
Discontinuous Galerkin Methods: Exploiting Superconvergence for Improved Time-stepping

A thesis submitted to the School of Mathematics at the
University of East Anglia in partial fulfilment of the
requirements for the degree of Doctor of Philosophy

Daniel J. Freat

June 2017

© This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that use of any information derived there from must be in accordance with current UK Copyright Law. In addition, any quotation or extract must include full attribution.

Abstract

The discontinuous Galerkin (DG) methods are one of the most extensively researched classes of numerical methods for solving partial differential equations that display convective or diffusive qualities and have been popularly adopted by the scientific and engineering communities as a method capable of achieving arbitrary orders of accuracy in space. The choice of numerical flux function plays a pivotal role in the successful construction of DG methods and has an intrinsic effect on the superconvergence properties. As an inherent property of the spatial discretisation, superconvergence can only be retained in the solution through a sensitive pairing with a time integrator. The results of the literature and of this work suggest that an improved pairing between the spatial and temporal discretisations is both desirable and possible.

We perform analysis of three different but related manifestations of superconvergence: the local, super-accurate points themselves; the subsequent global extraction via the Smoothness-Increasing Accuracy-Conserving (SIAC) filters; and the spectral properties that quantify, in terms of dispersion and dissipation errors, how accurately waves are convected. In order to explore the effect of the numerical flux function on superconvergence, we consider a generalisation of the “natural” upwind choice for a Method of Lines solution to the linear advection equation: the upwind-biased flux. We prove that the method is locally superconvergent at roots of a linear combination of the left- and right-Radau polynomials dependent on the value of a flux parameter and

that the use of SIAC filters is still able to draw out the superconvergence information and create a globally smooth and superconvergent solution. In exploring the coupling of DG with a time integrator, we introduce a new scheme to a class of multi-stage multi-derivative methods, following recent incorporation of local DG technologies to recover superconvergence and achieve improved wave propagation properties.

Contents

Abstract	1
List of Figures	7
List of Tables	9
Acknowledgements	11
1 Introduction	15
1.1 Problem Statement	17
1.2 Literature Review	18
1.2.1 Outline of thesis	26
1.3 Background	26
1.3.1 Notation and definitions	26
1.3.2 Hyperbolic conservation law	29
1.3.3 Tessellation	32
1.3.4 Discontinuous Galerkin semi-discretisation	34

1.3.5	Basis functions for the approximation space	35
1.3.6	Galerkin expansions	37
1.3.7	Numerical flux function	39
1.3.8	Linear advection: Fully resolved semi-discrete scheme	41
2	The Numerical Flux and Superconvergence	44
2.1	Pointwise Superconvergence	45
2.1.1	Pointwise error estimate	49
2.1.2	Numerical experiments	56
2.2	Superconvergence of Post-processed Solution	60
2.2.1	The convolution kernel	60
2.2.2	SIAC filtered error estimate	61
2.2.3	Numerical experiments	66
3	Timestepping	69
3.1	Strong Stability Preserving Runge-Kutta Methods	70
3.2	Multiderivative Methods: Lax-Wendroff DG	71
3.3	DG-TDRK4	73
3.4	A New DG-TDRK4 Method	76
3.4.1	Stability analysis	77
3.5	Numerical Experiments	80
3.5.1	Linear advection	81

3.5.2	Linear advection with discontinuous coefficient	86
4	Dispersion and Dissipation of DG Schemes	90
4.1	Fourier Analysis of Amplification Matrices	93
4.1.1	Non-dimensionalised presentation of eigenvalues	96
4.2	Numerical Process	98
4.3	Semi-discrete DG Scheme: Results	100
4.4	Fully-discrete DG Schemes: Formulation	104
4.4.1	Fully decoupled Runge-Kutta methods	105
4.4.2	Fully resolved forms: DG-TDRK4 with direct differentiation	106
4.4.3	Fully resolved forms: DG-TDRK4 with differentiation by DG	109
4.5	Fully-discrete DG Schemes: Results	110
4.5.1	Piecewise constant basis	111
4.5.2	Piecewise linear basis	114
4.5.3	Piecewise quadratic basis	116
4.6	Numerical Experiments	120
4.6.1	Linear advection: dispersion and dissipation	120
5	Conclusions	123
	Appendices	125
A	Mathematica Code	126

A.1 Semi-discrete example	126
A.2 Fully-discrete example	135

List of Figures

1.3.1 One-dimensional elements in a segment of the tessellation	32
1.3.2 Legendre polynomials up to order 3	36
2.1.1 Discretisation errors for DG solution to 1D linear hyperbolic equation with $p = 1$ and $N = 10$	57
2.1.2 Discretisation errors for DG solution to 1D linear hyperbolic equation with $p = 2$ and $N = 10$	57
2.1.3 Discretisation errors for DG solution to 1D linear hyperbolic equation with $p = 3$ and $N = 10$	58
2.1.4 Discretisation errors for DG solution to 1D linear hyperbolic equation with $p = 4$ and $N = 10$	59
2.2.1 The $2p + 1$ B-splines of order $p = 1$ (left) and $p = 2$ (right) and the corresponding convolution kernels	61
2.2.2 DG and filtered errors for $p = 2$ at time $T = 1$	67
2.2.3 DG and filtered errors for $p = 3$ at time $T = 1$	68

3.5.1 Time history of \mathcal{L}^2 -errors of DG solutions to the linear advection equation	85
3.5.2 Linear Advection with discontinuous coefficient. \mathcal{L}^2 -errors (left) and post-processed errors (right) of Modified TDRK4, TDRK4, RK(10,4)-DG ($\theta = 1, 1.5$) (from top to bottom). \mathcal{P}^3 solutions at $T = 157$, all with $\nu = 0.01$	89
4.0.1 A propagating square wave (black) approximated by first order finite difference (red) at $t = 32$ with $\theta = 1$	92
4.5.1 $p = 0$: Magnitude of coefficients of leading order error terms $X_2K^2 + iX_3K^3$	113
4.5.2 $p = 1$: Magnitude of coefficients of leading order error terms $iX_4K^4 + X^5K^5$	115

List of Tables

2.1	Approximations to roots ξ_i^+ , $i = 1, \dots, p + 1$, of the right Radau polynomial $R_{p+1}^+(\xi)$	48
2.2	Approximations to roots ξ_i^* , $i = 1, \dots, p + 1$, of $R_{p+1}^*(\xi)$	49
2.3	\mathcal{L}^2 - and \mathcal{L}^∞ -norms of errors before and after post-processing for case $p = 2$	67
2.4	\mathcal{L}^2 - and \mathcal{L}^∞ -norms of errors before and after post-processing for case $p = 3$	68
3.1	\mathcal{L}^2 and \mathcal{L}^∞ errors and order of accuracy for fourth-order DG solutions to the linear advection equation at time $T = 1$	82
3.2	\mathcal{L}^2 - and \mathcal{L}^∞ -errors and order of accuracy for fourth-order DG solutions to the linear advection equation at time $T = 157$	88
4.1	Leading dispersion and dissipation errors for $p = 0$	113
4.2	Leading dispersion and dissipation errors for $p = 1$	114
4.3	Leading dispersion and dissipation errors for $p = 1$	115
4.4	$p = 2$; Coefficient of trailing dispersion error term $X_7 K^7$ in equation (4.5.8)	119

4.5	$p = 2$; Coefficient of trailing dissipation error term $X_8 K^8$ in equation (4.5.8)	119
4.6	Linear advection with $u_0(x) = \cos(4x)$ at $T = 400\pi$. Dissipation errors of RKDG methods for $p = 1$ and $p = 2$ with different values of θ	120
4.7	Linear advection with $u_0(x) = \cos(4x)$ at $T = 400\pi$. Dissipation errors of methods for $p = 3$	121

Acknowledgements

“The only thing to do with good advice is pass it on. It is never any use to oneself.”

– Oscar Wilde

First and foremost, I would like to thank Jennifer Ryan, my primary supervisor. Her patience, support and faith in me saw me through the slow periods and low times and heightened the peaks. My journey along this otherwise undulating road was smoothed as a result. I greatly appreciate the freedom afforded me. I am also grateful to David Seal. He engaged with a problem I was working on and rapidly helped me move forward at a time when it was important for me to do so. The Air Force Office of Scientific Research provided funding for this project for which I am thankful. It has brought me into contact with many great people. Several of those characters were at UMass Dartmouth. Sigal Gottlieb is somebody one wants to impress but somebody who imposes not a trace of judgement in being so. This is a rare thing. Zack Grant was a stable central part of my good experience. I have shared philosophical discussions about finite difference schemes on short-distance roadtrips with few other people. Thanks also go to Xinghui Zhong, Jingmei Qiu and Wei Guo for sending me in the right direction when I first started looking at Fourier analysis of DG methods and for the immediate sympathy in their faces when I confided in them as to how problematic it can be. The Learning Support Office at UEA, in particular Sidney, should be recognised for their work and for

their good-spirited sufferance of their direct dealings with me. The HPC people worked to good effect in the shadows. Thanks to my examiners, Tristan and Richard, for reading this document and for some useful comments.

There are a great number of people to whom I owe a debt of gratitude for their indirect, to varying degrees, help in getting me here. Mathematically speaking, it wouldn't have happened without Dave Crocker. Every school needs a teacher with a continued and evident fundamental interest in maths. It is extremely valuable and I'm grateful for the time he afforded me. Every kid needs a chance (or two); I remain indebted to Ann Cox for her advocacy of me. For his impact on me from day one at UEA, I thank Mark Cooker: a pedagogical luminary whose contemplative approach and breadth of knowledge and interest is inspirational. For conversations about mathematics, I recognise Tom Coleman at UEA and the guys and gals from the "Wizard Lab" at UCSD – especially Dr. Perry and Harrison – and all the fellow survivors of the Differential Geometry class that introduced to me people who are truly both very clever and very hard-working. I appreciate the positive attributes of my academic family: "brother" Xiaozhou Li (who is never too far from a smile), "sister" Thea (who I thank for sharing and always being there if needed), "twin sister" Julia (whose love is tough) and "uncle" Xiong Meng, all of whom I hope to stay in touch with. To finish on the same day as my boy Alberto capped the whole experience nicely. I hope we get on a big wall together one day.

Thanks to Bastien for genuinely offering help. *Chapeau* to all those who've provided, knowingly or not, regular escapism: my dearest footballers for their tolerance of my unelected captaincy on the pitch; Roy and Paul at Broadland Chess Club, and Gerald – who saw DG around the time of its inception – for questioning the wisdom of miring myself in a Philidor swamp before being able to swim in the gnat's sea; Sam and Charlotte for making it so easy to third-wheel; the "Belay Babes"; Tom and Rob at SDMC for facilitating the best of summers; Tommy D for never giving up; Helen; and Eleni for the song of the butterfly. Thanks also to my office mates (and to Mal) for their quiet tolerance

and everyone who put me up towards the end, most notably Vanessa and Hannah. I also appreciate all those who've added some spice, one way or another: Mac; Ana; Stephen 'D'C; Claire; Jason; Emma; J & K, especially J; Ben & Amanda (if I'd have come to your wedding, I wouldn't be writing this); Lucy; SamD; Dan; ProMc; "Leon"; Snoop Moyes; Dolores; Kez; Sabrina; Imo; Smidge; Rich; Andy; Ed; Phil; ToM; Iz; Faye; Kerry; Ash; Phoebe; Alex; JohnHas; and, of course, Stace. This thesis is dedicated to the Hooe Boys (to whom I owe no debt of gratitude). My eternal love to Chelsea for "letting" me do this and for everything I learned from you.

Finally, to those who knowingly helped in times of need: thank you so much! Without Hannah's friendship and hospitality, everything would have been measurably more difficult. Appropriate words to express my gratitude to my parents is a most pressing and serious topic for further work.

*“I was not proud of what I had learned but I never doubted
that it was worth knowing.”*

– ‘Dr.’ Hunter S. Thompson, *The Rum Diary*

Introduction

“Our life is frittered away by detail. Simplify, simplify.”

– Henry David Thoreau, *Walden and Other Writings*

In the last few decades, the discontinuous Galerkin method has seen a steady rise in popularity in response to the pursuit of a class of stable methods qualified to properly resolve partial differential equations that display convective or diffusive qualities. Our interests have come to be set predominantly in determining the ability of a method to accurately propagate a wave: the degree to which amplitude and peaks are preserved. If the numerical wave number – perturbed by the discretisation of a continuous problem – does not equal its exact counterpart, the approximation will exhibit phase errors (dispersion) and unwanted changes in amplitude (dissipation). These types of errors can lead to a numerical solution of a qualitatively divergent nature to the true solution.

The discontinuous Galerkin (DG) method has good wave resolution properties since alongside its trump cards as a robust and computationally efficient method is a desirable peculiarity, an ace in the hole. The celebrated property of superconvergence – a faster than generally expected convergence rate at certain points – is a result of a felicitous choice of numerical flux function and, as such, it is an intrinsic property of the DG spatial discretisation. As these “hidden” points are evolved in time, seeking to exploit their localised fortunes may seem elusive – a *will-o'-the-wisp* – yet superconvergence can be extracted to a global measure, and subsequently unearthed from its cache in a negative-order norm,

with an application of a specialised filter at the final time.

The roles of the temporal discretisation and a numerical flux function in accurately simulating a wave can become more prominent with a long time integration. While superconvergence is created by the numerical flux function in the spatial discretisation, it can be destroyed by an indiscriminate time evolution if the time-stepping method, which is often completely decoupled from the DG discretisation itself, is not of sufficiently inflated order. In this work, we wish to view the selection and design of the time integrator as dependent on the spatial discretisation since we are, after all, solving a PDE. The results of the literature suggest that an improved pairing between the spatial and temporal discretisations is both desirable and possible and depends on the specific problem and time regime. Moreover, we aspire to exploit the inherent property of superconvergence to better pair DG with the time-stepping. We describe the framework within which a specialised pairing may be realised through analysis of three different but related manifestations of superconvergence: the local, super-accurate points themselves; the subsequent global extraction via the Smoothness-Increasing Accuracy-Conserving (SIAC) filters; and, climactically, the spectral properties that quantify, in terms of dispersion and dissipation errors, the varying degrees to which different pairings accurately convect a wave. Superconvergence of high-order methods for solving hyperbolic PDEs that exhibit wave-like behaviour is an area of enduring interest. This thesis seeks to illuminate the relation between the local, post-processed and spectral analyses of superconvergence of DG methods and the role of the flux function and time-stepping scheme in achieving a superconvergent error.

1.1 Problem Statement

In this work, we consider numerical solutions to the hyperbolic system

$$u_t + \sum_{i=1}^d f_i(u)_{x_i} = 0, \quad (\mathbf{x}, t) \in \Omega \times (0, T],$$

together with appropriate initial and boundary conditions. For such systems, the Jacobian $f'(u)$ is diagonalisable with real eigenvectors. Hyperbolic equations present particular challenges since the solution may contain discontinuities even in spite of a completely continuous initial condition. Furthermore, the weak formulation does not give rise to a unique solution. Throughout most of this document, we make the following simplifications:

- uniformity of the spatial mesh and the time-step size
- single spatial dimension ($d = 1$)
- linear flux function $f(u) = cu$, often with $c = 1$
- periodic boundary conditions
- smooth initial condition.

