RICE UNIVERSITY

Explicit Discontinuous Galerkin Methods for Linear Hyperbolic Problems

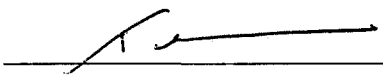
by

Thomas Reid Atcheson

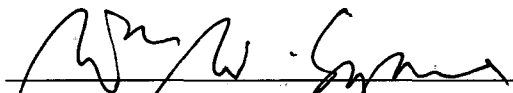
A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Masters of Arts

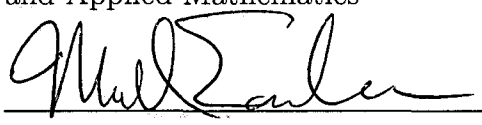
APPROVED, THESIS COMMITTEE:



Timothy Warburton
Associate Professor of Computational and Applied Mathematics



William W. Symes
Noah Harding Professor of Computational and Applied Mathematics



Mark Embree
Professor of Computational and Applied Mathematics

Houston, Texas

April, 2013

ABSTRACT

Explicit Discontinuous Galerkin Methods for Linear Hyperbolic Problems

by

Thomas Reid Atcheson

Discontinuous Galerkin methods have many features which make them a natural candidate for the solution of hyperbolic problems. One feature is flexibility with the order of approximation; a user with knowledge of the solution's regularity can increase the spatial order of approximation by increasing the polynomial order of the discontinuous Galerkin method. A marked increase in time-stepping difficulty, known as stiffness, often accompanies this increase in spatial order however. This thesis analyzes two techniques for reducing the impact of this stiffness on total time of simulation. The first, operator modification, directly modifies the high order method in a way that retains the same formal order of accuracy, but reduces the stiffness. The second, optimal Runge-Kutta methods, adds additional stages to Runge-Kutta methods and modifies them to customize their stability region to the problem. Three operator modification methods are analyzed analytically and numerically, the mapping technique of Kosloff/Tal-Ezer [61], the covolume filtering technique of Warburton/Hagstrom [100], and the flux filtering technique of Chalmers, et al. [19]. The covolume filtering and flux filtering techniques outperform mapping in that they negligibly impact accuracy but yield a reasonable improvement in efficiency. For optimal Runge-Kutta methods this thesis considers five top performing methods from the literature on hyperbolic problems and applies them to an unmodified method, a flux filtered method, and a

covolume filtered method. Gains of up to 80% are seen for covolume filtered solutions applied with optimal Runge-Kutta methods, showing the potential for efficient high order solutions of unsteady systems.

Acknowledgements

I would like to thank my advisor Dr. Timothy Warburton for his guidance and careful reading of my documents, through which my writing ability improved dramatically. I came to understand these topics from many different perspectives which were enormously valuable for the ultimate formulations that appear in this thesis, for this I thank Dr. Mark Embree for the many insightful discussions and seminars on eigenvalues and polynomial iterations, and also I thank Dr. Dan Sorensen for the many classes and discussions he has offered on similar topics. Furthermore I would like to thank Dr. William Symes for some of our discussions on both analysis and how often (or not so often) high order methods are applied in practice; I hope this thesis points in a direction for the resolution of some of these issues. Also my writing and presentation skills would not be where they are today without the Thesis Writing course through which Dr. Jan Hewitt, Dr. Heinkenschloss, and Dr. Timothy Warburton provided a tremendous amount of valuable feedback. Having different perspectives on discontinuous Galerkin also has proved to be very useful, and for this I thank Dr. Ḃeatrice M. Rivière for the many insightful seminars and classes.

This work was funded by the NSF VIGRE fellowship grant, and I am enormously grateful for the support.

Contents

Illustrations

Tables

MATLAB code

Nomenclature

$(\cdot,\cdot)_{D^k}$ Local L^2 inner product.

$[\mathbf{Q}]$ Jump of $\mathbf{Q}$ across element interface.

Γ^D Set of edges with Dirichlet boundary condition imposed.

Γ Set of unique edges in mesh.

$\mathbf{C}$ Symbol of a Friedrich system.

$\mathbf{G}_\lambda^k$ Geometric factors matrix. $\lambda = 0$ corresponds to unmapped case.

$\mathbf{I}$ Reference bi-unit square.

$\mathcal{T}_h$ Mesh of quadrilaterals.

$\|\cdot\|_{D^k}$ Local L^2 norm

$\|\cdot\|_{H^p(\mathcal{T}_h)}$ Broken Sobolev norm for a given mesh

Π^C L^2 projection from covolume space to primal space.

Π^P L^2 projection from primal space to covolume space.

D^k Quadrilateral element

$H^p(\mathcal{T}_h)$ Broken Sobolev space for a given mesh.

J_λ^k Jacobian of reference transformation. $\lambda = 0$ corresponds to unmapped case.

$V_{N,\lambda}^{k}$ Local solution space. $\lambda = 0$ corresponds to unmmaped case.

Chapter 1

Introduction

This thesis considers efficient time integration of hyperbolic conservation laws that have been discretized using a discontinuous Galerkin (DG) spatial discretization. Two strategies are presented: spatial operator modification and optimal Runge-Kutta time stepping methods. Within operator modification three techniques are compared, the mapping method of Kosloff and Tal-ezer [61], the filtering method of Warburton and Hagstrom [100], and the filtering technique of Chalmers et al.[19]. For optimal Runge-Kutta methods this thesis compares the optimized methods of Toulorge and Desmet [95] with those of Niegemann, Richard, and Kurt in [73].

1.1 Time integration for DG Discretizations of Hyperbolic Conservation Laws

The modern DG method was introduced by Reed and Hill in 1973 to solve the neutron transport problem in two spatial dimensions [84], and it was later analyzed by Lesaint and Raviart in [78] for general two dimensional linear hyperbolic systems. One of the novelties of this approach is the flexibility one has in handling information at element boundaries. In the linear case a simple approach takes the boundary conditions for the solution on an element to be the "inflow" values. In other words, one uses information about the direction of wave propagation that is generally known a priori for linear hyperbolic problems. Yoseph-Bar shows in [11] that this results in an

explicit numerical method in which computations can be done on an element-by-element basis by following these "characteristic" directions. This has the benefit of relative simplicity of implementation, and potential for parallelization. This thesis discusses mainly methods with these characteristics, as they have implications for parallelization on specialized computing architectures, as noted by Goedel, Klockner, and Warburton in the three papers [38],[60], [2].

Reed and Hill considered a steady state problem, and many subsequent advances adapted the use of DG for transient simulations [24]. One may categorize two broad strategies for handling the additional time variable: the first uses fully finite element space-time discretizations as by Yoseph-bar and Lowrie in [11],[70] respectively, the second applies the method of lines to the semidiscretization obtained when DG is used on the conservation law, a brief history of which may be found in Cockburn's survey paper [24].

It is difficult to render a fully explicit scheme using the space-time DG approach, as noted by Yoesph Bar in [11] and Lowrie in his PhD thesis [70]. An inherent difficulty is that unlike the case of Lesaint and Raviart, or Reed and Hill, the notion of "inflow" boundaries may not be well defined. The resolution that Lowrie takes in his thesis [70] and Richter takes in the paper [87] is to impose a strict "angle condition" on the space-time mesh. Yoseph Bar, instead of following this type of restrictive mesh condition to maintain a fully explicit scheme, opted to permit a broader range of space-time meshes which instead yields an implicit method, requiring iterative techniques to solve [11], [12]. It should be noted that although Yoseph Bar's method resulted in an implicit scheme, the discontinuous nature of the approximation still yields a block diagonal mass matrix which can be factorized on an element-by-element basis. The boundary conditions however required iteration, as inflow directions might depend both on

location in space and on the solution itself, and this iteration would inevitably involve globalized inner products, which present scaling challenges on parallel architectures, which this thesis attempts to avoid in order to remain forward compatible with for future work.

As an alternative the method of lines approach saw an early applicaion with explicit time integrators by Chavent and Salzano in [20] where they apply DG in space and forward Euler in time, but obtain a scheme which requires one to take prohibitively small timesteps in order to ensure stability [24]. Later, Cockburn and Shu in their sequence of papers [26],[25],[23], [27] combine the use of higher order Runge-Kutta methods in conjunction with DG and obtain a fully explicit scheme for nonlinear multidimensional systems of hyperbolic conservation laws. Henceforth, this thesis will only consider those schemes which are spatially discretized by DG and use an explicit method of lines time discretization.

1.2 Stiffness in High Order Methods

With a fully discrete explicit method in hand the next task is to choose a suitable timestep so that the scheme correctly resolves the physics of the problem. Given a mesh consisting of elements (intervals in one dimension, triangles or quadrilaterals in two dimensions, tetrahedra or hexahedra in three dimensions) and assuming the velocity of propagation for waves in the solution to hyperbolic conservation laws is generally known, a reasonable condition on timesteps might be one which requires that a wave does not fully cross any single element in that elapsed time, for otherwise compact stencil schemes like DG will completely miss that information as it communicates only with nearby elements.

Numerical experiments by Solomonoff in [92] however show that when high order

polynomial based methods are used the timestep must be taken significantly smaller than what the above "domain of dependence" argument suggests. These strict requirements are order-dependent, and are observed by Gottlieb et. al in [40], and rigorous asymptotic bounds given using eigenvalue techniques by Dubiner in [30], and also using approximation theoretic inequalities by Gottlieb in [41]. The basic results shows that if the order of polynomial approximation used is N, then the timestep must be taken to be $\Delta t = C/N^2$ for some $C > 0$. In the case of mesh based DG methods the constant C is mesh dependent, and in fact $C = O(h)$ where h is a mesh size parameter, further exacerbating the situation [63],[94],[46]. A detailed derivation of this timestep restriction is given in chapter 4 which provides L^2 estimates for the DG operator, and it follows the approach of Gottlieb in [41], and also that of Warburton and Hagstrom in [100], but extended to work beyond the first dimension.

An additional difficulty noted by Trefethen and Trummer in [96] shows that in addition to quadratic dependence on polynomial order, numerical roundoff effects also produce nonnegligible stability issues for methods based on polynomial approximation. An example of this phenomenon is the Legendre method produced earlier by Tal-Ezer in [93], which has a provable eigenvalue stability limit of $\Delta t = C/N$ for some $C > 0$, but in practice has a quadratic dependence on polynomial order. This may be explained by severe nonnormality of the resulting operator which causes perturbations to yield drastic changes to the spectra of the operator governing the system of ODEs [96]. To address this seeming contradiction with known stability results, Trefethen and Reddy extend the notion of absolute stability (defined later) to account for this discrepancy in [83]. This definition of stability will automatically be satisfied by the asymptotic arguments in this thesis by enforcing L^2 stability, which is a stricter stability requirement than Trefethen and Reddy's in [83].

The strict stability constraints from high order DG leads to a separation of scales: relevant physical information propagates at a certain speed, but timesteps must be chosen to resolve much faster spurious transient phenomena which are effectively artifacts of polynomial approximation. When this separation of scales exists, the system is said to be stiff, and this thesis chiefly considers techniques for overcoming this stiffness (operator modification strategies), or mitigating its effects (optimal Runge-Kutta methods).

1.3 Operator Modification

Two reasons may be given for stiffness in high order polynomial based methods. One explanation applies the domain of dependence argument seen earlier, but treats the polynomial method instead as a high order finite difference method. This approach is taken by Tal-Ezer in [93], and Kosloff and Tal-Ezer in [61]. The conclusion then comes from the fact that to guarantee stability of polynomial interpolation, one must use nodes which exhibit clustering near element boundaries, such as Chebyshev nodes or Legendre-Gauss-Lobatto nodes [14]. It turns out this minimal spacing is indeed $O(1/N^2)$ for order N approximations. An obvious approach to fixing this problem then would be to return to equally spaced nodes; unfortunately Solomonoff in [92] gives numerical evidence demonstrating that the resulting spectral methods are unstable, and more recently Platte, Trefethen et al [77] have shown that equally spaced nodes can not be stably used for any spectrally convergent linear approximation method (this includes as a subset all standard spectral methods).

A method for using equally spaced nodes however is still available, and is the subject of Kosloff and Tal-Ezer's paper [61]. The basic strategy to moves away from pure polynomial approximation, and instead composes the polynomial basis with a

function which serves to more uniformly distribute interpolation nodes. The resulting method can produce $O(1/N)$ timestep restrictions, which for high orders N does not harm the accuracy of the method (indeed, Kosloff and Tal-Ezer in [61] provide explicit formulas for maintaining predetermined precision, such as single precision). This mapping has seen extensive analysis by Solomonoff for its impact on high order spectral differentiation [29], Mead and Renaut for its implications on timestep stability [71], Abril-Raymundo and Garcia-Archilla in [6] for its approximation properties, Shen and Wang in [90] for its effect on error results, and a general explicit bound for points-per-wavelength required for resolution of the mapped method has been given by Liu and Shi in [99]. More recently this mapping approach has been generalized by Hale in his PhD thesis [43] which also gives a conformal mapping interpretation of the mapping's effect, and proceeds to construct additional mappings which provide similar effects. Only one mapping however is considered in this thesis, as it will be shown that the impact on accuracy is independent of the particular form of the mapping, and depends solely on its impact on grid spacing.

In all but one of the above citations for the mapping technique, the map itself is seen as a coordinate transformation which modifies the underlying semidiscrete equation's form to attain its effect. It should be noted that the approach for motivation and computations taken in this thesis is to take the *inverse* of this coordinate transformation, and instead produce a new basis with which to perform computations. The two approaches yield the same results, but the latter approach used here allows the use of well known orthogonal polynomial approximation results [90]. As will be seen in chapter 3, it also permits a motivation which unifies this approach with other techniques as an effective strategy for dampening large gradients of polynomials near element boundaries.

The Kosloff/Tal-Ezer mapping strategy has seen application in many areas of high order spectral methods. It has many virtues which may be exploited by its careful application: Solomonoff in [29] has used it to mitigate roundoff effects of spectral differentiation, Kosloff and Tal-Ezer in [61], Mead and Renaut in [71], and W.B. Liu et al. in [99] all have demonstrated that the mapping may improve resolution properties, and finally Kosloff and Tal-Ezer have demonstrated the timestep improvements [61]. For timestepping there have been many successful applications of this method: Jose in [54] used this method to obtain the promised $O(1/N)$ timestep restriction for the two dimensional elastic wave equation, Patrick Godon in [36] used it to similar effect with a slightly generalized form of the Kosloff/Tal-Ezer mapping strategy to model accretion discs subject to certain tidal effects, Hesthaven et al. used this method in the simulation of diffractive optical elements but instead chooses a less aggressive mapping strategy for more modest gains to ensure that spectral convergence remained unaffected, but still reported a stable doubling of timesteps [44].

The mapping strategy has also been used for spatial resolution properties, a strategy actually introduced before Kosloff and Tal-Ezer's paper by Bayliss et. al in [13] to accumulate points in spectral methods to locations in the solution which exhibit physically large gradients. An example of this application, specifically with that of the Kosloff/Tal-Ezer mapping is by Celik and Cangellaris in [18] where they simulate transmission lines, and adaptively apply the Kosloff/Tal-Ezer map when the line was a certain factor larger than the smallest wavelength (in this particular paper they chose the factor to be 4).

An alternative viewpoint on stiffness is given by Gottlieb in [41], where approximation theoretic inequalities are used to derive the timestep's quadratic dependence on order for pseudospectral methods. The basic inequality used is a special case of

the Markov inequality (see e.g. Lorentz [69] for the classical proof) which says that the maximum absolute value of the derivative of an $N-th$ degree polynomial grows like $O(N^2)$. On the face this does not appear to give a usable strategy for improving stiffness, but an observation by Warburton and Hagstrom in [100] suggests that another approximation theoretic inequality is at play: Bernstein's inequality, which shows that the $O(N^2)$ derivative scaling is at worst localized in small neighborhoods of the boundaries of the element, but elsewhere behaves more as $O(N)$. Combining both Markov and Bernstein's inequality one deduces that the gradients of polynomials are in general the largest near element boundaries, and in fact near element interiors behave more like $O(N)$. Thus the quadratic behavior is the result of large gradient behavior in small neighborhoods of element boundaries. This led Warburton and Hagstrom to staggered grid strategies which effectively use two approximations to the same solution, and combine them in such a way that information closest to element centers are used, thus enforcing a $O(N)$ gradient. Analytical and computational results show this method retains the approximation power of polynomials, but also can reduce the timestep restriction to $O(1/N)$ [100]. It is worth noting that this staggered grid strategy has appeared in various forms elsewhere without the approximation theoretic motivation, for example Hagstrom et. al in [39], and developments in stable polynomial approximation follow the trend of using staggered grids as well as seen in Boyd [53],[15] where the Runge pheneomenon is avoided by using staggered gridpoints. More recently Hagstrom et al. has extended the staggered grid Hermite methods given in [39] to an order adaptive method with similar timestep implications [22].

A newer technique which does not depend directly on grid spacing or derivative norm growth has been introduced by Chalmers et al. [19] whereby the timestep

restriction is improved by targeting the constants implied in the polynomial trace inequalities, which effectively bound boundary integrals in terms of volume integrals. It is shown in the paper [19] that by appropriately modifying the lifting operator in penalty methods so as to yield one of smaller norm can effectively increase the stable timestep, and it is furthermore proven that doing this in a certain way can yield a method with the same theoretical order of convergence in one dimension, a result that is numerically verified in this thesis for two dimensions whenever the resulting method is stable.

In chapter 3 this thesis analyzes numerically both the mapping strategy, and the staggered grid strategy under the approximation theoretic approach of Warburton and Hagstrom [100], and compares it by numerical experiment to the mapping strategies of Kosloff and Tal-ezer [61], and Hale [43]. These two methods along with the technique of Chalmers et. al [19] will be compared numerically. A contribution of this thesis is the extension of the numerical results of Chalmers et al. in [19], of Warburton and Hagstrom in [100] to a two dimensional problem, and the application of Kosloff and Tal-Ezer's mapping technique in [61] to discontinuous Galerkin. Furthermore in chapter 3 the filtering of Chalmers and the filtering of Warburton and Hagstrom will be combined with customized time-stepping routines to even further improve the benefits. Customized timesteppers are the topic of the rest of this introduction.

1.4 Optimal Runge-Kutta Methods

With the spatial discretization suitably modified to improve the timestep restrictions, there is an additional layer of modification to be made. The underlying explicit timestepper may be customized to the problem at hand so as to stably admit the largest possible timestep. This thesis focuses on Runge-Kutta methods because they

permit low storage implementations which are suitable to low memory environments such as graphics processors, a unified approach to low storage techniques given by Ketcheson in [56], but in principle the ideas here could carry forward to other methods such as the exponential timestepping methods analyzed by Saad in [88] or general linear methods analyzed by Butcher in [16] (also known as multistep Runge-Kutta methods, e.g. Renaut in [85]).

The early development of Runge-Kutta methods focused mainly on improving the efficiency of solving ordinary differential equations by increasing the order of Runge-Kutta method and using the minimal possible number of inner stage calculations to achieve this order [50], thus permitting larger timesteps to be used for greater accuracy. This approach however does not effectively apply to stiff systems of differential equations where timestep must be taken far below accuracy limits in order to satisfy stability constraints. To quantify this stability limit we associate to each Runge-Kutta method a "stability polynomial" (the seminal work of Butcher [16] investigates these in depth) and consider the region of the complex plane for which the absolute value of this stability polynomial has modulus not exceeding one. A timestep then is said to be stable for a particular linear system of differential equations if the spectrum of the linear operator scaled by that timestep is contained within the above mentioned region, which will be denoted the "stability region."

An early attempt at optimizing this stability region was made by Lawson in [64], and [65] where fifth and sixth order Runge-Kutta methods were constructed such that instead of using the minimal number of stages required to achieve fifth and sixth order, more stages were added than is strictly necessary in order to make the stability region larger. Using symbolic techniques a gain of 30% in efficiency in terms of right hand side evaluations was observed [64]. Following this was Lomax in 1968 [68] who

actually suggested a least squares approach for the adaptive selection of Runge-Kutta methods. This adaptivity idea has not resurfaced in recent literature for convection dominated problems except in a few cases, e.g. in [97] or [51].

Later development of customized Runge-Kutta methods can be largely categorized in two ways: techniques for targeting the stability region to a simple reference spectrum shape (such as a line, circle, ellipse), and analytic results concerning the stability polynomials themselves. Initially many of the techniques used were symbolic in nature and sought two key reference spectra: those for discretizations of parabolic and hyperbolic problems. For the former case it is desirable to include as much of the negative real axis as possible in the stability region, whereas for the latter case it is important to contain a portion of the imaginary axis owing to the typically skew-symmetric hyperbolic operators noted by Van Der Houwen in [98]. Both cases are important for DG approximations to hyperbolic problems, as numerical dissipation is often introduced for stability reasons, yielding eigenvalues with large negative real part (see e.g. Hesthaven and Warburton in [46]).

For optimal imaginary axis stability Runge-Kutta methods Van Der Houwen in [98] found symbolically those Runge-Kutta methods of first and second order which include the largest portion of the imaginary axis as possible, these methods contained the number of stages as an adjustable parameter (but only for odd number of stages) so that one can find the optimal number of inner stage evaluations for their problem. This work was continued by Kinnmark, Ingemar, and Gray in their sequence of papers [59],[58],[57] where they extend the allowable orders to third and fourth using ad hoc symbolic strategies. Later a general principle for construction of these methods would be given by Kinnmark and Ingemar [75].

Optimal negative real axis Runge-Kutta methods saw more activity, as many

results would turn to their favor. One of these results is the equal ripple property, which characterizes stability polynomials with optimal negative real axis inclusion in their stability region [76]. This result may be used directly to construct such polynomials, and in the first order case one is lead to shifted and scaled Chebyshev polynomials, which as noted by Van Der Houwen [76] is actually a different, but equivalent, formulation for Chebyshev acceleration of Richardson iteration [86] and the resulting method became known as the Runge-Kutta-Chebyshev method. One property of this method is that (like Chebyshev acceleration) it can be factorized into a series of forward Euler steps with varying timesteps. This idea would later be revived as supertimestepping by Alexiades in [8] and [9], where it would then be applied in a more adaptive way in [35],[32], [67], and closely related DUMKA schemes of and Medovikov [72], all of which require varying degrees of knowledge about the spectrum of the problems considered. It should be noted that although super-time-stepping was developed for parabolic problems, it has been succesfully applied to hyperbolic problems which have been discretized with a method that induces spurious eigenvalues of large negative real part [91]. Where super-time-stepping took advantage of the ability to factorize a high stage high real axis stability method as a sequence of forward Euler steps, another development by Abdulle et. al in [4],[3] would instead exploit the stable three term recurrence of Chebyshev polynomials and use shifted and scaled Chebyshev polynomials for the evaluation of even higher order Runge-Kutta methods with real axis stability, effectively solving the "inner stage stability" problem mentioned by Van Der Houwen in [76]. This development might see future activity, as Ketcheson in [55] has noted benefits in changing polynomial basis so that the basis is almost orthogonal on the spectrum of interest.

In the time period mentioned some important theoretical work has also been done

in addition to the explicit calculation of Runge-Kutta methods. The basic task was to characterize stability polynomials which were "optimal" in some predefined sense. The first theorems in this direction were bounds on the possible real axis and imaginary axis stability that may be had. For imaginary axis stability Vichnevetsky in [81] showed that for an S stage Runge-Kutta method, its stability region can not include an interval of length greater than $S-1$ of the imaginary axis. This bound is sharp in the sense that some optimal schemes attain it [76]. A result for inclusion of circles was shown in the same paper that said in effect that a disk tangent to the imaginary axis could only be included in a Runge-Kutta stability region with S stages if its radius was S, confirming a result saying essentially the same thing in [52]. For negative real axis stability the explicit construction of the Runge-Kutta-Chebyshev method shows by properties of Chebyshev polynomials that for an S stage method, its stability region can include at most an interval of length $2S^2$ of the negative real axis. For problems which include both wave propagation and dissipative characteristics however there is no known analytic bound.

More recently Runge-Kutta methods have started to be tailored to specific types of discretizations, for example in aeroacoustics Ramboer et al. constructed optimal six stage Runge-Kutta methods [82] and Allampali et al. went up to 7 stages in [10], this thesis will consider those optimal schemes of Niegemann et al. in [73] and Toulorge and Desmet in [95], which are both very well suited to DG approximations of hyperbolic operators (the last paper dealing specifically with that case). I present these timestepping methods with both modified and unmodified DG operators to see what manner of gains may be had, and I discuss the standard construction of these Runge-Kutta methods along with a new technique given by Ketcheson and Ahmadia in [55].

Chapter 2

Discontinuous Galerkin for Friedrichs' System

In this chapter I present the discontinuous Galerkin (DG) method for solving hyperbolic conservation laws, specifically the acoustic wave equation. This formulation will form the foundation for the remainder of the thesis, which will present modification strategies to the underlying weak formulation to alter the spectral properties of the method.

2.1 Introduction

This thesis will consider the solution of systems of equations of the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{AQ}}{\partial x} + \frac{\partial \mathbf{BQ}}{\partial y} = \mathbf{R} \tag{2.1.1}$$

on an open domain Ω with suitable initial conditions and boundary conditions. Here the matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{r \times r}$ are symmetric and independent of space. Such a system is known as a Friedrich system [33].

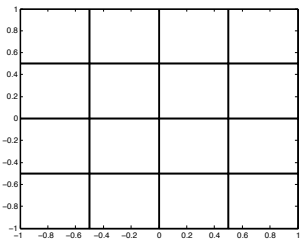


Figure 2.1.1 : Quadrilateral mesh

Initially one constructs a mesh $\mathcal{T}_h = \left\{ D^k \mid D^k \text{ is a quadrilateral}, k = 1, \ldots, K \right\}$ (see figure (2.1.1)), The idea behind DG is to approximate $\mathbf{Q}$ with a piecewise polynomial $\mathbf{Q}_N^h$ from a space P_h^N of piecewise polynomials by forming a suitable projection of the residual equation onto this space. A pure Galerkin approach would require that $\mathbf{Q}_N^h$ satisfy

$$\int_\Omega \mathbf{v} \left(\frac{\partial \mathbf{Q}_N^h}{\partial t} + \frac{\partial \mathbf{A}\mathbf{Q}_N^h}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}_N^h}{\partial y} \right) = \int_\Omega \mathbf{v}\mathbf{R}$$

for all $v \in P_h^N$. However, as the space P_h^N does not automatically require $\mathbf{u}_N$ to satisfy any sort of auxiliary conditions (boundary conditions, continuity), it is necessary to impose them in the projection step through a term which penalizes discontinuities which arise from the solution space P_h^N. For the special case of a Friedrich system where $\mathbf{A}, \mathbf{B}$ are continuous this penalization may be achieved by requiring that for all elements $D^k \in \mathcal{T}_h$ we calculate the local solution $\mathbf{Q}_N$ by requiring

$$\int_{D^k} \mathbf{v} \left(\frac{\partial \mathbf{Q}_N}{\partial t} + \frac{\partial \mathbf{A}\mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}_N}{\partial y} \right) - \int_{D^k} \mathbf{v}\mathbf{R} = \frac{1}{2} \int_{\partial D^k} \mathbf{v} \left(\mathbf{C} - \mathbf{C}^* \mathbf{C} \right) [\mathbf{Q}_N] \quad (2.1.2)$$

for all $\mathbf{v}$ polynomial on D^k. Here $\mathbf{C}$ is defined by

$$\mathbf{C} = n_x \mathbf{A} + n_y \mathbf{C}$$

where n_x, n_y are the components of the outward pointing normal of the element D^k and

$$[\mathbf{Q}] = \mathbf{Q}^- - \mathbf{Q}^+$$

is the difference between the solution on the element D^k and that of its neighboring element. It should be noted that the above discussion assumes the solution will indeed be a piecewise polynomial, but for quadrilateral meshes the basis functions local to

each element will be small perturbations of a polynomial basis. This detail will be handled carefully in chapter 4.

This thesis focuses on solving this system for $\mathbf{Q}_N$ in time using explicit timestepping methods. To see the difficulties which arise in doing this, note that for each $D^k \in \mathcal{T}_h$ one may view the global solution $\mathbf{Q}_N^h$ as a coupled system of ordinary differential equations

$$\frac{d\mathbf{Q}_N^h}{dt} = \mathbf{D}\mathbf{Q}_N^h + \mathbf{R},$$

which may be evolved in time by any explicit timestepper which has an appropriately shaped stability region. A well known fact about explicit timestepping methods is that they are for stability reasons limited in how large the timestep Δt may be. In fact, for any fixed explicit timestepping method the timestep must be inversely proportional to the spectral radius of $\mathbf{D}$, i.e. stability forces $\Delta t \approx \rho(\mathbf{D})^{-1}$. The difficulty then which arises from DG is that $\rho(\mathbf{D}) = O(N^2/h)$ where N is the polynomial order used, and h is the mesh spacing parameter. This can be shown directly from the weak form (2.1.2), and is derived in detail in chapter 4. The overall goal of this thesis is to investigate methods for reducing the impact of this spectral radius on the right-hand-side evaluations required to integrate the system to a desired final time.

