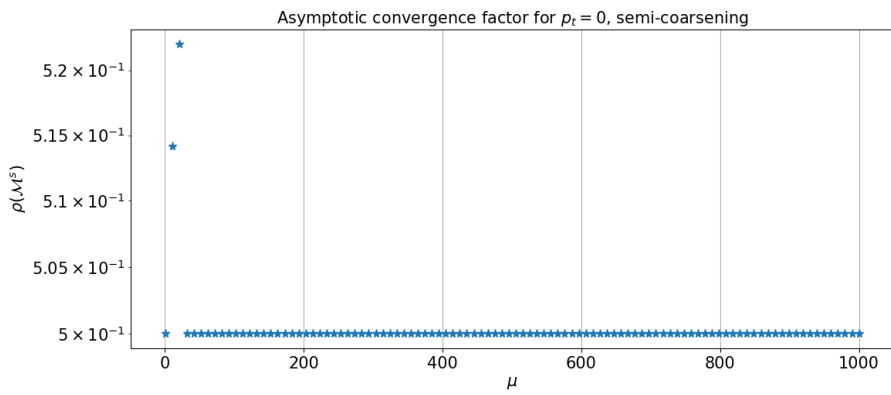


Figure 5.7: Two-grid asymptotic convergence factors for  $p_t = 0$  for different coarsening strategies, calculated with local Fourier analysis.



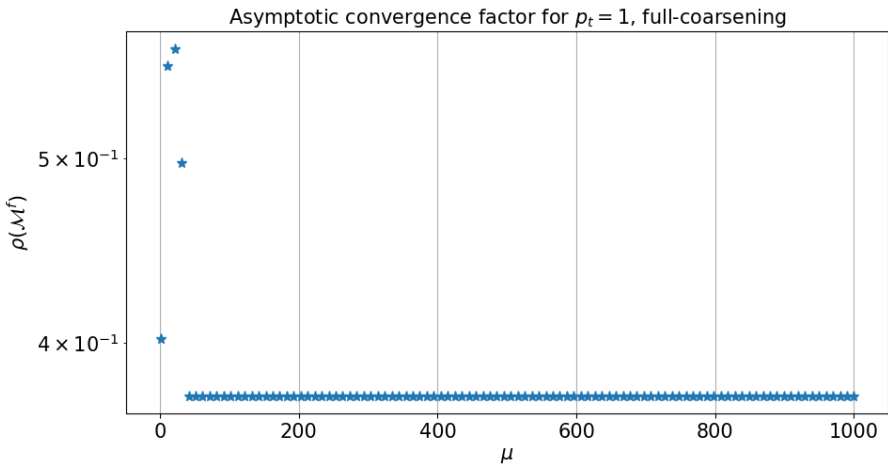
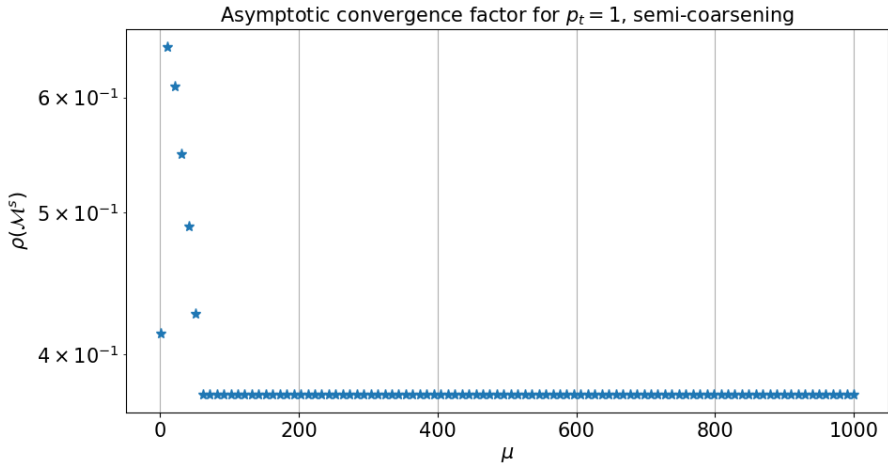


Figure 5.8: Two-grid asymptotic convergence factors for  $p_t = 1$  for different coarsening strategies, calculated with local Fourier analysis.

# Chapter 6

## Conclusions and Outlook

### I Conclusion

In this work we presented a method to construct multigrid based preconditioners for implicit DG-SEM discretizations. These discretizations need an efficient solver for the resulting nonlinear system since the size of the system is connected to the order of the method, i.e. the number of unknowns increases with increasing polynomial degree and dimension. In particular, this leads to large dense Jacobian blocks. Therefore we suggested to use a Jacobian-free Newton-Krylov solver, which is favorable with regards to memory consumption. However, in order to improve the convergence speed of the GMRES sub-solver, a preconditioner needs to be constructed.

Since we are interested in a Jacobian-free solver, the problem arose how to construct a good and efficient preconditioner without using the Jacobian. In this thesis we presented a novel idea for constructing a Jacobian-free preconditioner for the JFNK approach using multigrid methods. We suggested to construct a multigrid preconditioner based on a lower order replacement operator, i.e. a first order finite volume replacement operator. The choice of this replacement operator was motivated by the equivalence of a DG-SEM discretization and a high order FV discretization. With this ansatz, we avoided constructing the Jacobian of the original DG operator while keeping the number of degrees of freedom in the replacement operator the same. Moreover, using a FV replacement operator allows to apply available knowledge about fast multigrid schemes for FV discretizations, e.g. to equip these multigrid methods with a state of the art low memory  $W_3$  smoother. In the attached publications we showed the efficiency of the suggested preconditioner for both linear and nonlinear test problems and can therefore show the

potential of the suggested multigrid preconditioned JFNK solvers for conservation laws.

Recently, space-time DG methods have become of interest. In these methods the temporal direction is discretized using DG schemes as well, resulting in large nonlinear systems. In order to extend the efficient preconditioners to the space-time setting, we need space-time multigrid methods. These methods have to be constructed carefully since the spatial and temporal directions have different properties, e.g. with regards to information flow, but also require different coarsening strategies.

In order to improve the insight into such space-time multigrid methods, we presented and performed a Local Fourier Analysis. In the core part of this thesis we discussed an LFA for the one-dimensional linear advection equation as model problem, discretized with DG-SEM of variable order in time and order 0 in space. We showed how to calculate smoothing and convergence factors using the Fourier symbols of the operators, which is the only feasible option for large systems resulting from such discretizations. The results and insights we gained with this analysis are valuable for constructing efficient space-time multigrid based preconditioners.

## 2 Future work

There are several options for future directions for the work presented in this thesis. The most intuitive one is to generalize the preconditioners presented in the publications to a space-time framework. This might include constructing space-time multigrid preconditioners for space-time DG-SEM solvers using a replacement (space-time) FV operator as we suggested in the publications. When doing so, the option to coarsen in only some directions could be investigated with regards to the efficiency of the multigrid preconditioner. Moreover, considering space-time discretizations in two or three dimensions also implies the need for parallelization.

Then, the focus can be laid on either the construction of new numerical methods, i.e. optimizing the resulting preconditioner and analyzing the effects of different smoothers on the spatial and temporal direction. Another option is to consider a specific application and apply the suggested multigrid preconditioners to these problem, eventually using open source software for PDE solvers and extend them with multigrid based preconditioners.