Whilst the post-processed superconvergence proofs are quite robust with respect to these simplifications, without any one of them, the Fourier analysis in Chapter 4 becomes considerably more difficult, often prohibitively so. Translation invariance is required since we consider only a single mode within the infinite Fourier sum. Without periodicity, one can suffer from the Gibbs effect. Analyticity of the solution is also a commonly assumed property in proofs in order to allow for power series expansions. Finally, whilst non-linearity would make any result in this work more difficult to obtain, construction of a stable upwind-biased flux for non-linear equations is a topic of ongoing research so a similar study in this case is not yet possible.

The need for high-order methods is generally on the rise in light of the growing sophistication of the simulation sciences; required performance in these disciplines is rising more rapidly than the performance of existing computer architectures. Moreover, modern simulations often work with complex and large scale geometries which require a great deal of flexibility and robustness. Long time integration is often needed and calculations must be performed more and more accurately.

A numerical countermeasure to this situation is offered by discontinuous Galerkin schemes. However, a number of fundamental areas retain unresolved questions. How should the numerical flux be chosen? Are the schemes stable? How should the time-stepping scheme be paired with the DG spatial discretisation? This thesis asks how superconvergence is affected by, and it can in turn inform, the choice of flux and the coupling with a time-stepping scheme. In this sense, how can we get more from this class of high-order methods?

1.2 Literature Review

The discontinuous Galerkin methods (DGM) have become, since their inception in the 1970s, one of the most extensively researched classes of numerical methods for solving partial differential equations (PDE) and have been popularly adopted by the scientific and engineering communities as a method capable, given an appropriate choice of time integrator, of achieving arbitrary orders of accuracy in space. The method was introduced in 1973 by Reed and Hill [48] in the context of neutron transport equations while the first analysis of the DGM, where the optimal convergence rate for rectangular meshes was proved, was undertaken by LeSaint and Raviart [40] in the following year. Development of the theory supporting DG solutions to hyperbolic equations, including nonlinear conservation laws, was completed by Cockburn, Shu and others in [26, 27, 21, 45, 61, 3, 4, 5]. In these works, it was proposed that the DGM be paired with Runge-Kutta time-stepping methods to obtain the fully

discrete solution.

As a hybridisation of the finite element and finite volume schema, DGM seek to profit from assets of both of the parent frameworks. The finite element structure retains the ability to cope with complicated geometries while the monotone numerical fluxes (or approximate Riemann solvers) at cell boundaries, chosen in a finite volume manner using piecewise polynomial bases, allow for high resolution of discontinuities. The geometric flexibility afforded by allowing discontinuities across element boundaries comes at a considerable computational expense characterised in part by the restrictive Courant-Friedrichs-Lewy (CFL) condition. If higher order spatial derivatives are present in the original PDE, a particularly severe time-step restriction must be imposed. In this case, when an ingenuous application of DGM can result in a catastrophic breakdown of the method's stability, one may consider using the local discontinuous Galerkin (LDG) method, an extension of the DGM designed by Cockburn and Shu [27].

The DG solution can develop oscillations near discontinuities. In order to control these spurious phenomena, a great deal of work has been put into developing limiters, and these are often expensive to apply. The fewer stages involved in a time integrator, the fewer times we need to compute with the limiter. The treatment in the hyperbolic conservation law course of LeVeque [41] will suffice for this work as background on limiters as we apply them only when needed in our numerical experiments. A thorough detailing of work from the previous millennium is offered by the lecture notes of Cockburn, Karniadakis and Shu [24].

The choice of numerical flux function plays a pivotal role in the successful construction of DG methods. While this function is chosen to guarantee the stability of the scheme, it has an intrinsic effect on the acclaimed superconvergent properties. The vast majority of theory for DG schemes for conservation laws has been developed with the (somewhat habituated) choice of a monotone numerical flux. Design and criteria for selection of numerical flux functions is an area with a great deal of scope for further investigation.

Even for a Runge-Kutta (RK) DG solution to the linear advection equation, one can do better than the “natural” upwind flux. Recently, Meng, Shu and Wu [44] introduced in the context of DG methods for linear hyperbolic equations a more general flux function: the upwind-biased flux. This function parametrises the ratio of information taken from the left compared to the right of cell interfaces. This choice avoids the requirement of exact knowledge of the eigenstructure of the Jacobian and may reduce numerical dissipation (yielding a better approximation in smooth regions) but it is made at the cost of the loss of monotonicity. In [44], \mathcal{L}^2 -stability and optimal $\mathcal{O}(h^{p+1})$ convergence results, where p is the order of the piecewise polynomial basis and h is the element width, for the periodic and inflow boundary conditions and for multiple-dimensions were obtained and are comparable with those for the upwind scheme [49]. Numerical experiments include non-uniform meshes, while a treatment of nonlinear equations is left to further work. Other technical challenges set in [44] include defining a suitable projection of the exact solution into the approximation space such that superconvergence results analogous to those for the purely-upwind case may be derived. These results provide the theoretical foundations for our investigations into the choice of flux function for RK-DG methods.

As an inherent property of the spatial discretisation, superconvergence can only be retained in the solution by a sensitive pairing with a time integrator. The most popular option is to pair DG with a Runge-Kutta ODE solver to produce a Method of Lines (MoL). Due to the *strong stability preserving* (SSP) nature of the RK methods often employed, much of the analysis for the completely decoupled RK-DG schemes has been undertaken on the semi-discrete scheme only. The main alternative, favoured for diffusion problems, is to perform the spatial and temporal discretisations together by coupling DG with a Lax-Wendroff (LW) Taylor-type time discretisation. Motivated by the local discontinuous Galerkin (LDG) method of Cockburn and Shu (SINUM, 1998), the original LW-DG method was introduced in 2005 by Qiu et al. in [46]. This

scheme uses Taylor series to replace temporal derivatives by spatial ones before applying the DG discretisation, resulting in a single-step procedure which respects the original problem in the discretisations. However, this method is often outperformed by the RK-DG method and comes at a higher computational cost.

Guo, Qiu and Qiu (2015) [31] studied the superconvergence of Lax-Wendroff DG schemes. These Taylor-type methods are qualitatively different to a MoL in that the two discretisations are intertwined so analysis must be performed on the fully-discrete scheme. The original scheme of Qiu et al. (2005) computes higher-order spatial derivatives directly and is not superconvergent but, using auxiliary variables as in the LDG methodology, superconvergence similar to that seen in RK-DG schemes is recovered in [31].

The class of multi-stage multi-derivative methods presented in [51] can be considered as a unification of two opposing time-stepping procedures: the Runge-Kutta (Method of Lines) and Lax-Wendroff (Taylor) discretisations. In Chapter 3, we extend the approach of [31] to define a new multiderivative method with superconvergent properties. While we lose some of the portability of the original method, the new scheme has demonstrably superior wave propagation properties.

The DGM has been shown to be order $p + 1$ accurate [28], where p is the order of the piecewise polynomial basis. However, it turns out that at certain points, the DGM achieves a higher than expected order of accuracy: the so-called superconvergence property. Recently, it has been observed that the superconvergence property depends on the flux used to construct the discontinuous Galerkin method, specifically for the LW-DG method [47]. Several different approaches have been made to explore superconvergence. Results may be pigeon-holed into three categories: pointwise superconvergence, superconvergence in a negative-order norm, and superconvergence towards a special projection of the solution.

The pointwise superconvergence proofs include a wide class of equations (elliptic, parabolic and hyperbolic) [3, 4, 5, 7, 8, 11]. Original speculation regarding the superconvergence of DG approximations at Radau points was given by Biswas et al. [12]. For the upwind flux, these points are roots of right-Radau polynomials where the approximation exhibits $\mathcal{O}(h^{2p+1})$ superconvergence at the outflow edge and $\mathcal{O}(h^{p+2})$ at roots in the interior of the element [2, 6, 5]. Lowrie [43] also noted that a component of the error converges with $\mathcal{O}(h^{2p+1})$. One approach is to inspect the behaviour of the DGM when applied to an initial value problem or boundary value problem. Adjerid et al. [3] showed that, for the ODE $u' - cu = 0$, the DG solution on a uniform mesh has errors on the order of h^{p+2} at Radau points and on the order of h^{2p+1} at the downwind points. Moreover, the leading term of the spatial discretisation error is shown to be proportional to a Radau polynomial of degree $p + 1$ at the downwind points, and at these points the local and global errors are on the orders of h^{2p+2} and h^{2p+1} . Convergence to the Radau polynomial also occurs in the polynomial degree p .

We follow the procedure outlined in [11] to obtain *a posteriori* error estimates, defining a new Radau polynomial parametrised by the same measure as is used in the upwind-biased flux. This approach requires a global initial projection, as in [44], which is complicated by the multi-element nature of the upwind-biased flux. We prove that the method is locally $\mathcal{O}(h^{p+2})$ superconvergent at roots of a linear combination of the left- and right-Radau polynomials. This linear combination depends on the value of the parameter used in the flux. In order to define a proper global initial interpolation for odd-degree polynomials, the range of the flux parameter must be extended beyond that given in [44]. We demonstrate numerically that, for simple transport, as more information from the direction counter to the wind direction is included at cell boundaries, the discretisation errors can be improved. This choice does, however, have implications for the CFL number.

The “hidden” local accuracy of the DG solution may be extracted to a global measure by applying a post-processing at the final time. Bramble and Schatz [13] developed a local post-processing technique that utilises information offered by the negative-order Sobolev norm. Negative-order norm error estimates are related to extracting the “hidden” superconvergence from special points. In the context of DG approximations for linear hyperbolic equations, this technique was described by Cockburn, Luskin, Shu and Süli in [25], where it was shown that the DG solutions converge with order $2p + 1$ in the negative-order norm. It was later extended and developed by Ryan and others [56, 36] to treat nonlinear equations, non-periodic boundary conditions and non-uniform meshes. This extended version was relabelled as the Smoothness-Increasing Accuracy-Conserving (SIAC) filter.

The superconvergence result can also be achieved in the \mathcal{L}^2 -norm. This involves convolving the approximation against a specially designed kernel comprising a linear combination of B-splines, effecting increased smoothness by damping the non-physical eigenmodes of the DG operator and exploiting information concealed in the unwelcome fluctuations that characterise the numerical solution. Analysis of the SIAC filtered error, which is facilitated by a dual analysis in a similar fashion to [37], is largely uncomplicated by the upwind-biased flux. In support of the pointwise observations, we prove that the use of SIAC filters is still able to draw out the superconvergence information and create a globally smooth and superconvergent solution of $\mathcal{O}(h^{2p+1})$, demonstrating that the price paid for the introduction of the flux parameter is limited to a contribution to the constant attached to the post-processed error term.

Other superconvergence results include those of Cheng and Shu [21], Yang et al. [61, 62], Meng et al. [45] and Cao et al. [16]. These results include a description of $\mathcal{O}(h^{p+2})$ superconvergence towards a special projection of the solution (so-called *supercloseness*), a fruitful area of recent research which can also make use of the negative-order norm. In the case of a linear scalar problem with DG

approximation using piecewise linear polynomials and an upwind flux, Cheng and Shu [21] showed that the DG solution is superconvergent (in the \mathcal{L}^1 , \mathcal{L}^2 and \mathcal{L}^∞ norms) towards the Radau projection of the exact solution. Indeed, the DG solution is closer to the Radau projection of the exact solution than the solution itself. The DG error, which is shown not to grow over a long time period $\mathcal{O}(1/h)$, is decomposed into two parts: the superconvergent part which grows (at most) linearly in time, and another part which does not grow in time. This result for superconvergence is desired in the general case of p^{th} order polynomials. The error decomposition is connected in [32] to analysis ([68]) via the Fourier approach.

Recent interest [68, 64, 63, 52, 60, 32, 42] in analysis via a Fourier approach of DG solutions to the linear advection equation offers an alternative means by which to explore superconvergence. This analysis is limited to linear equations with periodic boundary conditions and a uniform mesh. However, as is justified by numerical examples [32], the results provide a guide for the behaviour of solutions in a more general setting. Stability and $(p + 1)^{\text{th}}$ order accuracy can be established via this approach while the analysis in [68] provides a quantitative superconvergent error estimate on the order of h^{p+2} at Radau points and on the order of h^{2p+1} at the downwind points.

A p^{th} -order DG solution to the linear advection equation has one physical mode and p spurious ones which are damped exponentially fast over time [34]. Guo et al. [32] derive the amplification matrix of a DG spatial discretisation and decompose the error e of the DG approximation at the final time $t = T$ into three parts:

$$\|e\| \leq C_1 T h^\sigma + C_2 h^{p+1} + C_3 \exp\left(-\frac{CT}{h}\right) h^{p+1}, \quad (1.2.1)$$

where $C_1, C_2, C_3, C \in \mathbb{R}_+$. The first term on the right-hand side of inequality (1.2.1), which dominates for $T = \mathcal{O}\left(\frac{1}{h^{\sigma-p-1}}\right)$, is attributed to the dispersion and dissipation errors of the physically relevant eigenvalues and grows linearly in time. The expected order of accuracy is $\sigma = p + 1$; however, a

judicious choice of numerical flux function in the semi-discrete DG scheme can yield a *superconvergent* order of accuracy as high as $\sigma = 2p + 1$. The third term, which decays exponentially fast over time with respect to h , accounts for dissipation of the spurious modes. The second term is due to projection of the initial condition and does not grow in time. Thus the error is on the order of $2p + 1$ for long time integration but only $p + 1$ over short time. At certain special points – the superconvergent points which change with the choice of numerical flux – the accuracy of inequality (1.2.1) can be increased to $\mathcal{O}(h^{p+2})$ by carefully interpolating the initial projection.

To illuminate the contribution of the flux function to accurate wave propagation, we include results for the upwind-biased scheme in our comparison of eigenvalues in Chapter 4. The flux parameter is chosen to obtain favourable dispersion and dissipation properties, in particular for long-time integration. While these results are comparable to those in [68] for the upwind flux, we demonstrate how to choose the parameter such that the coefficients in the expressions for the physically relevant eigenvalues are decreased. This is the first time that consideration of the flux has been directly included in dispersion analysis of DG schemes.

Fourier analysis for the fully discrete error under various SSPRK schemes is provided in [68]. This is used to compute the number of points per wavelength necessary to obtain a fixed error. Analytical dispersion and dissipation errors are derived in [60] for the RK-DG method (not SSPRK) and it is found that the DG spatial discretisation, with small CFL, contributes to superconvergence while the RK time discretisation reduces it. The authors suggest that this is due to the respective finite element and finite difference natures of the methods. A separate issue not treated in these papers is the consideration of time discretisation errors.