2.2 Mesh and Solution Space

One difficulty in using quadrilateral meshes is that the mapping between arbitrary quadrilaterals to a fixed reference element is mildly nonlinear. That distinguishes this type of mesh from triangular or tetrahedral where the mapping is affine. This means that the local solution space is not uniformly a polynomial space on each element, only on those whose reference mappings reduce to affine (e.g. parallelograms). This will

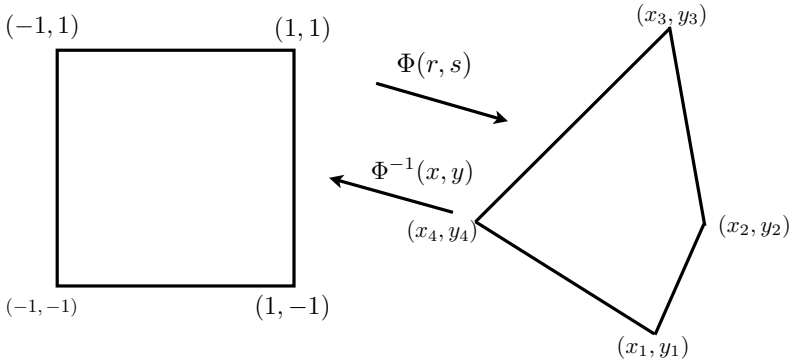


Figure 2.2.1 : Example quadrilateral

require a little more careful notation. The local solution spaces to each element $D^k \in \mathcal{T}_h$ will be constructed by taking the standard tensor product Legendre polynomial basis on the reference bi-unit square and through a coordinate transformation induced by the reference mapping create a new basis on the desired element.

The mesh in use is a quadrilateral mesh and will be denoted $\mathcal{T}_h$, with h parameterizing the maximum element diameter. Given an arbitrary quadrilateral specified by four vertices $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ labeled in anticlockwise manner one may map the reference biunit square onto this quadrilateral through the change of coordinates $x = \Phi_1^k(r, s), y = \Phi_2^k(r, s)$ with

$$\begin{aligned} \Phi_1^k(r,s) &= \frac{(1+r)(1+s)x_1 + (1-r)(1+s)x_2}{4} \\ &\quad + \frac{(1-r)(1-s)x_3 + (1+r)(1-s)x_4}{4} \\ \Phi_2^k(r,s) &= \frac{(1+r)(1+s)y_1 + (1-r)(1+s)y_2}{4} \\ &\quad + \frac{(1-r)(1-s)y_3 + (1+r)(1-s)y_4}{4} \end{aligned} \tag{2.2.1}$$

The transformation Φ^k from reference r, s coordinates into x, y coordinates on $D^k \in \mathcal{T}_h$ will have associated with it a transformation Jacobian matrix $\mathbf{J}^k$ and inverse

$\left(\mathbf{J}^k\right)^{-1} = \mathbf{G}^k$ given by the entries

$$\mathbf{G}^k = \begin{pmatrix} \frac{\partial r}{\partial x} & \frac{\partial s}{\partial x} \\ \frac{\partial r}{\partial y} & \frac{\partial s}{\partial y} \end{pmatrix} \tag{2.2.2}$$

and an associated Jacobian determinant (hereafter referred to as "Jacobian")

$$J^k = \frac{1}{\det \mathbf{G}^k} \tag{2.2.3}$$

These geometric factors are important in the calculation of integrals and derivatives on elements D^k, and effectively reduce calculation in an arbitrary quadrilateral to calculation on the reference bi-unit square scaled by appropriate metric quantities. One difficulty that will arise however is that (2.2.1) is nonlinear, and so the Jacobian which appears in integration after pulling back to reference coordinates will be seen later to damage orthogonality of the basis.

On the reference element, which will be denoted $\mathbf{I}$ throughout this thesis, an orthogonal polynomial basis is constructed through a tensor product procedure on one dimensional polynomials. Namely on the interval $[-1, 1]$ one has the Legendre orthogonal polynomials (L_i^*) which are defined through the recursion [5]

$$\begin{aligned} L_1^*(x) &= 1 \\ L_2^*(x) &= x \\ L_{i+1}^*(x) &= \frac{1}{n+1}\left((2n+1)\,xL_n^*(x) - nL_{n-1}^*(x)\right) \end{aligned}$$

and futhermore these polynomials will be normalized to have L^2 norm of one. The extension to two dimensions is straightforward by defining a new double-indexed basis

(L_{ij}) such that

$$L_\alpha(x,y) = L_{ij}(x,y) = L_i(x)\, L_j(y)$$

which is still an orthonormal basis. A linear indexing scheme for α may be imposed arbitrarily, the one used in this thesis is the one supposed by MATLAB's "kron" command, as all tensor products are computed with it to ensure consistency of indexing. The extension of this basis to a general quadrilateral D^k is obtained through the coordinate transformation Φ :

$$\begin{aligned} L_\alpha(x,y) &= L_\alpha\left(\left(\Phi_1^k\right)^{-1}(x), \left(\Phi_2^k\right)^{-1}(y)\right) \\ &= L_\alpha(r,s). \end{aligned}$$

Since the transformation Jacobian (2.2.3) is nonlinear, this new basis is nonpolynomial and furthermore not necessarily orthogonal. This may be seen through the equation

$$\int_{D^k} L_\alpha(x,y)\, L_\beta(x,y)\, dS = \int_{\mathbf{I}} L_\alpha(r,s)\, L_\beta(r,s)\, J_k(r,s).$$

Later this restriction will not be important, as some of the modification techniques introduced in chapter 3 will impose a structure restriction on the mesh, which will effectively turn Φ^k into an affine function. The theory developed however will in all cases seek to work on the general quadrilateral case, wherever applicable. Since the reference mapping Φ^k will be different for each k, the solution space becomes more complicated than simply finding piecewise polynomials. To account for this I introduce the following local solution spaces,

$$V_N^k = \operatorname{span}\left\{L_\alpha \circ \left(\Phi^k\right)^{-1} \mid \alpha = 1, \ldots, (N+1)^2\right\}. \tag{2.2.4}$$

In following DG formulations, the solution sought will locally belong to to one of these spaces, or to modifications of them.

2.3 A Representative Problem

The purpose of this section is to present the standard DG method in two dimensions for the acoustic wave equation in first order form. As explained in section (2.1) this will be achieved by first discretizing the domain in question with quadrilaterals D^k and then searching for a solution whose restriction to D^k is in the local solution space for each D^k (but not requiring any sort of inter-element continuity). The auxiliary conditions of inter-element continuity and of boundary conditions are then imposed weakly by penalizing jumps, the way this is accomplished will be reminiscent of finite volume methods and their use of the numerical flux [46, 66].

2.3.1 The Equations and Weak Formulation

The equations to solve are

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \nabla p = 0, \quad \frac{1}{\kappa}\frac{\partial p}{\partial t} + \nabla \cdot \mathbf{u} = R \tag{2.3.1}$$

with suitable initial conditions and boundary conditions. Here $\mathbf{u}$ is wave velocity, p is acoustic pressure, ρ is density, and κ is bulk modulus. The speed of sound given the material parameters κ, ρ is $c = \sqrt{\kappa/\rho}$. To see that this takes the form as (2.1.1), one may rewrite (2.3.1) as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{AQ}}{\partial x} + \frac{\partial \mathbf{BQ}}{\partial y} = \mathbf{R} \tag{2.3.2}$$

with

$$\mathbf{Q} = \begin{pmatrix} u_1 \\ u_2 \\ p \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 \\ \kappa & 0 & 0 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\rho} \\ 0 & \kappa & 0 \end{pmatrix}$$

Suppose now that the domain for the sought solution is $\Omega_h = \bigcup_{k=1}^{K} D^k$, with D^k being a quadrilateral for each k. Before following the discussion in the introduction, some important notation must be introduced. As the solution sought exists in a space of functions which does not guarantee continuity, it is important to refer to information inside a given element D^k and information outside this given element, as the two may be different (in contrast to traditional finite element methods). The *interior trace* value of a function f at a point $x \in \partial D^k$ is given by

$$f^{-}(x) = \lim_{\substack{y \to x \\ x \in D^k}} f(y), \tag{2.3.3}$$

and the *exterior trace* value at this same point is

$$f^+(x) = \lim_{\substack{y \to x \\ y \in adjacent\left(D^k\right)}} f(y), \tag{2.3.4}$$

here $adjacent\left(D^k\right)$ means the element D^j that shares the edge with D^k which contains x. Note that (2.3.4) is not technically well defined, but it will always be clear from context what values are being used. It will be useful as well to refer to the notion of jumps across element interfaces

$$[f] = f^- - f^+$$

Given the form (2.3.2) it is now possible to follow the idea in the introduction and write a weak form. To simplify notation the following inner products will be used throughout this thesis

$$(\mathbf{u},\mathbf{v})_{D^k} = \int_{D^k} \mathbf{u}\cdot\mathbf{v}, \qquad (\mathbf{u},\mathbf{v})_{\partial D^k} = \int_{\partial D^k} \mathbf{u}\cdot\mathbf{v},$$

along with their associated norms

$$\|\mathbf{u}\|_{D^k} = (\mathbf{u},\mathbf{u})_{D^k}, \qquad \|\mathbf{u}\|_{\partial D^k} = (\mathbf{u},\mathbf{u})_{\partial D^k}.$$

The weak form (2.1.2) gives rise to the weak form for the acoustic wave equation: for each k find $\mathbf{Q}_N \in \left(V_N^k\right)^3$ (dependence on k omitted) such that

$$\int_{D^k} \mathbf{v}\left(\frac{\partial \mathbf{Q}_N}{\partial t} + \frac{\partial \mathbf{A}\mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}_N}{\partial y}\right) - \int_{D^k} \mathbf{v}\mathbf{R} = \frac{1}{2}\int_{\partial D^k} \mathbf{v}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)[\mathbf{Q}_N] \tag{2.3.5}$$

holds for all $\mathbf{v} \in \left(V_N^k\right)^3$. The expression in the boundary integral may be interpreted as an upwinding flux, and in fact in the case of continuous coefficients evaluates exactly to what the upwinding flux should be [47]:

$$(\mathbf{C} - \mathbf{C}^*\mathbf{C}) \left[\mathbf{Q}^N\right] = \begin{pmatrix} n_x \frac{1}{\rho} [p_N] - \kappa^2 n_x (\mathbf{n} \cdot [\mathbf{u}_N]) \\ n_y \frac{1}{\rho} [p_N] - \kappa^2 n_y (\mathbf{n} \cdot [\mathbf{u}_N]) \\ \kappa \mathbf{n} \cdot [\mathbf{u}_N] - \mathbf{n} \cdot \left(\mathbf{n} \frac{1}{\rho^2}\right) [p_N] \end{pmatrix},$$

implementation details for this local semidiscrete form are postponed until chapter 7, which will also contain implementation details for other components of this thesis.

2.3.2 Boundary Conditions

Dirichlet boundary conditions will be imposed by requiring the exterior trace $\mathbf{Q}_N^+$ on boundary edges to satisfy the equation

$$\mathbf{Q}_N^+ = \mathbf{D}\mathbf{Q}_N^- \tag{2.3.6}$$

where the operator $\mathbf{D}$ is constrained to ensure that

$$\mathbf{Q}_N^- \cdot (\mathbf{C}^*\mathbf{C} - \mathbf{C})\, \mathbf{D}\mathbf{Q}_N^- \leq 0. \tag{2.3.7}$$

Condition (2.3.7) is revisted in chapter 4 where it will be seen to be necessary and sufficient for stability on an edge where a boundary condition of type (2.3.6) is imposed. Aside from periodic conditions a popular boundary condition used in acoustic wave simulations are reflecting conditions. For DG this is traditionally stated $p_N^+ = -p_N^-$,

the corresponding matrix $\mathbf{D}$ is

$$\mathbf{D} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

and one may verify by inspection that it satisfies (2.3.7).

Chapter 3

Stiffness Reduction through Operator Modification

In this chapter I deal with methods for handling the stringent requirements on timesteps which is inherent in higher order methods. I focus the strategy on attempting to reduce the operator L^2 norms of the projected derivative operator and the trace lifting operator, as these can be seen as key components to the contribution of stiffness in DG (see chapter 4). The methods presented in this chapter focus on directly altering the underlying semidiscrete formulation (12) in such a way as to yield a smaller spectral radius without damaging theoretical results of classical DG.

I give numerical results to investigate the usefulness of these methods. It will be seen that the mapping techniques generate significant error when they are applied aggressively enough to yield a nonnegligible impact on timestep size. This is in contrast to the covolume filtering method, which degrades solution quality as well if applied too aggressively, but it will be seen that there exists a cutoff which if used leaves the accuracy virtually unchanged. The timesteps are improved considerably under the covolume filtering strategy, and it appears to yield stable timesteps that decrease only linearly with polynomial order as opposed to quadratically. The third technique, which for lack of a name in the literature I will call "flux filtering" does improve timestep size without heavily impacting accuracy, however its effect is somewhat unpredictable with respect to how heavily it is applied. A light application of flux filteirng in some cases yields timesteps that are twice as large without significantly impacting accuracy, but unlike the one dimensional case [19] can yield an

unstable method after overapplication in two dimensions.

The contributions of this chapter are the extensions of covolume filtering and flux filtering to two dimensions, and the application of mapping techniques to discontinuous Galerkin. Furthermore the methods are all compared to one another in terms of their impact on accuracy.

3.1 Introduction

Key elements which yield the L^2 norm of $O\left(N^2/h\right)$ are two inequalities: the inverse trace inequality and the Markov inequality. The first two methods investigated in this chapter, the mapping method and the covolume filtering method are motivated by looking at the Markov inequality. As it turns out, even though the Markov inequality is sharp, the N^2 magnitude of gradients of polynomials is only seen near element boundaries. This can be seen through the Bernstein inequality, which says that given an order N polynomial P on the interval $[-1,1]$ then the following *pointwise* bound holds [69]

$$\left|P'\left(x\right)\right| \leq \frac{N}{\sqrt{1-x^2}}\|P\|_\infty$$

for each $x \in (-1,1)$. Thus near element midpoints the gradients grow only linearly with polynomial order, instead of quadratically.

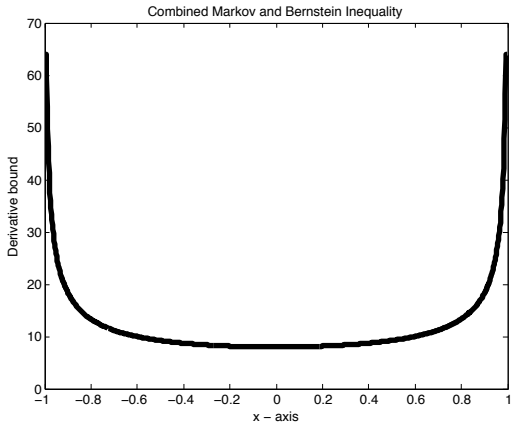


Figure 3.1.1 : Combined Markov and Bernstein inequality on the interval $[-1, 1]$

For mapped methods [61] a coordinate transformation is suggested with serves to 'stretch' out the interval $[-1, 1]$ near its endpoints, but to leave the interior near the midpoint relatively unchanged. A function which is expressed in this new coordinate system would have lower gradients near its boundaries than in the original system, and so hopefully lessening its impact on timestep considerations. The impact of this stretching can be dramatic, as seen below (compare to figure (3.1.1) above)

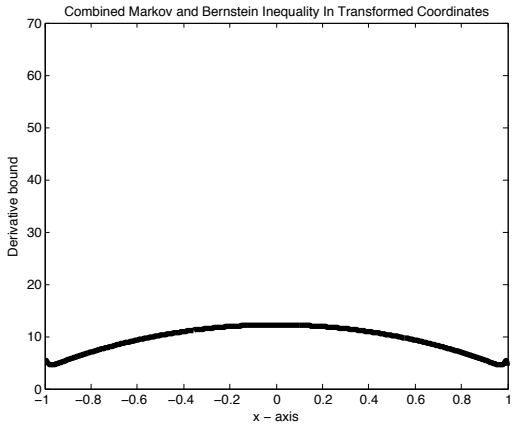


Figure 3.1.2 : Markov/Bernstein inequality under aggressive coordinate transformation

The covolume filtering strategy [100] works differently in that it involves projecting

a function between two grids, the *primal* grid which is the grid on which the PDE in question was originally posed in weak form, and the *covolume* grid which is staggered over the primal grid. In this way only information from element interiors are used, and spuriously large gradients near element boundaries are filtered out of the solution.

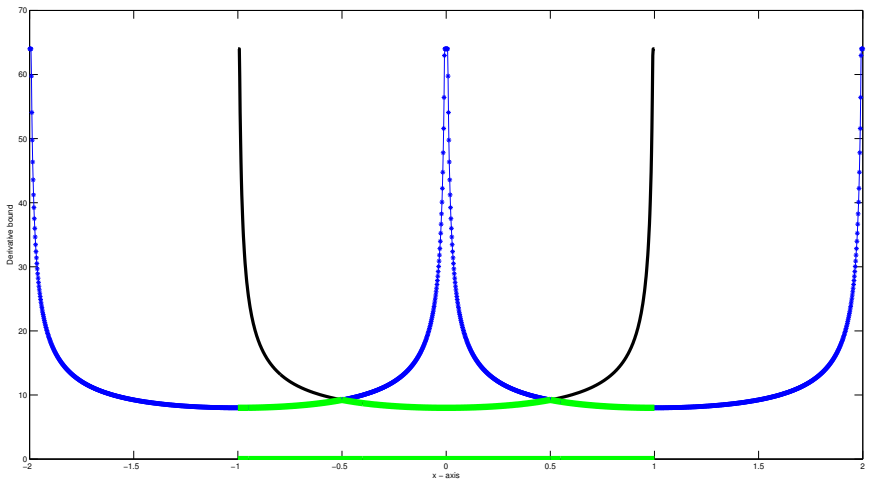


Figure 3.1.3 : Combined Markov/Bernstein with staggered grid.

The flux filtering strategy [19] does not directly impact the derivative operator as the mapped or covolume filtering strategies do. Instead it takes the lifted flux term in the DG formulation, and filters the top modes before adding it to the interior gradients. It was shown in the paper presenting this strategy that if only the top mode is filtered then the formal order of convergence remains unchanged in the case of periodic one dimensional advection. This method essentially reduces the spectral radius of the DG operator by constraining the solution to satsify a more restrictive inverse trace inequality. By not addressing the issue of spuriously large gradients however, it can not achieve the same order of magnitude reduction as the mapped methods or covolume filtering methods.

The numerical examples in this chapter will all focus on the two dimensional acoustic wave equation on a periodic spatial domain

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \nabla p &= 0 \\ \frac{\partial p}{\partial t} + \nabla \cdot \mathbf{u} &= 0 \end{cases} \quad \text{for } (x,y,t) \in (-1,1) \times (-1,1) \times (0,T)$$

with initial condition

$$\begin{aligned} p(0,x,y) &= \sum_{i=1}^{m} \sum_{j=1}^{n} p_{mn} \sin(j\pi x) \sin(i\pi y) \\ \mathbf{u}(0,x,y) &= \mathbf{0} \end{aligned}$$

and analytic solution

$$\begin{aligned} p(t,x,y) &= \sum_{i=1}^{m} \sum_{j=1}^{n} p_{mn} \sin(j\pi x) \sin(i\pi y) \cos\left(\pi t \sqrt{i^2+j^2}\right) \\ \mathbf{u} &= \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{p_{mn} \sin\left(t\pi \sqrt{i^2+j^2}\right)}{\sqrt{i^2+j^2}} \begin{pmatrix} j \cos(n\pi x) \sin(m\pi y) \\ i \sin(j\pi x) \cos(i\pi y) \end{pmatrix}. \end{aligned}$$

In this case I take $p_{ij} = 1$ for $1 \leq i,j \leq 6$. The following sequence of meshes will be used for h-convergence studies:

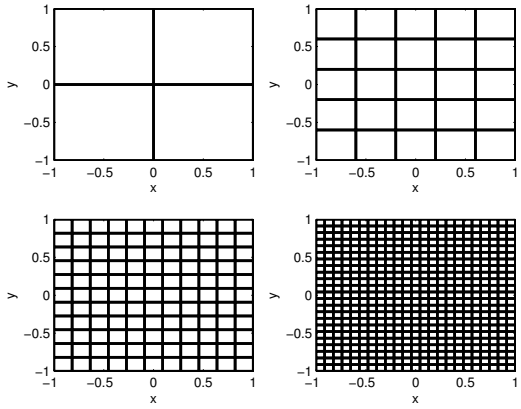


Figure 3.1.4 : Sequence of meshes used for h-convergence studies

3.2 Mapped Methods

The idea behind mapping is to find a coordinate transformation which will lessen the gradients of polynomials near boundaries while retaining the favorable gradients near midpoints. Such coordinate transformations have been investigated in a related problem of attempting to construct stable approximation techniques from equispaced datapoints [43, 77], the one investigated in this thesis will be that of Kosloff and Tal-Ezer [61], i.e. the one dimensional coordinate transformation

$$x = g_\lambda(y) = \frac{\sin^{-1}\lambda y}{\sin^{-1}\lambda}$$

which has inverse

$$y = h_\lambda(x) = \frac{\sin\left(x\sin^{-1}\lambda\right)}{\lambda}.$$

Two dimensional versions are readily constructed through tensor products in the case of the square. To see the impact this has on the derivative of a function f expressed

in the new coordinate system, apply the chain rule to find

$$\begin{aligned} f'(y) &= f'\left(\frac{\sin\left(x\sin^{-1}\lambda\right)}{\lambda}\right) \\ &= \frac{\sin^{-1}\lambda\cos\left(x\sin^{-1}\lambda\right)}{\lambda} f'(x) \end{aligned}$$

which shows significant reduction in the gradient of f near boundaries for $\lambda \to 1$

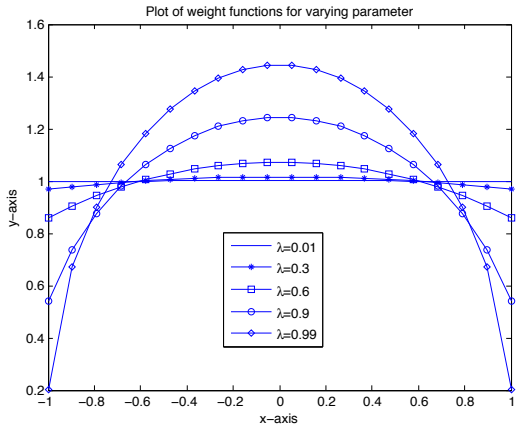


Figure 3.2.1 : $\frac{\sin^{-1}\lambda\cos\left(x\sin^{-1}\lambda\right)}{\lambda}$ for $\lambda \to 1$.

The behavior of this function near the midpoint of $[-1, 1]$ thus appears essentially bounded above and unchanged for changes in λ, but near boundaries it drops off dramatically, which is exactly the desired effect. The effects of this transformation will now be investigated.

3.2.1 Weak Formulation

To facilitate its use in DG I will no longer consider this mapping as a coordinate transformation, but rather as a non-polynomial basis which is used as an alternative to the polynomial basis. If using the Lagrange basis L_i on the reference square $\mathbf{I}$ the

new modified basis becomes through this transformation:

$$L_i^{\lambda}(x) = L_i(h_{\lambda}(x))$$

and the reference solution space becomes

$$P_{N,\lambda}(\mathbf{I}) = \left\{ v \circ h_{\lambda} \mid v \in P^N(\mathbf{I}) \right\}$$

with local solution spaces $V_{N,\lambda}^k$ constructed by coordinate transformations as in the polynomial case. Recalling the local weak form (2.3.5)

$$\int_{D^k} \mathbf{v} \left(\frac{\partial \mathbf{Q}_N}{\partial t} + \frac{\partial \mathbf{A}\mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}_N}{\partial y} \right) - \int_{D^k} \mathbf{v}\mathbf{R} = \frac{1}{2} \int_{\partial D^k} \mathbf{v} \left(\mathbf{C} - \mathbf{C}^* \mathbf{C} \right) [\mathbf{Q}_N]$$

which holds for every $k = 1, \ldots, K$ and $\mathbf{v} \in \left(V_N^k\right)^3$, one readily obtains the modified weak form by simply replacing V_N^k in the unmodified case with $V_{N,\lambda}^k$.

3.2.2 Numerical Results

Here I provide numerical results to determine whether the mapping technique is effective at reducing the stiffness of the DG operator without hurting accuracy. It will be seen that this is generally not the case for the polynomial orders considered in this thesis, as there will be order reduction for aggressive mapping.

3.2.2.1 Effect on Spectrum

Now computations are undertaken to suggest numerically the timestep improvement for appropriately chosen λ. It will be shown that choosing λ as a function of N, increasing so that $\lambda \to 1$ can yield a sequence of DG operators with spectral radius

growing only linearly with N for $N = 1, \ldots, 8$. As in the first chapter this requires bounds on the derivative operator and the trace lifting operator. First I investigate the derivative operator behavior. This I give numerically, as the operator is block diagonal in DG simulations, and so only the L^2 norms of a single block need to be analyzed for varying λ. The L^2 norms of this operator may be calculated explicitly through the generalized eigenvalue problem [74]

$$\mathcal{D}_{r,\lambda}^T \mathcal{M}_\lambda^{2D} \mathcal{D}_{r,\lambda}^T \mathbf{u} = \xi \mathcal{M}_\lambda^{2D} \mathbf{u}$$

and for the lifting operator $\mathcal{L}_\lambda$ a similar eigenvalue problem is obtained

$$\left(\mathcal{L}_\lambda \mathcal{Y}\right)^T \mathcal{M}_\lambda^{2D} \left(\mathcal{L}_\lambda \mathcal{Y}\right) \mathbf{u} = \xi \mathcal{M}_\lambda^{2D} \mathbf{u}$$

where $\mathcal{Y}$ is the linear operator defined as

$$\mathcal{Y}\mathbf{u} = \mathbf{u} \mid_{\partial D} .$$

Here the matrices $\mathcal{D}_{r,\lambda}, \mathcal{M}_\lambda^{2D}, \mathcal{L}_\lambda$ are the projected derivative operator, the two dimensional mass matrix, and the trace lifting operator for the reference solution space $P_{N,\lambda}\left(\mathbf{I}\right)$. Given that these matrices are very small compared to the whole problem this can be done with relative ease using MATLAB's "eig" command. Below are results for varying λ and polynomial order N. The results for varying N and λ are given by the below two figures

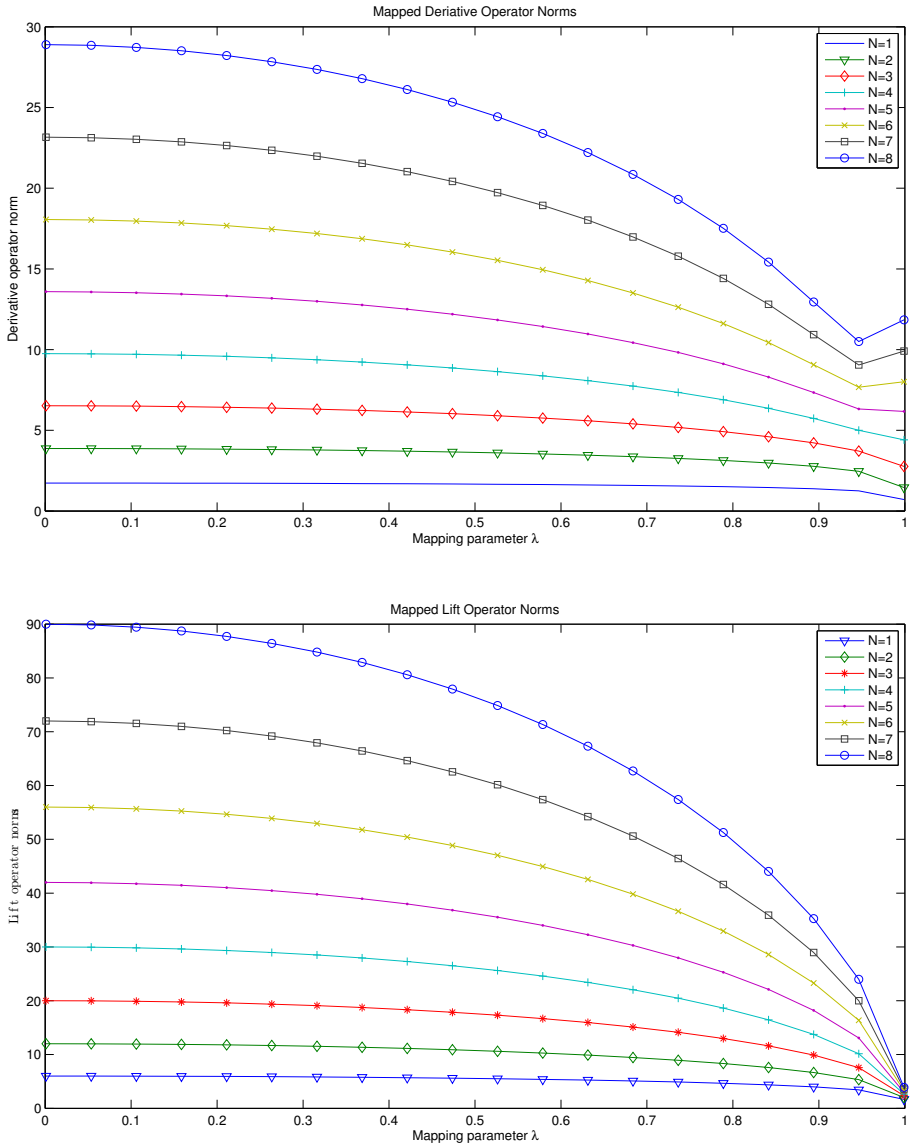


Figure 3.2.2 : L^2 norms of the operators $\mathcal{D}_{r,\lambda}$,$\mathcal{L}_\lambda$ for various N and λ

note the scale difference in the x-axis on both figures. It is also instructive to observe the imapct on the full spectrum of the DG operator. For example the following figure shows how mapping compresses the spectrum in the case of the acoustic wave equation operator posed on a periodic domain:

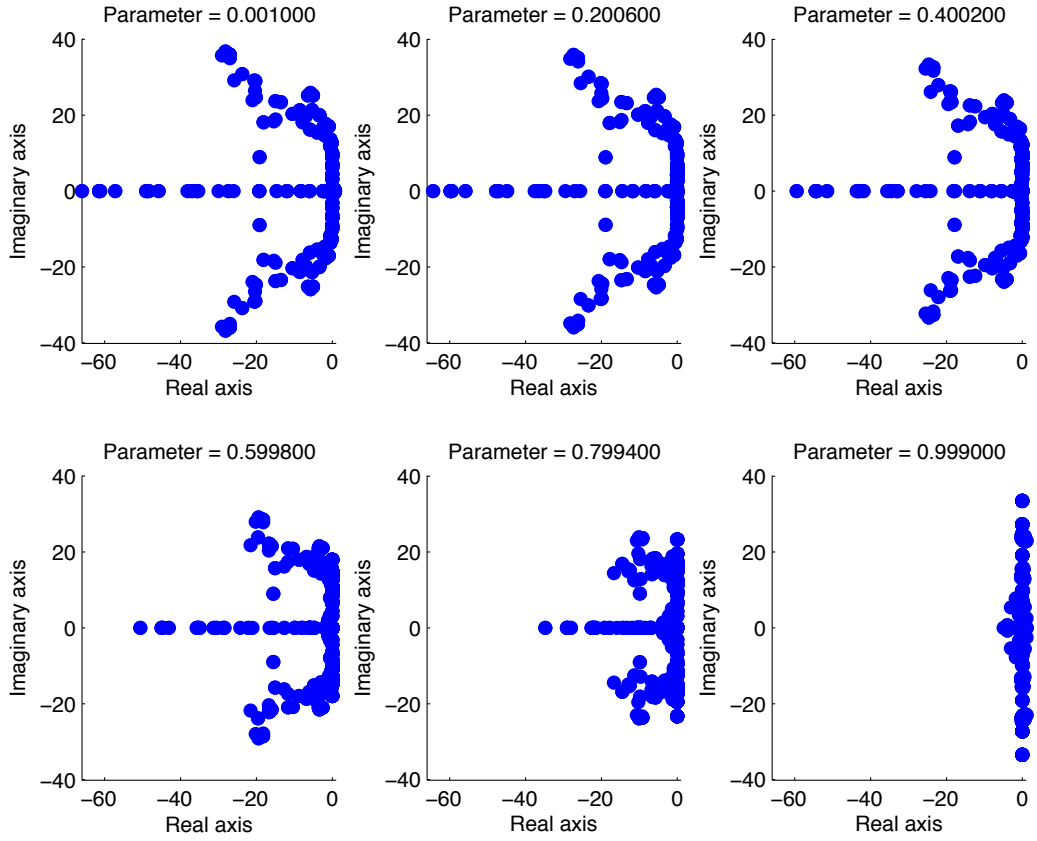


Figure 3.2.3 : Compression of spectrum through application of Kosloff/Tal-Ezer mapping

One can see above the dramatic effect of the mapping for parameter values of λ in a small neighborhood of one.