# References

- [1] Y. Allaneau, L. Y. Li, and A. Jameson. Convergence Acceleration of High Order Numerical Simulations using a Hybrid Spectral Difference-Finite Volume Multigrid Method. In *7th International Conference on Computational Fluid Dynamics (ICCFD7)*, pages 9–13, 2012.
- [2] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9(1):17–29, 1951. doi: 10.1090/qam/42792.
- [3] P. Birken. Optimizing Runge-Kutta smoothers for unsteady flow problems. *Electronic Transactions on Numerical Analysis*, 39(1):298–312, 2012.
- [4] P. Birken. Numerical methods for the unsteady compressible Navier-Stokes equations, Habilitationsschrift. 2012.
- [5] P. Birken and A. Jameson. On nonlinear preconditioners in Newton-Krylov methods for unsteady flows. *International Journal for Numerical Methods in Fluids*, 62(5): 565–573, 2010. doi: 10.1002/flid.2030.
- [6] P. Birken, G. Gassner, M. Haas, and C.-D. Munz. Preconditioning for modal discontinuous Galerkin methods for unsteady 3D Navier-Stokes equations. *Journal of Computational Physics*, 240:20–35, 2013. doi: 10.1016/j.jcp.2013.01.004.
- [7] P. Birken, J. Bull, and A. Jameson. Preconditioned smoothers for the full approximation scheme for the RANS equations. *Journal of Scientific Computing*, 78(2): 995–1022, 2019. doi: 10.1007/s10915-018-0792-9.
- [8] P. Birken, G. J. Gassner, and L. M. Versbach. Subcell finite volume multigrid preconditioning for high-order discontinuous Galerkin methods. *International Journal of Computational Fluid Dynamics*, pages 1–9, 2019. doi: 10.1080/10618562.2019.1667983.

- [9] P. D. Boom and D. W. Zingg. High-Order Implicit Time-Marching Methods Based on Generalized Summation-by-Parts Operators. *SIAM Journal on Scientific Computing*, 37:A2682–A2709, 2015. doi: 10.1137/15M1014917.
- [10] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977. doi: 10.2307/2006422.
- [11] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, 2000. doi: 10.1137/1.9780898719505.
- [12] M. H. Carpenter and D. Gottlieb. Spectral methods on arbitrary grids. *Journal of Computational Physics*, 129(1):74–86, 1996. doi: 10.1006/jcph.1996.0234.
- [13] M. H. Carpenter, D. Gottlieb, and S. Abarbanel. The stability of numerical boundary treatments for compact high-order finite-difference schemes. *Journal of Computational Physics*, 108(2):272–295, 1991. doi: 10.1006/jcph.1993.1182.
- [14] P. Deuffhard. *Newton Methods for Nonlinear Problems*. Springer, Berlin, Heidelberg, 2004. doi: 10.1007/978-3-642-23899-4.
- [15] S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing*, 17(1):16–32, 1996. doi: 10.1137/0917003.
- [16] R. P. Fedorenko. The speed of convergence of one iterative process. *USSR Computational Mathematics and Mathematical Physics*, 4(3):227–235, 1964. doi: 10.1016/0041-5553(64)90253-8.
- [17] K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal. p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 207(1):92–113, 2005. doi: 10.1016/j.jcp.2005.01.005.
- [18] T. C. Fisher and M. H. Carpenter. High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains. *Journal of Computational Physics*, 252:518–557, 2013. doi: 10.1016/j.jcp.2013.06.014.
- [19] T. C. Fisher, M. H. Carpenter, J. Nordström, N. K. Yamaleev, and C. Swanson. Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: Theory and boundary conditions. *Journal of Computational Physics*, 234:353–375, 2013. doi: 10.1016/j.jcp.2012.09.026.

- [20] L. Friedrich, G. Schnücke, A. R. Winters, D. C. R. Fernández, G. J. Gassner, and M. H. Carpenter. Entropy Stable Space–Time Discontinuous Galerkin Schemes with Summation-by-Parts Property for Hyperbolic Conservation Laws. *Journal of Scientific Computing*, 80(1):175–222, 2019. doi: 10.1007/s10915-019-00933-2.
- [21] M. J. Gander and M. Neumüller. Analysis of a new space-time parallel multigrid algorithm for parabolic problems. *SIAM Journal on Scientific Computing*, 38(4): A2173–A2208, 2016. doi: 10.1137/15M1046605.
- [22] G. Gassner and D. A. Kopriva. A comparison of the dispersion and dissipation errors of Gauss and Gauss-Lobatto discontinuous Galerkin spectral element methods. *SIAM Journal on Scientific Computing*, 33(5):2560–2579, 2011. doi: 10.1137/100807211.
- [23] G. J. Gassner. A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. *SIAM Journal on Scientific Computing*, 35(3):A1233–A1253, 2013. doi: 10.1137/120890144.
- [24] W. Hackbusch. Parabolic multi-grid methods. In R. Glowinski and J.-L. Lions, editors, *Computing Methods in Applied Sciences and Engineering IV*, pages 189–197. Elsevier Science Publisher B.V., Noth-Holland, 1984.
- [25] E. Hairer and G. Wanner. *Solving ordinary differential equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2010. doi: 10.1007/978-3-642-05221-7.
- [26] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006. doi: 10.1007/3-540-30666-8.
- [27] P. W. Hemker, W. Hoffmann, and M. Van Raalte. Two-level Fourier Analysis of a Multigrid Approach for Discontinuous Galerkin Discretization. *SIAM Journal on Scientific Computing*, 25(3):1018–1041, 2003. doi: 10.1137/S1064827502405100.
- [28] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer-Verlag New York, 2008. doi: 10.1007/978-0-387-72067-8.
- [29] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*, volume 21. Cambridge University Press, 2007. doi: 10.1017/CBO9780511618352.

- [30] F. Hindenlang, G. J. Gassner, C. Altmann, A. Beck, M. Staudenmaier, and C.-D. Munz. Explicit discontinuous Galerkin methods for unsteady problems. *Computers & Fluids*, 61:86–93, 2012. doi: 10.1016/j.compfluid.2012.03.006.
- [31] G. Horton and S. Vandewalle. A Space-Time Multigrid Method for Parabolic Partial Differential Equations. *SIAM Journal on Scientific Computing*, 16:848–864, 1995. doi: 10.1137/0916050.
- [32] H. T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*, page 4079, 2007. doi: 10.2514/6.2007-4079.
- [33] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In *10th Computational Fluid Dynamics Conference*, page 1596, 1991. doi: 10.2514/6.1991-1596.
- [34] A. Jameson. Evaluation of fully implicit Runge Kutta schemes for unsteady flow calculations. *Journal of Scientific Computing*, 73(2-3):819–852, 2017. doi: 10.1007/s10915-017-0476-x.
- [35] A. Jameson and D. Caughey. How many steps are required to solve the Euler equations of steady, compressible flow-In search of a fast solution algorithm. In *15th AIAA Computational Fluid Dynamics Conference*, page 2673, 2001. doi: 10.2514/6.2001-2673.
- [36] L. O. Jay. *Lobatto Methods*, pages 817–826. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. doi: 10.1007/978-3-540-70529-1\_123.
- [37] C. M. Klaij, M. H. van Raalte, H. van der Ven, and J. J. van der Vegt. h-Multigrid for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 227(2):1024–1045, 2007. doi: 10.1016/j.jcp.2007.08.034.
- [38] D. A. Knoll and D. E. Keyes. Jacobian-Free Newton-Krylov Methods: A Survey of Approaches and Applications. *Journal of Computational Physics*, 193(2):357–397, 2004. doi: 10.1016/j.jcp.2003.08.010.
- [39] D. A. Kopriva. *Implementing Spectral Methods for Partial Differential Equations*. Springer, 2009. doi: 10.1007/978-90-481-2261-5.
- [40] D. A. Kopriva, S. L. Woodruff, and M. Y. Hussaini. Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method. *International Journal for Numerical Methods in Engineering*, 53(1):105–122, 2002. doi: 10.1002/nme.394.