A Fourier-type analysis in [60] contrasts the RK-DG and the original LW-DG schemes: for the second-order methods, RK is more dissipative than LW and vice versa for the third-order methods. In Chapter 4, we study the dispersion and dissipation errors of a class of multi-stage multiderivative discontinuous Galerkin

methods that sit halfway between RK- and LW-DG for solving a linear advection equation over a long time period.

1.2.1 Outline of thesis

The outline of this thesis is as follows: in Section 1.3, we discuss preliminaries and review the construction of discontinuous Galerkin scheme, paying some particular attention to the upwind-biased flux. In Chapter 2, we present pointwise and post-processed superconvergence results for the upwind-biased flux. Options for time integration are discussed in Chapter 3. These include the establishment of a new multi-stage multi-derivative method which we call the modified TDRK4 scheme. In Chapter 4, we perform dispersion analysis via Fourier approach of the DG scheme with upwind-biased flux and the TDRK4-DG schemes. We support this analysis with numerical examples throughout and conclusions in Chapter 5.

1.3 Background

In the remainder of this chapter, we discuss preliminaries to the DG formulation and perform the semi-discretisation of the hyperbolic conservation law.

1.3.1 Notation and definitions

We first define the approximation spaces associated with the discretisation of the spatial domain.

Function spaces

The DG method seeks a numerical solution belonging to the approximation space

$$V_h^p = \{v \in \mathcal{L}^2(\Omega) : v|_S \in \mathcal{P}^p(S), \forall S \in \Omega_h\}, \quad (1.3.1)$$

where Ω_h is a tessellation of a domain $\Omega \subset \mathbb{R}^d$ into elements S , $\mathcal{P}^p(S)$ is the space of polynomials of degree at most p in each variable on an element S and where $\mathcal{L}^2(\Omega)$ is the space of square-integrable functions on the domain Ω . The piecewise polynomial nature of the approximation space V_h^p allows discontinuities across cell boundaries resulting in an ability to better approximate sharp gradients.

In multiple dimensions, due to the tensor product nature of the post-processing kernel, we also require the function space of tensor-product polynomials $\mathcal{Q}^p(S)$ of degree at most p in each variable. This necessitates the use of the following finite element spaces:

$$W_h^p = \{\phi \in \mathcal{L}^2(\Omega) : \phi|_S \in \mathcal{Q}^p(S), \forall S \in \Omega_h\},$$

$$\Sigma_h^p = \{\eta = (\eta_1, \dots, \eta_d)^T \in (\mathcal{L}^2(\Omega))^d : \eta_l \in \mathcal{Q}^p(S), l = 1, \dots, d; \forall S \in \Omega_h\}.$$

Nevertheless, we mention here that it has been observed ([50]) that the filter also works for the standard polynomial space $\mathcal{P}^p(S)$. Note that for a one-dimensional domain $\Omega = I$, these function spaces $\mathcal{Q}^p(S)$ and $\mathcal{P}^p(S)$ agree.

Operators on the function spaces

We list the following standard notations. The inner-product over Ω of two functions is defined as

$$(w, v)_\Omega = \sum_S \int_S wv \, dS; \quad (\mathbf{p}, \mathbf{q})_\Omega = \sum_S \int_S \mathbf{p} \cdot \mathbf{q} \, dS$$

depending on whether the functions take scalar or vector values. We denote by $\mathbb{P}_h v$ the usual \mathcal{L}^2 -projection of a function v .

The \mathcal{L}^2 -norm on the domain Ω and on the boundary $\partial\Omega$ is defined as

$$\|\eta\|_\Omega = \left(\int_\Omega \eta^2 \, d\mathbf{x} \right)^{1/2}; \quad \|\eta\|_{\partial\Omega} = \left(\int_{\partial\Omega} \eta^2 \, ds \right)^{1/2}$$

and the ℓ -norm and semi-norm of the Sobolev space $\mathcal{H}^\ell(\Omega)$ are defined respectively as

$$\|\eta\|_{\ell,\Omega} = \left(\sum_{|\alpha| \leq \ell} \|D^\alpha \eta\|_{\Omega}^2 \right)^{1/2}; \quad |\eta|_{\ell,\Omega} = \sum_{|\alpha| \leq \ell} \|D^\alpha \eta\|_{\infty,\Omega}, \quad \ell > 0,$$

where α is a d -dimensional multi-index of order $|\alpha|$ and where D^α denotes multi-dimensional partial derivatives. The definitions for the above norms for vector-valued functions are analogous to the scalar case.

The negative-order Sobolev norm is defined as

$$\|\eta\|_{-\ell,\Omega} = \sup_{\Phi \in \mathcal{C}_0^\infty(\Omega)} \frac{(\eta, \Phi)_\Omega}{\|\Phi\|_{\ell,\Omega}}.$$

Note that for all $\ell \geq 1$, by definition and the Cauchy inequality we easily obtain

$$\|\eta\|_{-\ell,\Omega} \leq \|\eta\|_{\Omega}.$$

The negative-order norm measures the higher order modes of a function and can be used to detect oscillations ([25]). In Section 2.2, we discuss the SIAC filter which smooths oscillations in the error and uses the negative-order norm as a means of obtaining \mathcal{L}^2 -error estimates for the filtered solution.

Finally, the difference quotients $\partial_{h,j} v$ are given by

$$\partial_{h,j} v(\mathbf{x}) = \frac{1}{h} \left[v(\mathbf{x} + \frac{1}{2} h \mathbf{e}_j) - v(\mathbf{x} - \frac{1}{2} h \mathbf{e}_j) \right],$$

where \mathbf{e}_j is the j^{th} component unit normal vector. For any multi-index $\alpha = (\alpha_1, \dots, \alpha_d)$, we define the α^{th} -order difference quotient by

$$\partial_{h,j}^\alpha v(\mathbf{x}) = \left(\partial_{h,1}^{\alpha_1} \cdots \partial_{h,d}^{\alpha_d} \right) v(\mathbf{x}).$$

Projection and interpolation properties and known error estimates

Theorem 1.3.1. *Cauchy-Schwarz Inequality:* For $w, v \in \mathcal{L}^2(\Omega)$, we have

$$|(w, v)_\Omega| \leq \|w\|_\Omega \|v\|_\Omega. \quad (1.3.2)$$

Theorem 1.3.2. *Young's Inequality with ϵ :* Let $1 < p, q < \infty$ with $\frac{1}{p} + \frac{1}{q} = 1$. Then, for all $\epsilon > 0$ and for all $a, b \in \mathbb{R}$, there holds

$$ab \leq \frac{\epsilon}{p} a^p + \frac{1}{\epsilon q} b^q. \quad (1.3.3)$$

Theorem 1.3.3. *Cauchy Inequality:*

$$\sum_i a_i b_i \leq \left(\sum_i a_i^2 \right)^{1/2} \left(\sum_i b_i^2 \right)^{1/2}. \quad (1.3.4)$$

1.3.2 Hyperbolic conservation law

Consider the hyperbolic system

$$u_t + \sum_{i=1}^d f_i(u)_{x_i} = 0, \quad (\mathbf{x}, t) \in \Omega \times (0, T], \quad (1.3.5)$$

for the conserved quantity $u(\mathbf{x}, t)$, where $\mathbf{x} = (x_1, \dots, x_d) \in \Omega \subset \mathbb{R}^d$. Since we do not seek to consider the effect of the boundary conditions, we assume periodicity. For the initial condition, much of the analyses require only that $u_0(\mathbf{x}) \in \mathcal{H}^{p+1}(\Omega)$ but for some of the proofs, we require infinite differentiability,

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \in \mathcal{C}^\infty(\Omega),$$

in order to write the DG solution as a Maclaurin series. Even with a completely continuous initial condition, equation (1.3.5) can develop discontinuities.

For simplicity and ease of exposition, much of the discussion will be framed in a

single spatial dimension with the equation

$$\begin{aligned} u_t + f(u)_x &= 0, & x \in I = [0, 2\pi], \quad t \in (0, T), \\ u(x, 0) &= u_0(x), & u(0, t) = u(2\pi, t), \end{aligned} \quad (1.3.6)$$

arising from setting $d = 1$ in equation (1.3.5). Furthermore, many of the results are limited to the *linear advection equation*

$$u_t + cu_x = 0, \quad c > 0, \quad u(x, 0) = u_0(x), \quad (1.3.7)$$

for which the exact solution

$$u(x, T) = u_0(x - cT)$$

is known analytically. This permits otherwise prohibitively complicated analysis of the accuracy of numerical solutions to this problem whilst the resulting observations can be used to provide guidance for more complicated settings.

Outline of the Method of Lines Approach

The predominant idea in treating the equation (1.3.6) is to discretise the problem in space and in time. A Method of Lines solver keeps separate the temporal and spatial discretisations. We discretise first in space

$$u_t = -f(u)_x,$$

approximating the continuous function u by a discrete numerical solution u_h . For hyperbolic conservation laws, which may develop shocks, a particularly attractive choice of spatial discretisation is the DG method, the details of which we present later in this chapter. In this way, we define a spatial discretisation operator

$$L(u_h) := -f(u_h)_x,$$

forming a large ODE system

$$\frac{\partial}{\partial t} u_h = L(u_h).$$

Once this system has been resolved into its component parts, the second stage of the process is completed by an ODE integrator. For hyperbolic conservation laws in particular, this is commonly taken to be an explicit RK method due to their ease of implementation and relatively high computational efficiency. With an abuse of notation, we now write

$$\frac{d}{dt} u_h = \mathcal{L}(u_h) \tag{1.3.8}$$

when the Fundamental Theorem of Calculus gives

$$u_h^{n+1} = u_h^n + \int_{t_n}^{t_{n+1}} \mathcal{L}(u_h) dt,$$

where $u_h^n = u_h(x, t_n)$ is the approximation after the n^{th} level of time-stepping.

We may approximate the above integral by an s -stage Runge-Kutta method

$$\int_{t_n}^{t_{n+1}} \mathcal{L}(u_h) dt \approx \Delta t \sum_{i=1}^s b_i \mathcal{L}(u_h(t_n + c_i \Delta t)),$$

the simplest of which is the single stage Euler's method:

$$u_h^{n+1} = u_h^n + \Delta t \mathcal{L}(u_h^n). \tag{1.3.9}$$

We describe a certain class of RK methods, named Strong Stability Preserving RK methods, and further discuss timestepping in Chapter 3. One of the main attractions of using SSP time discretisations is that they can be written as convex combinations of first-order forward Euler steps, significantly simplifying proofs for nonlinear stability. In the remainder of this chapter, we describe in detail the DG discretisation of the spatial derivative.

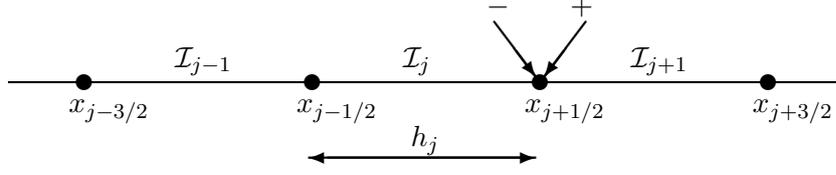


Figure 1.3.1: One-dimensional elements in a segment of the tessellation

1.3.3 Tessellation

Consider a discretisation \mathcal{I}_h of the one-dimensional bounded domain $I = [a, b]$ into N cells \mathcal{I}_j of length h_j such that the cells take the form

$$\mathcal{I}_j = \left[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}} \right], \quad j = 1, \dots, N,$$

with $a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \dots < x_{N+\frac{1}{2}} = b$. Denote the cell centres by

$$x_j = x_{j-\frac{1}{2}} + \frac{1}{2}h_j.$$

For a function $v(x)$ defined on both \mathcal{I}_j and \mathcal{I}_{j+1} , we define $v_{j+\frac{1}{2}}^- = \lim_{x \rightarrow x_{j+\frac{1}{2}}^-} v(x)$ and, similarly, $v_{j+\frac{1}{2}}^+ = \lim_{x \rightarrow x_{j+\frac{1}{2}}^+} v(x)$.

For computational convenience, we apply the linear mapping

$$\xi : \mathcal{I}_j \mapsto [-1, 1], \quad \xi(x) = \frac{2}{h_j}(x - x_j) \quad (1.3.10)$$

of each element to the canonical element $\xi \in [-1, 1]$.

For simplicity, and out of necessity for many of the approaches we take to analysis of the schemes, we consider cells of uniform length

$$h = \Delta x = \frac{b - a}{N}.$$

The Jacobian of the scaling (1.3.10) is $\frac{h}{2}$ in this case.

Much of the discussion that follows will focus on a single spatial dimension but the tessellation is easily extended to multiple dimensions. Specifically, let Ω_h be

a tessellation of a d -dimensional bounded domain Ω into elements S of regular quadrilateral-type shape. Denote by $\partial\Omega_h = \bigcup_{S \in \Omega_h} \partial S$ the union of boundary faces ∂S of the elements $S \in \Omega_h$. A face e internal to the domain has associated with it “left” and “right” elements S_L and S_R and exterior-pointing normal vectors $\mathbf{n}_L = (n_1^L, \dots, n_d^L)$ and $\mathbf{n}_R = (n_1^R, \dots, n_d^R)$ respectively as described in [37]. Given a function v defined on neighbouring elements S_L and S_R which share a face e , we refer to its restriction in S_L to the face e by writing $v^L := (v|_{S_L})|_e$ and similarly for v^R , the restriction of v to e in S_R .

In two dimensions for example, let $\Omega = [-1, 1]^2$ and construct $N_x \cdot N_y$ rectangular cells $S = I_i \times J_j$ comprising the product of intervals

$$I_i = \left(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right) \quad \text{and} \quad J_j = \left(y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}\right)$$

for $1 \leq i \leq N_x$ and $1 \leq j \leq N_y$ respectively. The cell lengths are given by $h_{x,i} = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, $1 \leq i \leq N_x$, and $h_{y,j} = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}$, $1 \leq j \leq N_y$, subject to the regularity condition $h_{x,i}, h_{y,j} \geq ch$, $c > 0$, where we define

$$h = \max \left\{ \max_{1 \leq i \leq N_x} h_{x,i}, \max_{1 \leq j \leq N_y} h_{y,j} \right\}.$$

In passing, we give inverse inequalities on the cells and cell boundaries.

Theorem 1.3.4. *Inverse estimates:*

$$\|v_x\|_{I_j} \leq \mu_1 h^{-1} \|v\|_{I_j} \tag{1.3.11}$$

$$\|v\|_{\partial I_j} \leq \mu_2 h^{-1/2} \|v\|_{I_j}. \tag{1.3.12}$$

With the spatial mesh defined, we are now ready to discuss the DG discretisation.

1.3.4 Discontinuous Galerkin semi-discretisation

The most straightforward pairing of the spatial and temporal discretisations is to treat them separately: discretise the spatial derivatives to transform the governing PDE to an ODE then step the solution forwards in time with an ODE solver as described in §1.3.2. An alternative is to intertwine the discretisations before specifying the spatial operator. In both cases, we require the DG discretisation of a spatial derivative w_x of some function w . The appropriate choice of DG discretisation amounts to selection of the numerical flux function. What follows is a full description of the DG semi-discretisation of the hyperbolic system (1.3.5).