3.2.2.2 h-convergence

Next it is important to see how the solution quality improves as one refines the mesh. What is seen here however is that a heavily mapped method yields significant order reduction for larger values of λ, at least for the polynomial orders considered here. The order values here are computed by using the mesh sequence (3.1.4). Because of the amount of information it is necessary to convey I omit presenting the exact error

for each mesh but instead just report the computed error decay p assuming it takes the form $\|u - u_N^h\| \leq Ch^p$.

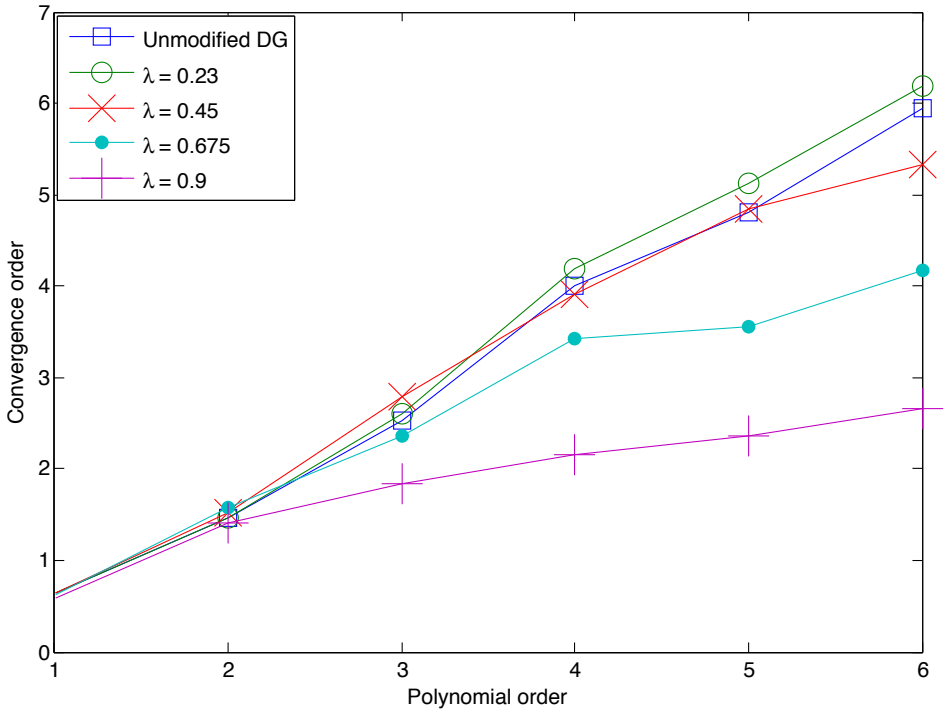


Figure 3.2.4 : h-convergence for different mapping parameters λ

One sees in the above figure a significant order reduction even for modest values of λ, and so I did not take values approaching one, but the order reduction becomes even steeper in this limit.

3.2.3 Explanation of Results

In the numerical results of this section it was demonstrated that mapping does not perform well in the low order regime in which DG is often applied. This might lead one to consider alternative mappings as Hale did in his thesis [43] (but on notably much higher order methods) or a basis that does not suffer the same Markov inequality

as polynomials as Hesthaven et al. did in [21]. It should be first noted that it is impossible to find a basis (ϕ_i) such that $\operatorname{span}\{\phi_i \mid i = 1, \dots\}$ is dense in $L^2(-1,1)$ which satisfies the inequality

$$\|\phi_N'\|_{L^\infty(-1,1)} \leq CN^{1-\epsilon} \|\phi_N\|_{L^\infty(-1,1)}$$

since one could always construct a linear combination $f = \sum_{i=1}^{N} \alpha_j \phi_j$ which is orthogonal to the characteristic functions of the intervals $\left[\frac{m}{N}, \frac{m+1}{N}\right]$ for $m = -N, \dots, N-1$, which would give

$$\begin{aligned} \|f\|_{L^\infty(-1,1)} &\leq \max_m \sup_{\left[\frac{m}{N}, \frac{m+1}{N}\right]} f - \inf_{\left[\frac{m}{N}, \frac{m+1}{N}\right]} f \\ &\leq \frac{1}{N} \|f'\|_{L^\infty(-1,1)} \\ &\leq C\frac{N^{1-\epsilon}}{N} \|f\|_{L^\infty(-1,1)} \end{aligned}$$

and applying the same argument to $(1/C) f$ yields a contradiction (construction adapted from a Math Overflow comment [7]). Furthermore if $\operatorname{span}\{\phi_i \mid i = 1, \dots\}$ is not just dense in $L^2(-1,1)$, but also satisfies similar error estimates as polynomials, and a Markov inequality of the type

$$\|\phi_N'\|_{L^2(-1,1)} \leq \lambda_N \|\phi_N\|_{L^2(-1,1)}$$

then by defining $\mathcal{U}_N$ to be the L^2 projection operator from $L^2(-1,1)$ to span $\{\phi_i \mid i = 1, \ldots, N\}$ and taking any polynomial $p_N \in P^N(-1,1)$ we have

$$\begin{aligned}
\|p_N'\|_{L^2(-1,1)} &= \left\|p_N' - (\mathcal{U}_N p_N)' + (\mathcal{U}_N p_N)'\right\|_{L^2(-1,1)} \\
&\leq \left\|p_N' - (\mathcal{U}_N p_N)'\right\|_{L^2(-1,1)} + \left\|(\mathcal{U}_N p_N)'\right\|_{L^2(-1,1)} \\
&\leq \left\|p_N' - (\mathcal{U}_N p_N)'\right\|_{L^2(-1,1)} + \lambda_N \left\|\mathcal{U}_N p_N\right\|_{L^2(-1,1)} \\
&\leq \left\|p_N' - (\mathcal{U}_N p_N)'\right\|_{L^2(-1,1)} + \lambda_N \left\|p_N\right\|_{L^2(-1,1)}
\end{aligned}$$

so that if the term left term above is bounded in N we see that λ grows at least quadratically in N by the traditional polynomial Markov inequality. Further information is provided in chapter 4 relating to blowup of constants in error estimates for mapped methods.

3.3 Covolume Filtering

The idea of covolume filtering is to introduce a new 'covolume' mesh which staggers the original mesh. Then two L^2 projection operators Π^P, Π^C are formed. The Π^C operator takes a piecewise polynomial function $u \in P^N\left(D^k\right)$ and projects it so that $\Pi^C u$ is a piecewise polynomial on the covolume grid $\mathcal{T}_h^C$. The Π^P operator does the same thing but instead projects it onto the space of piecewise polynomials on the 'primal' or original grid $\mathcal{T}_h^P$. The covolume filter operator then is

$$\Pi^P \Pi^C.$$

The filtering used for this thesis is

$$\Pi_\beta = (1-\beta)\, I + \beta \Pi^P \Pi^C$$

where I is the identity operator. The semidiscrete form 2.3.5 becomes the task to find $\mathbf{Q}_N \in V_N^k$ such that

$$\int_{D^k} \mathbf{v} \left(\frac{\partial \mathbf{Q}_N}{\partial t} + \frac{\partial \mathbf{A} \Pi_\beta \mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B} \Pi_\beta \mathbf{Q}_N}{\partial y} \right) - \int_{D^k} \mathbf{v}\mathbf{R} = \frac{1}{2} \int_{\partial D^k} \mathbf{v} \left(\mathbf{C} - \mathbf{C}^* \mathbf{C} \right) \left[\Pi_\beta \mathbf{Q}_N \right]$$

holds for all $\mathbf{v} \in \left(V_N^k \right)^3$.

3.3.1 Numerical Results

First an idea of how covolume filtering impacts the spectrum of the periodic acoustic wave opeartor, this shows that one should expect an improved timestep through such a filtering.

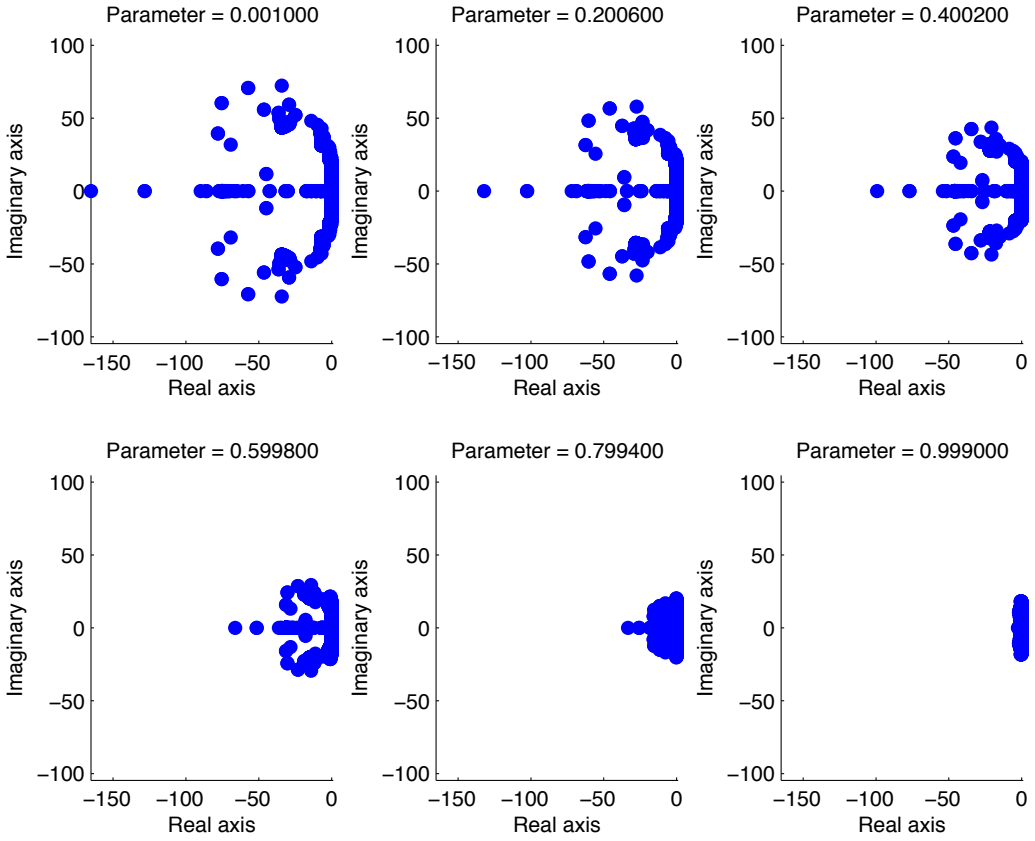


Figure 3.3.1 : Impact of covolume filtering on DG spectrum

I provide here h-convergence results for covolume filtering. Note the mild departure from standard DG

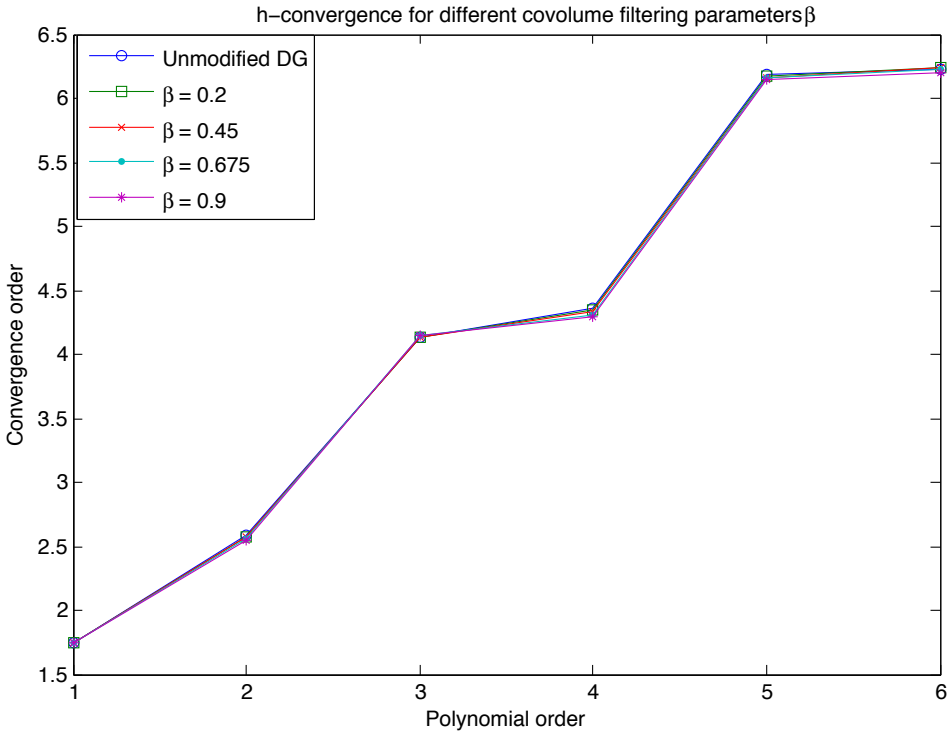


Figure 3.3.2 : h-convergence for varying covolume filter parameter β

3.4 Flux Filtering

In the flux filtering technique introduced by Chalmers, et. al [19], based on ideas from [62], the flux lifting term in the DG discretization is expressed in terms of a Legendre basis and then the highest order modes of this expansion are directly reduced. The operator used to achieve this is similar to traditional filtering used to improve stability when aliasing errors introduce spurious oscillations, the only difference is that the filter is applied only to the lifted flux term. Normally this filter would impact order of accuracy, but in [19] it is shown that the order of convergence remains unchanged in the one dimensional case if the filter is only used on the lifted flux terms, and if only the topmost modes are modified. The impact is to reduce the contribution of the lift operator to the spectral radius of the overall scheme, reducing its CFL number.

This section is shorter than others as the theory is undeveloped for more complicated problems, and its impact is less than that of mapping or covolume filtering. It is nevertheless useful to consider, as its impact on accuracy will be seen to be minimal, and among the other modification techniques considered here it involves the least change to existing code.

The idea behind flux filtering in one dimension is to take the elementwise semidiscrete form on an element D^k

$$\frac{d}{dt}u = \mathcal{D}u + \mathcal{L}\left(du\right)$$

with du as before referring to the field differences at the boundaries of D^k, accounting for upwinding, and then replace $\mathcal{L}$ with a filtered version $\mathcal{L}^\delta$ with $\delta \in [0,1]$. Given a polynomial u on the boundary of D^k one may express it as a sum of appropriately transformed Legendre polynomials P_i so that

$$\mathcal{L}\left(du\right) = \sum_{i=1}^{N_p} \alpha_i L_i$$

and so a filtered version of $\mathcal{L}\left(du\right)$ would be to take instead

$$\mathcal{L}^\delta\left(du\right) = \sum_{i=1}^{N_p-1} \alpha_i L_i + \delta\alpha_{N_p} L_{N_p}. \tag{3.4.1}$$

The resulting local weak form is

$$\frac{d}{dt}u = \mathcal{D}u + \mathcal{L}^\delta\left(du\right).$$

Furthermore for the case of one dimensional advection it is shown in [19] that this

modification also does not impact the theoretical h-convergence rates, but does potentially increase the constant implied in the error estimate. One contribution of this thesis will be to extend this idea to two dimensions. No rigorous error estimates or fully discrete stability will be proven in the accompanying theory chapter, instead I will only provide the numerical results to validate these facts

3.4.1 Flux Filtering in Two Dimensions

In two dimensions simply filtering the highest degree mode as in (3.4.1) is not sufficient for any significant gain, instead if one expresses a function $u \in P^N$ as a tensor product expansion of Legendre polynomials as

$$u = \sum_{1 \leq i,j \leq N} u_{ij} L_i L_j$$

with α being a multi-index, then the filtering in one dimension can be readily extended to two dimensions as such:

$$\mathcal{F}^\delta u = \sum_{\substack{1 \leq i,j \leq N \\ i+j \neq N}} u_{ij} L_i L_j + \delta \sum_{i+j=N} u_{ij} L_i L_j$$

with the resulting filtered lift operator $\mathcal{L}^\delta$ becoming

$$\mathcal{L}^\delta = \mathcal{F}^\delta \mathcal{L}.$$

3.4.2 Numerical Results

Here I provide similar reasoning for the improved timestep bounds as was used in the mapping section as well as h-convergence studies. The difference is that here the only operator affected is the lifting operator $\mathcal{L}$. This operator yields inverse trace inequalities which result in timestep bounds that result in asymptotic stability results, like that derived in chapter 4. Below I compute the operator norm $\left\|\mathcal{L}^{\delta}\right\|_{L^2}$ for various δ. It will be seen that the operator norm decreases monotonically for decreasing δ, corresponding to more aggressive filters.

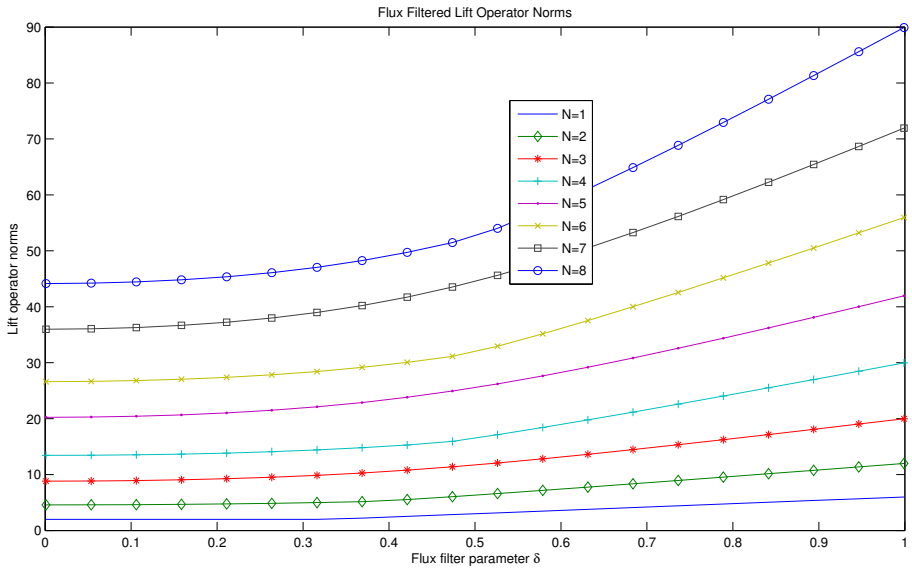


Figure 3.4.1 : L^2 norms $\left\|\mathcal{L}^{\delta}\right\|_{L^2}$ for various δ

one furthermore sees that the most aggressive effect yields at most a halving of $\left\|\mathcal{L}^{\delta}\right\|_{L^2}$, but this does not necessarily correspond to a doubling of stable timesteps as the Markov inequality will eventually dominate the estimates for L^2 stability. Furthermore, there is no theoretical guarantee that this even results in a stable method, as

seen in the below figure

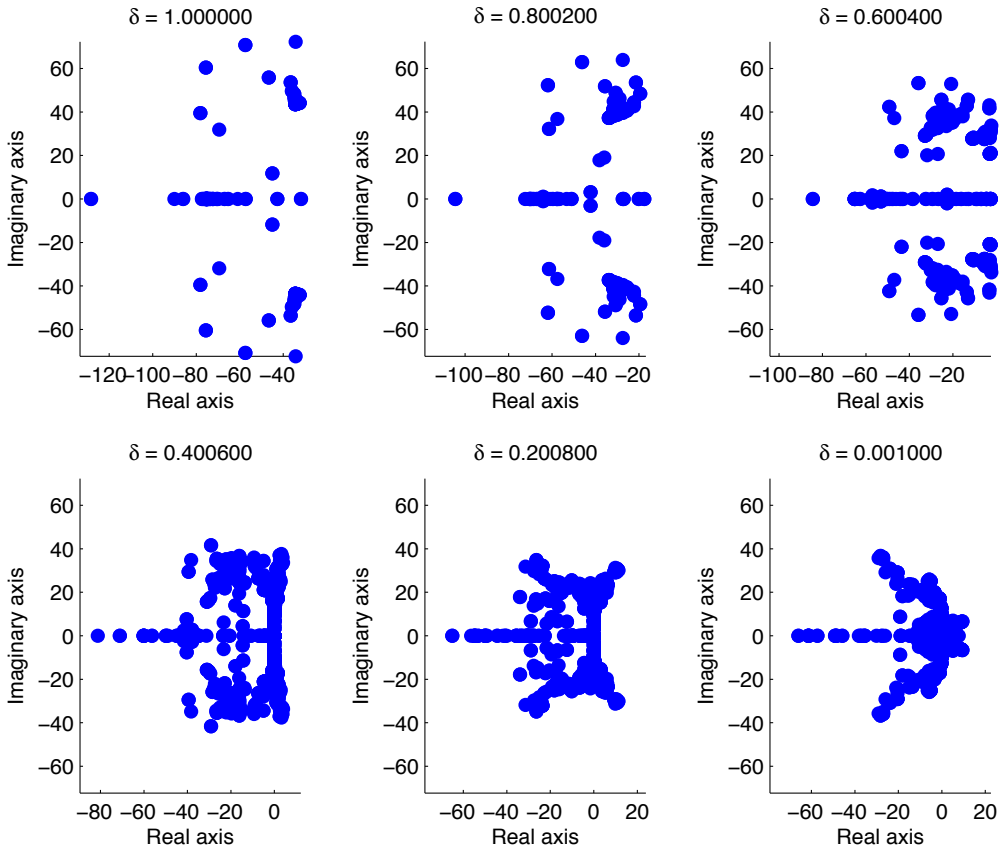


Figure 3.4.2 : Effect on spectrum of flux filtering

note that the last two are unstable.

Now I provide h-convergence results for flux filtering. Its departure from unmodified DG is seen to be essentially negligible

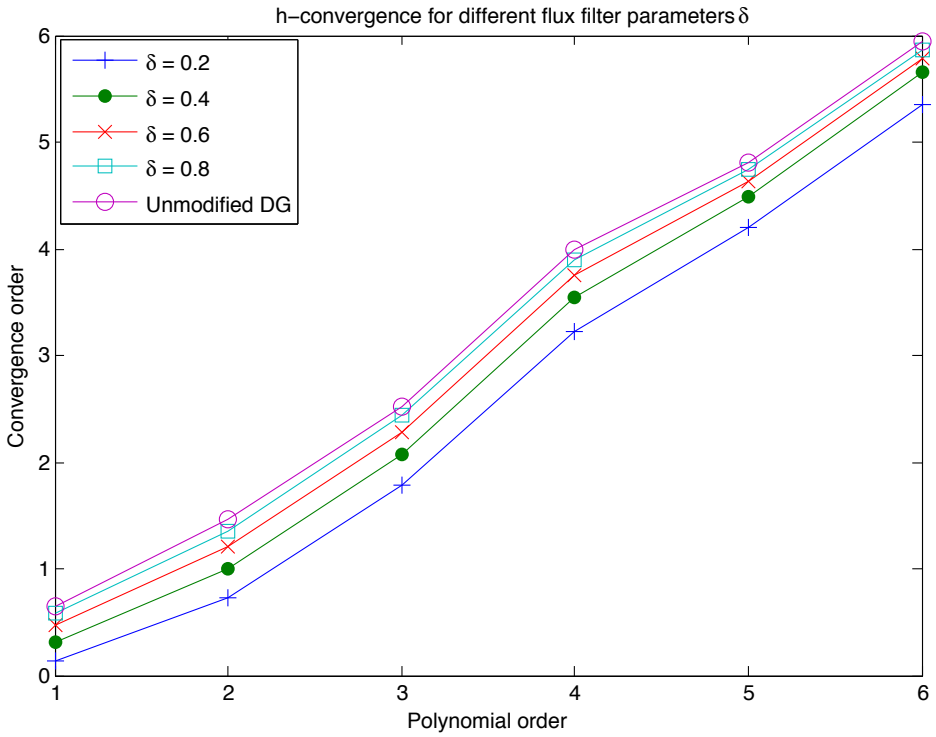


Figure 3.4.3 : h-convergence for different flux filtering parameters δ

3.5 Comparisons and Conclusions

Now I compare each of the three methods presented in this chapter so as to see which yields the most efficient method by measuring the number of right-hand-side evaluations as a function of accuracy. It will be seen that the covolume filtering method provides the most efficient for a modest impact to accuracy for the problems considered here, while the mapping technique negatively impacts accuracy to the point of yielding significant order reduction. In between these two methods is the flux filtering technique, the impact of which is less but accordingly has a lesser impact on accuracy as well. It has the added benefit of a simple implementation. The figure below was determined by fixing the tolerance on ODE45 to single precision, calculating the amount of RHS evaluations required to fully solve the system, and then calculating the spatial error of the result. Calculations were done for a fixed grid and fixed polynomial order of $N = 6$.

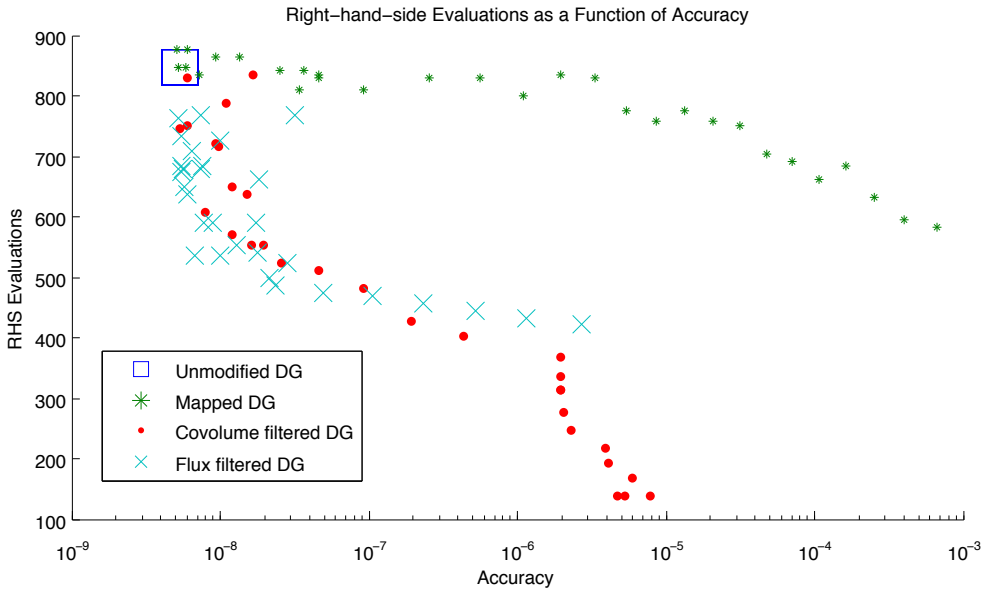


Figure 3.5.1 : RHS evaluations compared to accuracy achieved

The mapping technique should be expected to fail when used very aggressively, taking a less aggressive approach however does not yield helpful gains in timestep size, since the more dramatic effects of the mapping occur in a relatively small neighborhood of the parameter value $\lambda \to 1$.

The covolume filtering method avoids the difficulties of mapping while still addressing the same issue. It furthermore allows the full use of polynomial approximation results, which both in theory and in experiments have proved to yield solutions of similar quality to unmodified DG but with significantly improved timestep limitations. The experiments done in this thesis however focused on linear problems with smooth solutions, and as pointed out in the paper [100] one should not expect similar results in nonsmooth cases (e.g. discontinuous coefficients, or development of shocks) without a significant change to the technique. Furthermore the extension of this technique to an unstructured grid is not straightforward, and would require further study.