- [41] A. Kravchenko and P. Moin. On the effect of numerical errors in large eddy simulations of turbulent flows. *Journal of Computational Physics*, 131(2):310–322, 1997. doi: 10.1006/jcph.1996.5597.
- [42] H.-O. Kreiss and G. Scherer. Finite element and finite difference methods for hyperbolic partial differential equations. In *Mathematical aspects of finite elements in partial differential equations*, pages 195–212. Elsevier, 1974. doi: 10.1016/B978-0-12-208350-1.50012-1.
- [43] G. May, F. Iacono, and A. Jameson. A hybrid multilevel method for high-order discretization of the Euler equations on unstructured meshes. *Journal of Computational Physics*, 229(10):3938–3956, 2010. doi: 10.1016/j.jcp.2010.01.036.
- [44] W. A. Mulder. A new multigrid approach to convection problems. In *11th International Conference on Numerical Methods in Fluid Dynamics*, pages 429–433. Springer, 1989. doi: 10.1016/0021-9991(89)90121-6.
- [45] S. Osher and E. Tadmor. On the convergence of difference approximations to scalar conservation laws. *Mathematics of Computation*, 50(181):19–51, 1988. doi: 10.2307/2007913.
- [46] P.-O. Persson and J. Peraire. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 30(6):2709–2733, 2008. doi: 10.1137/070692108.
- [47] N. Qin, D. Ludlow, and S. Shaw. A matrix-free preconditioned Newton/GMRES method for unsteady Navier-Stokes solutions. *International Journal for Numerical Methods in Fluids*, 33:223–248, 2000. doi: 10.1002/(SICI)1097-0363(20000530)33:2<223::AID-FLD10>3.0.CO;2-V.
- [48] W. H. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Lab., N. Mex.(USA), 1973.
- [49] S. Rhebergen, J. J. van der Vegt, and H. van der Ven. Multigrid optimization for space-time discontinuous Galerkin discretizations of advection dominated flows. In *ADIGMA-A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, pages 257–269. Springer, Heidelberg, Berlin, 2010. doi: 10.1007/978-3-642-03707-8\_18.
- [50] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 2003. doi: 10.1137/1.9780898718003.
- [51] Y. Saad. Iterative methods for linear systems of equations: A brief historical journey. preprint on arXiv:1908.01083, 2019.



- [52] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986. doi: 10.1137/0907058.
- [53] M. Svärd and J. Nordström. Review of summation-by-parts schemes for initial-boundary-value problems. *Journal of Computational Physics*, 268:17–38, 2014. doi: 10.1016/j.jcp.2014.02.031.
- [54] R. C. Swanson, E. Turkel, and C.-C. Rossow. Convergence acceleration of Runge-Kutta schemes for solving the Navier-Stokes equations. *Journal of Computational Physics*, 224(1):365–388, 2007. doi: 10.1016/j.jcp.2007.02.028.
- [55] E. Tadmor. Skew-selfadjoint form for systems of conservation laws. *Journal of Mathematical Analysis and Applications*, 103(2):428–442, 1984. doi: 10.1016/0022-247X(84)90139-2.
- [56] E. Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Mathematics of Computation*, 49(179):91–103, 1987. doi: 10.2307/2008251.
- [57] E. Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numerica*, 12:451–512, 2003. doi: 10.1017/S0962492902000156.
- [58] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Elsevier Ltd., 2001.
- [59] L. M. Versbach, P. Birken, and G. J. Gassner. Finite volume based multigrid preconditioners for discontinuous Galerkin methods. *PAMM*, 18(1):e201800203, 2018. doi: 10.1002/pamm.201800203.
- [60] P. Vincent, F. Witherden, B. Vermeire, J. S. Park, and A. Iyer. Towards green aviation with python at petascale. In *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–11. IEEE, 2016. doi: 10.1109/SC.2016.1.
- [61] P. E. Vincent and A. Jameson. Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists. *Mathematical Modelling of Natural Phenomena*, 6(3):97–140, 2011. doi: 10.1051/mmnp/20116305.
- [62] P. Wesseling. *Multigrid Methods*. R.T. Edwards, 2004.
- [63] A. R. Winters, D. Derigs, G. J. Gassner, and S. Walch. A uniquely defined entropy stable matrix dissipation operator for high Mach number ideal MHD and compressible Euler simulations. *Journal of Computational Physics*, 332:274–289, 2017. doi: 10.1016/j.jcp.2016.12.006.

# Scientific publications

## Author contributions

### **Paper I: Finite volume based multigrid preconditioners for discontinuous Galerkin methods**

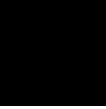
I did the implementation, produced the numerical results and did most of the writing.

### **Paper II: Subcell finite volume multigrid preconditioning for high-order discontinuous Galerkin methods**

I contributed to the development of the presented method, did the implementation and produced the numerical results. I did most of the writing.



Paper I





# Finite volume based multigrid preconditioners for discontinuous Galerkin methods

Lea Miko Versbach<sup>1,\*</sup>, Philipp Birken<sup>1,\*\*</sup>, and Gregor J. Gassner<sup>2,\*\*\*</sup>

<sup>1</sup> Centre for the mathematical sciences, Numerical Analysis, Lund University, Lund, Sweden

<sup>2</sup> Mathematisches Institut, Universität zu Köln, Weyertal 86-90, 50931 Köln, Germany

Our aim is to construct efficient preconditioners for high order discontinuous Galerkin (DG) methods. We consider the DG spectral element method with Gauss-Lobatto-Legendre nodes (DGSEM-GL) for the 1D linear advection equation. It has been shown in [4] that DGSEM-GL has the summation-by-parts (SBP) property and an equivalent finite volume (FV) discretization is presented in [3]. Thus we present a multigrid (MG) preconditioner based on a simplified FV discretization.

© 2018 Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim

## 1 Introduction

An efficient implicit DG variant is DGSEM, where the interpolation of the flux is collocated with the numerical quadrature used for the inner products, [4]. Key to an efficient algorithm is a fast solver with low memory footprint. Our aim is to construct a matrix-free preconditioner using approximations to the FV discretization. We solve the 1D advection equation with DGSEM using a right preconditioner based on an agglomeration multigrid method. We show first results for 2-, 3- and 4-stage Runge-Kutta (RK) smoothers with optimized parameters from [1].

## 2 Discontinuous Galerkin Spectral Element Method

To introduce the DGSEM method, we consider the one-dimensional linear scalar advection equation

$$u_t + au_x = 0 \tag{1}$$

with  $a > 0$ , periodic boundary and suitable initial conditions. We construct a grid with  $M$  elements and introduce a nodal polynomial approximation of  $N + 1$  degrees of freedom  $\{u_j\}_{j=0}^N$  located at the element grid nodes  $x_0^n, \dots, x_N^n$

$$u_n(x, t) \approx u_h(x, t) = \sum_{j=0}^N u_j(t) \ell_j(x), \tag{2}$$

in each element. We choose GL grid nodes and Lagrange basis functions  $\ell_j(x)$  of degree  $N$ . Multiplying with a test function  $\ell \in \{\ell_j\}_{j=0}^N$  and integrating over the element transformed to  $[-1, 1]$  gives the weak form of (1). Inserting the numerical approximation (2) and integration by parts yields

$$\int_{-1}^1 \dot{u}_h \ell_j dx + [u^* \ell_j]_{-1}^1 - \int_{-1}^1 u_h \ell_j' dx = 0, \quad j = 0, \dots, N \tag{3}$$

with  $\dot{u}_h$  the time derivative of (2),  $(\cdot)'$  the spatial derivative w.r.t  $x$  and  $u^*$  the numerical upwind flux between elements. Integration by numerical quadrature with GL nodes and Lagrange basis yields a matrix vector formulation on each element:

$$\frac{\Delta x_n}{2} M \dot{u} - D^T M u = B u^*, \tag{4}$$

$D_{ki} = \ell_i(x_k)$ ,  $i, k = 0, \dots, N$  and  $B = \text{diag}([-1, 0, \dots, 0, 1])$ .