Given a tessellation Ω_h of the d -dimensional domain Ω , multiply equation (1.3.5) by a test function v and integrate over an arbitrary element $S \in \Omega_h$, integrating by parts to obtain the *weak formulation*

$$\int_S u_t v \, dS - \sum_{i=1}^d \int_S f_i(u) v_{x_i} \, dS + \sum_{i=1}^d \int_{\partial S} f_i(u) v \, ds = 0. \quad (1.3.13)$$

Next, we assume that both the solution and test function belong to the finite dimensional approximation space of piecewise polynomials of degree at most p :

$$V_h^p := \{v \in \mathcal{L}^2(\Omega) : v|_S \in \mathcal{Q}^p(S), \forall S \in \Omega_h\}.$$

Note that functions $v \in V_h^p$ are allowed to be discontinuous across element boundaries. This is the distinguishing feature of DG schemes amongst finite element methods. Since the boundary terms in equation (1.3.13) are not well defined when u and v are in the approximation space V_h^p , they require special treatment. The test function values are taken from inside the cell and the solution values are often chosen to mimic properties of the system whilst ensuring stability.

By replacing in equation (1.3.13) the solution $u(\mathbf{x}, t)$ by a numerical

approximation $u_h(\mathbf{x}, t)$ such that $u_h(\cdot, t) \in V_h^p$, we obtain the discontinuous Galerkin method: find, for any $v \in V_h^p$ and for all elements S , the unique function $u_h(\cdot, t) \in V_h^p$ that satisfies

$$\int_S (u_h)_t v \, dS - \sum_{i=1}^d \int_S f_i(u_h) v_{x_i} \, dS + \sum_{i=1}^d \int_{\partial S} \widehat{f}_i(u_h) v \, ds = 0, \quad (1.3.14)$$

where \widehat{f} is a single-valued numerical flux function used to enforce weak continuity at the cell interfaces. We discuss the numerical flux function in §1.3.7.

Summing equation (1.3.14) over the elements S , we get a compact expression for the global scheme:

$$((u_h)_t, v)_{\Omega_h} + \mathcal{B}(u_h; v) = 0,$$

where we define for future use a bilinear operator

$$\mathcal{B}(u_h; v) := \sum_S \mathcal{B}_S(u_h; v), \quad (1.3.15)$$

where

$$\mathcal{B}_S(u_h; v) := - \sum_{i=1}^d (f_i(u_h), v_{x_i})_S + \sum_{i=1}^d (\widehat{f}_i(u_h) n_i, v)_{\partial S}. \quad (1.3.16)$$

Before discussing the numerical flux function \widehat{f} , we define the basis functions for the approximation space. For simplicity, we proceed with our discussion in a single dimension. For multiple dimensions, the basis simply consists of tensor products of the one-dimensional functions.

1.3.5 Basis functions for the approximation space

The numerical solution resulting from a DG discretisation of the domain can be written locally in terms of a set of basis functions for V_h^p . Different choices of basis may favour particular approaches to the analysis of the schemes. For the majority of this document, we employ the Legendre polynomials $P_n(\xi)$ that are

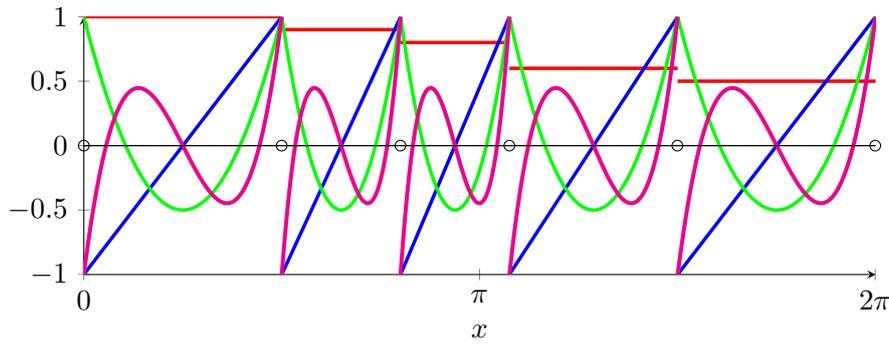


Figure 1.3.2: Legendre polynomials up to order 3

defined by the Rodrigues formula

$$P_n(\xi) = \frac{1}{2^n n!} \frac{d^n}{d\xi^n} ((\xi^2 - 1)^n), \quad -1 \leq \xi \leq 1, \quad n \geq 0,$$

and which satisfy the orthogonality condition

$$\int_{-1}^1 P_n(\xi) P_m(\xi) d\xi = \frac{2}{2n+1} \delta_{nm}, \quad (1.3.17)$$

where δ_{nm} is the Kronecker-delta function. The first few are

$$P_0(\xi) = 1; \quad P_1(\xi) = \sqrt{3}\xi; \quad P_2(\xi) = \frac{\sqrt{5}}{2}(3\xi^2 - 1); \quad P_3(\xi) = \frac{\sqrt{7}}{2}(5\xi^3 - 3\xi).$$

Further properties that we use include:

$$P_n(1) = 1; \quad P_n(-1) = (-1)^n; \quad \frac{d}{d\xi} P_n(1) = \frac{1}{2}n(n+1); \quad (1.3.18)$$

and

$$\frac{d}{d\xi} P_{n+1}(\xi) = (2n+1)P_n(\xi) + (2n-3)P_{n-2}(\xi) + (2n-7)P_{n-4}(\xi) + \dots \quad (1.3.19)$$

Figure 1.3.2 shows Legendre polynomials up to degree 3 on each of 5 elements in a tessellation of the interval $[0, 2\pi]$. The piecewise constant $P_0(\xi)$ is staggered to aid visualisation.

We also define here the right- and left-Radau polynomials

$$R_{p+1}^+(\xi) = P_{p+1}(\xi) - P_p(\xi), \quad R_{p+1}^-(\xi) = P_{p+1}(\xi) + P_p(\xi)$$

respectively. These are the difference and sum of consecutive Legendre polynomials. It is known that the roots

$$\xi_1^+ < \xi_2^+ < \dots < \xi_{p+1}^+ = 1$$

of $R_{p+1}^+(\xi)$ and the roots

$$-1 = \xi_1^- < \xi_2^- < \dots < \xi_{p+1}^-$$

of $R_{p+1}^-(\xi)$ are real, distinct and lie in the interval $[-1, 1]$.

1.3.6 Galerkin expansions

Modal discontinuous Galerkin methods track coefficients $U_j^{(\ell)}$ of basis functions $\varphi^{(\ell)}$. To ease computation, we usually map the element \mathcal{I}_j to $[-1, 1]$ via the scaling (1.3.10). In one dimension, the approximate solution when restricted to a single cell \mathcal{I}_j is given by the *Galerkin expansion*

$$u_h(x(\xi), t)|_{\mathcal{I}_j} := u_j(\xi, t) = \sum_{\ell=0}^p U_j^{(\ell)}(t) \varphi^{(\ell)}(\xi), \quad (1.3.20)$$

where p is the maximal polynomial order of the basis. Such an approximation is $(p+1)^{\text{st}}$ -order accurate.

We mention *en passant* that the Galerkin expansion of some given function, such as the flux function $f(u)$ or the initial condition $u(x, 0) = u_0(x)$, is computed by an \mathcal{L}^2 -projection onto the basis. That is, given $g(x, t)$, on each cell

\mathcal{I}_j require

$$\sum_{\ell=0}^p G_j^{(\ell)}(t) \int_{\mathcal{I}_j} \varphi_j^{(\ell)}(x) \varphi_j^{(m)}(x) \, dx = \int_{\mathcal{I}_j} g(x, t) \varphi_j^{(m)}(x) \, dx \quad (1.3.21)$$

for each $m = 0, \dots, p$. The integral on the left-hand side of equation (1.3.21) gives rise to a mass matrix which must be inverted while the integral on the right-hand side may be computed using a quadrature rule. The orthogonal Legendre basis leads to a diagonal mass matrix.

For example, given the approximate solution $u_h(x, t)$, compute the Galerkin expansion of $f(u_h)$ by the following process:

1. For each $j = 1, \dots, N$ and $i = 1, \dots, R$, compute at a set of R quadrature points $\bar{\xi}_i \in [-1, 1]$ on each cell \mathcal{I}_j the flux function point-values $f(u_j(x(\bar{\xi}_i), t))$.
2. After choosing local basis functions $\varphi^{(m)}(\xi) = P_m(\xi)$ of Legendre type, the Galerkin coefficients of $f(u)$ on the cell \mathcal{I}_j are given by

$$F_j^{(m)}(t) = \frac{2m+1}{2} \sum_{i=1}^R f(u_j(\bar{\xi}_i, t)) P_m(\bar{\xi}_i) w_i, \quad m = 0, \dots, p,$$

where w_i are the quadrature weights corresponding to the nodes $\bar{\xi}_i$.

3. The Galerkin expansion can be constructed from the coefficients:

$$f(u_h) = \sum_{\ell=0}^p F_j^{(\ell)}(t) \varphi^{(\ell)}(\xi).$$

To denote the vector of Galerkin coefficients $F_j^{(m)}(t)$ for a function f on a cell \mathcal{I}_j , we use a bold font:

$$\mathbf{F}_j := \left(F_j^{(0)}, \dots, F_j^{(p)} \right)^T.$$

The DG scheme (1.3.14) is parsed by substituting for u_h the Galerkin expansion (1.3.20) and setting the test function v to be a Legendre basis function $\varphi^{(m)}$, $m = 0, \dots, p$.

1.3.7 Numerical flux function

In order to ensure stability of the scheme (1.3.14), it remains to define the numerical flux functions \widehat{f} featured in the cell boundary terms. In general, $\widehat{f}(u_h^L, u_h^R)$ depends on values of the numerical solution from both sides of the cell interface. Construction of the function is often motivated by the energy stability proof and, traditionally ([26]), it is chosen to be a so-called monotone flux: a function that satisfies

- Lipschitz continuity
- consistency: $\widehat{f}(u, u) = f(u)$
- monotonicity: $\widehat{f}(\uparrow, \downarrow)$.

By monotonicity, we mean to say that the function is non-decreasing in its first argument u^- and non-increasing in its second argument u^+ .

The numerical flux function is the part of the DG approximation that is responsible for superconvergence. In the linear case, one can use characteristics to guide the design of the flux \widehat{f} . When $f(u) = cu$, $c \in \mathbb{R}$, in the hyperbolic equation (1.3.6), a single wind direction is determined and the natural choice to mimic advection when $c > 0$ is to satisfy the upwinding condition

$$\widehat{f}(u_h) = c\widehat{u}_h = cu_h^-$$

so that information propagates only from left to right. As a means of investigating the effect of the flux on the superconvergent properties of the schemes, we consider a more general function

$$\widehat{u}_h = \theta u_h^- + (1 - \theta)u_h^+, \quad \theta > \frac{1}{2}, \quad (1.3.22)$$

which incorporates a parameter θ that measures the amount of information included from the upwind direction. More information is taken from the left

than from the right of cell boundaries and, when $\theta = 1$, the function reduces to the purely upwind flux u_h^- . We do not allow $\theta = \frac{1}{2}$, which gives a central flux, since then the scheme becomes unstable. This function was named the *upwind-biased* flux by Meng et al. (2016) who introduced it in the context of DG solutions to linear hyperbolic equations. Here, it is defined for a problem with periodic boundary conditions but in [44, 15], Dirichlet conditions were also considered. Construction of an upwind-biased flux for nonlinear equations is the topic of ongoing work.

An upwind-biased flux may lead to a reduced numerical viscosity and may be easier to construct for more complicated problems [44]. It was recently shown in [15] that the upwind-biased DG scheme has many of the superconvergent properties enjoyed by the purely upwind scheme despite its irregular transfer of information in the direction counter to the wind direction. Moreover, for values $\theta > 1$, the flux (1.3.22) is not monotone. The main contribution of this thesis with respect to the upwind-biased flux is to show how the parameter θ can be varied in such a way that the spectral properties of the approximation are improved, especially over long time integration. That wave propagation can be improved in this way goes against one's instincts as to how flux functions should be designed.

For nonlinear problems and for some schemes with coupled spatial and temporal discretisations, we consider the generalised Rusanov-type numerical flux

$$\widehat{f}_{j+\frac{1}{2}} = \alpha\theta_1 u_{j+\frac{1}{2}}^- + \alpha(1 - \theta_1)u_{j+\frac{1}{2}}^+ + \theta_2 (f - \alpha u)_{j+\frac{1}{2}}^- + (1 - \theta_2) (f - \alpha u)_{j+\frac{1}{2}}^-, \quad (1.3.23)$$

where $\alpha = \max\{|f'(u_j)|, |f'(u_{j+1})|\}$ and $\theta_1, \theta_2 \in [0, 1]$ are fixed parameters. Per the caution in [60], the parameters $\theta_{1,2}$ must be carefully chosen in order to ensure stability of the scheme. A popular choice employed in [46, 60, 51] is to take $\theta_1 = 1$ and $\theta_2 = \frac{1}{2}$. This results in a function similar to the local Lax-Friedrichs (LLF)

flux

$$\widehat{f}_{j+\frac{1}{2}} := \widehat{f}(x_{j+\frac{1}{2}}) = \frac{1}{2} \left[f_{j+\frac{1}{2}}^+ + f_{j+\frac{1}{2}}^- - \alpha (u_{j+\frac{1}{2}}^+ - u_{j+\frac{1}{2}}^-) \right], \quad \alpha = \max_u |f'(u)|, \quad (1.3.24)$$

defined here for the one-dimensional case, although there are many other choices. The LLF flux adjusts the (mathematically) natural but unstable centered choice by adding sufficient numerical diffusion to stabilise the schemes.

1.3.8 Linear advection: Fully resolved semi-discrete scheme

To complete our discussion of the DG semi-discretisation, we bring together the components introduced thus far to explicitly derive the ODE form (1.3.8) given in §1.3.2 for the MoL approach to solving the linear advection equation in one dimension

$$u_t + cu_x = 0, \quad c > 0. \quad (1.3.25)$$

A DG discretisation of equation (1.3.25) along the lines of equation (1.3.14) with the upwind-biased flux (1.3.22) results in

$$\begin{aligned} \int_{\mathcal{I}_j} (u_h)_t v \, dx &= c \int_{\mathcal{I}_j} u_h v_x \, dx - c \left(\theta u_h(x_{j+1/2}^-) + (1 - \theta) u_h(x_{j+1/2}^+) \right) v(x_{j+1/2}^-) \\ &\quad + c \left(\theta u_h(x_{j-1/2}^-) + (1 - \theta) u_h(x_{j-1/2}^+) \right) v(x_{j-1/2}^+) \end{aligned} \quad (1.3.26)$$

for all $v \in V_h^p$. In this case, we alter the notation (1.3.16) for the DG spatial discretisation to include the choice of flux:

$$\begin{aligned} \mathcal{B}_j^\theta(u_h; v) &:= -c \int_{\mathcal{I}_j} u_h v_x \, dx + c \left(\theta u_h(x_{j+1/2}^-) + (1 - \theta) u_h(x_{j+1/2}^+) \right) v(x_{j+1/2}^-) \\ &\quad - c \left(\theta u_h(x_{j-1/2}^-) + (1 - \theta) u_h(x_{j-1/2}^+) \right) v(x_{j-1/2}^+) \end{aligned} \quad (1.3.27)$$

so that

$$\int_{\mathcal{I}_j} (u_h)_t v \, dx = -\mathcal{B}_j^\theta(u_h; v). \quad (1.3.28)$$

We reserve $\mathcal{B}_j^-(u_h; v)$ and $\mathcal{B}_j^+(u_h; v)$ for the upwind (when $\theta = 1$ in equation (1.3.27)) and downwind (when $\theta = 0$) cases respectively. While $\theta = 0$ is not a legitimate choice here when $c > 0$, the notation will be useful later.