The flux filtering strategy directly lessens the impact of the discrete lifting operator to the spectral radius of the DG operator, but as it does not address the issue involved with calculating derivatives on element interiors its effects are effectively limited to decreasing the constant implied in the notation $\rho(\mathbf{D}) = O(N^2/h)$. It nevertheless yields timestep restrictions which are in some cases 30% larger than unmodified DG while only marginally reducing accuracy, and with its relative ease of implementation it could potentially save time in codes which are having speed issues in time integration. Its effects however are somewhat difficult to predict, with a very strong filter it leads to unstable schemes.

Chapter 4

Theory

This chapter states and proves standard theoretical results in the new context of mapped domains. The theory behind covolume filtering is also developed, but as its use currently restricts one to tensor product grids it is more or less a direct extension of the results from [100] with slightly more restrictive regularity requirements between primal and covolume grids. As of yet no theory exists for flux filtering in two dimensions, seeing its reasonable impact on efficiency however this could be a useful direction for further research. The theory in this chapter is not necessarily new, as much of it is motivated by the preprint [101]. It is however the first time discontinuous Galerkin theory has been applied to mapping techniques, showing their theoretical convergence. Furthermore the theory for covolume filtering in two dimensions is new, as the previous analysis in [100] applied only in the one dimensional case.

4.1 Notation and Preliminary Results

Before stating and proving key theorems for DG formulations I will introduce the basic notation and machinery that will make their statements possible, and state the key theorems and lemmas which are important in error analysis.

This thesis makes use of standard theoretical items from the theory weak solutions. Given a quadrilateral D^k the function spaces $L^2\left(D^k\right)$ and $H^s\left(D^k\right)$ all carry their standard meanings, associated norms, seminorms, and inner products. Since however

the DG method further weakens a solution by requiring it only to only be *piecewise* continuous, the following extended space is also used: Given a mesh $\mathcal{T}_h$ the set $H^p(\mathcal{T}_h)$ is defined by

$$H^p(\mathcal{T}_h) = \left\{ v \mid v_{|_D} \in H^p(D) \ \ \forall D \in \mathcal{T}_h \right\}$$

and is called the *broken Sobolev space* of order p for the mesh $\mathcal{T}_h$ and it has along with it an associated broken Sobolev norm

$$\|\mathbf{u}\|^2_{H^p(\mathcal{T}_h)} = \sum_{k=1}^{K} \|\mathbf{u}\|^2_{H^p\left(D^k\right)}$$

and furthermore $L^2(\mathcal{T}_h)$ will be understood as taking $p = 0$ in the above definitions. To each element $D^k \in \mathcal{T}_h$ there is the associated element measure h_k

$$h_k = diam\left(D^k\right)$$

and furthermore the global results will be stated in terms of

$$h = \max_k h_k.$$

The set of unique edges of elements $D^k \in \mathcal{T}_h$ will be denoted as Γ. Recall that local solution spaces are denoted V_N^k and $V_{N,\lambda}^k$ respectively for the standard and mapped DG respectively, and for each element we have a local Jacobian $\mathbf{J}_\lambda^k$ which depends on the local element geometry.

Now I catalogue inequalities and projection estimates which are useful in the derivation of error estimates.

Theorem 1. *[Cauchy-Schwarz Inequality] Suppose that V is an inner product space*

with inner product $(\cdot,\cdot)$*. Then for all* $u, v \in V$ *the following inequality holds*

$$|(u,v)| \leq \|u\| \cdot \|v\|.$$

Theorem 2. *[Young's Inequality] Suppose that* $a, b, \epsilon \in \mathbb{R}$*. Then*

$$2ab \leq \frac{a^2}{\epsilon} + \epsilon b^2$$

holds for all $\epsilon > 0$*.*

Theorem 3. *[Castillo,Cockburn, et al.] Suppose that* R, A, B, χ *are nonnegative functions from* $[0,T]$ *to* $\mathbb{R}$*, that* B, χ *are measurable, and that*

$$\chi^2(t) + R(t) \leq A(t) + 2\int_0^t B(s)\,\chi(s)\,ds$$

Then given any $t \in [0,T]$ *we have*

$$\sqrt{\chi^2(t) + R(t)} \leq \sup_{0\leq\theta\leq t} \sqrt{A(\theta)} + \int_0^t B(\theta)\,d\theta.$$

Proof. See [17, pg. 465, lemma 3.11] □

Theorem 4. *[Multiplicative Trace Inequality] Suppose that* D^k *is a quadrilateral with* h_k *its associated element measure. Then there exists a constant* C *such that for all* $v \in H^1\left(D^k\right)$ *we have*

$$\|v\|^2_{\partial D^k} \leq C\left(\|v\|_{D^k}\,|v|_{H^1\left(D^k\right)} + \frac{1}{h}\|v\|^2_{D^k}\right)$$

The following theorems will be proven, as they are not already common in the

literature. They deal chiefly with approximation theoretic estimates such as inverse inequalities and truncation error estimates. The difference between theorems here and other well known theorems is the unknown constants do not depend on local element geometry. The dependence of the estimates on local geometry is explicitly quantified, which permits one to impose restrictions on a quadrilateral mesh which are sufficient to recover optimal convergence. The techniques of proof will mirror to some extent the techniques in Hughes, et al. in [1]. To motivate some of the below definitions recall that the local solution spaces $V_{N,\lambda}^{K}$ are nonpolynomial, but can in some sense be considered polynomial on a special reference element after reversing the coordinate transformation used to obtain $V_{N,\lambda}^{k}$. More specifically for each function $\phi, \psi \in V_{N,\lambda}^{k}$ we may calculate the inner products as

$$\int_{D^k} \phi(x,y)\,\psi(x,y)\,dxdy \;=\; \int_{\mathbf{I}} \phi(r,s)\,\psi(r,s)\,J_{\lambda}^{k}(r,s)\,drds \tag{4.1.1}$$

here J_{λ}^{k} is the Jacobian obtained after combining the coordinate transformations from the Kosloff/Tal-Ezer mapping and the reference element mapping. Note that $\phi(r,s), \psi(r,s)$ are polynomial on $\mathbf{I}$. It will also be important to evaluate the following inner product

$$\begin{aligned} & \int_{D^k} \frac{\partial}{\partial x}\phi(x,y)\,\frac{\partial}{\partial x}\psi(x,y)\,dxdy \\ = \;& \int_{D^k} \mathbf{G}_1^{k,\lambda}\nabla\phi(x,y)\,\mathbf{G}_1^{k,\lambda}\nabla\psi(x,y)\,dxdy \qquad (4.1.2) \\ = \;& \int_{\mathbf{I}} \mathbf{G}_1^{k,\lambda}\nabla\phi(r,s)\,\mathbf{G}_1^{k,\lambda}\nabla\psi(r,s)\,J_{\lambda}^{k}drds \qquad (4.1.3) \end{aligned}$$

where $\mathbf{G}_1^{k,\lambda}$ is the first row of the geometric factors matrix $\mathbf{G}^{k,\lambda}$ obtained again after combining the two mappings. This result follows similarly for $\partial/\partial y$.

To each $D^k \in \mathcal{T}_h$ I denote by $\mathcal{P}^k_{N,\lambda}$ the weighted L^2 projection operator defined for every $\mathbf{f} \in L^2\left(D^k\right)^3$ by requiring

$$(\mathbf{f}, \mathbf{v})_{D^k} = \left(\mathcal{P}^k_N \mathbf{f}, \mathbf{v}\right)_{D^k}$$

to hold for every $\mathbf{v} \in V^k_{N,\lambda}$. As a slight abuse of notation I consider $\mathcal{P}^k_{N,0} = \mathcal{P}^k_N$ to be the projection onto the unmapped local space V^k_N. The theorems here follow a general theme: express the L^2 projection operator for an arbitrary local solution space $V^{k,\lambda}_N$ as a weighted L^2 projection operator onto a space of polynomials. The result of this projection is well known to minimize the associated weighted residual over polynomials, and the use of Bramble-Hilbert will provide estimates on the unweighted L^2 norm which will be seen to be equivalent to the weighted norm. Note the form of Bramble-Hilbert in use will be that of Dupont and Scott in [31], which is easier to use when the approximation operator is known to satisfy a variational property.

I begin first with two generalizations of standard estimates, a generalized Markov inequality and a generalized inverse trace inequality. Techniques here mirror those taken by Warburton in the unpublished report[101].

Theorem 5 (Local Markov Inequality)**.** *There exists a constant C_λ independent of h_k, N such that for every $u \in V^k_{N,\lambda}$ the following estimate holds*

$$\left\| \frac{\partial}{\partial x} u \right\|_{D^k} \leq C_\lambda \frac{N^2}{h_k} \|u\|_{D^k}$$

Proof. Suppose $u \in V_{N,\lambda}^k$, then

$$\begin{aligned}
\left\| \frac{\partial}{\partial x} u \right\|_{D^k} &= \left\| \mathbf{G}_1^{k,\lambda} \nabla u \right\|_{D^k} \\
&\leq \left\| \mathbf{G}_1^{k,\lambda} \right\|_{D^k} \left\| \nabla u \right\|_{D^k} \\
&\leq \left\| \mathbf{G}_1^{k,\lambda} \right\|_{D^k} \left\| \sqrt{J_\lambda^k} \right\|_{L^\infty(\mathbf{I})} \left\| \nabla u \right\|_{\mathbf{I}} \\
&\leq CN^2 \left\| \mathbf{G}_1^{k,\lambda} \right\|_{D^k} \left\| \sqrt{J_\lambda^k} \right\|_{L^\infty\left(D^k\right)} \left\| u \right\|_{\mathbf{I}} \\
&\leq CN^2 \left\| \mathbf{G}_1^{k,\lambda} \right\|_{D^k} \left\| \frac{1}{\sqrt{J_\lambda^k}} \right\|_{L^\infty\left(D^k\right)} \left\| \sqrt{J_\lambda^k} \right\|_{L^\infty\left(D^k\right)} \left\| u \right\|_{D^k}
\end{aligned}$$

where I have applied the Markov inequality for polynomials (see e.g. [28]) to obtain the N^2 scaling. Now I apply the following assumptions on the metric quantities:

$$\begin{aligned}
\left\| \mathbf{G}_1^{k,\lambda} \right\|_{D^k} &= O\left(\frac{1}{h_k} \right) \\
\left\| \frac{1}{\sqrt{J_\lambda^k}} \right\|_{L^\infty\left(D^k\right)} \left\| \sqrt{J_\lambda^k} \right\|_{L^\infty\left(D^k\right)} &= O\left(1\right).
\end{aligned}$$

Variants of these assumptions will appear elsewhere also, they are not violated by any of the code used in this thesis. □

The next theorem estimates boundary norms in terms of volume norms.

Theorem 6 (Local Inverse Trace Inequality)**.** *There exists a constant L_λ independent of N, h_k such that for every $u \in V_{N,\lambda}^k$ the following bound holds*

$$\left\| u \right\|_{\partial D^k} \leq L_\lambda \frac{N}{\sqrt{h_k}} \left\| u \right\|_{D^k}$$

Proof. Suppose that $u \in V_{N,\lambda}^k$, then

$$\begin{aligned}
\|u\|_{\partial D^k} &\leq \left\| \sqrt{J_\lambda^{k,e}} \right\|_{L^\infty(\partial \mathbf{I})} \|u\|_{\partial \mathbf{I}} \\
&\leq L_\lambda N^2 \left\| \sqrt{J_\lambda^{k,e}} \right\|_{L^\infty(\partial \mathbf{I})} \|u\|_{\mathbf{I}} \\
&\leq L_\lambda N \left\| \sqrt{J_\lambda^{k,e}} \right\|_{L^\infty\left(\partial D^k\right)} \left\| \frac{1}{\sqrt{J_\lambda^k}} \right\|_{L^\infty\left(D^k\right)} \|u\|_{D^k}
\end{aligned}$$

where $J_\lambda^{k,e}$ is the jacobian of the reference mapping restricted to edges of D^k that appears in the pullback of the first inequality. The scaling of N appears from the standard polynomial inverse trace inequality on $\mathbf{I}$. Now apply the assumption

$$\left\| \sqrt{J_\lambda^{k,e}} \right\|_{L^\infty\left(\partial D^k\right)} \left\| \frac{1}{\sqrt{J_\lambda^k}} \right\|_{L^\infty\left(D^k\right)} = O\left(\frac{1}{\sqrt{h_k}} \right)$$

□

Now for projection error estimates on local elements.

Theorem 7. *There exists a constant C independent of h such that for every $u \in H^{N+1}\left(D^k\right)$ one has.*

$$\left\| u - \mathcal{P}_{N,\lambda}^k u \right\|_{L^2\left(D^k\right)} \leq C \cdot G_k \cdot h_k^{N+1} \|u\|_{H^{N+1}\left(D^k\right)}.$$

where $G_{k,\lambda}$ depends on the local geometry of D^k and the mapping parameter λ.

Proof. The definition of $\mathcal{P}_{N,\lambda}^k$ yields

$$\mathcal{P}_{N,\lambda}^k u = \arg \min_{\mathbf{v} \in V_N^{k,\lambda}} \|u - v\|_{L^2\left(D^k\right)}$$

and furthermore Bramble-Hilbert guarantees the existence of a polynomial p for which

$$\|u - p\|_{L^2(\mathbf{I})} \leq C \cdot |u|_{H^{N+1}(\mathbf{I})}$$

so

$$\begin{aligned}
\left\|u - \mathcal{P}^k_{N,\lambda} u\right\|_{L^2\left(D^k\right)} &= \left\|\sqrt{J^k}\left(u - \mathcal{P}^k_{N,\lambda} u\right)\right\|_{L^2(\mathbf{I})} \\
&\leq \left\|\sqrt{J^k}\,(u - p)\right\|_{L^2(\mathbf{I})} \\
&\leq \left\|\sqrt{J^k}\right\|_{L^\infty(\mathbf{I})} \|u - p\|_{L^2(\mathbf{I})} \\
&\leq C \left\|\sqrt{J^k}\right\|_{L^2(\mathbf{I})} \|u\|_{H^{N+1}(\mathbf{I})}
\end{aligned}$$

and provided the mesh is shape-regular the following scaling argument holds

$$\left\|\sqrt{J^k}\right\|_{L^\infty(\mathbf{I})} |u|_{H^{N+1}(\mathbf{I})} \leq C \left\|\sqrt{J^k}\right\|_{L^\infty\left(D^k\right)} \left\|\frac{1}{\sqrt{J^k}}\right\|_{L^\infty\left(D^k\right)} h_k^{N+1} \|u\|_{H^{N+1}\left(D^k\right)}$$

(see Ern and Guermond [34] pg.66), establishing the result with

$$G_k = C \left\|\sqrt{J^k}\right\|_{L^\infty\left(D^k\right)} \left\|\frac{1}{\sqrt{J^k}}\right\|_{L^\infty\left(D^k\right)} \tag{4.1.4}$$

□

Remark 8. The dependence of the estimate on local geometry is measured by the constant G_k in (4.1.4), and so for the estimate to retain its h-optimality it is necessary

that

$$\kappa_1 \leq G_k \leq \kappa_2 \tag{4.1.5}$$

holds for $0 < \kappa_1 \leq \kappa_2$ independent of h. Note however that the constant C depends on λ, which will not affect the a-priori analysis as λ is chosen independently of h, and is fixed during mesh refinement.

Next I provide a means to estimate truncation errors which involve boundary integrals, which will first require estimating the gradients of the truncation error to ensure that the loss of order in this case does not exceed expectations.

Theorem 9. *There exists a constant C independent of h such that for each $u \in H^{N+1}\left(D^k\right)$ one has*

$$\left\| \frac{\partial}{\partial x}\left(u - \mathcal{P}^k_{N,\lambda} u\right) \right\|_{L^2\left(D^k\right)} \leq C G^*_k h_k^{N+1} \left\|u\right\|_{H^{N+1}\left(D^k\right)} \tag{4.1.6}$$

where G^*_k depends only on local element geometry.

Proof. One has

$$\begin{aligned}
\left\| \frac{\partial}{\partial x}\left(u - \mathcal{P}^k_{N,\lambda} u\right) \right\|_{L^2\left(D^k\right)} &= \left\| \mathbf{G}^k_1 \nabla \left(u - \mathcal{P}^k_{N,\lambda} u\right) \right\|_{L^2\left(D^k\right)} \\
&= \left\| \frac{\sqrt{J^k}}{\sqrt{J^k}} \mathbf{G}^k_1 \nabla \left(u - \mathcal{P}^k_{N,\lambda} u\right) \right\|_{L^2\left(D^k\right)} \\
&\leq \left\| \mathbf{G}^k_1 \sqrt{J^k} \right\|_{L^\infty\left(D^k\right)} \left\| \frac{1}{\sqrt{J^k}} \nabla \left(u - \mathcal{P}^k_{N,\lambda} u\right) \right\|_{L^2\left(D^k\right)} \\
&= \left\| \mathbf{G}^k_1 \sqrt{J^k} \right\|_{L^\infty\left(D^k\right)} \left\| \nabla \left(u - \mathcal{P}^k_{N,\lambda} u\right) \right\|_{L^2(\mathbf{I})} \\
&\leq C \left\| \mathbf{G}^k_1 \sqrt{J^k} \right\|_{L^\infty\left(D^k\right)} |u|_{H^{N+1}(\mathbf{I})}
\end{aligned}$$

where I have again applied the Bramble-Hilbert lemma for the last inequality. The

result now follows from scaling with

$$G_k^* = \left\| \mathbf{G}_1^k \sqrt{J^k} \right\|_{L^\infty\left(D^k\right)}.$$

□

Remark 10. What looks like optimal recovery of order in (4.1.6) is misleading. The constant G_k^* will generally scale as $1/h^k$ for well behaved meshes and so the next assumption on our mesh will be

$$G_k^* \leq \frac{C}{h_k} \tag{4.1.7}$$

for C independent of h.

Theorem 11. *There exists a constant C independent of h such that for all $u \in H^{N+1}\left(D^k\right)$ one has*

$$\left\| u - \mathcal{P}_{N,\lambda}^k u \right\|_{L^2\left(\partial D^k\right)} \leq C h_k^{N+1/2} \left\| u \right\|_{H^{N+1}\left(D^k\right)}.$$

provided G_k, G_k^* defined above satisfy (4.1.5) and (4.1.7) respectively.

Proof. Applying the multiplicative trace inequality (4) one obtains

$$\begin{aligned} \left\| u - \mathcal{P}_{N,\lambda}^k u \right\|_{L^2\left(\partial D^k\right)}^2 &\leq C \left\| u - \mathcal{P}_{N,\lambda}^k \right\|_{L^2\left(D^k\right)} \left| u - \mathcal{P}_{N,\lambda}^k u \right|_{H^1\left(D^k\right)} \\ &\quad + C \frac{1}{h_k} \left\| u - \mathcal{P}_{N,\lambda}^k u \right\|_{L^2\left(D^k\right)} \\ &\leq C h_k^{N+1/2} \left\| u \right\|_{H^{N+1}\left(D^k\right)} \end{aligned}$$

the latter inequality following from estimates (7),(9), and assumptions (4.1.5) and (4.1.7). □

4.2 Unmodified and Mapped Theory

Here I present three standard theoretical items for the case of the DG formulation of the acoustic wave equation. Semidiscrete stability guarantees that if the system (2.3.5) is solved exactly in time, then the energy of the system does not increase in time if all source terms are zero, and otherwise increases only as fast as the time integral of the norm of the source term if it is nonzero. The error estimate gives an idea as to how one should expect solution quality to behave as the mesh is refined, or as polynomial order is increased. Finally bounds for fully discrete stability are derived which gives what stable timesteps must be asymptotically for a fixed explicit timestepping method. Recall that the matrices $\mathbf{A}, \mathbf{B}$ are symmetric, and assumed independent of space. The results here will apply equally to unmodified DG and to mapped DG, with special attention paid to the terms which arise out of mapped DG (e.g. through the Jacobian J^k_λ) which can potentially damage convergence. Many of the ideas used here were motivated by the preprint [101].

4.2.1 Stability

Stability proved in this thesis means specifically that the energy of the system has a well behaved bound. Specifically, defining the energy for a Friedrich system (2.1.1) to be

$$E \equiv \sum_k E^k \equiv \frac{1}{2} \sum_k \left(\|\mathbf{Q}_N\|^2_{D^k} \right)$$

then the energy is bounded in time. Furthermore, L^2 bounds are obtained for a CFL-like condition to be imposed on timesteps.

4.2.1.1 Semidiscrete Stability

Theorem 12 (Semidiscrete Stability)**.** *Suppose that the domain Ω_h is periodic and that $\mathbf{Q}^N$ satisfies the weak form (2.3.5) for each $k = 1, \ldots, K$ and for all $t \in [0, T]$. Then the energy E satisfies the differential inequality*

$$\frac{d}{dt}\frac{1}{2}E \leq C\left(E + \sum_k \|\mathbf{R}\|^2_{D^k}\right).$$

In the event that $R = 0$, then $C \leq 0$.

To prove this theorem, I first prove it for the case of two adjacent elements.

Lemma 13 (Shared edge stability)**.** *Suppose that the domain Ω_h consists of two neighboring quadrilaterals D^1, D^2. Then the differential inequality in theorem (12) holds.*

Proof. Without loss of generality assume $\mathbf{R} = \mathbf{0}$ and consider the local weak form on D^1 :

$$\int_{D^1} \mathbf{v}\left(\frac{\partial \mathbf{Q}_N}{\partial t} + \frac{\partial \mathbf{A}\mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}_N}{\partial y}\right) = \frac{1}{2}\int_{\partial D^1} \mathbf{v}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)[\mathbf{Q}_N]$$

which holds for all functions $\mathbf{v}$ in the local solution space $Q^N(D^1)$. Thus we may take $\mathbf{v} = \mathbf{Q}^N$, and sum the components of this vector equation to obtain a dot product. This may be concisely written

$$\left(\mathbf{Q}_N, \frac{\partial \mathbf{Q}_N}{\partial t} + \frac{\partial \mathbf{A}\mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}_N}{\partial y}\right)_{D^1} = \left(\mathbf{Q}_N^-, \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)[\mathbf{Q}_N]\right)_{\partial D^1} \tag{4.2.1}$$

applying the product rule on the time derivative in (4.2.1) yields

$$\frac{1}{2}\frac{d}{dt}\|\mathbf{Q}_N\|^2_{D^1} + \left(\mathbf{Q}_N, \frac{\partial \mathbf{A}\mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}_N}{\partial y}\right)_{D^1} = \left(\mathbf{Q}_N^-, \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)[\mathbf{Q}_N]\right)_{\partial D^1} \tag{4.2.2}$$

and by symmetry of $\mathbf{A}, \mathbf{B}$ integration by parts says that

$$\left(\mathbf{Q}_N, \frac{\partial \mathbf{A}\mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}_N}{\partial y}\right)_{D^1} = \left(\mathbf{Q}_N^-, \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)\left[\mathbf{Q}_N\right] - \frac{1}{2}\mathbf{C}\mathbf{Q}_N^-\right)$$

so that (4.2.2) becomes

$$\frac{1}{2}\frac{d}{dt}\left\|\mathbf{Q}_N\right\|_{D^1}^2 = \frac{1}{2}\left(\mathbf{Q}_N^-, \left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)\left[\mathbf{Q}_N\right] - \mathbf{C}\mathbf{Q}_N^-\right)$$

to obtain the energy at the shared edge $e = e^1 \cap e^2$ one sums the contributions from both elements and uses the fact that on e^1 one has $\mathbf{C}^+ = -\mathbf{C}^-, \left[\mathbf{Q}_N\right]^+ = -\left[\mathbf{Q}_N\right]^-$

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\sum_{i=1}^{2}\left\|\mathbf{Q}_N\right\|_{D^i}^2 &= \frac{1}{2}\left(\mathbf{Q}_N^-, \left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)\left[\mathbf{Q}_N\right] - \mathbf{C}\mathbf{Q}_N^-\right)_{e^1} \\ &\quad + \frac{1}{2}\left(\mathbf{Q}_N^+, \left(\mathbf{C} + \mathbf{C}^*\mathbf{C}\right)\left[\mathbf{Q}_N\right] + \mathbf{C}\mathbf{Q}_N^+\right)_{e^1}. \qquad (4.2.3)\end{aligned}$$

Thus the energy will be bounded in time provided the boundary integrals above are nonnegative, it is sufficient to show that the integrand is nonnegative for any choice of $\mathbf{Q}_N^+, \mathbf{Q}_N^-$. The term inside the integral of (4.2.3) is a quadratic form, and recalling that $\left[\mathbf{Q}_N\right] = \mathbf{Q}_N^- - \mathbf{Q}_N^+$, it may be written

$$\begin{aligned}\begin{pmatrix} \mathbf{Q}_N^- & \mathbf{Q}_N^+ \end{pmatrix}\begin{pmatrix} -\mathbf{C}^*\mathbf{C} & \mathbf{C}^*\mathbf{C} - \mathbf{C} \\ \mathbf{C} + \mathbf{C}^*\mathbf{C} & -\mathbf{C}^*\mathbf{C} \end{pmatrix}\begin{pmatrix} \mathbf{Q}_N^- \\ \mathbf{Q}_N^+ \end{pmatrix} &= -\left[\mathbf{Q}_N\right]^T\mathbf{C}^*\mathbf{C}\left[\mathbf{Q}_N\right] \\ &\leq 0\end{aligned}$$

where I have used the fact that $x^T A x = x^T \frac{1}{2}\left(A + A^T\right)x$ when the entries of A, x are real. □

Corollary 14. *In the absence of a source term* $\mathbf{R}$*, the energy on interior edges* e *of*

the penalized DG discretization of a Friedrich system evolves as

$$\frac{1}{2}\frac{d}{dt}E = -\sum_{e\in\Gamma} \|\mathbf{C}\left[\mathbf{Q}_N\right]\|_e^2 . \tag{4.2.4}$$

Corollary 14 may be alternatively interpreted as saying: DG decreases energy when jumps appear between element interfaces, and the magnitude of reduction is proportional to the magnitude of the jump. The addition of a source term may be accounted for easily through the Cauchy-Schwarz and Young inequalities, this is summarized in the below corollary:

Corollary 15. *If the source term* $\mathbf{R}$ *is nonzero, then the energy on interior edges* e *of the penalized DG discretization of a Friedrich system satisfies the following differential inequality:*

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}E &= -\sum_{e\in\Gamma}\frac{1}{2}\|\mathbf{C}\left[\mathbf{Q}_N\right]\|_e^2 + \sum_{k=1}^{K}\left(\mathbf{Q}_N,\mathbf{R}\right)_{D^k} \\
&\leq -\sum_{e\in\Gamma}\frac{1}{2}\|\mathbf{C}\left[\mathbf{Q}_N\right]\|_e^2 + \sum_{k=1}^{K}\|\mathbf{Q}_N\|_{D^k}\|\mathbf{R}\|_{D^k} \\
&\leq -\sum_{e\in\Gamma}\frac{1}{2}\|\mathbf{C}\left[\mathbf{Q}_N\right]\|_e^2 + \sum_{k=1}^{K}\left(\frac{1}{2}\|\mathbf{Q}_N\|_{D^k}^2 + \frac{1}{2}\|\mathbf{R}\|_{D^k}^2\right) \\
&= -\sum_{e\in\Gamma}\frac{1}{2}\|\mathbf{C}\left[\mathbf{Q}_N\right]\|_e^2 + \frac{1}{2}E + \sum_{k=1}^{K}\frac{1}{2}\|\mathbf{R}\|_{D^k}^2
\end{aligned}$$

I now complete the semidiscrete analysis by showing the method is stable on boundary edges when boundary conditions of type (2.3.6) are imposed.

Lemma 16 (Dirichlet Boundary Stability)**.** *Suppose that on boundary edges the following equation is enforced:*

$$\mathbf{Q}_N^+ = \mathbf{D}\mathbf{Q}_N^-$$

where

$$(\mathbf{C}^*\mathbf{C} - \mathbf{C})\,\mathbf{D}$$

is negative definite. Then the DG method is semidiscrete stable on these edges.

Proof. Following the same arguments above, integrate by parts and apply the product rule to obtain

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}E &= \frac{1}{2}\left(\mathbf{Q}^-, (\mathbf{C} - \mathbf{C}^*\mathbf{C})\,[\mathbf{Q}] - \mathbf{C}\mathbf{Q}^-\right)_e \\
&= \frac{1}{2}\left(\mathbf{Q}^-, -\mathbf{C}^*\mathbf{C}\mathbf{Q}^- + (\mathbf{C} - \mathbf{C}^*\mathbf{C})\,\mathbf{D}\mathbf{Q}^-\right)_e \\
&= -\frac{1}{2}\left\|\mathbf{C}\mathbf{Q}^-\right\|_e + \int_e \mathbf{Q}^- \cdot (\mathbf{C} - \mathbf{C}^*\mathbf{C})\,\mathbf{D}\mathbf{Q}^- \\
&\leq 0
\end{aligned}$$

as claimed. □

4.2.1.2 Fully Discrete Stability

The next task in stability analysis is fully discrete stability, that is stability in the presence of inexact timestepping. The way this is usually accomplished is through eigenvalue estimates of the discrete weak form. To do this I must first introduce a lifting operator, and estimate its norm. Define $\mathcal{L}_N$ to be the operator such that for every $u \in L^2\left(\partial D^k\right)$ the following equation holds

$$\begin{aligned}
(u, v)_{\partial D^k} &= (\mathcal{L}_N u, v)_{D^k} \\
\mathcal{L}_N u &\in V_{N,\lambda}^k
\end{aligned} \tag{4.2.5}$$

and one has also the following bound

Theorem 17 (Norm of Continuous Lifting Operator)**.** *The operator $\mathcal{L}_N$ is bounded on $V_{N,\lambda}^k$ for every fixed N, λ and satisfies*

$$\|\mathcal{L}_N u\|_{D^k} \leq \frac{CN^2}{h_k} \|u\|_{D^k}$$

for every $u \in V_{N,\lambda}^k$.