## 3 Finite Volume Based Agglomeration Multigrid Preconditioner

We construct a preconditioner for (2) based on a MG method applied to a simple FV discretization of (1). The grid points of the FV cells coincide with the DGSEM elements plus additional  $N$  equidistant points inside of each element. An equidistant FV discretization for (1) with mesh width  $\Delta \tilde{x}$  reads

$$\dot{u}_i + \frac{a}{\Delta \tilde{x}} (u_i - u_{i-1}) = 0, \tag{5}$$

\* Corresponding author: e-mail lea\_miko.versbach@math.lu.se, phone +46 46 222 6811

\*\* e-mail philipp.birken@na.lu.se

\*\*\* e-mail gassner@math.uni-koeln.de

for the cell average  $u_i$  on cell  $i = 1, \dots, K$ . Implicit Euler time stepping yields a linear system to be solved in each time step:

$$u^{n+1} - u^n + \frac{a\Delta t}{\Delta x} C u^{n+1} = 0 \Leftrightarrow A u^{n+1} = u^n, \tag{6}$$

for  $A = I + \frac{\nu}{\Delta x} C \in \mathcal{R}^{K \times K}$  and  $\nu = a\Delta t$ . The two-grid preconditioner for a smoother  $x^{k+1} = M_S x^k + N_S^{-1} b$  reads

$$(I - P_1^0 A_0^{-1} R_0^1 A_1) N_S^{-1} + P_1^0 A_0^{-1} R_0^1 A_1. \tag{7}$$

We consider  $s$ -stage RK schemes with initial condition  $u_0 = u^n$  as smoothers:

$$u_j = u^n + \alpha_j \Delta t^* (u^n - A u_{j-1}), \quad j = 1, \dots, s - 1, \\ u_{n+1} = u^n + \Delta t^* (u^n - A u_{s-1}).$$

On each grid level  $l$ , the explicit pseudo time step is defined by  $\Delta t_l^* = c\Delta x_l/\nu$  for mesh width  $\Delta x_l$ . We use optimized  $s$ -stage RK parameters from [1].

### 4 Numerical Results

We test the preconditioner on (1) with  $a = 2$  on the interval  $[0, 2]$  with periodic boundary conditions and initial condition  $\sin(\pi x)$ . We keep the number of unknowns constant 300 and perform one time step with  $CFL = 1$ . In figure 1 we see

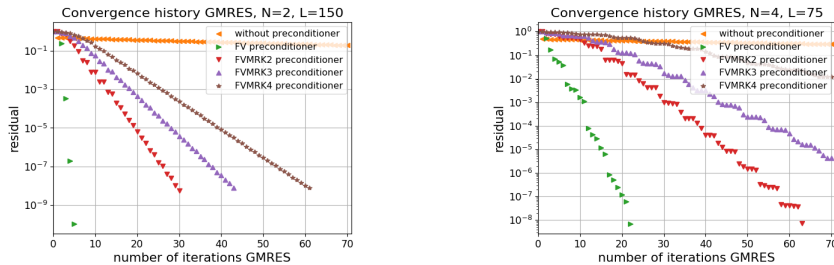


Fig. 1: Performance of preconditioned DGSEM for different optimized smoothers

the effect of the preconditioner. As expected, using the FV method itself results in the best preconditioner. FV MG based preconditioners perform also well, with a visible influence of the smoother. In this setting, the optimized RK2 smoother gives the best results. While unpreconditioned GMRES does not terminate in 70 iterations, using the FV based MG preconditioner with RK2 smoother GMRES terminates in less than 70 iterations. It is moreover noticeable that the preconditioner performs better for smaller  $N$ , namely in the left figure. This might be caused by the optimized smoothing parameters, which are constructed for the case  $N = 1$ .

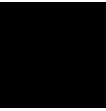
### 5 Conclusions and Outlook

We suggested an FV based MG preconditioner for DGSEM to solve the advection equation in 1D. Using explicit 2-, 3- and 4-stage RK smoothers we have achieved an improved performance compared to the unpreconditioned case. The relatively large influence of the smoother choice suggests to consider more advanced smoothers, such as W-smoothers [2], and to find new optimal parameters for higher order DGSEM discretizations.

### References

- [1] P. Birken, Optimizing Runge-Kutta smoothers for unsteady flow problems, *Electronic Transactions on Numerical Analysis* **39**, pp. 298-312 (2012).
- [2] P. Birken, J. Bull, A. Jameson, Preconditioned smoothers for the full approximation scheme for the RANS equations, *J. Sci. Comput.* preprint (2016).
- [3] T. C. Fisher, M. H. Carpenter, J. Nordström, N. K. Yamaleev, C. Swanson, Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: Theory and boundary conditions, *Journal of Computational Physics* **234**, pp. 353-375 (2013).
- [4] G. J. Gassner, A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods, *J. Sci. Comput.* **35**, pp. A1233-A1253 (2012).
- [5] D. A. Kopriva, *Implementing Spectral Element Methods for Partial Differential Equations*, Scientific Computing, Springer, Dordrecht (2009).

Paper II







## Subcell finite volume multigrid preconditioning for high-order discontinuous Galerkin methods

Philipp Birken<sup>a</sup>, Gregor J. Gassner<sup>b</sup> and Lea M. Versbach<sup>a</sup>

<sup>a</sup>Numerical Analysis, Centre for the Mathematical Sciences, Lund University, Lund, Sweden; <sup>b</sup>Department for Mathematics and Computer Science, Center for Data and Simulation Science, University of Cologne, Köln, Germany

### ABSTRACT

We suggest a new multigrid preconditioning strategy for use in Jacobian-free Newton–Krylov (JFNK) methods for the solution of algebraic equation systems arising from implicit Discontinuous Galerkin (DG) discretisations. To define the new preconditioner, use is made of an auxiliary first-order finite volume discretisation that refines the original DG mesh, but can still be implemented algebraically. As smoother, we consider the pseudo-time iteration W3 with a symmetric Gauss–Seidel-type approximation of the Jacobian. As a proof of concept numerical tests are presented for the one-dimensional Euler equations, demonstrating the potential of the new approach.

### ARTICLE HISTORY

Received 14 December 2018  
Accepted 7 August 2019

### KEYWORDS


Discontinuous Galerkin;  
multigrid preconditioner;  
W smoother; Euler equations

### 1. Introduction

The goal of our research is the construction of efficient Jacobian-free preconditioners for high order Discontinuous Galerkin (DG) discretisations with implicit time integration. One of our main interests is three-dimensional unsteady compressible flow. High-order DG methods (and related methods such as Flux Reconstruction (FR) discretisations) offer great potential for Large Eddy Simulation (LES) of turbulent flows with geometries, such as jet engines. The idea of DG (or FR) is to approximate the solution element-wise using a polynomial, which is allowed to be discontinuous across element interfaces, see Kopriva (2009) and Huynh (2007). Communication and coupling of degrees of freedom (DOF) is only across faces, whereas the element-local computations are very dense. As a result, DG methods are very well suited for domain-decomposition-based parallelisation (see, e.g. Hindenlang et al. 2012; Vincent et al. 2016). The specific variant we consider is the *DG Spectral Element Method* (DG-SEM), e.g. Kopriva, Woodruff, and Hussaini (2002). We use a Lagrange-type (nodal) basis with Gauss–Lobatto (GL) quadrature nodes with the collocation of the discrete integration. These choices yield DG operators that satisfy the summation-by-parts (SBP) property (see Gassner 2013), which is the discrete analogue to integration-by-parts. SBP is key

to construct methods that are discretely entropy stable and/or kinetic energy preserving.