To give the global operator in terms of jumps

$$\llbracket v \rrbracket_{j+\frac{1}{2}} := v_{j+\frac{1}{2}}^+ - v_{j+\frac{1}{2}}^- \quad (1.3.29)$$

in the boundary values, which can be useful for proofs, sum the scheme (1.3.26) over the elements \mathcal{I}_j and use the periodic boundary conditions:

$$\int_{\mathcal{I}_h} (u_h)_t v \, dx = -\mathcal{B}^\theta(u_h; v), \quad (1.3.30)$$

where

$$\mathcal{B}^\theta(u_h; v) := -c \int_{\mathcal{I}_h} u_h v_x \, dx - c \sum_{j=1}^N \left(u_h(x_{j+\frac{1}{2}}^-) + (1-\theta) \llbracket u_h \rrbracket_{j+\frac{1}{2}} \right) \llbracket v \rrbracket_{j+\frac{1}{2}}. \quad (1.3.31)$$

With this presentation, we make apparent the additional contribution of a non-unity flux parameter θ . By replacing u_h by its Galerkin expansion (1.3.20), scaling the element \mathcal{I}_j to the canonical element $[-1, 1]$ and inverting a (sparse) mass matrix, the DG weak formulation (1.3.14) can be resolved into a matrix ODE system for the vector \mathbf{U}_j of solution coefficients $U_j^{(\ell)}(t)$:

$$\frac{d}{dt} \mathbf{U}_j = \frac{1}{\Delta x} \mathcal{D}^\theta \mathbf{U}_j, \quad (1.3.32)$$

where we define an operator \mathcal{D}^θ that acts on the vector of Galerkin coefficients:

$$\mathcal{D}^\theta \mathbf{W}_j := M^{-1} [(\theta A_1 + (1-\theta) A_2) \mathbf{W}_j + \theta B \mathbf{W}_{j-1} - (1-\theta) C \mathbf{W}_{j+1}]. \quad (1.3.33)$$

The entries of the $(p+1) \times (p+1)$ matrices A_1, A_2, B and C are defined as follows:

$$\begin{aligned} (A_1)_{m\ell} &= S_{m\ell} - \varphi^{(\ell)}(1) \varphi^{(m)}(1); & (A_2)_{m\ell} &= S_{m\ell} + \varphi^{(\ell)}(-1) \varphi^{(m)}(-1); \\ B_{m\ell} &= \varphi^{(\ell)}(1) \varphi^{(m)}(-1); & C_{m\ell} &= \varphi^{(\ell)}(-1) \varphi^{(m)}(-1), \end{aligned} \quad (1.3.34)$$

where the elements of the mass and stiffness matrices are given respectively by

$$M_{m\ell} = \frac{1}{2} \int_{-1}^1 \varphi^{(\ell)} \varphi^{(m)} \, d\xi; \quad S_{m\ell} = \int_{-1}^1 \varphi^{(\ell)} \frac{\partial \varphi^{(m)}}{\partial \xi} \, d\xi. \quad (1.3.35)$$

For the special cases of a purely upwind ($\theta = 1$) or downwind ($\theta = 0$) flux in equation (1.3.33), we reserve the notations \mathcal{D}^- and \mathcal{D}^+ respectively.

In certain settings (such as in Chapter 4), it can be advantageous to consider the mathematically equivalent form of equation (1.3.26) obtained by performing an extra integration by parts:

$$\int_{\mathcal{I}_j} (u_h)_t v \, dx = -c \int_{\mathcal{I}_j} (u_h)_x v \, dx - c(1 - \theta) \llbracket u_h \rrbracket_{j+\frac{1}{2}} v_{j+\frac{1}{2}}^- - c\theta \llbracket u_h \rrbracket_{j-\frac{1}{2}} v_{j-\frac{1}{2}}^+. \quad (1.3.36)$$

For later use, we modify the DG operator notation accordingly. Define

$$\bar{\mathcal{D}}^- \mathbf{W}_j := M^{-1} [\bar{A}_1 \mathbf{W}_j + B \mathbf{W}_{j-1}] \quad (1.3.37)$$

$$\bar{\mathcal{D}}^+ \mathbf{W}_j := M^{-1} [\bar{A}_2 \mathbf{W}_j - C \mathbf{W}_{j+1}], \quad (1.3.38)$$

where, denoting by S^T the transpose of the stiffness matrix S ,

$$(\bar{A}_1)_{m\ell} = -S_{m\ell}^T + \varphi^{(\ell)}(1) \varphi^{(m)}(1); \quad (\bar{A}_2)_{m\ell} = -S_{m\ell}^T - \varphi^{(\ell)}(-1) \varphi^{(m)}(-1)$$

and where B and C are as in equations (1.3.34).

In Chapter 2, we take a close-up look at how the upwind-biased flux changes some of the well-known properties of the upwind scheme. While the view we take is somewhat divorced from the temporal discretisation, it is important to better understand the ordonnance of the spatial discretisation if we are, ultimately, to ask of the time-stepping method a more caring treatment of the prized superconvergent points.

The Numerical Flux and Superconvergence

“As if you could kill time without injuring eternity.”

– Henry David Thoreau, *Walden*

In this chapter, we address how the superconvergence properties of the DG discretisation depend on the numerical flux function. For linear problems, can we do better than the purely upwind flux? The first section takes a view from physical space and provides a description of how the superconvergent points of the semi-discrete description change with the upwind-biased flux parameter θ . We prove pointwise superconvergent accuracy when the upwind-biased flux is employed. This property directly contributes to the global error constants in the negative-order norm. These can be reduced by carefully choosing the value of θ . This also indicates improved dispersion properties of the fully-discrete schemes, a topic we address in Chapter 4.

This work was performed independently of and concurrently to the developments of Waixiang Cao, Yang Yang et al. [15] who also performed analysis for the one-dimensional linear hyperbolic equation solved by DG with upwind-biased flux. On a quasi-uniform mesh, they considered a special interpolation function u_I along the lines of Zhimen Zhang’s previous work (for example [16]). This function is the difference between the Radau projection

discussed by Xiong Meng et al. [44] and a correction function w built on Legendre polynomials and an integral projection. The correction function w is of high order as h decreases so that it is dominated by the Radau projection. In this chapter, we instead provide a concrete study of how the superconvergent points change with θ and how the flux function can be chosen to improve the approximation.

Secondly, we consider superconvergence of the approximation under the negative order norm after a post-processing by the SIAC filters developed by Ryan and others ([13, 25, 35]). These two areas of study are directly connected; the superconvergent error estimate for the post-processed solution relies on the values of the approximation at the element boundaries and is thus dependent on the choice of flux.

Within this chapter, we consider the linear advection equation discretised by the discontinuous Galerkin method with upwind-biased flux. By tracking the flux parameter θ , we shall better understand the role of the flux function in determining superconvergence in both physical space and the space governed by the negative-order norm and be able to provide guidance as to how to choose θ such that error constants associated to the approximation may be reduced in magnitude.

2.1 Pointwise Superconvergence

In this section, we demonstrate that when the flux in the DG scheme is chosen to be the upwind-biased flux, the leading order term in the error is proportional to a sum, dependent upon θ , of left- and right-Radau polynomials. At certain special points—roots of this sum of Radau polynomials—the solution displays higher than expected accuracy. Knowledge of how these superconvergent points change with the flux function can be used to inform developments in other areas of the method. For example, it may be advantageous to base interpolating Lagrange

polynomials on roots of the special Radau polynomial defined below. We briefly illustrate this in Chapter 4.

The main result, Theorem 2.1.3, is an extension of the observation, for example of Adjerid, Baccouch and others ([6, 5]), that the superconvergent points for the purely upwind DG scheme are generated by roots of right-Radau polynomials. To this end, we define a “special” Radau polynomial

$$R_{p+1}^*(\xi) := \theta R_{p+1}^+(\xi) + (-1)^p(1 - \theta)R_{p+1}^-(\xi), \quad \xi \in [-1, 1], \quad \frac{1}{2} < \theta. \quad (2.1.1)$$

We show that roots of $R_{p+1}^*(\xi)$, which change with the value of θ , generate superconvergent points on the order of h^{p+2} for the upwind-biased scheme. Interestingly, our results indicate that if the choice of flux is not chosen sensitively in relation to the parity of the polynomial degree, one of these “superconvergent points” lies outside the element $[-1, 1]$ when convergence to the expected order is slow.

In the following Lemma, we describe the roots of $R_{p+1}^*(\xi)$. For this argument, we consider the polynomials $P_n(\xi)$ arising from the Rodrigues formula (1.3.5) and then extend their domain of definition beyond $[-1, 1]$. Of course, any root that we find to be outside $[-1, 1]$ will not directly manifest as a superconvergent point of the DG solution.

Lemma 2.1.1. *Let $p \in \mathbb{N}$ and consider the special Radau polynomial*

$$R_{p+1}^*(\xi) = \theta R_{p+1}^+(\xi) + (-1)^p(1 - \theta)R_{p+1}^-(\xi), \quad \xi \in \mathbb{R}, \quad \theta > \frac{1}{2}.$$

All roots of $R_{p+1}^(\xi)$ lie in the interval $[-1, 1]$ provided that $\frac{1}{2} < \theta \leq 1$ when p is even and that $\theta \geq 1$ when p is odd. Otherwise, exactly one root of $R_{p+1}^*(\xi)$ is greater than 1 whilst all other roots lie in the interval $[-1, 1]$.*

Proof. We split the proof into two cases, writing the special Radau polynomial

as

$$R_{p+1}^* = \begin{cases} P_{p+1} - (2\theta - 1)P_p, & \text{when } p \text{ is even,} \\ (2\theta - 1)P_{p+1} - P_p, & \text{when } p \text{ is odd.} \end{cases} \quad (2.1.2)$$

This *special Radau polynomial* corresponds to the “generalised Radau polynomial” $G_\alpha = P_{p+1} - \alpha P_p$ analysed in [15]. Note that when p is even, $G_{\alpha=2\theta-1} = R^*$ and when p is odd, $G_{\alpha=1/(2\theta-1)} = \frac{1}{2\theta-1}R^*$.

Suppose that p is even and let $\frac{1}{2} < \theta$. Note that by equation (1.3.19) we have, for $\xi > 1$,

$$\begin{aligned} \frac{d}{d\xi} R_{p+1}^*(\xi) &= [(2p+1)P_p - (2\theta-1)(2p-1)P_{p-1}] \\ &\quad + [(2p-3)P_{p-2} - (2\theta-1)(2p-5)P_{p-3}] + \cdots + P_0 > 0. \end{aligned}$$

Thus, since $R_{p+1}^*(1) = 2(1-\theta)$, there exists a root $\xi > 1$ of $R_{p+1}^*(\xi)$ only if $\theta > 1$. Similarly, there is no root $\xi < -1$ since $R_{p+1}^*(-1) = -2\theta < 0$ and $\frac{d}{d\xi} R_{p+1}^*(\xi) > 0$. Suppose instead that the polynomial degree p is odd. Note that $R_{p+1}^*(1) = 2(\theta-1)$ and that $\lim_{\xi \rightarrow \infty} R_{p+1}^*(\xi) = +\infty$. Thus, when $\theta \geq 1$, all the roots of $R_{p+1}^*(\xi)$ must satisfy $\xi \leq 1$. On the other hand, when $\theta < 1$ the Intermediate Value Theorem implies the existence of a root $\xi > 1$. Furthermore, there is only one such root since, for all $\xi > 1$, property (1.3.19) gives

$$\begin{aligned} \frac{d}{d\xi} R_{p+1}^*(\xi) &\geq (2\theta-1)[(2p+1)(P_p - P_{p-1}) + (2p-3)(P_{p-2} - P_{p-3}) \\ &\quad + \cdots + 3(P_1 - P_0)] > 0. \end{aligned}$$

A similar argument shows that there are no roots $\xi < -1$: while $R_{p+1}^*(-1) = 2\theta > 0$, we have that $\frac{d}{d\xi} R_{p+1}^*(\xi) < 0$ for all $\xi < -1$. \square

Recall that for the upwind flux (when $\theta = 1$), one of the superconvergent points is the *strongly superconvergent* downwind end ξ_{p+1}^+ where the approximation is on the order of h^{2p+1} . Table 2.1 gives approximations to the roots, which are the superconvergent points when $\theta = 1$, of the right-Radau polynomial of various

Table 2.1: Approximations to roots ξ_i^+ , $i = 1, \dots, p + 1$, of the right Radau polynomial $R_{p+1}^+(\xi)$.

p	ξ_i^+				
1	$-\frac{1}{3}$		1		
2	-0.69	0.29	1		
3	-0.82	-0.18	0.58	1	
4	-0.89	-0.45	0.17	0.72	1

degrees. Relative to these on a number line, when p is even, the roots of $R_{p+1}^*(\xi)$ shift to the left with decreasing values of θ . On the other hand, when p is odd, the points shift to the right and $\xi_{p+1}^* > 1$. This observation is reversed for increasing $\theta > 1$. For example, when $p = 1$, the roots of $R_{p+1}^*(\xi)$ are given by

$$\xi_{1,2}^* = \frac{1 \mp 2\sqrt{1 - 3\theta + 3\theta^2}}{3(2\theta - 1)}.$$

In other words, when the basis polynomial degree has odd parity, we require $\theta \geq 1$ in order for all superconvergent points (in particular, for the strongly superconvergent point) to be physically manifest within the elements. As a numerical demonstration of this observation, Table 2.2 gives approximations to the roots of $R_{p+1}^*(\xi)$ when $\theta = 0.55, 1.45$: that is, when we employ an upwind-biased flux satisfying

$$\hat{u}_{j+\frac{1}{2}} = u_{j+\frac{1}{2}}^- \mp 0.45[[u]]_{j+\frac{1}{2}}. \quad (2.1.3)$$

Note the roots that lie outside the interval $[-1, 1]$. We emphasise that this suggests choosing $\theta \in (\frac{1}{2}, 1]$ when the degree of the polynomial basis is even and $\theta \geq 1$ when p is odd. In the numerical results section §2.2.3, we plot the discretisation errors for $\theta = 1, 0.55, 1.45$ in Figure 2.1.3.