Proof. We have

$$\begin{aligned} \|\mathcal{L}_N u\|_{D^k}^2 &= (\mathcal{L}_N u, \mathcal{L}_N u)_{D^k} \\ &= (\mathcal{L}_N u, u)_{\partial D^k} \\ &\leq \|\mathcal{L}_N u\|_{\partial D^k} \cdot \|u\|_{\partial D^k} \\ &\leq \left(K_\lambda \frac{N}{\sqrt{h_k}}\right) \|\mathcal{L}_N u\|_{D^k} \left(K_\lambda \frac{N}{\sqrt{h_k}}\right) \|u\|_{D^k} \end{aligned}$$

from the local trace inverse inequality (theorem 6). □

Now to show fully discrete stability I will assume for simplicity that periodic boundary conditions are imposed and that the solution $\mathbf{Q}_N$ has the form

$$\mathbf{Q}_N(t, x, y) = \exp(t\mathbf{H})\, \mathbf{w}(x, y).$$

From the assumption that $\mathbf{Q}_N$ satisfies the weak form and the fact that $\exp(t\mathbf{H})$ is invertible for all t the following holds

$$\left(\mathbf{v}, \mathbf{H}\mathbf{w} + \frac{\partial \mathbf{A}\mathbf{w}}{\partial x} + \frac{\partial \mathbf{B}\mathbf{w}}{\partial y}\right)_{D^k} = \left(\mathbf{v}, \frac{1}{2}(\mathbf{C} - \mathbf{C}^*\mathbf{C})[\mathbf{w}]\right)_{\partial D^k} \tag{4.2.6}$$

to obtain L^2 estimates for $\mathbf{H}\mathbf{w}$ test with $\mathbf{H}\mathbf{w}$ and use the lifting operator (4.2.5) to

turn everything into volume integrals:

$$\begin{aligned}
\|\mathbf{Hw}\|_{D^k}^2 &= -\left(\mathbf{Hw}, \frac{\partial \mathbf{Aw}}{\partial x} + \frac{\partial \mathbf{Bw}}{\partial y}\right)_{D^k} + \left(\mathbf{Hw}, \mathcal{L}_N \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)[\mathbf{w}]\right)_{D^k} \\
&\leq \|\mathbf{Hw}\|_{D^k} \left\| \frac{\partial \mathbf{Aw}}{\partial x} + \frac{\partial \mathbf{Bw}}{\partial y} \right\|_{D^k} \qquad (4.2.7) \\
&\quad + \|\mathbf{Hw}\|_{D^k} \left\| \mathcal{L}_N \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)[\mathbf{w}] \right\|_{D^k} \qquad (4.2.8)
\end{aligned}$$

Unfortunately the norm in (4.2.7) can not be directly estimated because $[\mathbf{w}]$ has no regularity requirement, with positive trace values coming from nearby elements. We can however apply the triangle inequality and estimate the derivative terms with the Markov inequality (theorem 5) The task then becomes to estimate the term $\left\|\mathcal{L}_N \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)[\mathbf{w}]\right\|_{D^k}^2$. We have to globalize the estimate by summing over all elements to obtain

$$\begin{aligned}
&\sum_{D^k \in \mathcal{T}_h} \left\| \mathcal{L}_N \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)[\mathbf{w}] \right\|_{D^k}^2 \\
\leq\ & \sum_{D^k \in \mathcal{T}_h} \left\| \mathcal{L}_N \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)\mathbf{w}^- \right\|_{D^k}^2 + \left\| \mathcal{L}_N \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)\mathbf{w}^+ \right\|_{D^k}^2 \\
\leq\ & C \sum_{D^k \in \mathcal{T}_h} \left\| \mathcal{L}_N \frac{1}{2}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)\mathbf{w} \mid_{\partial D^k} \right\|_{D^k}^2 \qquad (4.2.9) \\
\leq\ & C \sum_{D^k \in \mathcal{T}_h} \frac{N^2}{h_k} \|\mathbf{C} - \mathbf{C}^*\mathbf{C}\|_{D^k} \|\mathbf{w}\|_{D^k}
\end{aligned}$$

the final inequality following from the lifting operator bound found in theorem 17. Combining the estimate from Markov's inequality and from (4.2.9) above yields

$$\|\mathbf{Hw}\|_{L^2(\mathcal{T}_h)} \leq C \frac{N^2}{h}$$

with C absorbing all other constants. Taking the supremum over all initial conditions $\mathbf{w}$ yields

$$\|\mathbf{H}\|_{L^2(\mathcal{T}_h)} \leq C\frac{N^2}{h}$$

which yields the following condition on Δt:

$$\Delta t \leq O\left(\frac{h}{N^2}\right).$$

Remark 18. Note the explicit appearance of the Markov and inverse trace inequality. Their application here is sharp in the sense that the bounds can not be otherwise improved theoretically, so the asymptotic analysis here is sharp in the sense that the quadratic growth in N is necessary. Therefore attempts to reduce this impact must change those elements of the formulation in some way in order to improve this analysis. This however only applies to L^2 stability, which is what the above argument provides. If instead one can directly bound the spectral radius of the DG operator $\mathbf{H}$ the bound could very well be smaller.

4.2.2 Error Estimates

This section is designed to prove an error estimate of the form

$$\|\mathbf{Q}_N^h - \mathbf{Q}\|_{\mathcal{E}} \leq Ch^{N+1/2} \|\mathbf{Q}\|_{H^{N+1}(\mathcal{T}_h)}$$

with C independent of the mesh, and $\|\cdot\|_{\mathcal{E}}$ a suitably defined energy norm. One approach to construct such an estimate is to apply the stability estimate proven earlier, but applied to a suitable projection of the truncation error. The resulting differential inequality can then be used to prove the desired estimate through application of

Gronwall's inequality [80]. This will be the approach taken here, but it should be noted that if willing to sacrifice an order reduction of 1/2 in the estimate, a constant C can be obtained which grows at most linearly in time [45]. The basic approach is to consider the element-local error term

$$\mathbf{Q}_N - \mathbf{Q}$$

and realize that it satsifies the weak form 2.3.5 with zero source term. This error is then split into two terms, a truncation error plus a local error:

$$\begin{aligned} \mathbf{Q}_N - \mathbf{Q} &= \mathbf{Q} - \mathcal{P}_N^k \mathbf{Q} + \mathcal{P}_N^k \mathbf{Q} - \mathbf{Q}_N \\ &\equiv \epsilon + \eta \end{aligned}$$

where $\mathcal{P}_N^k$ is some projection onto the local solution space. The truncation error ϵ will have known approximation theoretic estimates, and standard inequalities will be used to express the local error in terms of these estimates. In order for this to work, a standard assumption is made on the analytic solution $\mathbf{Q}$ that it properly satisfies the weak form (2.1.2) posed on the infinite dimensional space $H^1(\mathcal{T}_h)$. This ensures that the error satisfies the homogeneous acoustic wave equation weakly.

For the error analysis dealing with potential nonpolynomial functions can make handling derivatives more difficult, and the more general formulation of the Friedrichs' system makes the task a little more difficult. I used two techniques of Warburton's unpublished report [101] to handle both of these issues. In that paper using orthogonality properties of projectors gives a condition to recover optimal order convergence, and a manipulation provides a way to ensure that jumps in the local error can be

contributed back to the left-hand-side without scaling problems.

Theorem 19. *Suppose* $\mathbf{Q} \in H^{N+1}(\mathcal{T}_h)$ *satisfies the weak form (2.1.2) posed on* $H^1(\mathcal{T}_h)$, *and that* $\mathbf{Q}_N$ *satisfies the weak form posed on* P^N. *Then there exists a constant* C *independent of* h *such that*

$$\|\mathbf{Q} - \mathbf{Q}_N\|_{\mathcal{E}} \leq Ch^{N+1/2}\|\mathbf{Q}\|_{H^{N+1}(\mathcal{T}_h)}$$

Proof. Decompose the error

$$\begin{aligned} \mathbf{Q}_N - \mathbf{Q} &= \mathbf{Q} - \mathcal{P}^k_{N,\lambda}\mathbf{Q} + \mathcal{P}^k_{N,\lambda}\mathbf{Q} - \mathbf{Q}_N \\ &\equiv \epsilon + \eta \end{aligned}$$

where the projection $\mathcal{P}^k_{N,\lambda}$ projects the solution onto the local solution space, and can be either the mapped projection operator or standard L^2 projection depending on if the underlying DG method is mapped or unmapped. Note that $\epsilon + \eta$ satisfies the homogeneous Friedrich system weakly, which means that for all $\mathbf{v} \in \left(V_N^k\right)^3$ we have

$$\left(\mathbf{v}, \frac{\partial \epsilon + \eta}{\partial t} + \frac{\partial \mathbf{A}(\epsilon + \eta)}{\partial x} + \frac{\partial \mathbf{B}(\epsilon + \eta)}{\partial y}\right)_{D^k} = (\mathbf{v}, (\mathbf{C} - \mathbf{C}^*\mathbf{C})[\epsilon + \eta])_{\partial D^k}$$

applying bilinearity yields the following equation

$$\begin{aligned} \left(\mathbf{v}, \frac{\partial \eta}{\partial t} + \frac{\partial \mathbf{A}\eta}{\partial x} + \frac{\partial \mathbf{B}\eta}{\partial y}\right)_{D^k} - (\mathbf{v}, (\mathbf{C} - \mathbf{C}^*\mathbf{C})[\eta])_{\partial D^k} &= -\left(\mathbf{v}, \frac{\partial \epsilon}{\partial t} + \frac{\partial \mathbf{A}\epsilon}{\partial x} + \frac{\partial \mathbf{B}\epsilon}{\partial y}\right)_{D^k} \\ &\quad - (\mathbf{v}, (\mathbf{C} - \mathbf{C}^*\mathbf{C})[\epsilon])_{\partial D^k} \end{aligned}$$

testing with $\mathbf{v} = \eta$ and applying the same product rule + integration by parts argu-

ment as in the stability proof to the left hand side yields

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}\left\|\eta\right\|^2_{L^2(\mathcal{T}_h)} + \sum_{\text{Unique } e}\left\|\mathbf{D}\left[\eta\right]\right\|^2_e &= \sum_{D^k\in\mathcal{T}_h} -\left(\eta, \frac{\partial\epsilon}{\partial t} + \frac{\partial \mathbf{A}\epsilon}{\partial x} + \frac{\partial \mathbf{B}\epsilon}{\partial y}\right)_{D^k} \\
&\quad - \sum_{D^k\in\mathcal{T}_h}\left(\eta, \left(\mathbf{C}-\mathbf{C}^*\mathbf{C}\right)\left[\epsilon\right]\right)_{\partial D^k} \\
&= \sum_{D^k\in\mathcal{T}_h} -\left(\eta, \frac{\partial\epsilon}{\partial t}\right)_{D^k} + \left(\frac{\partial\eta}{\partial x}, \mathbf{A}\epsilon\right)_{D^k} \\
&\quad - \sum_{D^k\in\mathcal{T}_h}\left(\eta, \left(\mathbf{C}-\mathbf{C}^*\mathbf{C}\right)\left[\epsilon\right] - \mathbf{C}\epsilon^-\right)_{\partial D^k} \\
&= \sum_{D^k\in\mathcal{T}_h} T_1^k + T_2^k + T_3^k
\end{aligned} \tag{4.2.10}$$

where

$$\begin{aligned}
T_1^k &= -\left(\eta, \frac{\partial\epsilon}{\partial t}\right)_{D^k} \\
T_2^k &= \left(\frac{\partial\eta}{\partial x}, \mathbf{A}\epsilon\right)_{D^k} + \left(\frac{\partial\eta}{\partial y}, \mathbf{B}\epsilon\right)_{D^k} \\
T_3^k &= -\left(\eta^-, \left(\mathbf{C}-\mathbf{C}^*\mathbf{C}\right)\left[\epsilon\right] - \mathbf{C}\epsilon^-\right)_{\partial D^k}.
\end{aligned}$$

Bounding T_1^k is simplest by Cauchy-Schwarz:

$$\left|T_1^k\right| \le \left\|\eta\right\|_{D^k} \cdot \left\|\frac{\partial\epsilon}{\partial t}\right\|_{D^k} \tag{4.2.11}$$

The next term T_2^k may be bounded by realizing that the truncation error ϵ is orthog-

onal to the local solution space $V_{N,\lambda}^k$, so that

$$\begin{aligned}\left(\frac{\partial \eta}{\partial x}, \mathbf{A}\epsilon\right)_{D^k} &= \left(\mathbf{A}\frac{\partial \eta}{\partial x} - \mathcal{P}_{N,\lambda}^k \mathbf{A}\frac{\partial \eta}{\partial x}, \epsilon\right) \\ &\leq \left\|\mathbf{A}\frac{\partial \eta}{\partial x} - \mathcal{P}_{N,\lambda}^k \mathbf{A}\frac{\partial \eta}{\partial x}\right\|_{D^k} \cdot \|\epsilon\|_{D^k} \\ &\leq C\,\|\eta\|_{D^k} \cdot \|\epsilon\|_{D^k} \qquad (4.2.12)\end{aligned}$$

Similarly for the term $\left(\frac{\partial \eta}{\partial y}, \mathbf{B}\epsilon\right)_{D^k}$. The step (4.2.12) could use a standard inverse inequality for the finite dimensional space $V_{N,\lambda}^k$, however the scaling of the derivatives will yield a suboptimal error result. Since the norm is measuring the truncation instead one is left with the constant C effectively measuring the deviation of the local element D^k from an affine element. If the mesh is assumed to be asymptotically affine, then this term could be seen as yielding an optimal estimate. Also one can assume that C scales no worse than

$$C \sim \frac{Q}{\sqrt{h}}$$

(where Q is a generic h independent constant) and still recover an optimal estimate, since the loss of $1/2$ order is unavoidable through the use of the multiplicative trace inequality (4) (which I will use in bounding T_3^k). The technique used for equation (4.2.12) was repurposed from the preprint [101].

Bounding T_3^k I will use a manipulation found in [101] (equation (4.2.13)) to ensure that jumps in η can be contributed back to the left-hand-side of (4.2.10) , and then I will apply Cauchy-Schwarz and the Young inequality twice to obtain

$$
\begin{aligned}
T_3^k &= -\left(\eta^-, \left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)[\epsilon] - \mathbf{C}\epsilon^-\right)_{\partial D^k} \\
&= \frac{1}{4}\left(\mathbf{C}[\eta], \epsilon^+ + \epsilon^-\right)_{\partial D^k} - \frac{1}{4}\left(\mathbf{C}[\eta], \mathbf{C}[\epsilon]\right)_{\partial D^k} \qquad (4.2.13) \\
&\leq \frac{1}{4}\|\mathbf{C}[\eta]\|_{\partial D^k}\left(\left\|\epsilon^+ + \epsilon^-\right\|_{\partial D^k} + \|\mathbf{C}[\epsilon]\|_{\partial D^k}\right)_{\partial D^k} \\
&\leq \frac{\alpha+\beta}{8}\|\mathbf{C}[\eta]\|^2_{\partial D^k} + \frac{1}{8\alpha}\left\|\epsilon^+ + \epsilon^-\right\|^2_{\partial D^k} + \frac{1}{8\beta}\|\mathbf{C}[\epsilon]\|^2_{\partial D^k} \qquad (4.2.14)
\end{aligned}
$$

the step (4.2.13) used symmetry of $\mathbf{C}$. Since T_3^k effectively is the term coupling the local elements together it is not enough to give a local bound as with T_1^k, T_2^k. Furthermore the term containing jumps in η may be effectively ignored since the constants α, β will be chosen so that it may be used as a contribution to the same term on the left-hand-side of (4.2.10). Sum over all elements to obtain

$$
\begin{aligned}
&\textstyle\sum_{D^k\in\mathcal{T}_h} \frac{1}{8\alpha}\|\epsilon^+ + \epsilon^-\|^2_{\partial D^k} + \frac{1}{8\beta}\|\mathbf{C}[\epsilon]\|^2_{\partial D^k} \\
\leq\ &\textstyle\sum_{D^k\in\mathcal{T}_h}\left(\frac{1}{8\alpha} + \frac{\|\mathbf{C}\|^2_{\partial D^k}}{8\beta}\right)\left(\|\epsilon^-\|^2_{\partial D^k} + \|\epsilon^+\|^2_{\partial D^k}\right) \qquad (4.2.15)
\end{aligned}
$$

bounding the norm of the interior traces in (4.2.15) can be done directly through the use of the multiplicative trace inequality (theorem (4)). However the regularity requirements of the multiplicative trace inequality could be violated by the exterior trace term, as it is the result of contributions from four neighboring elements without any sort of a-priori regularity. To circumvent this note that on a given shared edge $e = e^1 \cap e^2$ that ϵ^+ on e^1 is simply ϵ^- on e^2. Summing over all unique interior edges

$e = e^1 \cap e^2$ and using the fact that $\mathbf{C}$ is continuous on the whole domain we obtain

$$\begin{aligned} & \sum_{\mathrm{e}\in\,\Gamma} \left(\tfrac{1}{8\alpha} + \tfrac{\|\mathbf{C}\|_e^2}{8\beta}\right) \left(\|\epsilon^-\|_e^2 + \|\epsilon^+\|_e^2\right) \\ = \; & \sum_{\mathrm{e}\in\,\Gamma} \left(\tfrac{1}{8\alpha} + \tfrac{\|\mathbf{C}\|_e^2}{8\beta}\right) \left(\|\epsilon^-\|_{e^1}^2 + \|\epsilon^-\|_{e^2}^2\right) \\ = \; & \sum_{\mathrm{e}\in\,\Gamma} C \left(\tfrac{1}{8\alpha} + \tfrac{\|\mathbf{C}\|_e^2}{8\beta}\right) \left(\|\epsilon^-\|_e^2\right) \end{aligned}$$

and a similar argument applies for boundary edges applying the condition $\mathbf{Q}^+ = \mathbf{D}\mathbf{Q}^-$. The final bound obtained is

$$\begin{aligned} \sum_{D^k\in\mathcal{T}_h} T_3^k \;\leq\; & \sum_{D^k\in\mathcal{T}_h} \frac{\alpha+\beta}{8} \|\mathbf{C}\,[\eta]\|_{\partial D^k}^2 \\ & + \sum_{\mathrm{e}\in\,\Gamma} C \left(\frac{1}{8\alpha} + \frac{\|\mathbf{C}\|_e^2}{8\beta}\right) \left(\|\epsilon^-\|_e^2\right) \\ & + \sum_{\text{boundary } e} \left(\frac{1}{8\alpha} + \frac{\|\mathbf{C}\|_e^2}{8\beta}\right) \left(\|\epsilon^-\|_e^2 + \|\mathbf{D}\mathbf{Q}^- - \mathcal{P}_{N,\lambda}^k \mathbf{D}\mathbf{Q}^-\|_e^2\right) \end{aligned}$$

so that finally one obtains for suitable α, β

$$\begin{aligned} \frac{1}{2}\frac{d}{dt}\|\eta\|_{L^2(\mathcal{T}_h)}^2 + C_1 \sum_{\text{Unique } e} \|\mathbf{D}\,[\eta]\|_e^2 \;\leq\; & \sum_{D^k\in\mathcal{T}_h} T_1^k + T_2^k + T_3^k \\ \leq\; & \sum_{D^k\in\mathcal{T}_h} \|\eta\|_{D^k} \cdot \left\|\frac{\partial\epsilon}{\partial t}\right\|_{D^k} \\ + \; & \sum_{D^k\in\mathcal{T}_h} C_2 \|\eta\|_{D^k} \cdot \|\epsilon\|_{D^k} \\ + \; & \sum_{\mathrm{e}\in\,\Gamma} C_3 \left(\|\epsilon^-\|_e^2\right) \\ + \; & \sum_{\text{boundary } e} C_4 \left(\|\epsilon^-\|_e^2 + \|\mathbf{D}\mathbf{Q}^- - \mathcal{P}_{N,\lambda}^k \mathbf{D}\mathbf{Q}^-\|_e^2\right) \end{aligned}$$

Applying projection estimates(7), (11) yields

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\left\|\eta\right\|^2_{L^2(\mathcal{T}_h)} + C_1 \sum_{\text{Unique } e} \left\|\mathbf{C}\left[\eta\right]\right\|^2_e &\leq C_5 h^{N+1} \sum_{D^k \in \mathcal{T}_h} \left\|\eta\right\|_{D^k} \cdot \left\|\frac{\partial \mathbf{Q}}{\partial t}\right\|_{H^{N+1}\left(D^k\right)} \\ &+ C_6 h^{N+1} \sum_{D^k \in \mathcal{T}_h} \left\|\eta\right\|_{D^k} \cdot \left\|\mathbf{Q}\right\|_{H^{N+1}\left(D^k\right)} \\ &+ C_7 h^{2N+1} \sum_{D^k \in \mathcal{T}_h} C_3 \left\|\mathbf{Q}\right\|^2_{H^{N+1}\left(D^k\right)}\end{aligned}$$

where I have absorbed the Sobolev norm of the boundary condition operator $\mathbf{D}$ into C_7. Now integrate in time and apply the Gronwall inequality from theorem 3 to obtain at time T

$$\begin{aligned}\sqrt{\frac{1}{2}\left\|\eta\right\|_{L^2(\mathcal{T}_h)} + \int_0^T C_1 \sum_{\text{Unique } e} \left\|\mathbf{C}\left[\eta\right]\right\|^2_e \, dt} &\leq C_7 h^{N+1/2} \sqrt{\sup_{0\leq t\leq T} C_3 \left\|\mathbf{Q}\right\|^2_{H^{N+1}(\mathcal{T}_h)}} \\ &+ C_5 h^{N+1} \int_0^T \left\|\frac{\partial \mathbf{Q}}{\partial t}\right\|_{H^{N+1}(\mathcal{T}_h)} \\ &+ C_6 h^{N+1} \int_0^T \left\|\mathbf{Q}\right\|_{H^{N+1}(\mathcal{T}_h)} \\ &+ \left\|\eta\left(0\right)\right\|_{L^2(\mathcal{T}_h)} \qquad (4.2.16)\end{aligned}$$

completing the error estimate. Here the energy norm $\left\|\cdot\right\|_{\mathcal{E}}$ is defined by the left-hand-side of (4.2.16). □

4.3 Covolume Filtering

Now I provide theoretical justification for the covolume filtering technique. Since the covolume filtering operator creates additional communication, it will be important to globalize inner products in order to correctly state the new weak form. With this in

mind, I define

$$\begin{aligned}(u,v)_{\mathcal{T}_h} &= \sum_{D^k\in\mathcal{T}_h}(u,v)_{D^k}\\ (u,v)_{\partial\mathcal{T}_h} &= \sum_{D^k\in\mathcal{T}_h}(u,v)_{\partial D^k}\end{aligned}$$

this method relies on a staggered grid Recall that the covolume filtering operator Π_β is defined as

$$\Pi_\beta = (1-\beta)\, I + \beta\Pi^P\Pi^C$$

with I the identity operator and Π^P, Π^C the standard L^2 projections onto the global solution spaces associated with the primal and covolume grids $\mathcal{T}_h^P, \mathcal{T}_h^C$ respectively. The space of piecewise polynomials on either of these sets can be interpreted as a space of piecewise polynomial on the same domain $\Omega_h = \bigcup \mathcal{T}_h$, and so we can use one inner product when referring to either space of functions. The distinction is more important in higher order Sobolev spaces however and so when relevant the norms $\|\cdot\|_{H^p\left(\mathcal{T}_h^P\right)}$, $\|\cdot\|_{H^p\left(\mathcal{T}_h^C\right)}$ will represent the broken Sobolev norms corresponding to the two different spaces respectively. Thus, $\Pi^C u$ for $u \in L^2\left(\Omega_h\right)$ is defined as

$$\left(v, \Pi^C u\right)_{\mathcal{T}_h} = (v,u)_{\mathcal{T}_h}$$

for every piecewise polynomial v on $\mathcal{T}_h^C$, similarly for Π^P we require

$$\left(v, \Pi^P u\right)_{\mathcal{T}_h} = (v,u)_{\mathcal{T}_h}$$

for every piecewise polynomial v on $\mathcal{T}_h^C$. This leads to an important discrete adjoint relationship: If v is a piecewise polynomial on $\mathcal{T}_h^C$ and u is a piecewise polynomial on

$\mathcal{T}_h^P$ then

$$\begin{aligned}\left(v, \Pi^C u\right)_{\mathcal{T}_h} &= (v,u)_{\mathcal{T}_h} \\ \left(\Pi^P v, u\right)_{\mathcal{T}_h} &= (v,u)_{\mathcal{T}_h}\end{aligned}$$

yields

$$\left(v, \Pi^C u\right)_{\mathcal{T}_h} = \left(\Pi^P v, u\right)_{\mathcal{T}_h} \tag{4.3.1}$$

for every $D^k \in \mathcal{T}_h$. The covolume filtering is applied after each right-hand-side evaluation of the semidiscrete form, so in other words, assuming zero source term and given the local weak form (2.3.5):

$$\int_{D^k} \mathbf{v}\left(\frac{\partial \mathbf{Q}_N}{\partial t} + \frac{\partial \mathbf{A}\mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}_N}{\partial y}\right) = \frac{1}{2}\int_{\partial D^k} \mathbf{v}\left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)\left[\mathbf{Q}_N\right]$$

holding for all $\mathbf{v} \in V_N^{k,P}$. A semidiscrete version of a covolume projected method filters the solution $\mathbf{Q}_N$, and is obtained through the following modification

$$\left(\mathbf{v}, \frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{A}\Pi_\beta \mathbf{Q}}{\partial x} + \frac{\partial \mathbf{B}\Pi_\beta \mathbf{Q}}{\partial y}\right)_{\mathcal{T}_h} = \frac{1}{2}\left(\mathbf{v}, \left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)\left[\Pi_\beta \mathbf{Q}\right]\right)_{\partial \mathcal{T}_h} \tag{4.3.2}$$

Theorem 20. *The covolume filtered method (4.3.2) is semidiscrete stable.*

Proof. From the weak form

$$\left(\mathbf{v}, \frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{A}\Pi_\beta \mathbf{Q}}{\partial x} + \frac{\partial \mathbf{B}\Pi_\beta \mathbf{Q}}{\partial y}\right)_{\mathcal{T}_h} = \frac{1}{2}\left(\mathbf{v}, \left(\mathbf{C} - \mathbf{C}^*\mathbf{C}\right)\left[\Pi_\beta \mathbf{Q}\right]\right)_{\partial \mathcal{T}_h}$$

one may test with $\mathbf{v} = \Pi_\beta \mathbf{Q}$ and apply the standard stability argument of theorem

12 to obtain in the periodic case

$$\beta\left(\Pi^P\Pi^C\mathbf{Q},\frac{\partial\mathbf{Q}}{\partial t}\right)_{\mathcal{T}_h}+\frac{1-\beta}{2}\frac{d}{dt}\left\|\mathbf{Q}\right\|^2_{L^2(\mathcal{T}_h)} \;=\; -\frac{1}{2}\sum_{e\in\Gamma}\left\|\mathbf{C}\left[\Pi_\beta\mathbf{Q}\right]\right\|_e$$

and applying the discrete adjoint relationship(4.3.1) yields

$$\begin{aligned}\left(\Pi^P\Pi^C\mathbf{Q},\frac{\partial\mathbf{Q}}{\partial t}\right)_{\mathcal{T}_h} &= \left(\Pi^C\mathbf{Q},\Pi^C\frac{\partial\mathbf{Q}}{\partial t}\right)_{\mathcal{T}_h}\\ &= \frac{1}{2}\frac{d}{dt}\left\|\Pi^C\mathbf{Q}\right\|^2_{L^2(\mathcal{T}_h)}\end{aligned}\tag{4.3.3}$$

combining (4.3.3) and (**??**) gives the following energy equation

$$\beta\frac{d}{dt}\left\|\Pi^C\mathbf{Q}\right\|^2_{L^2(\mathcal{T}_h)}+(1-\beta)\frac{d}{dt}\left\|\mathbf{Q}\right\|^2_{L^2(\mathcal{T}_h)}=-\frac{1}{2}\sum_{e\in\Gamma}\left\|\mathbf{C}\left[\Pi_\beta\mathbf{Q}\right]\right\|_e\tag{4.3.4}$$

analogous to the unmodified energy equation. □

To obtain an error estimate akin to theorem (19) three technical lemmas will be necessary. This lemma gives the bound on the error incurred by the filter Π^β on the interior of the domains, on their boundaries, and for derivatives of filtered solutions. They are effectively tensor product analogues to the one dimensional case in [100]. Note that from here on out the mesh $\mathcal{T}_h$ will be assumed uniform.