DG discretisation in space results in a big system of ODEs. Due to geometry features and thin boundary layers that occur in challenging compressible turbulent flow applications as the design of jet engines, aeroplanes and wind turbines, the resulting large system of ODEs is stiff. Implicit time integrators can overcome the deficiency of explicit time integrators with restrictive CFL conditions. However, efficiency can only be restored when the arising large non-linear systems are solved efficiently in terms of CPU time, but also regarding memory consumption. Vincent and Jameson mention that solvers for linear and non-linear equation systems are severely lacking for 3D DG applications as one of four major obstacles that need to be solved if high-order methods are to be widely adopted by, e.g. industry, Vincent and Jameson (2011). Candidates for solvers are *FAS-Multigrid* (full-approximation multigrid scheme) and *preconditioned Jacobian-Free Newton–Krylov methods* (JFNK) (Knoll and Keyes 2004) where multigrid can be used as a preconditioner (see Birken and Jameson 2010). The JFNK technology is in general interesting, as the memory use is minimised. Although DG systems have a weakly coupled block structure, the blocks themselves can be large. In particular, the problem for high-order

**CONTACT** Lea M. Versbach  [lea@maths.lth.se](mailto:lea@maths.lth.se), [lea\\_miko.versbach@math.lu.se](mailto:lea_miko.versbach@math.lu.se)

© 2019 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group  
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

DG methods is that the number of unknowns per element increases dramatically with increasing polynomial degree and dimension, leading to large dense Jacobian blocks (see Birken et al. 2013; Birken 2012). For a finite volume method, the block size is  $2+d$  with dimension  $d$ , whereas for a DG-SEM with  $p$ th degree polynomials, it is  $(d+2)(p+1)^d$ . For degree 2 in 3D, this is already 135. The favourable memory consumption of the JFNK approach is obsolete if the preconditioner requires the storage of (parts of) the DG system Jacobian.

Hence in this article, we present a novel idea for the construction of a well-performing preconditioner for the JFNK approach, while retaining the low memory use, i.e. a Jacobian-free preconditioner. The main ingredient consists in the construction of a simplified replacement operator. A motivation for this is a previously proved equivalence between a DG-SEM discretisation and a high order FV discretisation, see Fisher et al. (2013). One could, for instance, choose a different polynomial order in the element to generate a replacement operator as in Fidkowski et al. (2005) and Birken et al. (2013). However, we aim to retain the number of DOFs in our replacement operator by introducing subcells in each element, namely  $p+1$  in each spatial direction. On this subcell-element grid, the simplest replacement operator is a first-order finite volume (FV) discretisation. In some sense, we reinterpret the nodal values as input for an FV method. This gives a semi-structured–unstructured approximation, where the elements are unstructured, but inside the element the subcells are structured (Versbach, Birken, and Gassner 2018). We extend the idea of this paper to the Euler equations. In the resulting approximate Jacobian, we now only have  $(d+2)(p+1)(2d+1)$  entries (Birken 2012). Furthermore, it allows to use the available knowledge about fast multigrid (MG) methods for FV discretisations on (block-)structured meshes. As a smoother for our FV discretisation, we use a state of the art low memory W3 smoother from Birken, Bull, and Jameson (2018).

A related approach was proposed in Allaneau, Li, and Jameson (2012) for spectral difference (SD) methods, where the replacement operator is also an FV discretisation on a potentially fine grid. However, there the FV grid is not embedded in the high-order grid, but overlaid. Thus, it is necessary to interpolate the solution (and the residual) in-between different grids (with different topologies), which needs interpolation

and reconstruction operators similar to Chimera techniques. In contrast, we want to harness in particular the semi-unstructured–structured mesh-topology: our FV discretisation basically lives on the same DOFs as the nodal high-order DG method.

In the remainder of the paper, we first describe our prototype problem, the one-dimensional compressible Euler equations. We then present the DG-SEM and the FV subcell discretisations as well as the multigrid solver. In the last part of the paper, we show numerical experiments to validate the approach and draw our conclusions.

## 2. One-dimensional Euler equations

As a prototype problem for our novel idea, we consider the one-dimensional compressible Euler equations for a perfect gas

$$\begin{pmatrix} \rho \\ m \\ \rho E \end{pmatrix}_t + \begin{pmatrix} m \\ mv + p \\ Hm \end{pmatrix}_x = 0, \quad (1)$$

with appropriate initial and boundary conditions. Here  $\rho$  is the density,  $m = \rho v$  the momentum,  $p$  the pressure,  $E$  the energy and  $H = E + p/\rho$  the enthalpy. Defining  $U = (\rho, m, \rho E)^T$  and  $f(U) = (m, mv + p, Hm)^T$ , we can write the Euler equations in vector form:

$$U_t + f(U)_x = 0. \quad (2)$$

## 3. Spatial discretisation

### 3.1. DG-SEM

For the spatial discretisation, we introduce a grid with  $K$  elements  $e^k$  of width  $\Delta x_k$ ,  $k = 1, \dots, K$ . Each element is transformed to the reference element  $[-1, 1]$  by a linear mapping with Jacobian  $J_k := \Delta x_k/2$ . The solution is approximated by a nodal polynomial in reference space with degree  $p$  in each element  $e^k$

$$U(\xi, t)|_{e^k} \approx U^k(\xi, t) = \sum_{j=1}^{p+1} U_j^k(t) \psi_j(\xi), \quad (3)$$

where the interpolation nodes are the GL nodes  $\{\xi_j\}_{j=1}^{p+1}$ . We use the element mapping to transform the Euler equations into reference space

$$J_k U_t + f(U)_\xi = 0, \quad (4)$$

and insert our ansatz (3). Next, we integrate over the reference element, use integration-by-parts for the flux term and replace the fluxes at the element interfaces with so-called numerical flux functions  $f^*$  to arrive at

$$\int_{-1}^1 J_k U_t^k \psi(\xi) d\xi + [f^* \psi(\xi)]_{-1}^1 - \int_{-1}^1 f(U^k) \psi_{\xi}(\xi) d\xi = 0. \quad (5)$$

As a numerical flux function, we choose the Rusanov flux (or local Lax–Friedrich flux)

$$f^*(U^-, U^+) = \frac{f(U^-) + f(U^+)}{2} - \frac{\lambda_{\max}}{2} (U^+ - U^-), \quad (6)$$

where  $U^-$ ,  $U^+$  are the values left and right at an element interface and  $\lambda_{\max}$  is an estimate of the maximum wave speed at the interface.