Table 2.2: Approximations to roots ξ_i^* , $i = 1, \dots, p + 1$, of $R_{p+1}^*(\xi)$.

p	ξ_i^* when $\theta = 0.55$					ξ_i^* when $\theta = 1.45$				
1	-0.05	6.72				-0.43	0.78			
2	-0.76	0.03	0.79			-0.66	0.42	1.38		
3	-0.78	-0.02	0.76	5.76		-0.83	-0.24	0.47	0.91	
4	-0.90	-0.52	0.02	0.55	0.91	-0.87	-0.41	0.24	0.81	1.30

Following the lines of [5], we interpolate the initial condition at roots of $R_{p+1}^*(\xi)$, where we must restrict $\theta \leq 1$ when p is even and $\theta \geq 1$ when p is odd due to the result in Lemma 2.1.1. When one of the roots of $R_{p+1}^*(\xi)$ lies outside of $[-1, 1]$, for example when $\theta < 1$ and p is odd, one can instead define a global projection similar to [44] but we leave this as further work.

2.1.1 Pointwise error estimate

Lemma 2.1.2. *Let $p \in \mathbb{Z}_+$ and suppose that $u \in \mathcal{C}^{p+1}([0, h])$. Let $\xi_j^* \in [-1, 1]$, $j = 1, \dots, p + 1$, be the roots of $R_{p+1}^*(\xi)$ as defined by equation (2.1.1) with $\theta \leq 1$ if p is even and $\theta \geq 1$ if p is odd. Consider the p^{th} -degree Lagrange polynomial*

$$\pi^* u(x) = \sum_{n=1}^{p+1} L_n(x), \quad L_n(x) = u(x_n^*) \prod_{\substack{j=1 \\ j \neq n}}^{p+1} \frac{x - x_j^*}{x_n^* - x_j^*}, \quad x \in [0, h],$$

interpolating u at the (distinct) roots $x_j^* = \frac{h}{2}(\xi_j^* + 1)$ of the shifted special Radau polynomial $R_{p+1}^*(x)$ on $[0, h]$. Then the interpolation error satisfies

$$u(x(\xi)) - \pi^* u(x(\xi)) = h^{p+1} c_{p+1} R_{p+1}^*(\xi) + \sum_{\ell=p+2}^{\infty} Q_\ell(\xi) h^\ell, \quad (2.1.4)$$

where $Q_\ell(\xi)$ is a polynomial of degree at most ℓ .

Proof. The standard Lagrangian interpolation theory yields, for $x \in [0, h]$, an

$s = s(x) \in (0, h)$ such that

$$e(x) = u(x) - \pi^* u(x) = \frac{1}{(p+1)!} u^{(p+1)}(s(x)) \prod_{j=1}^{p+1} (x - x_j^*).$$

By the linear mapping $x = \frac{h}{2}(1 + \xi)$, we obtain

$$e(x(\xi)) = \frac{h^{p+1} u^{(p+1)}(s(x(\xi)))}{2^{p+1}(p+1)!} \prod_{j=1}^{p+1} (\xi - \xi_j^*), \quad \xi \in [-1, 1]. \quad (2.1.5)$$

The Maclaurin series of $u^{(p+1)}(s(x(\xi)))$ with respect to h gives the leading order term in the error so that equation (2.1.5) becomes

$$e(x(\xi)) = h^{p+1} \frac{u^{(p+1)}(0)}{2^{p+1}(p+1)!} \prod_{j=1}^{p+1} (\xi - \xi_j^*) + \sum_{m=1}^{\infty} Q_m(\xi) h^{m+p+1}, \quad (2.1.6)$$

where $Q_m(\xi)$ comprises the product of $R_{p+1}^*(\xi)$ and a polynomial of degree m in ξ :

$$Q_m(\xi) = \frac{\frac{d^m}{dh^m} \frac{\partial^{p+1} u(s(x(\xi)))}{\partial x^{p+1}} \Big|_{h=0}}{2^{p+1}(p+1)! m!} \prod_{j=1}^{p+1} (\xi - \xi_j^*).$$

□

The interpolatory polynomials described in Lemma 2.1.2 are used as initial conditions in the proof of Theorem 2.1.3. Numerical results in §2.1.2 confirm that there are only p superconvergent points in each element if the value of θ is not chosen carefully with respect to the polynomial degree.

Remark. The strong assumption of a uniform mesh, in this case, limits generalisation of our observations. Cao et al. [15] argued that it may be necessary to define a local as well as a global flux parameter θ . They showed that for certain *local* flux parameters θ_j , the following supercloseness result holds

$$|(u - u_I)(\xi_{j,\ell}^*)| \leq h_j^{p+2} \|u\|_{p+2, \infty}.$$

It follows that the superconvergent points of the projection and so the correction function u_I depend on the lengths h_j of the cells.

Theorem 2.1.3. Let $p \in \mathbb{Z}_+$. Consider the approximate solution u_h to the one-dimensional linear advection equation obtained by a DG scheme (1.3.14) using p^{th} -order basis functions, a uniform mesh and the upwind-biased flux with $\theta \leq 1$ if p is even and $\theta \geq 1$ if p is odd. Let the numerical initial condition be the interpolating polynomial $\pi^*u(x, 0)$ described in Lemma 2.1.2.

Let $\xi = \frac{2}{h}x - 1$ be the scaling between the cell \mathcal{I}_j and the canonical element $[-1, 1]$.

Then the error $e = u - u_h$ satisfies

$$e(\xi, h, t) = \sum_{\ell=p+1}^{\infty} Q_{\ell}(\xi, t)h^{\ell}, \quad Q_{\ell}(\cdot, t) \in \mathcal{P}^{\ell}([-1, 1]), \quad (2.1.7)$$

with

$$Q_{p+1}(\xi, t) = c_{p+1}(u, h, t)R_{p+1}^*(\xi),$$

where

$$R_{p+1}^*(\xi) = \left(\theta R_{p+1}^+(\xi) + (-1)^p(1 - \theta)R_{p+1}^-(\xi) \right).$$

Proof. Without loss of generality, assume the tessellation \mathcal{I}_h comprises a single element $[0, h]$. To facilitate the analysis, subtract the approximating scheme (1.3.14) from equation (1.3.13) to obtain a DG orthogonality condition for the error e :

$$\int_0^h e_t v \, dx - \int_0^h c e v_x \, dx + c \widehat{e} v \Big|_{x=0}^{x=h} = 0, \quad (2.1.8)$$

where $\widehat{e} = u - \widehat{u}_h$. The flux terms in equation (2.1.8) can be evaluated using the periodicity of the boundary conditions as follows:

$$\begin{aligned} \widehat{e}|_{x=0} &= \widehat{e}|_{x=h} = \theta(u - u_h^-)|_{x=h} + (1 - \theta)(u - u_h^+)|_{x=h} \\ &= \theta e|_{x=h} + (1 - \theta)e|_{x=0}. \end{aligned} \quad (2.1.9)$$

Substitution of the cell boundary evaluations (2.1.9) into equation (2.1.8) yields

for all $v \in V_h^p$, after a scaling to the canonical element $[-1, 1]$,

$$\frac{h}{2} \int_{-1}^1 e_t v \, d\xi - \int_{-1}^1 c e v_\xi \, d\xi + c(\theta e|_{\xi=1} + (1-\theta)e|_{\xi=-1})(v(1) - v(-1)) = 0. \quad (2.1.10)$$

Next, we reformat equation (2.1.10) as a scheme for the leading order terms of the error.

Step One: The DG solution within an element is clearly analytic as a function of h . Since we assume an initial condition of class C^∞ , the advecting exact solution is also smooth so the (local) DG solution is analytic in ξ . Hence we can expand the (local) error $e = u - u_h$, which is analytic, as a Maclaurin series with respect to h :

$$e(\xi, h, t) = \sum_{\ell=0}^{\infty} Q_\ell(\xi, t) h^\ell, \quad (2.1.11)$$

where $Q_\ell(\cdot, t) = \sum_{m=0}^{\ell} b_m P_m(\xi)$ is a polynomial of degree at most ℓ .

Next, substitute the expansion (2.1.11) into the scaled scheme (2.1.10) for the error and collect terms in powers of h to obtain the following equations:

$$- \int_{-1}^1 Q_0 v_\xi \, d\xi + (\theta Q_0(1, t) + (1-\theta)Q_0(-1, t))(v(1) - v(-1)) = 0; \quad (2.1.12a)$$

$$\begin{aligned} \frac{1}{2} \int_{-1}^1 (Q_{\ell-1})_t v \, d\xi - \int_{-1}^1 c Q_\ell v_\xi \, d\xi \\ + c(\theta Q_\ell(1, t) + (1-\theta)Q_\ell(-1, t))(v(1) - v(-1)) = 0, \quad \ell \geq 1. \end{aligned} \quad (2.1.12b)$$

Since $Q_0(\xi, t) = Q_0(t) = \theta Q_0(t) + (1-\theta)Q_0(t)$, the Fundamental Theorem of Calculus immediately satisfies equation (2.1.12a). It is from equation (2.1.12b), by inductively testing against functions $v \in V_h^p$, that the rest of the argument is extracted.

Step Two: Substitute $\ell = 1$ in equation (2.1.12b) and choose $v = 1$ to obtain

$$\int_{-1}^1 (Q_0)_t \, d\xi = (Q_0)_t \int_{-1}^1 \, d\xi = 0.$$

Thus we must have $(Q_0)_t(\xi, t) = 0$. Any $(p + 1)$ -node interpolating initial condition $\pi u_0(x)$ leads to the first $p + 1$ coefficients in the expansion (2.1.11) vanishing initially:

$$Q_\ell(\xi, 0) = 0, \quad \ell = 0, \dots, p.$$

In particular, since $(Q_0)_t(\xi, t) = 0$, we have $Q_0(\xi, t) = 0$ for all t . This last observation forms the base step for an induction on p in equation (2.1.12b) with the hypothesis $Q_\ell(\xi, t) = 0$, $\ell = 1, \dots, p - 1$.

Step Three: To show that $Q_p(\xi, t) = 0$, consecutively substitute $\ell = p$ and $\ell = p + 1$ in equation (2.1.12b) and choose $v = \xi$ and $v = 1$ respectively to obtain in turn

$$-\int_{-1}^1 Q_p \, d\xi + 2(\theta Q_p(1, t) + (1 - \theta)Q_p(-1, t)) = 0, \quad (2.1.13a)$$

$$\int_{-1}^1 (Q_p)_t \, d\xi = 0. \quad (2.1.13b)$$

After differentiating equation (2.1.13a) with respect to t , equation (2.1.13b) yields

$$\frac{d}{dt}(\theta Q_p(1, t) + (1 - \theta)Q_p(-1, t)) = 0. \quad (2.1.14)$$

Since $Q_p(\xi, 0) = 0$, the integral in time of equation (2.1.14) leaves, for any t ,

$$\theta Q_p(1, t) + (1 - \theta)Q_p(-1, t) = 0. \quad (2.1.15)$$

The terms in equation (2.1.15) feature in equation (2.1.12b) when $\ell = p$ so that

$$\int_{-1}^1 Q_p v_\xi \, d\xi = 0, \quad v \in V_h^p.$$

Hence Q_p is orthogonal to all $v \in \mathcal{P}^{p-1}$ and, if we write $Q_p(\cdot, t)$ and $v(\xi)$ as sums of Legendre polynomials $P_\ell(\xi)$, the orthogonality properties (1.3.17) yield

$$Q_p(\xi, t) = \sum_{\ell=0}^p b_\ell(t) P_\ell(\xi) = b_p(t) P_p(\xi). \quad (2.1.16)$$

The expansion (2.1.16) must satisfy the flux condition (2.1.15). It follows that

$$\theta b_p P_p(1) + (1 - \theta) b_p P_p(-1) = 0$$

so, by Legendre properties (1.3.18), $b_p = 0$. This completes the induction:

$$Q_\ell(\xi, t) = 0, \quad \ell = 0, 1, \dots, p.$$

Step Four: We now consider the term Q_{p+1} , following the same process as before. That is, consecutively substitute $\ell = p + 1$ and $\ell = p + 2$ in equation (2.1.12b) and choose $v = \xi$ and $v = 1$ respectively to obtain in turn

$$\int_{-1}^1 Q_{p+1} \, d\xi = 2(\theta Q_{p+1}(1, t) + (1 - \theta) Q_{p+1}(-1, t)), \quad (2.1.17a)$$

$$\int_{-1}^1 (Q_{p+1})_t \, d\xi = 0. \quad (2.1.17b)$$

Next, differentiate equation (2.1.17a), equate to zero using equation (2.1.17b) and apply the Fundamental Theorem of Calculus to obtain

$$\theta Q_{p+1}(1, t) + (1 - \theta) Q_{p+1}(-1, t) = \theta Q_{p+1}(1, 0) + (1 - \theta) Q_{p+1}(-1, 0). \quad (2.1.18)$$

To see that the right-hand side of equation (2.1.18) vanishes, recall that the leading order term in the interpolation error $u_0(x) - \pi^* u_0(x)$ satisfies

$$Q_{p+1}(\xi, 0) = c_{p+1} R_{p+1}^*(\xi)$$

then note that, irrespective of the value of p , the following equates to zero:

$$\begin{aligned} \theta R_{p+1}^*(1) + (1 - \theta) R_{p+1}^*(-1) &= \theta(1 - \theta) \left[(-1)^p R_{p+1}^-(1) + R_{p+1}^+(-1) \right] \\ &= \theta(1 - \theta) \left[2(-1)^p + 2(-1)^{p+1} \right]. \end{aligned}$$

It follows from equation (2.1.18) that we also have, for all $t \geq 0$,

$$\theta Q_{p+1}(1, t) + (1 - \theta) Q_{p+1}(-1, t) = 0. \quad (2.1.19)$$

Thus, the flux terms in equation (2.1.12b) with $\ell = p + 1$ vanish, leaving

$$\int_{-1}^1 Q_{p+1} v_\xi \, d\xi = 0, \quad v \in V_h^p.$$

Hence Q_{p+1} is orthogonal to all $v \in \mathcal{P}^{p-1}$ and we can write, at any given $t > 0$,

$$Q_{p+1}(\xi, t) = b_{p+1}(t)P_{p+1}(\xi) + b_p(t)P_p(\xi). \quad (2.1.20)$$

If we require of the expansion (2.1.20) the conditions (2.1.19), then we must satisfy

$$\theta [b_{p+1} + b_p] + (1 - \theta) [(-1)^{p+1}b_{p+1} + (-1)^p b_p] = 0,$$

from which it follows that

$$b_p = \begin{cases} -(2\theta - 1)b_{p+1}, & \text{when } p \text{ is even} \\ -\frac{1}{2\theta - 1}b_{p+1}, & \text{when } p \text{ is odd.} \end{cases}$$

In the case of an even polynomial degree p , we have

$$\begin{aligned} Q_{p+1}(\xi, t) &= b_{p+1}P_{p+1}(\xi) + (1 - 2\theta)b_{p+1}P_p(\xi) \\ &= b_{p+1} \left[\theta R_{p+1}^+(\xi) + (1 - \theta)R_{p+1}^-(\xi) \right] \\ &= b_{p+1}R_{p+1}^*(\xi) \end{aligned}$$

and, for an odd polynomial degree p , we have

$$\begin{aligned} Q_{p+1}(\xi, t) &= \frac{1}{2\theta - 1} [(2\theta - 1)b_{p+1}P_{p+1}(\xi) - b_{p+1}P_p(\xi)] \\ &= \frac{1}{2\theta - 1} b_{p+1} \left[\theta R_{p+1}^+(\xi) - (1 - \theta)R_{p+1}^-(\xi) \right] \\ &= \frac{b_{p+1}}{2\theta - 1} R_{p+1}^*(\xi), \end{aligned}$$

where b_{p+1} depends on t . □

Remark. For simplicity of exposition, Theorem 2.1.3 was restricted to the

one-dimensional case but extension of the results to multiple dimensions, when the approximation space consists of piecewise continuous tensor polynomials, is reasonably straightforward. For a linear advection system, the superconvergent points are just tensors of the roots of $R_{p+1}^*(\xi)$ in each dimension.