Lemma 21. *There exists a constant C independent of h such that for every $u\in H^{N+1}\left(\mathcal{T}_h^P\right)\cap H^{N+1}\left(\mathcal{T}_h^C\right)$ one has*

$$\left\|u-\Pi_\beta u\right\|_{L^2\left(\mathcal{T}_h^P\right)}\leq Ch_k^{N+1}\left(\left\|u\right\|_{H^{N+1}\left(\mathcal{T}_h^P\right)}+\left\|u\right\|_{H^{N+1}\left(\mathcal{T}_h^C\right)}\right)$$

Proof. We have

$$\begin{aligned}\left\|u-(1-\beta)\,u-\beta\Pi^P\Pi^C u\right\|_{L^2\left(\mathcal{T}_h^P\right)} &= \beta\left\|u-\Pi^P\Pi^C u\right\|_{L^2\left(\mathcal{T}_h^P\right)}\\ &= \beta\left\|u-\Pi^P u+\Pi^P u-\Pi^P\Pi^C u\right\|_{L^2\left(\mathcal{T}_h^P\right)}\\ &\leq \beta\left\|u-\Pi^P u\right\|_{L^2\left(\mathcal{T}_h^C\right)}+\beta\left\|u-\Pi^C u\right\|_{L^2\left(\mathcal{T}_h^P\right)}\end{aligned}$$

where I have used the fact that Π^P is a projection. The result now follows from standard L^2 projection estimates. □

Lemma 22. *Suppose that* $u\in H^{N+1}\left(\mathcal{T}_h^P\right)\cap H^{N+1}\left(\mathcal{T}_h^C\right)$. *Then there exists a constant* C *independent of* h *such that*

$$\left\|\frac{\partial}{\partial x}\left(u-\Pi_\beta u\right)\right\|_{L^2\left(\mathcal{T}_h^P\right)}\leq Ch_k^N\left(\|u\|_{H^{N+1}\left(\mathcal{T}_h^P\right)}+\|u\|_{H^{N+1}\left(\mathcal{T}_h^C\right)}\right)$$

Proof. Following similar reasoning as above

$$\begin{aligned}\left\|\frac{\partial}{\partial x}\left(u-\Pi_\beta u\right)\right\|_{L^2\left(\mathcal{T}_h^P\right)} &= \beta\left\|\frac{\partial}{\partial x}\left(u-\Pi^P u\right)+\frac{\partial}{\partial x}\left(\Pi^P u-\Pi^P\Pi^C u\right)\right\|_{L^2\left(\mathcal{T}_h^P\right)}\\ &\leq \beta\left\|\frac{\partial}{\partial x}\left(u-\Pi^P u\right)\right\|_{L^2\left(\mathcal{T}_h^P\right)}+\beta\frac{C}{h}\left\|\Pi^P u-\Pi^P\Pi^C u\right\|_{L^2\left(\mathcal{T}_h^P\right)}\end{aligned}$$

where in the last step I used the Markov inequality for the finite dimensional solution space. The conclusion follows from classical estimates and the fact that Π^P is a projection. □

Lemma 23. *There exists a constant* C *independent of* h *such that for all* $u\in H^{N+1}\left(\mathcal{T}_h^P\right)\cap H^{N+1}\left(\mathcal{T}_h^C\right)$ *the following holds:*

$$\sum_{D^k \in \mathcal{T}_h^P} \|u - \Pi_\beta u\|_{L^2(\partial D^k)} \leq C h_k^{2N+1} \left(\|u\|^2_{H^{N+1}(\mathcal{T}_h^P)} + \|u\|^2_{H^{N+1}(\mathcal{T}_h^C)} \right)$$

Proof. Once again

$$\begin{aligned} \sum_{D^k \in \mathcal{T}_h^P} \|u - \Pi_\beta u\|^2_{L^2(\partial D^k)} &\leq \sum_{D^k \in \mathcal{T}_h^P} \beta \left\|u - \Pi^P u\right\|^2_{L^2(\partial D^k)} + \beta \left\|\Pi^P u - \Pi^P \Pi^C u\right\|^2_{L^2(\partial D^k)} \\ &\leq \sum_{D^k \in \mathcal{T}_h^P} \beta \left\|u - \Pi^P u\right\|^2_{L^2(\partial D^k)} + \frac{\beta}{\sqrt{h}} \left\|u - \Pi^C u\right\|^2_{L^2(D^k)} \end{aligned}$$

where I have applied an inverse trace inequality for the finite dimensional solution space, and the estimate follows from standard projection estimates and the multiplicative trace inequality. □

I now prove the main theorem in this section, an error estimate for covolume filtered DG.

Theorem 24. *There exists a constant C such that if $\mathbf{Q}_N$ is the covolume filtered solution and $\mathbf{Q} \in H^{N+1}\left(\mathcal{T}_h^P\right) \cap H^{N+1}\left(\mathcal{T}_h^C\right)$ is continuous at element bisectors of the primal and covolume meshes, and satisfies the continuous weak form, then*

$$\|\mathbf{Q}_N - \mathbf{Q}\|_{\mathcal{E}} \leq C h^N \|\mathbf{Q}\|_{H^{N+1}(\mathcal{T}_h^P)}.$$

where $\|\cdot\|_{\mathcal{E}}$ is a suitably defined energy norm.

Proof. Suppose that $\mathbf{Q} \in H^{N+1}\left(\mathcal{T}_h^P\right) \cap H^{N+1}\left(\mathcal{T}_h^C\right)$ satisfies the weak form (2.1.2) in $H^1(\mathcal{T}_h)$, i.e. for each D^k we have

$$\begin{aligned} \left(\mathbf{v}, \frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{A}\mathbf{Q}}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}}{\partial y}\right)_{D^k} &= (\mathbf{v}, (\mathbf{C} - \mathbf{C}^* \mathbf{C})[\mathbf{Q}])_{\partial D^k} \\ &= 0 \end{aligned}$$

for all $\mathbf{v} \in H^1(\mathcal{T}_h)^3$, which means that the L^2 projection satisfies

$$\left(\mathbf{v}, \frac{\partial \Pi^P \mathbf{Q}}{\partial t} + \Pi^P \left(\frac{\partial \mathbf{A}\mathbf{Q}}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}}{\partial y}\right)\right)_{D^k} = 0 \tag{4.3.5}$$

for each D^k and polynomial $\mathbf{v}$. Now suppose that T_h is polynomial on each D^k such that

$$\left(\mathbf{v}, \frac{\partial \Pi^P \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{A}\Pi_\beta \Pi^P \mathbf{Q}}{\partial x} + \frac{\partial \mathbf{B}\Pi_\beta \Pi^P \mathbf{Q}}{\partial y}\right)_{D^k} - \left(\mathbf{v}, (\mathbf{C} - \mathbf{C}^*\mathbf{C})\left[\Pi_\beta \Pi^P \mathbf{Q}\right]\right)_{\partial D^k} = (\mathbf{v}, T_h)_{D^k} \tag{4.3.6}$$

holds for each D^k and polynomial $\mathbf{v}$. Subtracting (4.3.5) from (4.3.6), summing over all elements and testing with $\mathbf{v} = T_h$ yields

$$\begin{aligned} \|T_h\|^2_{L^2(\mathcal{T}_h)} \;\leq\; & \|T_h\|_{L^2(\mathcal{T}_h)} \cdot \left\| \frac{\partial \mathbf{A}\Pi_\beta \Pi^P \mathbf{Q}}{\partial x} + \frac{\partial \mathbf{B}\Pi_\beta \Pi^P \mathbf{Q}}{\partial y} - \left(\frac{\partial \mathbf{A}\mathbf{Q}}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}}{\partial y}\right) \right\|_{L^2(\mathcal{T}_h)} \\ & + \sum_{D^k \in \mathcal{T}_h} \|T_h\|_{\partial D^k} \left\| (\mathbf{C} - \mathbf{C}^*\mathbf{C})\left[\Pi_\beta \Pi^P \mathbf{Q}\right] \right\|_{\partial D^k} \end{aligned}$$

From lemma 22 the first term is bounded by

$$\left\| \frac{\partial \mathbf{A}\Pi_\beta \Pi^P \mathbf{Q}}{\partial x} + \frac{\partial \mathbf{B}\Pi_\beta \Pi^P \mathbf{Q}}{\partial y} - \left(\frac{\partial \mathbf{A}\mathbf{Q}}{\partial x} + \frac{\partial \mathbf{B}\mathbf{Q}}{\partial y}\right) \right\|_{L^2(\mathcal{T}_h)} \leq C h^N \|\mathbf{Q}\|_{H^{N+1}(\mathcal{T}_h)}$$

and utilizing continuity of $\mathbf{Q}$ at bisectors the second term satisfies

$$\begin{aligned} & \sum_{D^k \in \mathcal{T}_h} \|T_h\|_{\partial D^k} \left\| (\mathbf{C} - \mathbf{C}^*\mathbf{C})\left[\Pi_\beta \Pi^P \mathbf{Q}\right] \right\|_{\partial D^k} \\ = \; & \sum_{D^k \in \mathcal{T}_h} \|T_h\|_{\partial D^k} \left\| (\mathbf{C} - \mathbf{C}^*\mathbf{C})\left[\Pi_\beta \Pi^P \mathbf{Q} - \mathbf{Q}\right] \right\|_{\partial D^k} \\ \leq \; & \sqrt{\sum_{D^k \in \mathcal{T}_h} \|T_h\|^2_{\partial D^k}} \sqrt{\sum_{D^k \in \mathcal{T}_h} \|(\mathbf{C} - \mathbf{C}^*\mathbf{C})\left[\Pi_\beta \Pi^P \mathbf{Q} - \mathbf{Q}\right]\|_{\partial D^k}} \end{aligned} \tag{4.3.7}$$

by the discrete Cauchy-Schwarz inequality. The polynomial trace inequality tells us that the first factor in (4.3.7) satisfies

$$\begin{aligned}\sqrt{\sum_{D^k\in\mathcal{T}_h}\|T_h\|^2_{\partial D^k}} &\leq \sqrt{\sum_{D^k\in\mathcal{T}_h}\frac{C}{h}\|T_h\|^2_{D^k}}\\ &\leq \frac{C^{1/2}}{h^{1/2}}\|T_h\|_{L^2(\mathcal{T}_h)}\end{aligned}$$

and the second factor in (4.3.7) may be bounded by separating out traces and applying theorem (4), projection estimates yields the final truncation estimate

$$\|T_h\|_{L^2(\mathcal{T}_h)} \leq Ch^N \|\mathbf{Q}\|_{H^{N+1}(\mathcal{T}_h)}.$$

Therefore considering the numerical solution $\mathbf{Q}_N$ to the filtered weak form (4.3.2), which satisfies

$$\left(\mathbf{v}, \frac{\partial \mathbf{Q}_N}{\partial t} + \frac{\partial \mathbf{A}\Pi_\beta \mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\Pi_\beta \mathbf{Q}_N}{\partial y}\right)_{D^k} = \left(\mathbf{v}, \left(\mathbf{C}-\mathbf{C}^*\mathbf{C}\right)\left[\Pi_\beta \mathbf{Q}_N\right]\right)_{\partial D^k}$$

for all $D^k \in \mathcal{T}_h$ and $\mathbf{v}$ will also satisfy a local error equation

$$\begin{aligned}&\left(\mathbf{v}, \frac{\partial \Pi^P \mathbf{Q}}{\partial t} - \frac{\partial \mathbf{Q}_N}{\partial t}\right)_{D^k}\\ =\ &-\left(\mathbf{v}, \frac{\partial \mathbf{A}\Pi_\beta \mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\Pi_\beta \mathbf{Q}_N}{\partial y} - \frac{\partial \mathbf{A}\Pi_\beta \Pi^P \mathbf{Q}}{\partial x} - \frac{\partial \mathbf{B}\Pi_\beta \Pi^P \mathbf{Q}}{\partial y}\right)_{D^k}\\ &-\left(\mathbf{v}, T_h\right)_{D^k} - \left(\mathbf{v}, \left(\mathbf{C}-\mathbf{C}^*\mathbf{C}\right)\left[\Pi_\beta \mathbf{Q}_N - \Pi_\beta \Pi^P \mathbf{Q}\right]\right)_{\partial D^k}\end{aligned} \tag{4.3.8}$$

and observe that by the definition of Π_β the inner product on the left-hand-side satisfies

$$\left(\mathbf{v}, \frac{\partial \Pi^P \mathbf{Q}}{\partial t} - \frac{\partial \mathbf{Q}_N}{\partial t}\right)_{D^k} = \left(\mathbf{v}, \Pi_\beta\left(\frac{\partial \Pi^P \mathbf{Q}}{\partial t} - \frac{\partial \mathbf{Q}_N}{\partial t}\right)\right)_{D^k}$$

focusing on the second term of (4.3.8) one sees that

$$\begin{aligned} & -\left(\mathbf{v}, \frac{\partial \mathbf{A}\Pi_\beta \mathbf{Q}_N}{\partial x} + \frac{\partial \mathbf{B}\Pi_\beta \mathbf{Q}_N}{\partial y} - \frac{\partial \mathbf{A}\Pi_\beta \Pi^P \mathbf{Q}}{\partial x} - \frac{\partial \mathbf{B}\Pi_\beta \Pi^P \mathbf{Q}}{\partial y}\right)_{D^k} \\ = & -\left(\mathbf{v}, \frac{\partial \mathbf{A}\Pi_\beta \left(\mathbf{Q}_N - \Pi^P \mathbf{Q}\right)}{\partial x} + \frac{\partial \mathbf{B}\Pi_\beta \left(\mathbf{Q}_N - \Pi^P \mathbf{Q}\right)}{\partial y}\right)_{D^k} \end{aligned}$$

we may therefore apply stability of the covolume filtered operator (energy equation (4.3.4)) to get

$$\begin{aligned} & \beta \frac{d}{dt} \left\| \Pi^C \left(\mathbf{Q}_N - \Pi^P \mathbf{Q}\right) \right\|^2_{L^2(\mathcal{T}_h)} + (1-\beta) \frac{d}{dt} \left\| \mathbf{Q}_N - \Pi^P \mathbf{Q} \right\|^2_{L^2(\mathcal{T}_h)} \\ & + \frac{1}{2} \sum_{\Gamma^e} \left\| \mathbf{C} \left[\Pi_\beta \left(\mathbf{Q}_N - \Pi^P \mathbf{Q}\right) \right] \right\|_e \\ \leq & \left\| T_h \right\|_{L^2(\mathcal{T}_h)} \cdot \left\| \Pi_\beta \left(\mathbf{Q}_N - \Pi^P \mathbf{Q}\right) \right\|_{L^2(\mathcal{T}_h)} \\ \leq & \left\| T_h \right\|_{L^2(\mathcal{T}_h)} \left(\beta \left\| \Pi^C \left(\mathbf{Q}_N - \Pi^P \mathbf{Q}\right) \right\|_{L^2(\mathcal{T}_h)} + (1-\beta) \left\| \mathbf{Q}_N - \Pi^P \mathbf{Q} \right\|_{L^2(\mathcal{T}_h)} \right) \end{aligned}$$

the final result following from an application of Gronwall's inequlity, as done in the unmodified case. Here however the use of the Markov inequality and the trace inequality in the estimation of the truncation T_h only gives us order n, instead of $n + 1/2$. $\square$

Chapter 5

Time-stepping for DG

In chapter 3 I investigated ways to directly manipulate the spatial discretization so as to lower the CFL restriction. An additional strategy not yet attempted is to find time-steppers which are most appropriate for the type of physics to be solved, ideally one custom chosen to the problem at hand should give a more efficient method. There are conflicting goals in time-stepping however, and so there is no proper "best" choice for this reason, but rather a list of choices which balance these many traits. One helpful characteristic for example would be a time-stepper which minimizes the number of right-hand-size evaluations (matrix-vector multiplications, in this case) that it takes to fully integrate a system to a desired final time. This goal may however be at odds with memory limitations, or it might even cause one to construct a numerically unstable time-stepper [49, 76].

The explicit time-steppers which require the least amount of function evaluations per timestep are the Adams-Bashforth linear multistep methods, but they each require a history of previous time derivative values of the solution state, increasing memory needs. Instead of requiring previous time derivative values of the solution, an alternative is to produce the needed values on the fly. This is what one-step methods such as Runge-Kutta methods do. An additional benefit of Runge-Kutta methods is they are easily adapted to the problem at hand, leading to many optimized schemes which minimize the right-hand-side evaluation requirement by increasing the maximum stable timestep. These optimized Runge-Kutta methods can be further constrained to

fulfill a memory saving from which requires at any step to store only two full solution states [56]. It will be these low-storage versions of Runge-Kutta methods which are investigated in this thesis, with the exception of classical RK4 which is used as a baseline.

In this chapter I will present the basic theory of explicit one-step timesteppers and explain how it permits their optimization and low-storage varieties. Then five optimized timesteppers from the literature which are specially constructed for the upwind DG operator will be compared to classical RK4 and Adams-Bashforth methods. All numerical experiments in this chapter focus on the two dimensional acoustic wave equation with a point source, but the timestepper theory itself will deal with a generic linear system of ODEs.

Here I will consider numerical results for unmodified DG, flux filtered DG, and covolume filtered DG, so as to see what the total gain one might expect in a high order simulation. Mapped methods are excluded due to their order reduction which occurs for parameter values that yield a nonnegligible timestep improvement. The optimal RK methods by themselves are seen to yield around a 50% improvement over classical RK4 in the unmodified case, but as one further modifies the method this improvement becomes as high as 80%.

While the time-steppers here are not new, and have been applied to discontinuous Galerkin in the past, this chapter shows how many of them can be still a significant improvement over classical time-steppers when applied to operator-modified DG. The numerical dissipation of DG makes the question of what sort of stability region will be optimal for your problem a little more difficult since it is not constrained by a simple geometric shape. I show here that even with the altered spectra produced by modification, the timesteppers which were optimal for unmodified DG still yield

considerable gain for modified DG, notably covolume filtered DG.

5.1 Optimal Low-Storage Runge-Kutta Methods

Given a system of ordinary differential equations $\mathbf{q}' = \mathbf{f}(t, \mathbf{q})$, explicit Runge-Kutta methods of order p are derived by forming the Taylor expansion of $\mathbf{y}$ about the timestep Δt and then forming a discrete recurrence involving evaluations of $\mathbf{f}$ which matches this Taylor expansion up to the p-th term. A special subset of these recurrences are those which may be evaluated using only $2N$ storage, given that $\mathbf{q}(t) \in \mathbb{R}^N$. These low storage varieties of Runge-Kutta methods will be those that are explored here. The general form of a low storage Runge-Kutta method used here will be the same used in Toulorge and Desmet [95] and Niegemann et al. [73]. It is given as

$$\begin{aligned} \tilde{\mathbf{q}}^i &= A_i \tilde{\mathbf{q}}^{i-1} + \Delta t \mathbf{f}\left(t_n + c_i \Delta t, \mathbf{q}^{i-1}\right) \\ \mathbf{q}^i &= \mathbf{q}^{i-1} + B_i \tilde{\mathbf{q}}^i \end{aligned}$$

for $i = 1, \ldots, s-1$ with s the number of stages associated with the Runge-Kutta method and the coefficients A_i, B_i, c_i taken from tabulations of precomputed values arrived at through a specified optimization procedure. This is implemented in the general low-storage Runge-Kutta code *LSRK.m* (see chapter 7, listing 7.6)

To analyze the stability region of this method, observe that in being a Runge-Kutta method it sends a linear ODE system $\mathbf{q}' = \mathbf{f}(\mathbf{t}, \mathbf{q}) = \mathbf{H}\mathbf{q}$ to a truncated Taylor expansion of the exponential function, i.e. a p-th order s stage Runge-Kutta method will yield

$$\mathbf{q}(\Delta t) \approx \sum_{j=0}^{p} \frac{\Delta t^j \mathbf{H}^j}{j!} \mathbf{q}(0) + \sum_{j=p+1}^{s} \Delta t^j \eta^j \mathbf{H}^j \mathbf{q}(0) = \exp(\Delta t \mathbf{H}) \mathbf{q}(0) + O(\Delta t^p).$$

Observe that this is nothing more than a polynomial of the matrix $\mathbf{H}$, i.e. one may more concisely write

$$\mathbf{q}(\Delta t) \approx R_{p,s}(\Delta t \mathbf{H})\,\mathbf{q}(0)$$

with $R_{p,s}$ a polynomial of degree s. Repeated application of the Runge-Kutta method means repeated application of the operator $R_{p,s}(\Delta t\mathbf{H})$ to $\mathbf{q}(0)$, so that to arrive at a desired time $M\Delta t$ one obtains

$$\mathbf{q}(M\Delta t) \approx [R_{p,s}(\Delta t \mathbf{H})]^M \mathbf{q}(0).$$

A well known fact of matrix analysis states that this system is unbounded in M only if the spectral radius of $R_{p,s}(\Delta t\mathbf{H})$ exceeds or possibly equals 1. Thus a condition for stability would be to choose Δt such that

$$\rho(R_{p,s}(\Delta t \mathbf{H})) \leq 1,$$

or equivalently to find Δt such that the function $|R_{p,s}|$ does not exceed 1 on the spectrum of $\Delta t\mathbf{H}$.

The exact timesteppers used have their coefficients A_i, b_i, c_i tabulated in chapter 7, tables 7.6 through 7.10 which can then be provided to LSRK.m to integrate a general system of ordinary differential equations. The methods will be those of Toulorge and Desmet, using the same names *(RK84,RKC84,RKC73)* the naming scheme indicating by the first number the number of stages and by the second number the order. Furthermore these will be compared to higher stage timesteppers constructed in Niegemann, et al. which were designed for schemes involving numerical dissipation (as in the case of upwinding used in this thesis). These will be compared by means

of their required function evaluations to integrate the upwind DG discretization to the two dimensional acoustic wave equation. Accuracy at designated times (using maximum possible timestep) will then be considered by comparison to the output of MATLAB's *ODE45* at its most stringent error tolerance. It should be noted that the differences should not be expected to be great among the Runge-Kutta methods, as stability considerations dominate accuracy for the choice of the timestep; the spatial discretization contributes the most to error in the simulation. However the accumulation of error in time for Runge-Kutta methods and Adams-Bashforth methods could be different, considering their different origins.

5.2 Adams-Bashforth Methods

Adams Bashforth methods have the virtue of only requiring one function evaluation per step, but the tradeoff is necessity to store additional derivatives of previous solution states. The construction of these methods works differently than Runge-Kutta methods in that instead of devising a recurrence to match a truncated Taylor expansion to the exponential operator, a polynomial interpolant is constructed of previous time derivatives and an appropriate quadrature rule is applied in time to determine the next solution state. As a recurrence, an Adams-Bashforth recurrence with s previous derivatives stored takes the form

$$\mathbf{q}^{i+s+1} = \mathbf{q}^{i+s} + \Delta t \sum_{k=1}^{s} b_k \mathbf{f}\left(k\Delta t, \mathbf{q}^{i+k}\right)$$

where by taking a constant Δt requires one only to evaluate $\mathbf{f}\left(s\Delta t, \mathbf{q}^{i+s}\right)$ for each solution at time indexed by i.

There is little opportunity for optimizing the coefficients b_k in the above recur-

rence, as their choice is effectively uniquely determined by order and stability constraints. Thus only classical Adams-Bashforth methods are investigated here, with precomputed values for b_k. These precomputed values are tabulated in chapter 7 for reference. Furthermore, taking $s = 1$ yields Forward Euler which is known to be unstable for hyperbolic problems discretized by upwind DG, similarly for $s = 2$. Thus only Adams-Bashforth of orders $3, 4, 5$ are considered in this thesis. The general code for Adams-Bashforth methods in MATLAB is given by the file ABM.m (see chapter 7 listing 7.7)

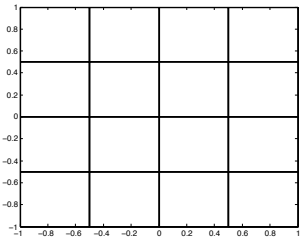
The stability analysis of Adams-Bashforth methods for systems of ordinary differential equations is a little more complicated than for Runge-Kutta methods, as its stability depends on how one "starts" the method by computing the initial derivative values. This can be done with a Runge-Kutta method with the same timestep, but in this thesis I chose to use the adaptive MATLAB timestepper ode45 with a stringent error tolerance, so as to minimize the impact of the starting scheme on the validity of results given later. How stable timesteps were computed is discussed in the numerical results section, the strategy follows the Runge-Kutta strategy closely.

Numerical results will follow the same strategy as for Runge-Kutta methods, to facilitate their ultimate comparison. It will be seen that for $s = 3$, Adams-Bashforth yields a competitive integration strategy, but $s = 4, 5$ yield very strict stability constraints on Δt, so that even by only performing one function evaluation per step they end up performing far more steps to yield the same final time result as the low storage Runge-Kutta methods.

5.3 Numerical Results

The problem under consideration here is the solution of the two dimensional wave equation 2.3.1 on page 23 on a square domain with periodic boundary conditions.

Figure 5.3.1 : Discretized Domain



The material parameters are taken to be constantly $\rho = 1, \kappa = 1$. The stable timesteps are calculated by constructing directly for Runge-Kutta methods the companion matrix

$$H(\Delta t) \equiv R_{p,s}(\Delta t \mathbf{H})$$

and for Adams-Bashforth methods the companion matrix is taken to be the matrix resulting from applying Adams-Bashforth with initial condition being the identity matrix, and time-stepping forward until the dynamic behavior of the Adams-Bashforth recurrence dominates the behavior given by the manner in which the initial derivatives are computed (if the initial states are computed to stringent tolerance as in this thesis, a stability analysis shows the method to be almost $A-$stable if only one Adams-Bashforth step is taken, this does not however represent its full dynamic behavior for long time integration). Next the timestep is found by solving the constrained maxi-

mization problem

$$\begin{aligned} \max_{\Delta t} \quad & \Delta t \\ \text{s.t.} \quad & \rho\left(H\left(\Delta t\right)\right) \leq 1 \end{aligned}$$

using MATLAB's fmincon (minimizing instead the function $-\Delta t$). In what follows below the timesteppers will be given the brief names:

RK4	Classical RK4
RKC73	Toulorge + Desmet 7th order 3 stage
RKC84	Toulorge + Desmet 8th order 4 stage
RKF84	Toulorge + Desmet 8th order 4 stage
NRK13E	Niegemann et al. Elliptical stability region
NRK14C	Niegemann et al. Circular stability region
AB3	Third order Adams-Bashforth
AB4	Fourth order Adams-Bashforth
AB5	Fifth order Adams-Bashforth

Table 5.1 : Naming Schemes for Timesteppers

First I provide a qualitative idea of what one should expect from the methods by giving their stability regions (scaled by number of function evaluations per timestep). I provide here stability regions for the different timesteppers under consideration, all superimposed on the RK4 stability region for comparison purposes.

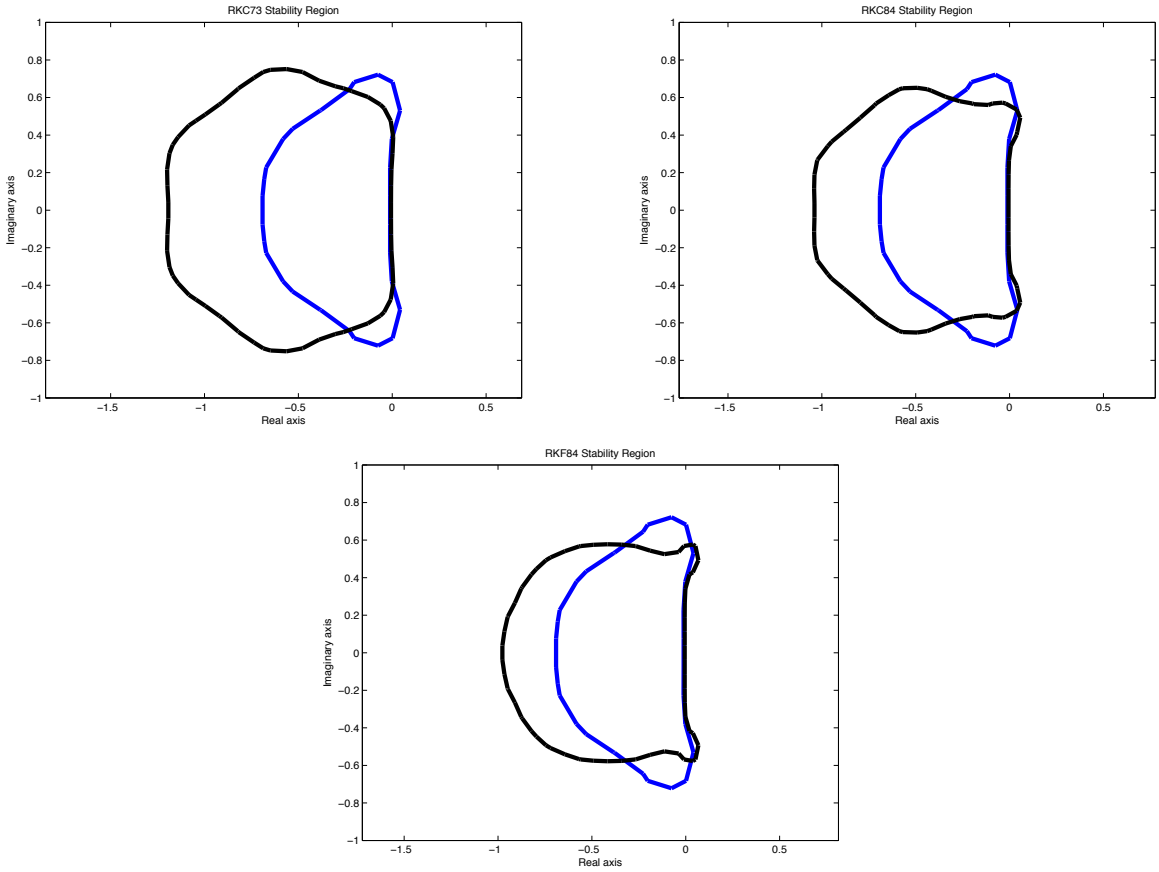


Figure 5.3.2 : RKC73,RKC84,RKF84 stability regions (black) scaled by number of stages

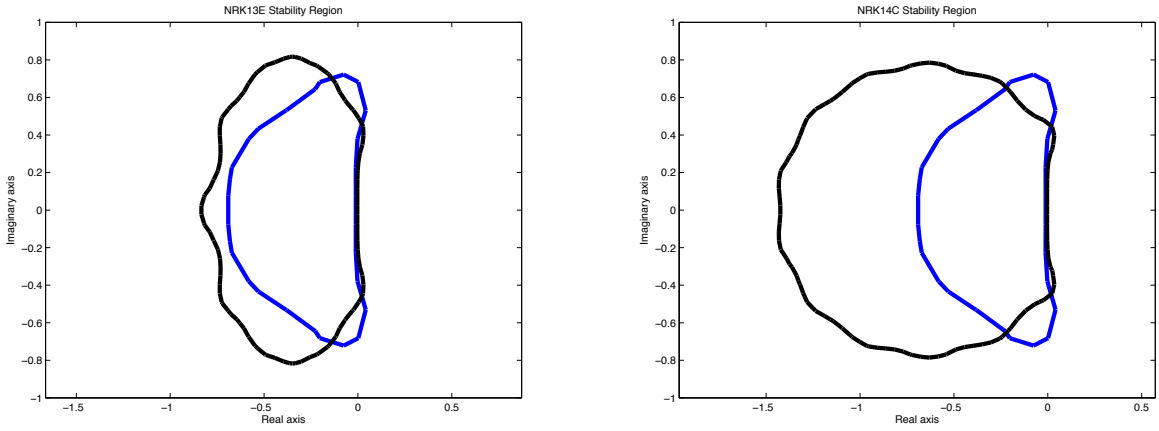


Figure 5.3.3 : NRK13E,NRK14C stability regions (black) scaled by number of stages.