As stated above, the main idea of the DG-SEM is collocation. We use collocation for our discrete integration, i.e. we replace the integrals in (5) with GL quadrature rules at the same location as our interpolation, which can be interpreted as a collocation of the non-linear fluxes  $f(U)$ . With this choice, the DG-SEM operators simplify a lot: we get the diagonal mass matrix  $M_{ij} = \delta_{ij} \omega_i$ ,  $i, j = 1, \dots, p+1$ , the diagonal boundary matrix  $\mathbf{B} = \text{diag}(-1, 0, \dots, 0, 1)$  and the derivative matrix  $D_{ij} := (\psi_j)_{\xi}(\xi_i)$ ,  $i, j = 1, \dots, p+1$ . Replacing the integrals in (5) and using the definitions of the DG-SEM operators, we arrive at the DG-SEM method in the matrix–vector formulation for one cell  $e^k$

$$\dot{\mathbf{U}}^k = -\frac{1}{J_k} (\mathbf{M}^{-1} \mathbf{B} \mathbf{f}^* - \mathbf{M}^{-1} \mathbf{D}^T \mathbf{M} \mathbf{f}). \quad (7)$$

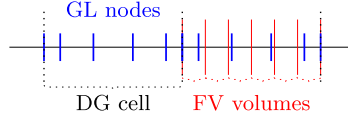
The elemental residuals are coupled through the numerical flux  $f^*$  with DOFs from other elements connected via the interfaces. Collecting the equations for all elements in one big system with unknown  $\mathbf{u}$  and applying implicit Euler in time gives

$$\mathbf{u}^{n+1} - \mathbf{u}^n - \Delta t \mathbf{G}(\mathbf{u}^{n+1}) =: \mathbf{F}(\mathbf{u}^{n+1}) = \mathbf{0}, \quad (8)$$

where  $\mathbf{G}(\mathbf{u}^{n+1})$  collects the DG-SEM residuals, i.e. the right-hand side of (7).

### 3.2. Finite volume discretisation

Based on the DG-SEM discretisation, we define an FV discretisation on a subcell mesh with  $p+1$  cells



**Figure 1.** Two DG cells with 6 GL nodes each. An equidistant FV mesh is shown in the right cell.

(see Figure 1). The discretisation on this semi-structured–unstructured grid is used to define the preconditioner and to construct the multigrid method. An FV method is based on approximating the cell averages in a subcell  $i$  in an element  $e^k$

$$\frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} U(x, t^n) dx \approx U_i^{k,n} \quad (9)$$

at each time level  $t^n$ . For a subcell in the element  $e^k$  with subcell size  $\Delta x_i$ , the subcell FV discretisation reads

$$\dot{U}_i^k + \frac{1}{\Delta x_i} (f_{i+1/2}^* - f_{i-1/2}^*) = 0, \quad i = 1, \dots, p+1; k = 1, \dots, K. \quad (10)$$

The numerical flux function  $f_{i+1/2}^*$  is again Rusanov flux (6), with the values left and right being not the polynomial values, but the subcell average values instead.

## 4. Preconditioned Jacobian-free Newton-GMRES

The resulting system of nonlinear equations (8) is solved using Newton’s method, written for the equation  $\mathbf{F}(\mathbf{u}) = \mathbf{0}$ :

$$\text{solve } \frac{\partial \mathbf{F}(\mathbf{u})}{\partial \mathbf{u}} \Big|_{\mathbf{u}^{(k)}} \mathbf{s} = -\mathbf{F}(\mathbf{u}^{(k)}), \quad (11)$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{s}, \quad k = 0, 1, \dots$$

for a given initial guess  $\mathbf{u}^{(0)}$ . The linear system in (11) is solved using right preconditioned GMRES with a relative tolerance.

In order to not compute the Jacobian in each iteration (11), we replace the matrix–vector products appearing in GMRES by a difference quotient:

$$\frac{\partial \mathbf{F}(\mathbf{u})}{\partial \mathbf{u}} \mathbf{q} \approx \frac{\mathbf{F}(\mathbf{u} + \epsilon \mathbf{q}) - \mathbf{F}(\mathbf{u})}{\epsilon}, \quad \text{with } \epsilon = \frac{1e-7}{\|\mathbf{q}\|}. \quad (12)$$



authors. We follow a recent version from Birken, Bull, and Jameson (2018).

The first step is to approximate the Jacobian by using a different first-order discretisation of the linearised Euler equation. It is based on a splitting  $\mathbf{A} = \mathbf{A}^+ + \mathbf{A}^-$  of the flux Jacobian. This is evaluated in the average of the values on both sides of the interface. The split Jacobians correspond to positive and negative eigenvalues and can be written in terms of the matrix of right eigenvectors  $\mathbf{Q}$  as

$$\mathbf{A}^+ = \mathbf{Q}\mathbf{\Lambda}^+\mathbf{Q}^{-1}, \quad \mathbf{A}^- = \mathbf{Q}\mathbf{\Lambda}^-\mathbf{Q}^{-1},$$

where  $\mathbf{\Lambda}^\pm$  are diagonal matrices containing the positive and negative eigenvalues, respectively. These are then bounded away from zero using a parabolic function which takes care when the modulus of the eigenvalue  $\lambda$  is smaller or equal to a fraction  $ad$  of the speed of sound  $a$  with free parameter  $d \in [0, 1]$ :

$$|\lambda| = \frac{1}{2} \left( ad + \frac{|\lambda|^2}{ad} \right), \quad |\lambda| \leq ad. \quad (21)$$

With this, an upwind discretisation of the split Jacobian is given in cell  $i$  by

$$\begin{aligned} \mathbf{u}_{i*} + \mathbf{u}_i + \frac{\Delta t}{\Delta x_i} ((\mathbf{A}_{ii}^+ \mathbf{u}_i + \mathbf{A}_{i,i+1}^- \mathbf{u}_{i+1}) \\ - (\mathbf{A}_{i-1,i}^+ \mathbf{u}_{i-1} + \mathbf{A}_{ii}^- \mathbf{u}_i)) = 0. \end{aligned} \quad (22)$$

The corresponding approximation of the Jacobian is then used to construct a preconditioner. Specifically, we consider the block SGS preconditioner

$$\underline{\mathbf{W}}^{-1} = (\underline{\mathbf{D}} + \underline{\mathbf{L}})^{-1} \underline{\mathbf{D}} (\underline{\mathbf{D}} + \underline{\mathbf{U}})^{-1}, \quad (23)$$

where  $\underline{\mathbf{L}}$ ,  $\underline{\mathbf{D}}$  and  $\underline{\mathbf{U}}$  are block matrices with  $3 \times 3$  blocks.

We have  $\underline{\mathbf{L}} + \underline{\mathbf{D}} + \underline{\mathbf{U}} = \mathbf{I} + \eta \Delta t^* \mathbf{J}$  and obtain

$$\mathbf{L}_{i-1,i} = -\frac{\eta \Delta t \Delta t_i^*}{\Delta x_i} \mathbf{A}_{i-1,i}^+, \quad \mathbf{U}_{i,i+1} = \frac{\eta \Delta t \Delta t_i^*}{\Delta x_i} \mathbf{A}_{i,i+1}^-, \quad (24)$$

$$\underline{\mathbf{D}}_{ii} = \mathbf{I} + \eta \Delta t^* \mathbf{I} + \frac{\eta \Delta t \Delta t_i^*}{\Delta x_i} (\mathbf{A}_{ii}^+ - \mathbf{A}_{ii}^-).$$

Applying this preconditioner requires solving  $3 \times 3$  systems coming from the diagonal, which can be done directly. A fast implementation is obtained by transforming first to a certain set of symmetrising variables (see Swanson, Turkel, and Rossow 2007).