2.1.2 Numerical experiments

We present a numerical discussion for the test equation

$$\begin{aligned} u_t + u_x &= 0, & (x, t) &\in [0, 2\pi] \times (0, T], \\ u(x, 0) &= \sin(x), & u(0, T) &= u(2\pi, T). \end{aligned}$$

Figures 2.1.1 -2.1.4 show the DG discretisation errors on a grid of $N = 10$ elements for various values of θ and for polynomial degrees $p = 1, 2, 3, 4$. Marked by the red crosses are the theoretical superconvergent points which are roots of $R_{p+1}^*(\xi)$ and which change with the value of $\theta \in (\frac{1}{2}, 1]$. The error curves cross the zero axis near these roots. Furthermore, the intersection points appear to align more closely as p increases, an observation shared by Adjerd et al. in [4].

For even polynomial degree ($p = 2$ and $p = 4$), we observe $p + 1$ superconvergent points while for the odd cases ($p = 1$ and $p = 3$), in general, the error curves cross the zero axis only p times. Furthermore, as the value of θ reduces, we see an overall reduction in the magnitude of the errors for even p . On the other hand, when p is odd the magnitude of the errors in general increases for smaller values of θ .

Inside certain anomalous elements, for example the fifth and tenth elements in Figure 2.1.2, the curves miss the crosses or we observe an additional intersection and this may be due to the initial condition $\sin(x)$.

Figure 2.1.1: Discretisation errors for DG solution to 1D linear hyperbolic equation with $p = 1$ and $N = 10$.

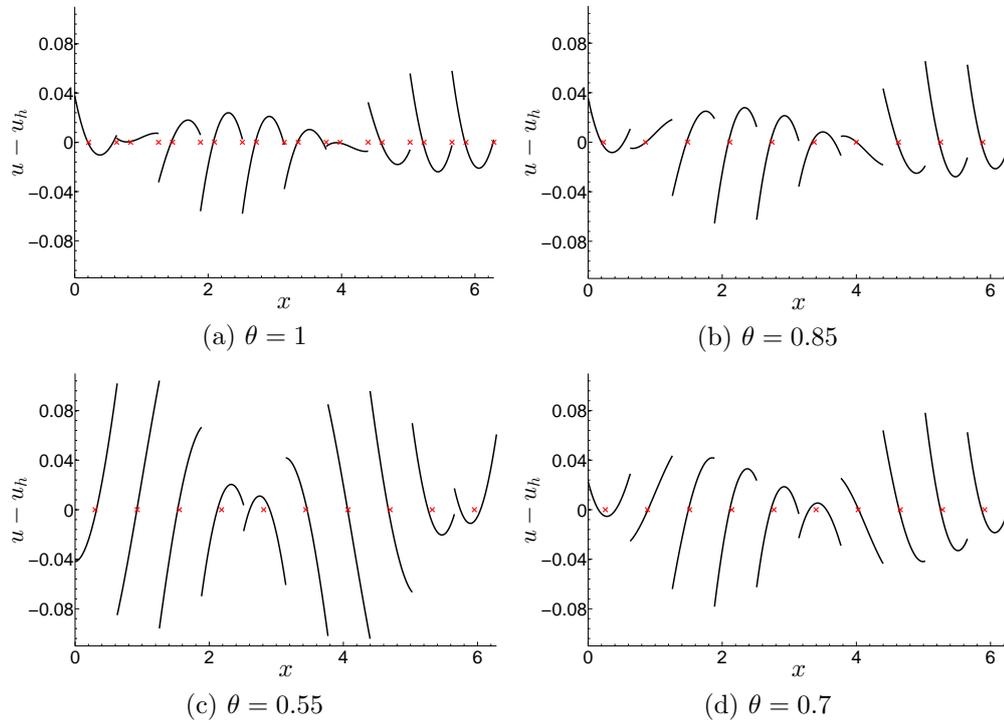


Figure 2.1.2: Discretisation errors for DG solution to 1D linear hyperbolic equation with $p = 2$ and $N = 10$.

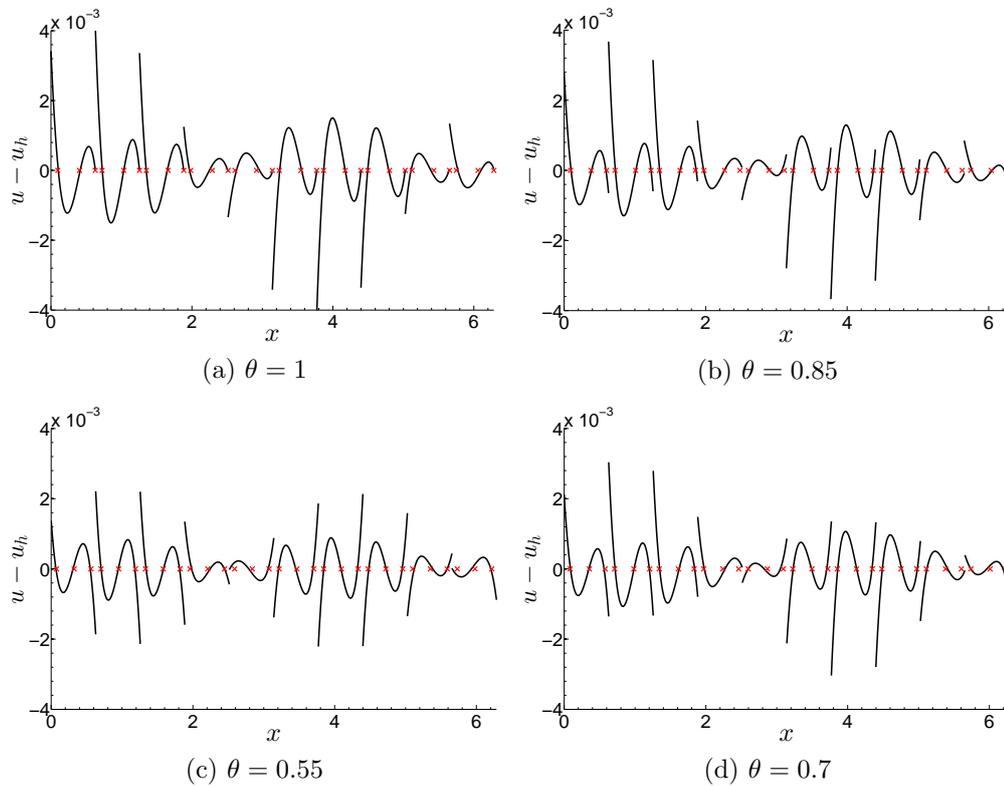


Figure 2.1.3: Discretisation errors for DG solution to 1D linear hyperbolic equation with $p = 3$ and $N = 10$.

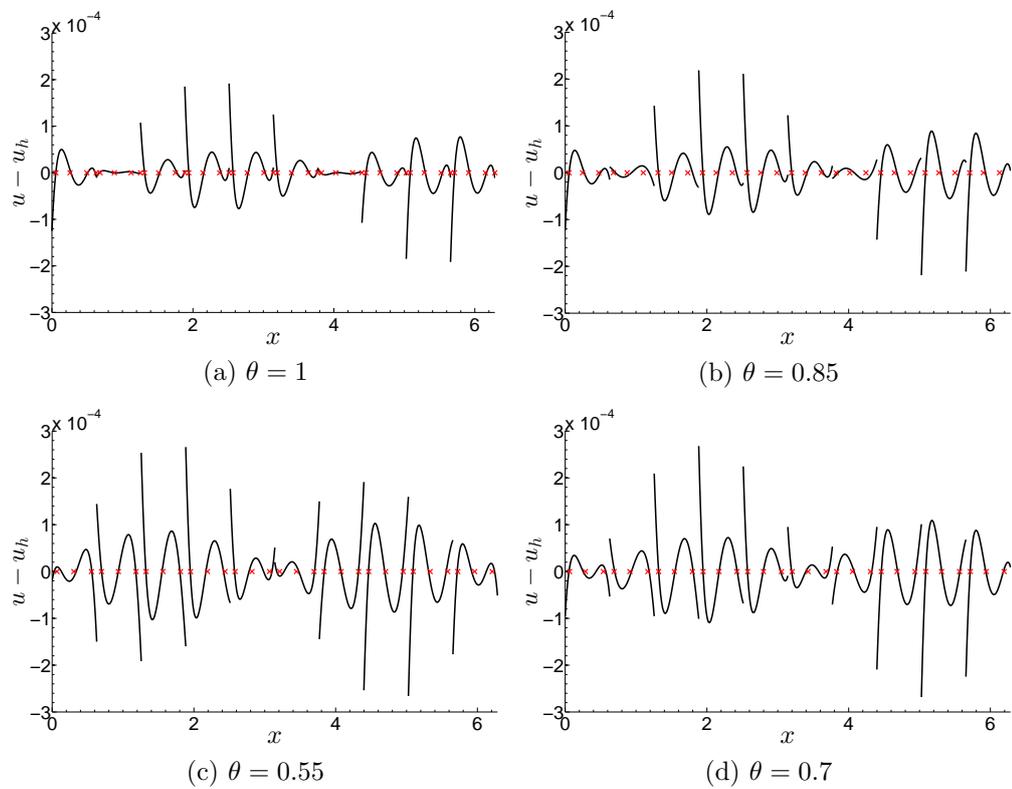
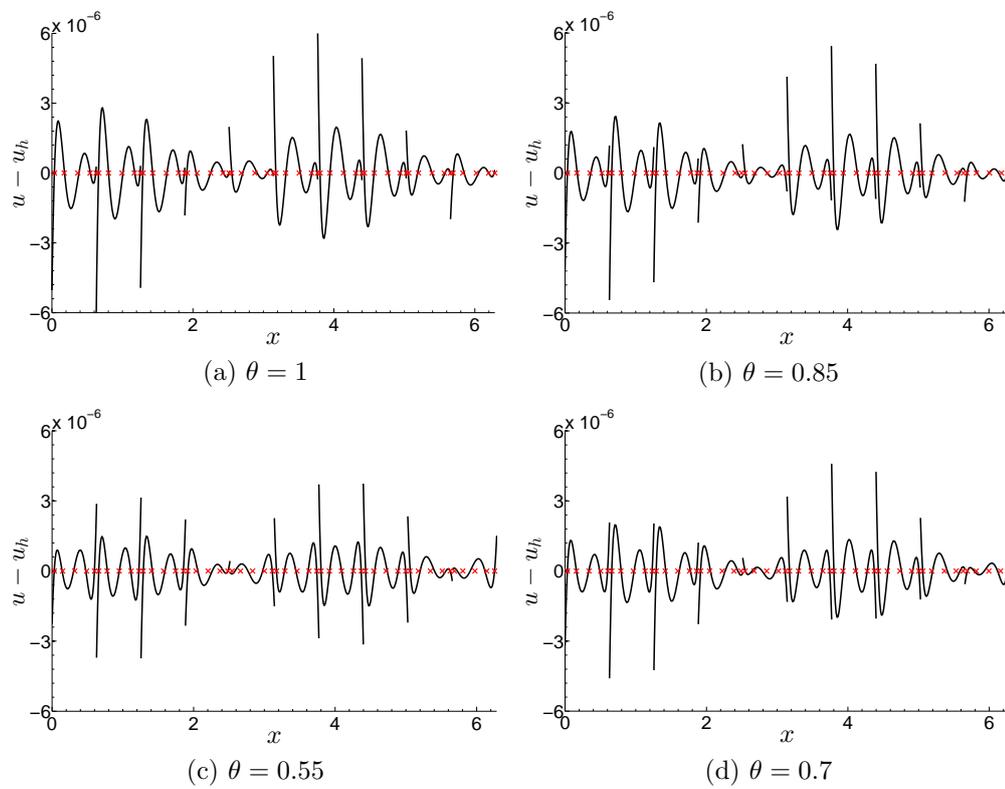


Figure 2.1.4: Discretisation errors for DG solution to 1D linear hyperbolic equation with $p = 4$ and $N = 10$.



2.2 Superconvergence of Post-processed Solution

The hidden local accuracy of the DG solution may be extracted to a global measure by applying the Smoothness-Increasing Accuracy-Conserving (SIAC) filter introduced in [56]. In this section, we show that $\mathcal{O}(h^{2p+1})$ superconvergent accuracy in the negative-order norm, as is observed ([36]) for the upwind flux, still occurs when the upwind-biased DG method is used to solve linear hyperbolic conservation laws.