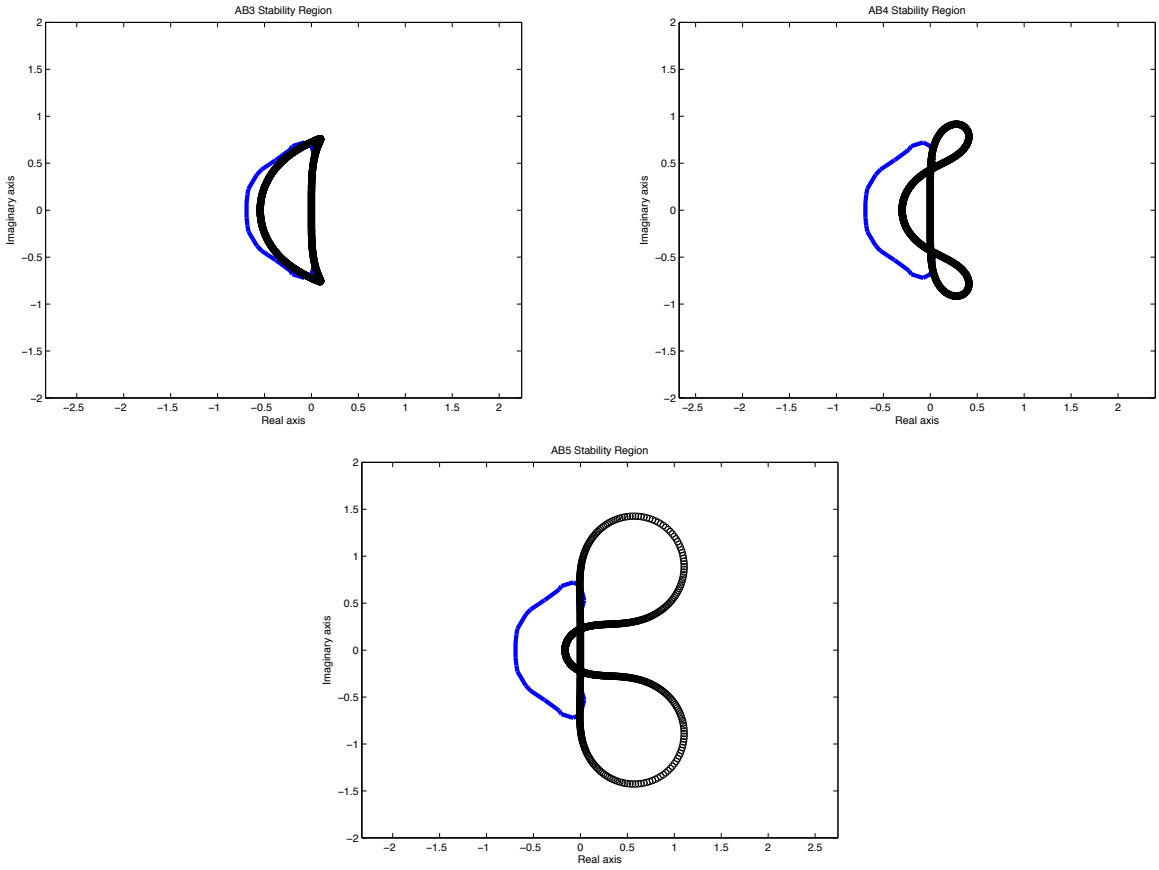


Figure 5.3.4 : AB3,AB4,AB5 stability regions (black) superimposed on RK4 (blue)

Now I consider the efficiency of each timestepper by calculating the number of function evaluations required to reach a final time of $t = 5$, and calculating respective L^2 errors for time $t = 5$. The result is summarized in the below table. Note that while I report the errors, this is only to illustrate that they are roughly all same order of magnitude, and so not a relevant factor in the choice of a proper timestepper. Computations here were done with order $N = 8$.

	#RHS	L^2 Error	Relative Cost Compared to RK4
NRK14C	5040	$0.4 \cdot 10^{-4}$	0.57
RKC73	5530	$0.1 \cdot 10^{-3}$	0.59
RKC84	6360	$0.9 \cdot 10^{-5}$	0.68
RKF84	6720	$0.7 \cdot 10^{-5}$	0.71
NRK13E	7930	$0.9 \cdot 10^{-5}$	0.84
RK4	9500	$0.6 \cdot 10^{-5}$	1.0
AB3	10665	$0.4 \cdot 10^{-4}$	–
AB4	32745	$0.8 \cdot 10^{-5}$	–
AB5	41010	$0.7 \cdot 10^{-5}$	–

Table 5.2 : Timestepers ordered in terms of RHS evaluations

Considering the memory overhead of Adams-Bashforth type methods, the optimal Runge-Kutta methods are the higher performing of those methods considered here, with a particularly high performing Runge-Kutta method by Niegemann et al. [73]. The relatively modest increase in RHS evaluations for AB3 however might be overcome through its viability in local timestepping situations where there are significant spatial scale differences that can be exploited to ensure that physics on larger elements is not needlessly overresolved [67, 89, 94, 37].

Next I consider only the optimized timesteppers, applied with flux filtering with the parameter chosen as aggressivley as possible but maintaining roughly the same order of magnitude error as in the unmodified case. Here that was $\delta = 0.5$.

	#RHS	Relative Cost Compared to Unmodified + RK4
NRK14C	3640	0.38
RKC73	4760	0.50
RKC84	5600	0.58
RKF84	5760	0.59
NRK13E	6500	0.68

Table 5.3 : Timestepers with flux filtering ordered in terms of RHS evaluations

so that one sees a 12% increase in efficiency for the highest performing timestepper. Finally I present the same list for covolume filtering, with the parameter β chosen in the same way. Here that was $\beta = 0.8$.

	#RHS	Relative Cost Comapred to Unmodified+RK4
NRK14C	1950	0.21
RKC73	2800	0.29
RKC84	3120	0.33
RKF84	4704	0.50
NRK13E	5070	0.53

Table 5.4 : Timestepers with covolume filtering ordered in terms of RHS evaluations

5.4 Conclusions

The optimal timesteppers used here yielded a reasonable gain in efficiency even in the case of unmodified DG. These methods have not been applied to modified DG methods before this thesis, and in the case of modifications even further gains were seen. One point to note is that although the modified methods mostly serve to reduce the impact of numerical dissipation on stiffness, the methods more highly optimized for less dissipative methods (NRK13E, for example) still did not perform as well as those more optimized for numerical dissipation, even in the case of flux filtering or covolume filtering. This can be attributed to the fact that in the limit as these methods practically eliminate dissipation they also create an unfavorable spatial error, and the parameters were chosen here so as to not impact spatial error significantly. Flux filtering is the easier to implement over covolume filtering, and the gains to be had from it are accordingly modest. This might still be a reasonable strategy to try should code be taking a very long time to run (on the order of days, for example). If code is taking even longer to run, say weeks, then the additional

effort in implementing a covolume filter might be worthwhile in reducing the total time of simulation without severely impacting solution quality.

Chapter 6

Conclusions and Future Work

Even though DG can suffer from many of the difficulties that other high order methods do, this thesis shows that at least in principle this is not a necessity. DG has many redeeming virtues which make its application to hyperbolic problems very natural. Like traditional continuous finite element methods (CFEM), the variational form makes possible error estimates based on a standard decomposition of total error into local and truncation errors. Unlike CFEM however DG permits a natural decoupling of elements for a compact scheme which is easily scaled on parallel architectures [2].

This thesis had two goals in addition to improving the timestep restriction, and I believe they have been achieved for the problem considered. The first goal was to retain the qualities of DG which make it a good candidate for solving hyperbolic problems, and the second goal was to avoid significant changes to an existing code-base. The first goal was shown to be achieved through extensive theory and numerical results in the operator modification chapter. Much of the theory in that chapter is new, particularly the mapping techniques, which have never been applied to discontinuous Galerkin before, but also the flux filtering and covolume filtering techniques, which have never been applied in two dimensions before. The second goal I believe also was satisfied, the techniques applied did not require a significant overhaul of an existing DG code. The filtering codes just required the addition of an alternate filtering routine which could be added directly to the right-hand-side evaluation without any further change, and the mapping codes just required that one evaluate the

mass matrix adaptively. The timesteppers presented automatically satisfy both goals since the method of lines makes DG compatible with any explicit timesteppers, that routine can be changed at will with very little change to surrounding DG code.

In terms of efficiency, the flux filtering and covolume filtering techniques clearly win over mapping for the polynomial orders considered in this thesis. Between flux filtering and covolume filtering however there are other more intangible factors to consider if one is to compare them. While covolume filtering had the strongest impact on timestep reduction, it relied on the tensor product structure of the mesh and furthermore can currently only handle simpler boundary conditions. Its extension to arbitrary boundary conditions or to unstructured meshes is not straightforward, and could be the topic of future research. Flux filtering did not have the same impact on timesteps that covolume filtering had, but its ease of implementation and natural incorporation of boundary conditions could be an important factor in whether one chooses to apply it to their problem. It generally only doubled the timestep before impacting accuracy of the solution, but it is a mesh and boundary condition independent method, meaning it has a wider range of possible applications than covolume filtering currently does.

The optimal timestepping methods also provide a useful way to improve efficiency without significantly changing existing code. One point to make here is that these timesteppers were generally intended for nonlinear problems, as the papers in which they are constructed enforce a nonlinear set of equations to ensure that formal order of convergence would be retained even in a nonlinear simulation. For the linear problem however one may observe that applying a timestepper is equivalent to evaluating the action of a matrix polynomial on a given starting vector. This makes possible the application of now well-researched polynomial methods for matrices, and

the order conditions for the linear problem can be much more easily enforced without imposing the standard nonlinear constraints. This is a direction that could be useful in combination with operator modification. The precomputed optimal Runge-Kutta methods did a good job capturing the spectrum of the modified methods, but in more complicated situations it may be more difficult to anticipate which method to use without costly trial-and-error. A possible future direction here would be to use some kind of modified Arnoldi factorization (e.g. using the energy inner product instead of the discrete l^2 inner product) with the desired truncation term as the starting vector to automatically enforce the truncation error, but using acquired Krylov information to optimize the next few terms to use. This would have some similarities with exponential timestepping [79, 88, 48]. It has already been seen that optimized timesteppers can yield a considerable gain, but further automating their selection could prove valuable and potentially yield even more efficient simulations.

The overall goal of this thesis has been met. Explicit timestepping has been used in a way that is significantly more efficient than before, the techniques have been applied for the first time in two dimensions, and where theory did not exist it has been included. There is possibility of further gain, and from the initial results contained in this thesis it seems that could be a valuable direction to pursue future research.

Chapter 7

Implementation

In this chapter I show how one can implement some of the ideas presented in this thesis in MATLAB code. The mesh is assumed uniform, and so the reference mapping becomes affine and the local solution spaces V_N^k all become polynomial spaces. This eases the implementation considerably since the coordinate transformed Legendre basis in local coordinates on D^k retain their orthogonality property. This orthogonality property translates to easily inverted mass matrices and a lower storage requirement. While in principle one could extend the ideas here to at least a perturbed uniform mesh, that was not pursued in this thesis. Covolume filtering still currently requires a structured grid, and mapping techniques already have performance issues without the added difficulties of non-affine elements.

For reference below is a table providing some of the primitive information that is known before any sort of construction is performed:

Variable	Value	Description
N_vQ	Mesh input	Number of vertices
KQ	Mesh input	Number of elements
K_b	Mesh input	Number of boundary elements
NQ	User input	Polynomial order
N_pQ	$(NQ+2)^2$	Nodes per element
$N_{fp}Q$	$4(NQ+1)$	Boundary DOFs per element
N_bQ	Mesh dependent	Total nodes on boundary

Table 7.1 : Indexing primitves

Note that by removing the appended Q to the variables, the result is the same associated variable for a one dimensional simulation. This will be important as many two dimensional operators can be constructed through tensor products of one dimensional operators.

7.1 Nodal Element

When implementing DG on a general mesh, most operations can be precomputed on a given reference element and then mapped to a general element through a coordinate transformation. In the case of quadrilaterals used for this thesis the mapping is quadratic in the reference coordinates, but if the mesh is assumed to be uniform it reduces to a affine transformation [42, 102].

Taking the reference element to be the biunit square $I = [-1, 1]^2$ the mapping to an element D^k of a uniform mesh becomes

$$\Phi_k(\mathbf{x}) = \mathbf{G}_k\mathbf{x} + \mathbf{r}_k$$

Where the geometric factors matrix $\mathbf{G}_k$ introduced in chapter 2 reduces here to a linear function, and $\mathbf{r}_k$ is the barycenter of the quadrilateral.

7.2 Index maps

In order to facilitate communication between the boundary degrees of freedom on each element, a global index set is given for each node and index maps provide a convenient way to directly access those global indices. A summary is given below for each map and what job it performs in the code.

Name	Dimensions	Description
mapPQ	$N_{fp}Q \cdot KQ \times 1$	Elemental boundary nodes to +trace map
mapMQ	$N_{fp}Q \cdot KQ \times 1$	Elemental boundary nodes to −trace map
vmapPQ	$N_{fp}Q \cdot KQ \times 1$	Full node to +trace map
vmapMQ	$N_{fp}Q \cdot KQ \times 1$	Full node to −trace map
mapBQ	$N_bQ \cdot K_b \times 1$	Elemental boundary to mesh boundary map

Table 7.2 : Index maps

7.3 Polynomial Basis

In the code I freely move between two types of basis: modal, and nodal. A nodal basis simply takes a Lagrange type basis for the local solution spaces V_N^k to ensure that the expansion coefficients correspond to solution values at specified nodal points. A modal basis is one which does not have this property, and is often an orthogonal or nearly-orthogonal basis (i.e. Legendre polynomials.) The input and output format of the code is nodal, that is the solution values at the predetermined $KQ \cdot N_pQ$ nodal points, (the Lagrange basis). However, most computation is actually done in a modal basis, that is an orthogonal basis on the reference element I. This basis is formed on a general quadrilateral by forming a tensor product of the Legendre polynomial basis in one dimension. The normalized polynomials can be computed with the code JacobiP.m taking $\alpha = \beta = 0$:

Listing 7.1: JacobiP.m

```
1 function [P] = JacobiP(x,alpha,beta,N);
2 % function [P] = JacobiP(x,alpha,beta,N)
3 % Purpose: Evaluate Jacobi Polynomial of type (alpha,beta) > -1
4 %          (alpha+beta <> -1) at points x for order N and
     returns P[1:length(xp))]
5 % Note   : They are normalized to be orthonormal.
6 % Turn points into row if needed.
7 xp = x; dims = size(xp);
8 if (dims(2)==1) xp = xp'; end;
9 PL = zeros(N+1,length(xp));
```

```
% Initial values P_0(x) and P_1(x)
gamma0 = 2^(alpha+beta+1)/(alpha+beta+1)*gamma(alpha+1)*...
    gamma(beta+1)/gamma(alpha+beta+1);
PL(1,:) = 1.0/sqrt(gamma0);
if (N==0) P=PL'; return; end;
gamma1 = (alpha+1)*(beta+1)/(alpha+beta+3)*gamma0;
PL(2,:) = ((alpha+beta+2)*xp/2 + (alpha-beta)/2)/sqrt(gamma1);
if (N==1) P=PL(N+1,:)'; return; end;
% Repeat value in recurrence.
aold = 2/(2+alpha+beta)*sqrt((alpha+1)*(beta+1)/(alpha+beta+3))
   ;
% Forward recurrence using the symmetry of the recurrence.
for i=1:N-1
  h1 = 2*i+alpha+beta;
  anew = 2/(h1+2)*sqrt( (i+1)*(i+1+alpha+beta)*(i+1+alpha)*...
      (i+1+beta)/(h1+1)/(h1+3));
  bnew = - (alpha^2-beta^2)/h1/(h1+2);
  PL(i+2,:) = 1/anew*( -aold*PL(i,:) + (xp-bnew).*PL(i+1,:));
  aold =anew;
end;
P = PL(N+1,:)';
return
```

and their derivatives may be calculated by recognizing that the derivatives of the Legendre basis form another orthogonal basis under a nonconstant weight. The new basis can also be computed with JacobiP.m taking $\alpha = \beta = 1/2$. This code comes from the codes from the book [46].

Finally a point should be made on how Lagrange polynomials are computed through the normalized Legendre basis. As mentioned initially, the input and output values of the code are in fact nodal, not modal. Therefore coefficients of the Legendre basis of the solution are not immediately available. These can be computed by assuming that a given vector $\mathbf{u}$ are the solution values at predetermined nodal locations

(x_i) and then forming a generalized Vandermonde matrix

$$\mathbf{V}_{ij} = L_{j-1}(x_i)$$

and then the Legendre coefficients of the polynomial $\mathbf{u}$ are given by the nterpolation [46]

$$\mathbf{c} = \mathbf{V}^{-1}\mathbf{u}.$$

Thus, if additionally it is desired that the solution $\mathbf{u}$ be evaluated at points (y_j) that are not (x_i), then one again forms the associated Vandermonde $\tilde{\mathbf{V}}_{ij} = L_{j-1}(y_i)$ and the nodal values of $\mathbf{u}$ at the points (y_i) are given as

$$\tilde{\mathbf{u}} = \tilde{\mathbf{V}}\mathbf{V}^{-1}\mathbf{u}.$$

Tensor product versions of the Vandermonde operators can easily be computed with MATLAB's command "kron." It automatically forms the tensor product of two operators, and this is how the operators are constructed in associated code with this thesis.

7.4 Precomputing Operators

Deriving precomputed operators for the two dimensional code in this thesis will first require precomputed one dimensional operators, due to the tensor product structure of the two dimensional simulations. To see why this is so, consider the Tensor product basis

$$\phi_{ij} = \phi_i \phi_j$$

where ϕ_i (single indexed) is the Lagrange polynomial on a given interval $[x_l, x_r]$ associated with a node x_i. Then given a bilinear form $a(\cdot,\cdot)$ and the task to find u such that

$$a(u,v) \quad = \quad f(v)$$

holds for all polynomials v below a prescribed order, then the problem effectively becomes the task to find coefficients $\mathbf{c}$ such that

$$\sum_{i,j=1}^{N_p} c_{ij} a(\phi_{ij}, \phi_{\alpha\beta}) = f(\phi_{\alpha\beta})$$

so that if a can be separated in the following fashion:

$$a(fg, hk) = a^1(f,h)\, a^2(g,k)$$

the problem becomes effectively one dimensional

$$\sum_{i=1}^{N_p} c_i a^1(\phi_i, \phi_\alpha) \sum_{j=1}^{N_p} c_j a^2(\phi_j, \phi_\beta) = f(\phi_{\alpha\beta}).$$

Before giving the relevant a^1, a^2 for the various two dimensional operators in the acoustic wave equation, I give a reference table for the two dimensional operators obtained in this way, and their associated tensor product form:

Operator	Tensor Product	Operation	MATLAB
$\mathcal{M}^{2D}$	$\mathcal{M}^{1D} \odot \mathcal{M}^{1D}$	Mass	MassMatrixQ
$\mathcal{S}_r$	$\mathcal{M}^{1D} \odot \mathcal{S}$	r-Stiffness	StiffnessMatrixrQ
$\mathcal{S}_s$	$\mathcal{S} \odot \mathcal{M}^{1D}$	s-Stiffness	StiffnessMatrixsQ

Table 7.3 : Tensor product operators

From these operators we obtain associated derived operators $\mathcal{D}_r$, $\mathcal{D}_s$ and $\mathcal{L}$, respectively the r, s projected partial derivative operators and the L^2 polynomial trace lifting operator. In order to compute the lifting operator $\mathcal{L}$ it is necessary to be able to apply the one dimensional mass matrix $\mathcal{M}^{1D}$ to the boundary degrees of freedom, the operator for this action will be denoted $\mathcal{E}$. For reference I put these in table format as well

Operator	Formula	Operation	MATLAB
$\mathcal{D}_r$	$\left(\mathcal{M}^{2D}\right)^{-1}\mathcal{S}_r$	L^2 projected $\frac{\partial}{\partial r}$	DrQ
$\mathcal{D}_s$	$\left(\mathcal{M}^{2D}\right)^{-1}\mathcal{S}_s$	L^2 projected $\frac{\partial}{\partial s}$	DsQ
$\mathcal{L}$	$\left(\mathcal{M}^{2D}\right)^{-1}\mathcal{E}$	L^2 trace lifting operator	LIFTQ

Table 7.4 : Derived operators

and now I present a table for the formulas of the one dimensional operators

Operator	Matrix entries	Operation	MATLAB
$\mathcal{M}^{1D}$	$\int_{-1}^{1}\phi_i(x)\,\phi_j(x)\,dx$	1D Mass	MassMatrix
$\mathcal{S}$	$\int_{-1}^{1}\left[\frac{d}{dx}\phi_i(x)\right]\phi_j(x)\,dx$	1D Stiffness	StiffnessMatrix

Table 7.5 : One dimensional operators

The codes for Mass1D.m is a little overly general for the polynomial case, it adaptively applies gaussian quadratures until convergence. For polynomials this is unnecessary, but as noted in the theory on chapter 4 it is important for the integrals to be computed exactly. Thus for the Kosloff/Tal-Ezer mapped basis the adaptive strategy is employed to be as exact as possible. Numerous experiments performed demonstrate that this is enough.

Listing 7.2: Mass1D.m

```
1 function M = Mass1D;
```

```
Globals1D;
tol=1e-15;
err=1;
qord=N;
umr=JacobiGL(0,0,N);
while(err>tol)
    %First quadrature rule mass matrix.
    [qr,qw]=lgwt(qord,-1,1);
    qr=sort(qr);
    umrV=Vandermonde1D(N,umr);
    VQUAD=Vandermonde1D(N,qr);
    W=diag(qw);
    [~,dmx]=MapFun(qr,MapParam);
    MW=diag(dmx);
    M1=(MW*VQUAD/umrV)'*(W*MW*VQUAD/umrV);
    %Second quadrature rule mass matrix.
    [qr,qw]=lgwt(qord+1,-1,1);
    qr=sort(qr);
    umrV=Vandermonde1D(N,umr);
    VQUAD=Vandermonde1D(N,qr);
    W=diag(qw);
    [~,dmx]=MapFun(qr,MapParam);
    MW=diag(dmx);
    M2=(MW*VQUAD/umrV)'*(W*MW*VQUAD/umrV);
    %Calculate discrete L^2 norm of difference.
    %e=eig((M1-M2)'*(M1-M2),MassMatrixQ);
    %err=max(sqrt(e));
    err=max(max(abs(M1-M2)));
    qord=qord+1;
end
M=M2;
```

and

Listing 7.3: Stiffness1D.m

```
function S = Stiffness1D
Globals1D;
umr = JacobiGL(0,0,N);
umV = Vandermonde1D(N,umr);
umDr=Dmatrix1D(N,umr,umV);
S = (umV') \ (umV\umDr);
```

The stiffness matrix does not need adaptivity for the nonpolynomial case considered in this thesis, because the geometric factors from the derivative and from the coordinate transformation cancel out in the integral. For linear problems with constant coefficients these operators are all that will be needed to fully evaluate the spatial discretization and ensure the validity of semidiscrete stability results.

7.5 Time-stepping

The final thing needed for a full simulation is a time marching scheme. Supposing for the moment that the PDE under consideration is scalar (each component of the solution may be treated as such for time-stepping), one would obtain a semidiscretization of the form

$$\frac{d\mathbf{u}}{dt} = \beta_1 \mathcal{D}_r \mathbf{u} + \beta_2 \mathcal{D}_s \mathbf{u} + \mathcal{L}\left(d\mathbf{u}\right)$$

where d above calculates the inter-element boundary conditions (given by the operator $\mathbf{C} - \mathbf{C}^*\mathbf{C}$). Thus given a function to evaluate the right hand side above, one can use any explicit timestepping code whose input need only be a right hand side evaluation.

Listing 7.4: AcousticRHS2D.m

```
1  function [rhsUx, rhsUy, rhsP] = AcousticRHS2D(Ux,Uy,P,rho,kappa
       ,S,time,alpha,CovolumeFilter)
2  %function [rhsUx, rhsUy, rhsP] = AcousticRHS2D(Ux,Uy,P,Z,S,
       alpha)
3  %Evaluates RHS of acoustic wave equation.
4  %Ux-x velocity.
5  %Uy-y velocity.
6  %P - Pressure.
7  %rho - Density.
8  %kappa - Bulk modulus.
9  %S - Source term.
10 GlobalsQuad2D;
11 NRHS=NRHS+1;
12 %Calculate relative impedence. Assumes references values of 1.
```

```
13 Z = zeros(NfpQ*NfacesQ,KQ); Z(:) = sqrt( rho(vmapMQ).*kappa(
       vmapMQ));
14 ZP =zeros(NfpQ*NfacesQ,KQ); ZP(:) = Z(mapPQ);
15 ZM = zeros(NfpQ*NfacesQ,KQ); ZM(:) = Z(mapMQ);
16 %First order absorbing condition.
17 %P(vmapBQ)=nxQ(mapBQ).*Ux(vmapBQ) + nyQ(mapBQ).*Uy(vmapBQ);
18 %Calculate field differences at faces.
19 dUx = zeros(NfpQ*NfacesQ,KQ); dUx(:) = Ux(vmapMQ)-Ux(vmapPQ);
20 dUy = zeros(NfpQ*NfacesQ,KQ); dUy(:) = Uy(vmapMQ)-Uy(vmapPQ);
21 dP = zeros(NfpQ*NfacesQ,KQ); dP(:) = P(vmapMQ)-P(vmapPQ);
22 %Impose reflective boundary conditions. (P+=-P-).
23 %dUx(mapBQ) = 0;
24 %dUy(mapBQ) = 0;
25 dP(mapBQ)=2*P(vmapBQ);
26 %Averages of acoustic impedences at faces.
27 Zavg = zeros(NfpQ*NfacesQ,KQ); Zavg(:) = (Z(mapMQ)+Z(mapPQ))/2;
28 %Evaluate upwind flux.
29 ndotdU = nxQ.*dUx+nyQ.*dUy;
30 fluxU = (1./(2*Zavg)) .* (ZP.*ndotdU - alpha*dP);
31 fluxPx=-(ZM./(2*Zavg)) .* nxQ.*(alpha*ZP.*ndotdU - dP);
32 fluxPy=-(ZM./(2*Zavg)) .* nyQ.*(alpha*ZP.*ndotdU - dP);
33 %Local derivatives of fields
34 dUxdr = DrQ*Ux;
35 dUxds = DsQ*Ux;
36 dUydr = DrQ*Uy;
37 dUyds = DsQ*Uy;
38 dPdr = DrQ*P;
39 dPds = DsQ*P;
40 dPdx = rxQ.*dPdr + sxQ.*dPds;
41 dPdy = ryQ.*dPdr + syQ.*dPds;
42 dUxdx= rxQ.*dUxdr+ sxQ.*dUxds;
43 %dUxdy= ryQ.*dUxdr+ syQ.*dUxds;
44 %dUydx= rxQ.*dUydr+ sxQ.*dUyds;
45 dUydy= ryQ.*dUydr+ syQ.*dUyds;
46 %Compute right hand sides of PDE system.
47 %Note: Interpolates S rather than performing L^2 projection.
48 %Does not impact stability results.
49 rhsUx = (1./rho).*(-dPdx + LIFTQ*(FscaleQ.*fluxPx));
50 rhsUy = (1./rho).*(-dPdy + LIFTQ*(FscaleQ.*fluxPy));
51 rhsP  = (kappa).*(-dUxdx - dUydy + LIFTQ*(FscaleQ.*fluxU) + S(
       time));
52 %Now do covolume filter
```

```
53 if(CovolumeFilter>0)
54   temp=CovolumeFilter2D(rhsUx);
55   rhsUx = (1-CovolumeFilter)*rhsUx + CovolumeFilter*temp;
56   temp=CovolumeFilter2D(rhsUy);
57   rhsUx = (1-CovolumeFilter)*rhsUy + CovolumeFilter*temp;
58   temp=CovolumeFilter2D(rhsP);
59   rhsUx = (1-CovolumeFilter)*rhsP + CovolumeFilter*temp;
60 end
```

and in order to be able to use this with the way timesteppers are generally written (assuming a long vector of dimensions as $mn \times 1$ rather than $m \times n$), an associated marshalling function is provided alongside this:

Listing 7.5: AcousticOdefun2D.m

```
1  function [uout] = AcousticOdefun2D(uin,rho,kappa,S,time,alpha,
       CovolumeFilter)
2  GlobalsQuad2D;
3  temp=reshape(uin,length(uin)/3,3);
4  Ux=reshape(temp(:,1),NpQ,KQ);
5  Uy=reshape(temp(:,2),NpQ,KQ);
6  P=reshape(temp(:,3),NpQ,KQ);
7  [rhsUx,rhsUy,rhsP] = AcousticRHS2D(Ux,Uy,P,rho,kappa,S,time,
       alpha,CovolumeFilter);
8  uout = [rhsUx(:);rhsUy(:);rhsP(:)];
9  return;
10 end
```