## 6. Numerical results

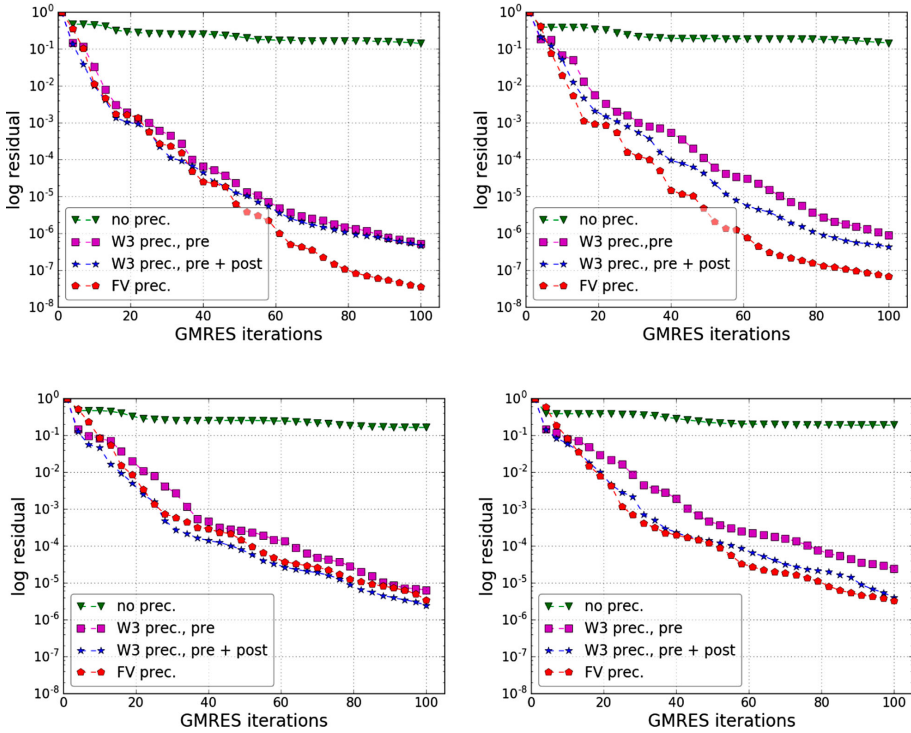
We consider the one-dimensional Euler equations on the interval  $[0, 10]$  and study one step of implicit Euler where we look at the convergence rate of GMRES with maximal 100 iterations in the first Newton step. All results are produced in Python. Measuring the CPU time will not give a great insight into performance in the one-dimensional case and is therefore not discussed in the following, but is important to consider in higher dimensions.

We equip the Euler equations with periodic boundary conditions and consider two different initial conditions: A subsonic case  $(\rho_0, v_0, p_0) = (1 + 0.1 \sin(2\pi x/10), 1, 28)$  with Mach number 0.16 and a transonic case  $(\rho_0, v_0, p_0) = (1 + 0.1 \sin(2\pi x/10), 1, 1)$  with Mach number 0.85. We choose a CFL number of 100 and test the discussed MG preconditioner with W3 smoother for a V cycle, see (15). A simple block Jacobi preconditioner with blocks corresponding to the  $3 \times 3$  systems does not improve the convergence rate of the GMRES cycle compared to no preconditioner. This motivates to consider more sophisticated preconditioners for the given problem. We also note that applying the W3 presmoothing step several times gives almost no convergence improvement while being very expensive in terms of computational costs. The same holds for using a W cycle or two consecutive V cycles, as well as for the method of nonsymmetric Restriction Aggregation (NSR) from Sala and Tuminaro (2008).

We test the framework for 4th- and 8th-order DG methods with 240 and 480 DOFs, respectively. In order to have a reference for efficiency, we construct a preconditioner based on the Jacobian of FV discretisation (12). The new multigrid preconditioner approximates the inverse of the FV Jacobian and thus cannot be expected to behave superior to the reference preconditioner. The reference preconditioner is applied by using GMRES with tolerance  $1e-14$  and maximal 300 iterations. This is very expensive in terms of computations and only suggested to compare how well the proposed MG preconditioner approximates the inverse of the FV Jacobian.

### 6.1. Subsonic case

In the subsonic case, the pseudo-CFL number for the W3 smoother is  $c^* = 10$ ,  $\eta = 1.4$  and  $d = 0.1$ . We consider two different MG preconditioners: one with only one presmoothing step on each level and one with one

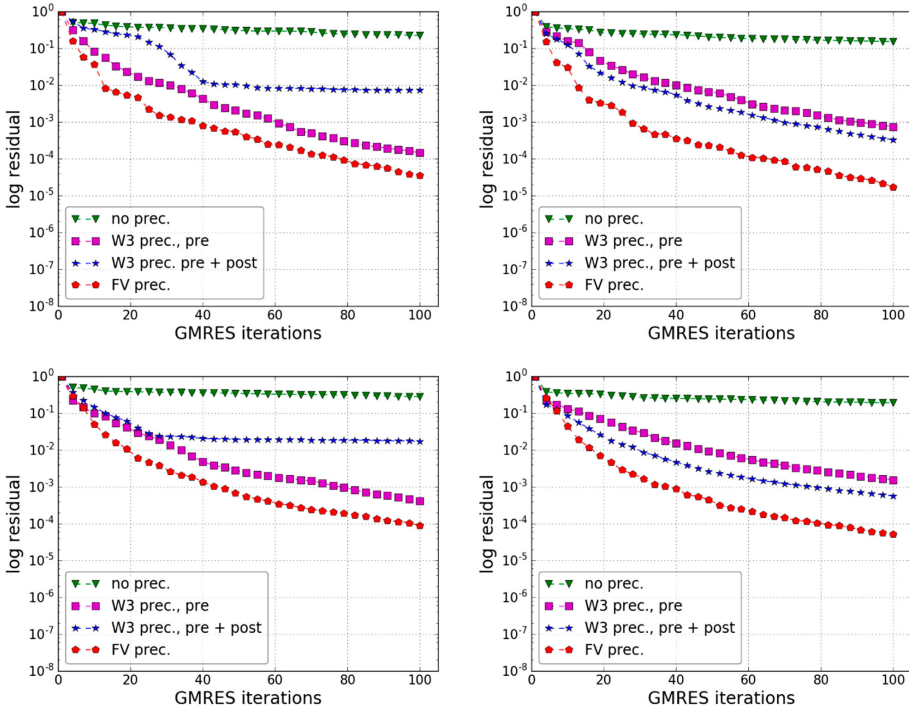


**Figure 2.** Convergence history for GMRES for one Newton iteration, subsonic case, DG order 4 (top) and 8 (bottom), 240 DOFs (left) and 480 DOFs (right).

pre- and one postsmoothing step on each level. The convergence results are shown in Figure 2.

We see that the reference preconditioner (FV) gives very good results for all four test cases, but works better for 4th order. In particular, there is fast initial convergence, which is crucial to get fast termination within an inexact Newton’s method. The suggested MG preconditioners yield a very good approximation to the reference preconditioner, especially within the first 20 GMRES iterations. The MG preconditioner with pre- and postsmoothing gives the best results, outperforming the MG preconditioner equipped with only pre-smoothing slightly. This holds for DG solvers of 4th and 8th order as well as for DOFs 240 and 480. Increasing the DOFs does not have a visible impact on the convergence rate of the reference preconditioner for both 4th and 8th order while we can notice small differences in

the behaviour of the MG preconditioners. The residual after 100 GMRES iterations differs slightly more for the two MG preconditioners when increasing the DOFs for both 4th and 8th order. Since the FV replacement operator is of the first order, the question arises how it behaves for the increased order of the DG method. When going from 4th to 8th order for 240 DOFs, there are two orders of magnitude decrease of the reference preconditioner, whereas the decrease for the W3 preconditioner is very small and they behave very similarly. For 480 DOFs, there is a 1.5 order of magnitude decrease of the reference preconditioner and the MG preconditioners behave similarly. It is noticeable that the MG preconditioners mimic the FV preconditioner better when increasing the order of the DG discretisation. We expect a performance improvement of the smoother for optimised  $c^*$ ,  $\eta$  and  $d$ .