Often this thesis uses the built in MATLAB solver ode45, but another comopnent of this thesis is the use of Adams-Bashforth and optimal low-storage Runge-Kutta methods. The codes for these are listed below for reference, along with corresponding coefficients for the Runge-Kutta methods

Listing 7.6: LSRK.m

```
1  function [u nf] = LSRK(odefun,init,dt,FinalTime,A,B,c,callback,
       varargin)
2  %Performs low-storage RK method associated with A,B,c. (See
       Ketcheson et. al.)
3  %ARGUMENTS:
4  %
```

```
5  % odefun - function for ODE.
6  % init - initial value.
7  % dt  - timestep. (use evalstabledt to find CFL constrained dt)
8  % FinalTime - final time.
9  %A,B,c - LSRK tabulated coefficients.
10 %callback - Function to call at specified timesteps
11 %RETURNS:
12 %
13 % u - solution.
14 % nf - #function evals.
15 %
16 %USAGE:
17 % [u nf] = LSRK(odefun,init,dt,FinalTime,A,B,c)
18 tol=1e-5;
19 time = 0;
20 Nsteps = ceil(FinalTime/dt);
21 dt = FinalTime/Nsteps;
22 u = init;
23 m = length(A);
24 nf = 0;
25 tstep=1;
26 while(time < FinalTime-tol)
27   callback(time,u,varargin{:});
28   K2 = zeros(size(u));
29   K1 = u;
30   for i = 1 : m
31     K2 = A(i)*K2 + dt*odefun(time+c(i)*dt,K1);
32     nf=nf+1;
33     K1 = K1 + B(i)*K2;
34   end
35   tstep=tstep+1;
36   u = K1;
37   time=time+dt;
38 end
```

Table 7.6 : Tabulated RKC73 coefficients

A_i	B_i	c_i
0	0.01197052673097840	0
−0.8083163874983830	0.8886897793820711	0.01197052673097840
1.503407858773331	0.4578382089261419	0.1823177940361990
−1.053064525050744	0.5790045253338471	0.5082168062551849
−1.463149119280508	0.3160214638138484	0.6532031220148590
−0.6592881281087830	0.2483525368264122	0.8534401385678250
−1.667891931891068	0.06771230959408840	0.9980466084623790

Table 7.7 : Tabulated RKC84 coefficients

A_i	B_i	c_i
0	0.216593673678085	0
−0.7212962482279240	0.1773950826411583	0.2165936736758085
−0.01077336571612980	0.01802538611623290	0.2660343487538170
−0.516258469830970	0.08473476372541490	0.2840056122522720
−1.730100286632201	0.8129106974622483	0.325126684378870
−5.200129304403076	1.903416030422760	0.4555149599187530
0.7837058945416420	0.1314841743399048	0.7713219317101170
−0.5445836094332190	0.2082583170674149	0.9199028964538660

Table 7.8 : Tabulated RKF84 coefficients

A_i	B_i	c_i
0	0.08037936882736950	0
−0.5534431294501569	0.5388497458569843	0.08037936882736950
0.01065987570203490	0.01974974409031960	0.3210064250338430
−0.551812888932000	0.09911841297339970	0.3408501826604660
−1.885790377558741	0.7466920411064123	0.385036482485470
−5.701295742793264	1.679584245618894	0.5040052477534100
2.113903965664793	0.2433728067008188	0.6578977561168540
−0.5339578826675280	0.1422730459001373	0.9484087623348481

Table 7.9 : Tabulated NRK13E coefficients

A_i	B_i	c_i
0	0.0271990297818803	0
−0.6160178650170565	0.1772488819905108	0.0271990297818803
−0.4449487060774118	0.0378528418949694	0.0952594339119365
−1.0952033345276178	0.6086431830142991	0.1266450286591127
−1.2256030785959187	0.2154313974316100	0.1825883045699772
−0.2740182222332805	0.2066152563885843	0.3737511439063931
−0.0411952089052647	0.0415864076069797	0.5301279418422206
−0.1797084899153560	0.219891884310925	0.5704177433952291
−1.1771530652064288	0.9893081222650993	0.5885784947099155
−0.4078831463120878	0.00631990119859826	0.6160769826246714
−0.8295636426191777	0.3749640721105318	0.6223252334314046
−4.7895970584252288	1.6080235151003195	0.6897593128753419
−0.6606671432964504	0.0961209123818189	0.9126827615920843

Table 7.10 : Tabulated NRK14C coefficients

A_i	B_i	c_i
0	0.0367762454319673	0
−0.718801208672410	0.3136296607553959	0.0367762454319673
−0.7785331173421570	0.1531848691869027	0.1249685262725025
−0.0053282796654044	0.0030097086818182	0.2446177702277698
−0.8552979934029281	0.3326293790646110	0.2476149531070420
−3.9564138245774565	0.2440251405350864	0.2969311120382472
−1.5780575380587385	0.3718879239592277	0.3978149645802642
−2.0837094552574054	0.6204126221582444	0.5270854589440328
−0.7483334182761610	0.1524043173028741	0.6981269994175695
−0.7032861106563359	0.0760894927419266	0.8190890835352128
0.0013917096117681	0.0077604214040978	0.8527059887098624
−0.0932075369637460	0.0024647284755382	0.8604711817462826
−0.9514200470875948	0.0780348340049386	0.8627060376969976
−7.1151571693922548	5.5059777270269628	0.8734213127600976

a function for computing Adams-Bashforth solutions for orders $1, 2, 3, 4, 5$ is also supplied below:

Listing 7.7: ABM.m

```
function [u nf] = ABM(odefun,init,dt,FinalTime,type,callback,
    varargin)
%Implements third order adams-bashforth
tol=1e-5;
time = 0;
Nsteps = ceil(FinalTime/dt);
dt = FinalTime/Nsteps;
u = init;
%Pick AB scheme
switch type
  %Euler's method
  case 1
    b = [1];
  %AB2
  case 2
    b = [-1/2;3/2];
  %AB3
  case 3
    b = [5/12;-4/3;23/12];

  %AB4
  case 4
    b = [-3/8;37/24;-59/24;55/24];
  %AB5
  case 5
    b = [251/720;-637/360;109/30;-1387/360;1901/720];
end
%Can't remember lowest relative tolerance, this will default to
    it.
options=odeset('RelTol',1e-15);
n=length(b);
%Obtain previous data
[T U] = ode45(@(t,u)odefun(t,u),[0:dt:(n-1)*dt],init,options);
for i = 1 : n
  fu(:,i)=odefun(time,U(i,:)');
end
u=U(end,:)';
time=time+(n-1)*dt;
nf=n;
while(time < FinalTime-tol)
```

```
39    callback(time,u,varargin{:});
40    u = u + dt*(fu*b);
41    time=time+dt;
42    fu(:,1:end-1)=fu(:,2:end);
43    fu(:,end)=odefun(time,u);nf=nf+1;
44  end
```

7.6 Covolume Filtering

The covolume filter of Warburton and Hagstrom [100] is applied by utilizing the tensor product structure of the structured grids in use to decompose the two dimensional filter into a sequence of one dimensional filters. The one and two dimensional filter MATLAB codes are given below

Listing 7.8: CovolumeFilterPeriodic1D.m

```
1  function v = CovolumeFilterPeriodic1D(u)
2  Globals1D;
3  %Assumes periodic mesh
4  v = P1*u(:,[2:end,1]) + P2*u;
5  v = P1*v + P2*v(:,[end,1:(end-1)]);
6  return;
7  end
```

and its associated 2D version in CovolumeFilter2D.m:

Listing 7.9: CovolumeFilter2D.m

```
1  function v = CovolumeFilter2D(u,CovolumeFilter)
2  GlobalsQuad2D;
3  v=zeros(size(u));
4  temp=zeros(size(u));
5  n=size(xgrid,3);
6
7
8
9  %First do horizontal sweep.
10 for i = 1 : n
11   v(xgrid(:,:,i)) = CovolumeFilterPeriodic1D(u(xgrid(:,:,i)));
12 end
```

```
13
14 %Next do vertical sweep.
15
16 for i = 1 : n
17    v(ygrid(:,:,i)) = CovolumeFilterPeriodic1D(v(ygrid(:,:,i)));
18 end
19
20
21 end
```

The operators in use in the code are precomputed L^2 projection operators and indexing sets for two dimensions. Currently this code works only for periodic boundary conditions.

7.7 Flux Filtering

The filter in use is implemented in Filter2D.m listed below

Listing 7.10: Filter2D.m

```
1  function [F] = Filter2D(Norder,scale)
2  %Constructs simple modal filter.
3  GlobalsQuad2D;
4  filterdiag = ones( (Norder+1)^2,1);
5  index=1:(Norder+1)^2;
6  index=reshape(index,Norder+1,Norder+1);
7  %filterdiag(end)=scale;
8  for i = 0 : Norder
9    for j = 0 : Norder
10     if( (i+j) >= Norder)
11       filterdiag(index(i+1,j+1))=scale;
12     end
13   end
14 end
15 F = VQ * diag(filterdiag) / VQ;
16 return;
```

and if $\mathcal{F}^\delta$ represents the filter operator, then the filtered lifting operator $\mathcal{L}^\delta$ is given by

$$\mathcal{L}^\delta = \mathcal{F}^\delta \mathcal{L},$$

once the lifting operator is computed, it is automatically used in the right-hand-side evaluation functions and so no further modification is required.

7.8 A Complete Script

With the explanations given above I am in a position to demonstrate how to construct a simulation script. In the script one must define the mesh desired polynomial order. After this a startup script (StartUpQuad2D.m) is run to evaluate the operators, construct index maps, and calculate geometric transformation factors. Next the user must define any initial conditions and source terms, and then input the right hand side evaluation function with these terms into the desired timestepper. This is shown in the following driver script

Listing 7.11: AcousticDriverExample.m

```
1  %Driver script to solve acoustic wave equation in conservation
       form in 2D using quadrilaterals.
2  GlobalsQuad2D;
3  %Polynomial order to use.
4  N = 8;
5  %Mapping function to use.
6  MapFun=@(r,lambda)identity(r,lambda);
7  %Flux filtering amount.
8  FluxFilter=1;
9  %Generate dummy 1D mesh for StartUp1D
10 [Nv, VX, K, EToV] = MeshGen1D(0.0,2.0,2);
11 Nx=5;
12 Ny=5;
13 %Construct uniform grid.
14 [NvQ VXQ VYQ KQ EToVQ] = BuildRegularQuadMesh2D(Nx,Ny);
15 %Construct airfoil grid.
16 %[NvQ, VXQ, VYQ, KQ, EToVQ] = BuildJoukouskyQuadMesh(Nx,Ny);
17 StartUpQuad2D;
18 %Impose periodic boundary conditions.
19 BuildPeriodicMapsQuad2D(2,2);
20 %Initial velocity = 0.
21 Ux = zeros(NpQ,KQ);
```

```
Uy = zeros(NpQ,KQ);
%Initial pressure.
P=cos(pi*xQ).*sin(pi*yQ) + sin(pi*xQ).*cos(pi*yQ);
%Basic material properties.
rho = ones(NpQ,KQ);
kappa = ones(NpQ,KQ);
%Zero source term.
S=@(t)zeros(NpQ,KQ);
%Desired times for solution outputs.
timesamples=0:1:5;
%Upwinding parameter
alpha=1;
%How much to apply covolume filtering.
CovolumeFilter=0;
odefun=@(t,u)AcousticOdefun2D(u,rho,kappa,S,t,alpha,
   CovolumeFilter);
init=[Ux(:);Uy(:);P(:)];
%Solve with ODE45
options=odeset('RelTol',1e-4);
[T U] = ode45(odefun,timesamples,init,options);
```

Bibliography

[1] Explicit trace inequalities for isogeometric analysis and parametric hexahedral finite elements - springer.

[2] T. Warburton A. Kloeckner and J. S. Hesthaven. High-order discontinuous galerkin methods by GPU metaprogramming. Technical Report 2011-13, Scientific Computing Group, Brown University, Providence, RI, USA, June 2011.

[3] Assyr Abdulle. Fourth order chebyshev methods with recurrence relation. *SIAM Journal on Scientific Computing*, 23(6):2041–2054, January 2002.

[4] Assyr Abdulle and Alexei A. Medovikov. Second order chebyshev methods based on orthogonal polynomials. *Numerische Mathematik*, 90(1):1–18, November 2001.

[5] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth dover printing, tenth GPO printing edition, 1964.

[6] M.R. Abril-Raymundo and B. Garcı?a-Archilla. Approximation properties of a mapped chebyshev method. *Applied Numerical Mathematics*, 32(2):119–136, February 2000.

[7] Ben Adcock\phantomx(mathoverflow.net/users/19011). *Markov-type inequalities with arbitrary exponents*. Published: MathOverflow

http://mathoverflow.net/questions/81933 (version: 2011-11-26).

[8] V. Alexiades. Overcoming the stability restriction of explicit schemes via super-time-stepping. *Proceedings of Dynamic Systems and Applications*, 2:39–44, 1995.

[9] Vasilios Alexiades, GeneviÃ©ve Amiez, and Pierre-Alain Gremaud. Super-time-stepping acceleration of explicit schemes for parabolic problems. *Com. Num. Meth. Eng*, 12:12–31, 1996.

[10] Vasanth Allampalli, Ray Hixon, M. Nallasamy, and Scott D. Sawyer. High-accuracy large-step explicit runge-kutta (HALE-RK) schemes for computational aeroacoustics. *Journal of Computational Physics*, 228(10):3837 – 3850, 2009.

[11] P. Bar-Yoseph. Space-time discontinuous finite element approximations for multi-dimensional nonlinear hyperbolic systems. *Computational Mechanics*, 5(2):145–160, March 1989.

[12] Pinhas Bar-Yoseph and David Elata. An efficient l2 galerkin finite element method for multi-dimensional non-linear hyperbolic systems. *International Journal for Numerical Methods in Engineering*, 29(6):1229–1245, 1990.

[13] A Bayliss and B.J Matkowsky. Fronts, relaxation oscillations, and period doubling in solid fuel combustion. *Journal of Computational Physics*, 71(1):147–168, July 1987.

[14] John P. Boyd. *Chebyshev and Fourier Spectral Methods*. 1999.

[15] John P. Boyd and Jun Rong Ong. Exponentially-convergent strategies for defeating the runge phenomenon for the approximation of non-periodic functions,

part two: Multi-interval polynomial schemes and multidomain chebyshev interpolation. *Applied Numerical Mathematics*, 61(4):460 – 472, 2011.

[16] J.C. Butcher and J. Wiley. *Numerical methods for ordinary differential equations*. Wiley Online Library, 2008.

[17] Paul Castillo, Bernardo Cockburn, Dominik Schotzau, Dominik, and Christoph Schwab. Optimal a priori error estimates for the hp-version of the local discontinuous galerkin method for convection-diffusion problems. *Math. Comput.*, 71(238):455–478, April 2002.

[18] M. Celik and A.C. Cangellaris. Simulation of multiconductor transmission lines using krylov subspace order-reduction techniques. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 16(5):485–496, 1997.

[19] N. Chalmers, L. Krivodonova, and R. Qin. Relaxing the CFL number of the discontinuous galerkin method.

[20] G. Chavent and G. Salzano. A finite-element method for the 1-d water flooding problem with gravity. *Journal of Computational Physics*, 45(3):307–344, March 1982.

[21] Q. Y. Chen, D. Gottlieb, and J. S. Hesthaven. Spectral methods based on prolate spheroidal wave functions for hyperbolic PDEs. *SIAM Journal on Numerical Analysis*, 43(5):1912–1933, January 2005.

[22] Ronald Chen and Thomas Hagstrom. P-adaptive hermite methods for initial value problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(03):545–557, 2012.

[23] Bernardo Cockburn, Suchung Hou, and Chi-Wang Shu. The runge-kutta local projection discontinuous galerkin finite element method for conservation laws. IV: the multidimensional case. *Mathematics of Computation*, 54(190):pp. 545–581, 1990.

[24] Bernardo Cockburn, George E. Karniadakis, and Chi-Wang Shu. *The development of discontinuous Galerkin methods*. 1999.

[25] Bernardo Cockburn, San-Yih Lin, and Chi-Wang Shu. TVB runge-kutta local projection discontinuous galerkin finite element method for conservation laws III: one-dimensional systems. *Journal of Computational Physics*, 84(1):90 – 113, 1989.

[26] Bernardo Cockburn and Chi-Wang Shu. TVB runge-kutta local projection discontinuous galerkin finite element method for conservation laws II: general framework. *Mathematics of Computation*, 52(186):pp. 411–435, 1989.

[27] Bernardo Cockburn and Chi-Wang Shu. The runge-kutta discontinuous galerkin method for conservation laws v: Multidimensional systems. *Journal of Computational Physics*, 141(2):199 – 224, 1998.

[28] Z. Ditzian. Multivariate bernstein and markov inequalities. *Journal of Approximation Theory*, 70(3):273 – 283, 1992.

[29] Wai Sun Don and Alex Solomonoff. Accuracy enhancement for higher derivatives using chebyshev collocation and a mapping technique. *SIAM J. Sci. Comput.*, 18(4):1040–1055, July 1997.

[30] Moshe Dubiner. Asymptotic analysis of spectral methods. *Journal of Scientific Computing*, 2(1):3–31, 1987. 10.1007/BF01061510.

[31] Todd Dupont and Ridgway Scott. Polynomial approximation of functions in sobolev spaces. *Mathematics of Computation*, 34(150):441–463, April 1980. ArticleType: research-article / Full publication date: Apr., 1980 / Copyright © 1980 American Mathematical Society.

[32] Kenneth Eriksson, Claes Johnson, and Anders Logg. Explicit time-stepping for stiff ODEs. *SIAM J. Sci. Comput.*, 25(4):1142–1157, April 2003.

[33] A. Ern and J.-L. Guermond. Discontinuous galerkin methods for friedrichs' systems. i. general theory. *SIAM Journal on Numerical Analysis*, 44(2):753–778, January 2006. ArticleType: research-article / Full publication date: 2006 / Copyright © 2006 Society for Industrial and Applied Mathematics.

[34] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements*, volume 159. Springer, 2004.

[35] C. W. Gear and Ioannis G. Kevrekidis. Projective methods for stiff differential equations: Problems with gaps in their eigenvalue spectrum. *SIAM J. Sci. Comput.*, 24(4):1091–1106, April 2002.

[36] Patrick Godon. Numerical modeling of tidal effects in polytropic accretion disks. *The Astrophysical Journal*, 480(1):329, 1997.

[37] N. Goedel, S. Schomann, T. Warburton, and M. Clemens. Local timestepping discontinuous galerkin methods for electromagnetic RF field problems. page 2149–2153, 2009.

[38] N. Goedel, T. Warburton, and M. Clemens. GPU accelerated discontinuous galerkin FEM for electromagnetic radio frequency problems. pages 1 –4, June 2009.

[39] John Goodrich, Thomas Hagstrom, and Jens Lorenz. Hermite methods for hyperbolic initial-boundary value problems. *Mathematics of Computation*, 75(254):595–630 (electronic), 2006.

[40] David Gottlieb and Steven A. Orszag. *Numerical Analysis of Spectral Methods*. Society for Industrial and Applied Mathematics, Philadephia, PA, 1977.

[41] David Gottlieb and Eitan Tadmor. The CFL condition for spectral approximations to hyperbolic initial-boundary value problems. *Mathematics of Computation*, 56(194):565–588, 1991.

[42] Ben-yu Guo and Hong-li Jia. Pseudospectral method for quadrilaterals. *Journal of Computational and Applied Mathematics*, 236(5):962–979, October 2011.

[43] Nicholas Hale. *On The Use Of Conformal Maps To Speed Up Numerical Computations*. PhD thesis, University of Oxford, 2009.

[44] J. S. Hesthaven, P. G. Dinesen, and J. P. Lynov. Spectral collocation time-domain modeling of diffractive optical elements. *J. Comput. Phys.*, 155(2):287–306, November 1999.

[45] J. S. Hesthaven and T. Warburton. High-order nodal discontinuous galerkin methods for the maxwell eigenvalue problem. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 362(1816):pp. 493–524, 2004.

[46] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[47] J.S Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids: I. time-domain solution of maxwell's equations. *Journal of Computational Physics*, 181(1):186 – 221, 2002.

[48] Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

[49] PJ Houwen. *Construction of integration formulas for initial value problems*. North-Holland Pub. Co.(Amsterdam and New York and New York), 1977.

[50] Butcher J.C. A history of runge-kutta methods. *Applied Numerical Mathematics*, 20(3):247–260, March 1996.

[51] R. Jeltsch and M. Torrilhon. Flexible stability domains for explicit runge-kutta methods. *Some topics in industrial and applied mathematics*, 8:152, 2007.

[52] Rolf Jeltsch and Olavi Nevanlinna. Largest disk of stability of explicit runge-kutta methods. *BIT Numerical Mathematics*, 18(4):500–502, 1978. 10.1007/BF01932030.

[53] Boyd John. Exponentially accurate runge-free approximation of non-periodic functions from samples on an evenly spaced grid. *Applied Mathematics Letters*, 20(9):971 – 975, 2007.

[54] Carcione Jose. A 2D chebyshev differential operator for the elastic wave equation. *Computer Methods in Applied Mechanics and Engineering*, 130(1–2):33–45, March 1996.

[55] David Ketcheson and Aron Ahmadia. Optimal runge-kutta stability regions. 2012.

[56] David I. Ketcheson. Runge-kutta methods with minimum storage implementations. *J. Comput. Phys.*, 229(5):1763–1773, March 2010.

[57] Ingemar P. E. Kinnmark and William G. Gray. Fourth-order accurate one-step integration methods with large imaginary stability limits. *Numerical Methods for Partial Differential Equations*, 2(1):63–70, 1986.

[58] Ingemar P.E. Kinnmark and William G. Gray. One step integration methods of third-fourth order accuracy with large hyperbolic stability limits. *Mathematics and Computers in Simulation*, 26(3):181 – 188, 1984.

[59] Ingemar P.E. Kinnmark and William G. Gray. One step integration methods with maximum stability regions. *Mathematics and Computers in Simulation*, 26(2):87 – 92, 1984.

[60] A. Klöckner. *High-Performance High-Order Simulation of Wave and Plasma Phenomenon*. PhD thesis, Brown University, 2010.

[61] Dan Kosloff and Hillel Tal-Ezer. A modified chebyshev pseudospectral method with an o(1/N) time step restriction. *J. Comput. Phys.*, 104(2), February 1993.

[62] L. Krivodonova and R. Qin. An analysis of the spectrum of the discontinuous galerkin method. *Applied Numerical Mathematics*.

[63] Ethan J. Kubatko, Clint Dawson, and Joannes J. Westerink. Time step restrictions for runge-kutta discontinuous galerkin methods on triangular grids. *J. Comput. Phys.*, 227(23):9697–9710, December 2008.

[64] J. Douglas Lawson. An order five runge-kutta process with extended region of stability. *SIAM Journal on Numerical Analysis*, 3(4):pp. 593–597, 1966.

[65] J. Douglas Lawson. An order six runge-kutta process with extended region of stability. *SIAM Journal on Numerical Analysis*, 4(4):620–625, December 1967. ArticleType: research-article / Full publication date: Dec., 1967 / Copyright © 1967 Society for Industrial and Applied Mathematics.

[66] Randall Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[67] Anders Logg. Multi-adaptive time integration. *Applied Numerical Mathematics*, 48(3-4):339 – 354, 2004. Workshop on Innovative Time Integrators for PDEs.

[68] Harvard Lomax. On the construction of highly stable, explicit, numerical methods for integrating coupled ordinary differential equations with parasitic eigenvalues. Technical report, Ames Research Center, Moffet Field, California, 1968.

[69] G.G. Lorentz. *Approximation of functions*. Chelsea Publishing Company, Incorporated, 2005.

[70] Robert Byron Lowrie, Professor Philip, L. Roe, and Professor Bram Van Leer. *Compact Higher-Order Numerical Methods For Hyperbolic Conservation Laws*. PhD thesis, 1996.

[71] Jodi L. Mead and Rosemary A. Renaut. Accuracy, resolution, and stability properties of a modified chebyshev method. *SIAM J. Sci. Comput.*, 24(1):143–160, January 2002.

[72] Alexei Medovikov. High order explicit methods for parabolic equations. *BIT Numerical Mathematics*, 38(2):372–390, June 1998.

[73] Jens Niegemann, Richard Diehl, and Kurt Busch. Efficient low-storage runge-kutta schemes with optimized stability regions. *Journal of Computational Physics*, (0):–, 2011.

[74] Sevtap Ozisk, Beatrice Riviere, and Tim Warburton. On the constants in inverse inequalities in l^2. Technical 19, Rice University, 2010.

[75] Ingemar P.E and Kinnmark. A principle for construction of one-step integration methods with maximum imaginary stability limits. *Mathematics and Computers in Simulation*, 29(2):87 – 106, 1987.

[76] Van Der Houwen P.J. The development of runge-kutta methods for partial differential equations. *Applied Numerical Mathematics*, 20(3):261 – 272, 1996.

[77] Rodrigo B. Platte, Lloyd N. Trefethen, and Arno B. J. Kuijlaars. Impossibility of fast stable approximation of analytic functions from equispaced samples. *SIAM Review*, 53(2):308–318, 2011.

[78] P.Raviart P.Lesaint. On a finite element method for solving the neutron transport equation. *Mathematical aspects of finite elements in partial differential equations*, 1974.

[79] David A. Pope. An exponential method of numerical integration of ordinary differential equations. *Commun. ACM*, 6(8):491–493, 1963.

[80] J. Proft and B. Riviere. Discontinuous galerkin methods for convection-diffusion equations for varying and vanishing diffusivity. *Int. J. Numer. Anal. Model*, 6(4):533–561, 2009.

[81] R and Vichnevetsky. New stability theorems concerning one-step numerical methods for ordinary differential equations. *Mathematics and Computers in Simulation*, 25(3):199 – 205, 1983.

[82] Jan Ramboer, Tim Broeckhoven, Sergey Smirnov, and Chris Lacor. Optimization of time integration schemes coupled to spatial discretization for use in CAA applications. *Journal of Computational Physics*, 213(2):777–802, April 2006.

[83] Satish C. Reddy and Lloyd N. Trefethen. Stability of the method of lines. *Numerische Mathematik*, 62(1):235–267, 1992. 10.1007/BF01396228.

[84] W.H Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. 1973.

[85] R. A. Renaut. Two-step runge-kutta methods and hyperbolic partial differential equations. *Mathematics of Computation*, 55(192):pp. 563–579, 1990.

[86] L. F. Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210:pp. 307–357, 1911.

[87] Gerard R. Richter. An explicit finite element method for the wave equation. *Appl. Numer. Math.*, 16(1-2):65–80, December 1994.

[88] Y. Saad. Analysis of some krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):pp. 209–228, 1992.

[89] S. Schomann, N. GoÌˆ anddel, T. Warburton, and M. Clemens. Local timestepping techniques using taylor expansion for modeling electromagnetic wave propagation with discontinuous galerkin-FEM. *Magnetics, IEEE Transactions on*, 46(8):3504 –3507, August 2010.

[90] Jie Shen and Li-Lian Wang. Error analysis for mapped legendre spectral and pseudospectral methods. *SIAM Journal on Numerical Analysis*, 42(1):326–349, January 2005. ArticleType: research-article / Full publication date: 2005 / Copyright © 2005 Society for Industrial and Applied Mathematics.

[91] Y. Shi, L. Li, and C.H. Liang. Multidomain pseudospectral time-domain algorithm based on super-time-stepping method. *Microwaves, Antennas and Propagation, IEE Proceedings* -, 153(1):55 – 60, February 2006.

[92] A. Solomonoff and E. Turkel. Global properties of pseudospectral methods. *Journal of Computational Physics*, 81(2):239 – 276, 1989.

[93] Hillel Tal-Ezer. A pseudospectral legendre method for hyperbolic equations with an improved stability condition. *Journal of Computational Physics*, 67(1):145 – 172, 1986.

[94] T. Toulorge and W. Desmet. CFL conditions for runge-kutta discontinuous galerkin methods on triangular grids. *Journal of Computational Physics*, 230(12):4657 – 4678, 2011.

[95] T. Toulorge and W. Desmet. Optimal Runge–Kutta schemes for discontinuous galerkin space discretizations applied to wave propagation problems. *Journal of Computational Physics*, 231(4):2067–2091, February 2012.

[96] Lloyd N. Trefethen and Manfred R. Trummer. An instability phenomenon in spectral methods. *SIAM Journal on Numerical Analysis*, 24(5):pp. 1008–1023, 1987.

[97] F. X. Trias and O. Lehmkuhl. A self-adaptive strategy for the time integration of navier-stokes equations. *Numerical Heat Transfer, Part B: Fundamentals*, 60(2):116–134, 2011.

[98] P. J. van der Houwen. Explicit runge-kutta formulas with increased stability boundaries. *Numerische Mathematik*, 20(2):149–164, 1972. 10.1007/BF01404404.

[99] Michael Ng W.-B. Liu and Zhong-Ci Shi, editors. *Spatial Resolution Properties of Mapped Spectral Chebyshev Methods.* Science Press (Beijing), 2007. p. 179-188.

[100] T. Warburton and T. Hagstrom. Taming the CFL number for discontinuous galerkin methods on structured meshes. *SIAM J. Numer. Anal.*, 46(6), September 2008.

[101] Timothy Warburton. Analysis of low storage curvilinear discontinuous galerkin method for wave problems. Preprint, Rice University.

[102] Ben yu Guo and Hong-Li Jia. Spectral method on quadrilaterals. *Math. Comput.*, 79(272):2237–2264, 2010.