**Figure 3.** Convergence history for GMRES for one Newton iteration, transonic case, DG order 4 (top) and 8 (bottom), 240 DOFs (left) and 480 DOFs (right).

## 6.2. Transonic case

In the transonic case, the pseudo-CFL number for the W3 smoother is  $c^* = 2$ ,  $\eta = 0.7$  and  $d = 0.1$ . We consider the same two different MG preconditioners as in the transonic case. The convergence results are shown in Figure 3.

Both reference FV and MG preconditioners perform worse than in the subsonic case. Such a loss of performance has been reported for a  $p$ -multigrid method in Premasathan et al. (2009). In the transonic case, the reference preconditioner works very similar for 4th and 8th order and different DOFs. We notice only a slight decrease in performance when increasing the order as well as when increasing the DOF. For 240 DOFs, the first MG preconditioner mimics the performance of the reference preconditioner with approximately 0.5 order of magnitude degradation and works slightly better for the 4th order. The second

MG preconditioner does not work well for 240 DOFs and the 4th and 8th order, respectively. When increasing the DOFs, the MG preconditioner with pre- and postsmoothing slightly outperforms the one with presmoothing only. In the case of 480 DOFs, the second MG preconditioner works around 1 order of magnitude worse than the reference preconditioner, giving for both 4th and 8th order methods a result approximately 0.5 order of magnitude degradation compared to 240 DOFs. Again the order affects the performance only slightly for different DOFs.

## 7. Conclusions

We presented a new multigrid preconditioning strategy for use in JFNK methods for the solution of equation systems arising from DG discretisations. The core idea is to make use of an auxiliary first-order FV



discretisation that refines the original DG mesh, but can still be implemented algebraically. As smoother, we consider W3. Numerical results show the potential of the approach as a proof of concept for the one-dimensional Euler equations. A simple block Jacobi preconditioner does not improve the convergence rate compared to no preconditioner at all, which justifies the necessity of using this more sophisticated preconditioner. The convergence results of the proposed preconditioner are promising, being close to a quasi-exact preconditioner in the subsonic case. Our results indicate that the performance of the preconditioner is only weakly influenced by the order of the DG discretisation. A possible extension of the preconditioner to multiple spatial dimensions could be based on a tensor product strategy, which is also the natural approach for the extension of DG-SEM to multiple spatial dimensions. It should be noted that W3 smoothers are designed for high aspect ratio grids and have been shown to achieve even better performance on those compared to equidistant meshes (Birken, Bull, and Jameson 2018). This is of special interest for Navier–Stokes equations with a boundary layer.

### Disclosure statement

No potential conflict of interest was reported by the authors.

### Funding

Philipp Birken and Lea Versbach gratefully acknowledge the support of the Swedish Research Council – Vetenskapsrådet [grant number 2015-04133]. Gregor Gassner has been supported by the European Research Council (ERC) under the EUs 8th Framework Program Horizon 2020 with the research project Extreme [ERC grant agreement no. 714487].

### References

- Allaneau, Yves, Lala Y. Li, and Antony Jameson. 2012. “Convergence Acceleration of High Order Numerical Simulations using a Hybrid Spectral Difference – Finite Volume Multigrid Method.” In *7th International Conference on Computational Fluid Dynamics (ICCFD7)*, Big Island, Hawaii, July 9–13.
- Birken, Philipp. 2012. “Numerical Methods for the Unsteady Compressible Navier-Stokes Equations.” Habilitation Thesis, University of Kassel.
- Birken, Philipp, Jonathan Bull, and Antony Jameson. 2018. “Preconditioned Smoothers for the Full Approximation Scheme for the RANS Equations.” *Journal of Scientific Computing* 78 (2): 995–1022.
- Birken, Philipp, Gregor Gassner, Mark Haas, and Claus-Dieter Munz. 2013. “Preconditioning for Modal Discontinuous Galerkin Methods for Unsteady 3D Navier-Stokes Equations.” *Journal of Computational Physics* 240: 20–35.
- Birken, Philipp, and Antony Jameson. 2010. “On Nonlinear Preconditioners in Newton-Krylov-Methods for Unsteady Flows.” *International Journal of Numerical Methods in Fluids* 62: 565–573.
- Fidkowski, Krzysztof J., Todd A. Oliver, James Lu, and David L. Darmofal. 2005. “p-Multigrid Solution of High-order Discontinuous Galerkin Discretizations of the Compressible Navier-Stokes Equations.” *Journal of Computational Physics* 207: 92–113.
- Fisher, Travis C., Mark H. Carpenter, Jan Nordström, Nail K. Yamaleev, and Charles Swanson. 2013. “Discretely Conservative Finite-difference Formulations for Nonlinear Conservation Laws in Split Form: Theory and Boundary Conditions.” *Journal of Computational Physics* 234: 353–375.
- Gassner, Gregor J. 2013. “A Skew-symmetric Discontinuous Galerkin Spectral Element Discretization and Its Relation to SBP-SAT Finite Difference Methods.” *Journal of Scientific Computing* 35: A1233–A1253.
- Hindenlang, Florian, Gregor J. Gassner, Christoph Altmann, Andrea Beck, Mark Staudenmaier, and Claus-Dieter Munz. 2012. “Explicit Discontinuous Galerkin Methods for Unsteady Problems.” *Computers Fluids* 61 (May): 86–93.
- Huynh, H. T. 2007. “A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods.” In *AIAA, 2007–4079*, Miami, FL.
- Knoll, Dana A., and David E. Keyes. 2004. “Jacobian-free Newton-Krylov Methods: A Survey of Approaches and Applications.” *Journal of Computational Physics* 193: 357–397.
- Kopriva, David A. 2009. *Implementing Spectral Element Methods for Partial Differential Equations*. Dordrecht: Springer Scientific Computing.
- Kopriva, David A., S. L. Woodruff, and M. Y. Hussaini. 2002. “Computation of Electromagnetic Scattering with a Non-conforming Discontinuous Spectral Element Method.” *International Journal for Numerical Methods in Engineering* 53: 105–122.
- Premasathan, Sachin, Chunlei Liang, Antony Jameson, and Z. J. Wang. 2009. “A P-Multigrid Spectral Difference Method for Viscous Compressible Flow Using 2D Quadrilateral Meshes.” *AIAA Paper 2009-950*, Orlando, FL.
- Sala, Marzio, and Raymond S. Tuminaro. 2008. “A New Petrov-Galerkin Smoothed Aggregation Preconditioner for Non-symmetric Linear Systems.” *SIAM Journal on Scientific Computing* 31 (1): 143–166.
- Swanson, R. Charles, Eli Turkel, and Cord-Christian Rossow. 2007. “Convergence Acceleration of Runge-Kutta Schemes for Solving the Navier-Stokes Equations.” *Journal of Computational Physics* 224 (1): 365–388.
- Versbach, Lea M., Philipp Birken, and Gregor J. Gassner. 2018. “Finite Volume Based Multigrid Preconditioners for Discontinuous Galerkin Methods.” *PAMM*, Vol. 18. Munich.

Vincent, Peter E., and Antony Jameson. 2011. "Facilitating the Adoption of Unstructured High-Order Methods Amongst a Wider Community of Fluid Dynamicists." *Mathematical Modelling of Natural Phenomena* 6 (3): 97–140.

Vincent, Peter E., Freddie Witherden, Brian Vermeire, Jin S. Park, and Arvind Iyer. 2016. "Towards Green Aviation with Python at Petascale." SC16: International Conference for High Performance Computing, Networking, Storage and Analysis. Article No. 1. Salt Lake City, UT.