2.2.1 The convolution kernel

We detail the component parts of the SIAC filter as defined in [36]. A B-spline $\psi^{(\ell)}$ of order ℓ is defined recursively by

$$\psi^{(1)} = \chi_{[-\frac{1}{2}, \frac{1}{2}]}; \quad \psi^{(\ell)} = \psi^{(\ell-1)} \star \chi_{[-\frac{1}{2}, \frac{1}{2}]}, \quad \ell \geq 2,$$

where $\chi_{[-\frac{1}{2}, \frac{1}{2}]}$ is the characteristic function on the interval $[-\frac{1}{2}, \frac{1}{2}]$ and where the operator \star denotes convolution:

$$f(x) \star g(x) = \int_{\mathbb{R}} f(x-y)g(y) \, dy.$$

For a multi-index α and given a point $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$, we define

$$\psi^{(\alpha)}(\mathbf{x}) = \psi^{(\alpha_1)}(x_1) \cdots \psi^{(\alpha_d)}(x_d); \quad \psi^{(\ell)}(\mathbf{x}) = \psi^{(\ell)}(x_1) \cdots \psi^{(\ell)}(x_d).$$

In this way, we construct a convolution kernel

$$\mathcal{K}_h^{(r+1, \ell)}(\mathbf{x}) = \sum_{\gamma \in \mathbb{Z}^d} \mathbf{c}_\gamma^{r+1, \ell} \psi^{(\ell)}(\mathbf{x} - \gamma) \quad (2.2.1)$$

that comprises a linear combination of $r+1$ B-splines $\psi^{(\ell)} \in \mathcal{C}^{\ell-2}$ of order ℓ such that $\mathcal{K}_h^{(r+1, \ell)}$ has compact support and reproduces (by convolution) polynomials of degree strictly less than r . Typically, $r = 2p$ and $\ell = p+1$, where p is the degree of the polynomial basis. The coefficients \mathbf{c}_γ are tensor products of

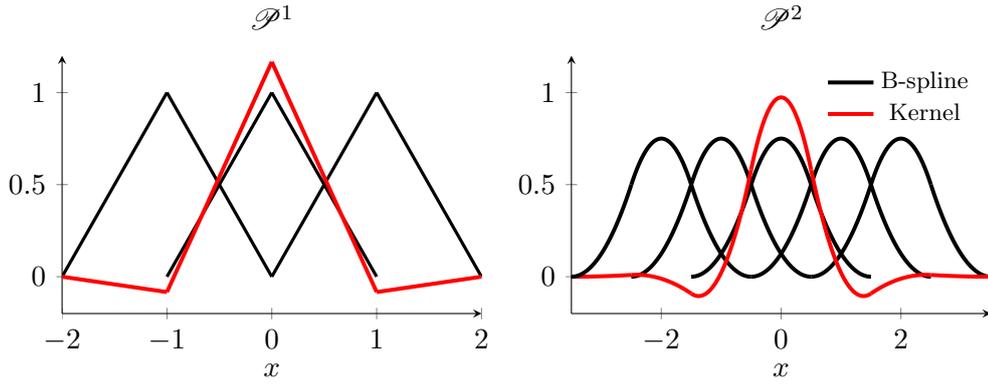


Figure 2.2.1: The $2p + 1$ B-splines of order $p = 1$ (left) and $p = 2$ (right) and the corresponding convolution kernels

the coefficients c_γ found by requiring the reproduction of polynomials property $\mathcal{K}_h^{(r+1,\ell)} \star x^q = x^q$, $q < r$, in the one-dimensional case. It is important to note that, due to properties of B-splines derivatives of a convolution with this kernel may be written in terms of difference quotients:

$$D^\alpha \left(\psi_h^{(\beta)} \star v \right) = \psi_h^{(\beta-\alpha)} \star \partial_h^\alpha v, \quad \beta_i \geq \alpha_i, \quad (2.2.2)$$

where $\psi_h^{(\beta)}(x) = \psi_h^{(\beta/h)}/h^d$. Further properties of the kernel may be found in [37]. By convolving the approximation with the kernel, we obtain the SIAC filtered solution

$$u_h^*(\bar{x}, t) := \mathcal{K}_h^{(r+1,\ell)}(\bar{x}) \star u_h(\bar{x}, t), \quad (2.2.3)$$

which displays increased accuracy and reduced oscillations in the error. The results in this paper treat only the symmetric kernel where the nodes γ are uniformly spaced. Extension to the one-sided filter given in [36] and [56] is a straight-forward task.

2.2.2 SIAC filtered error estimate

To begin, we observe that an error bound in the \mathcal{L}^2 -norm follows from a negative-order norm error estimate. Let

$$u_h^* = \mathcal{K}_h^{(2p+1,p+1)} \star u_h$$

be the DG solution to the hyperbolic system (1.3.5) post-processed with the convolution kernel at the final time. Denote by $e_h = u - u_h$ the usual DG error and consider the \mathcal{L}^2 -norm of the error $e_h^* := u - u_h^*$ associated with the filtered solution:

$$\|u - u_h^*\|_\Omega = \|u - \mathcal{K}_h \star u\|_\Omega + \|\mathcal{K}_h \star u - u_h^*\|_\Omega. \quad (2.2.4)$$

The first term on the right-hand side of (2.2.4) is bounded by Ch^{r+1} from the integral form of Taylor's theorem and from the reproduction of polynomials property of the convolution (Lemma 5.1, [36]). Thus we need only consider the second term for which

$$\begin{aligned} \|\mathcal{K}_h \star u - u_h^*\|_\Omega &= \|\mathcal{K}_h \star e_h\|_\Omega \leq \sum_{|\alpha| \leq \ell} \|D^\alpha(\mathcal{K}_h \star e_h)\|_\Omega \\ &\leq \sum_{|\alpha| \leq \ell} \|(D^\alpha \mathcal{K}_h) \star e_h\|_\Omega \\ &\leq \sum_{|\alpha| \leq \ell} \|(\partial^\alpha \mathcal{K}_h) \star e_h\|_\Omega \\ &\leq \sum_{|\alpha| \leq \ell} \|\tilde{\mathcal{K}}_h \star \partial^\alpha e_h\|_\Omega \leq \sum_{|\alpha| \leq \ell} \|\tilde{\mathcal{K}}_h\|_1 \|\partial_h^\alpha e_h\|_{-\ell} \end{aligned} \quad (2.2.5)$$

by kernel properties of the α^{th} derivative D^α , the kernel's relation to the divided difference ∂^α and by Young's inequality for convolutions. The tilde on $\tilde{\mathcal{K}}_h$ in inequality (2.2.5) signals that the kernel uses B-splines on the order of $\ell - |\alpha|$, which is a result of the property $D^\alpha \psi^{(\ell)} = \partial_h^\alpha \psi^{(\ell-\alpha)}$.

Note that $\|\tilde{\mathcal{K}}_h\|_1 = \sum_{i=0}^r |c_i|$ is just the sum of the kernel coefficients so we only need to show that $\|\partial_h^\alpha e_h\|_{-\ell} \leq Ch^{2p+1}$. Furthermore, the formulation of the DG scheme for the solution is similar to that for the divided differences and, as speculated in [25],

$$\|\partial_h^\alpha(u - u_h)\|_{-\ell, \Omega} \leq C \|\partial_h^\alpha u_0\|_{\ell, \Omega} h^{2p+m}, \quad m \in \{0, 1/2, 1\}. \quad (2.2.6)$$

This allows us to only have to consider the negative-order norm of the solution

itself; superconvergent accuracy in the negative-order norm gives superconvergent accuracy in the \mathcal{L}^2 -norm for the post-processed solution. The following result provides the required negative-order norm error estimate.

Remark. Notice that the superconvergent points for the upwind-biased scheme, as described in the one-dimensional case in Lemma 2.1.2, change with the value of θ . However, the global superconvergence in the negative-order norm occurs regardless of the value of θ . Furthermore, the proof of the following result does not differ between odd and even polynomial degrees.

Theorem 2.2.1. *Let u_h be the numerical solution to the linear hyperbolic conservation law (1.3.5) with smooth initial condition obtained via a DG scheme (1.3.14) with upwind-biased flux with parameter $\theta = (\theta_1, \dots, \theta_d)$. Then*

$$\|\partial_h^\alpha (u - u_h)(T)\|_{-k-1, \Omega} \leq C(u_0, \theta, T)h^{2p+1}, \quad \alpha < \ell. \quad (2.2.7)$$

Proof. The case when $\theta_i = 1, i = 1, \dots, d$, is covered in [25]. In the following, we point out the differences when the more general upwind-biased flux is used. For simplicity, we consider the case when $\alpha = 0$. The case for $\alpha > 0$ is similar ([25, 62]). In order to extract information about the error at the final time, we work with the dual equation: find a continuous and analytic $\phi(\mathbf{x}, t)$ such that

$$\phi_t + \sum_{i=1}^d a_i \phi_{x_i} = 0; \quad \phi(\mathbf{x}, T) = \Phi(\mathbf{x}), \quad (\mathbf{x}, t) \in \Omega \times [0, T]. \quad (2.2.8)$$

The term appearing in the definition of the negative-order norm can be split as

$$\begin{aligned} (u - u_h, \Phi)_\Omega(T) &= (u, \phi)_\Omega(T) - (u_h, \phi)_\Omega(T) \\ &= (u, \phi)_\Omega(0) - (u_h, \phi)_\Omega(0) - \int_0^T \frac{d}{dt} (u_h, \phi)_\Omega \, dt \\ &= (u - u_h, \phi)_\Omega(0) - \int_0^T \frac{d}{dt} (u_h, \phi)_\Omega \, dt. \end{aligned} \quad (2.2.9)$$

The bounding of this projection term $(u - u_h, \phi)_\Omega(0)$ is no different to that in [36]; this term is not affected by the choice of flux in the DG approximation. The \mathcal{L}^2 -

projection $\mathbb{P}_h u_0$ of the initial condition onto the solution space can be used as an initial condition to write

$$\begin{aligned} (u(0) - u_h(0), \phi(0))_\Omega &= (u_0 - \mathbb{P}_h u_0, \phi(0))_\Omega \\ &= (u_0 - \mathbb{P}_h u_0, \phi(0) - \mathbb{P}_h \phi(0))_\Omega. \end{aligned}$$

Here, we have used the fact that the difference $u_0 - \mathbb{P}_h u_0$ is orthogonal to the approximation space. By the Cauchy-Schwarz inequality,

$$\begin{aligned} |(u(0) - u_h(0), \phi(0))_\Omega| &\leq \|u_0 - \mathbb{P}_h u_0\|_\Omega \|\phi(0) - \mathbb{P}_h \phi(0)\|_\Omega \\ &\leq C_\alpha h^{p+1} \|u_0\|_{p+1} C_\beta h^{p+1} \|\phi(0)\|_{p+1} \\ &= C_1 h^{2p+2} \|u_0\|_{p+1} \|\phi(0)\|_{p+1}. \end{aligned} \quad (2.2.10)$$

Denote by $[[v]]_{\partial S} = v_{\partial S}^R \mathbf{n}_R + v_{\partial S}^L \mathbf{n}_L$ the jump in ϕ . Returning to equation (2.2.9), we use the dual equation, the DG approximation and the continuity of ϕ to rewrite the integrand:

$$\begin{aligned} &\frac{d}{dt} (u_h, \phi)_\Omega \\ &= ((u_h)_t, \phi)_\Omega + (u_h, \phi_t)_\Omega \\ &= ((u_h)_t, \phi - \chi)_\Omega + ((u_h)_t, \chi)_\Omega - \sum_{i=1}^d ((u_h, a_i \phi_{x_i})_\Omega \\ &= ((u_h)_t, \phi - \chi)_\Omega - \sum_{i=1}^d (a_i u_h, (\phi - \chi)_{x_i})_\Omega + \sum_S \sum_{i=1}^d \int_{\partial S} a_i \widehat{u}_h [[\chi]]_{\partial S} \, ds \\ &= ((u_h)_t, \phi - \chi)_\Omega - \sum_{i=1}^d (a_i u_h, (\phi - \chi)_{x_i})_\Omega - \sum_S \sum_{i=1}^d \int_{\partial S} a_i \widehat{u}_h [\phi - \chi]_{\partial S} \, ds \\ &= ((u_h)_t, \phi - \chi)_\Omega + \sum_{i=1}^d (a_i (u_h)_{x_i}, \phi - \chi)_\Omega + \sum_S \sum_{i=1}^d \int_{\partial S} a_i [[u_h(\phi - \chi)]]_{\partial S} \, ds \\ &\quad - \sum_S \sum_{i=1}^d \int_{\partial S} a_i \widehat{u}_h [\phi - \chi]_{\partial S} \, ds. \end{aligned} \quad (2.2.11)$$

Let $\chi = \mathbb{P}_h \phi$ be the projection of ϕ onto the approximation space. Since the

projection error $\phi - \mathbb{P}_h\phi$ is orthogonal to the approximation space, it holds that

$$((u_h)_t, \phi - \mathbb{P}_h\phi)_\Omega + \sum_{i=1}^d (a_i(u_h)_{x_i}, \phi - \mathbb{P}_h\phi)_\Omega = 0.$$

This leaves us only needing to bound the flux terms:

$$\frac{d}{dt}(u_h, \phi)_\Omega = \sum_S \sum_{i=1}^d a_i \llbracket u_h(\phi - \mathbb{P}_h\phi) \rrbracket_{\partial S} - \sum_S \sum_{i=1}^d a_i \widehat{u}_h \llbracket \phi - \mathbb{P}_h\phi \rrbracket_{\partial S}. \quad (2.2.12)$$

When we choose \widehat{u}_h to be the upwind-biased flux, the the above equation becomes

$$\frac{d}{dt}(u_h, \phi)_\Omega = \sum_S \sum_{i=1}^d a_i \int_{\partial S} \llbracket u_h \rrbracket_{\partial S} ((\phi - \mathbb{P}_h\phi)_{\partial S}^R \mathbf{n}_R) - (1 - \theta_i) \llbracket \phi - \mathbb{P}_h\phi \rrbracket_{\partial S} \, ds. \quad (2.2.13)$$

Let $C_\theta = \max\{|1 - \theta_1|, \dots, |1 - \theta_d|\}$. Then

$$\begin{aligned} \left| \frac{d}{dt}(u_h, \phi)_\Omega \right| &\leq C \left(\frac{1}{h} \int_0^T \llbracket u_h \rrbracket_{\partial S}^2 dt \right)^{1/2} \left[\left(\int_0^T \|\phi - \mathbb{P}_h\phi\|_{\partial S}^2 dt \right)^{1/2} \right. \\ &\quad \left. + C_\theta \left(\frac{1}{h} \int_0^T \llbracket \phi - \mathbb{P}_h\phi \rrbracket_{\partial S}^2 dt \right)^{1/2} \right] \\ &\leq C \left(\frac{1}{h} h^{2(p+1)} \right)^{1/2} \left[h^{p+\frac{1}{2}} \|\phi\|_{p+1} + C_\theta h^{p+\frac{1}{2}} \|\phi\|_{p+1} \right] \\ &= C (1 + C_\theta) h^{2p+1} \|\phi\|_{p+1}(T) \\ &= C_2 h^{2p+1} \|\Phi\|_{p+1}, \end{aligned} \quad (2.2.14)$$

where the constant C_2 depends on θ .

Combining the estimates (2.2.10) and (2.2.14) and using the periodicity of the boundary conditions, we conclude with a bound on the numerator in the definition of the negative-order norm:

$$(u - u_h, \Phi)_\Omega(T) \leq C_1 h^{2p+2} \|u_0\|_{p+1} \|\phi(0)\|_{p+1} + C_2 h^{2p+1} \|\Phi\|_{p+1}.$$

□

Remark. The penalty for using the new flux is limited to a contribution to the

constant attached to the order term in the negative-order norm error estimate and we can extract the same global order of accuracy, $\mathcal{O}(h^{2p+1})$, for any polynomial degree p . This is in contrast to the changing local behaviour seen in the pointwise analysis in the first half of this chapter. We leave to further work the investigation of a tighter bound that explains how to choose θ so that the error constant associated to the post-processed approximation is minimised.

2.2.3 Numerical experiments

We present a numerical discussion for the test equation

$$\begin{aligned} u_t + u_x &= 0, & (x, t) &\in [0, 2\pi] \times (0, T), \\ u(x, 0) &= \sin(x), & u(0, T) &= u(2\pi, T) \end{aligned} \quad (2.2.15)$$

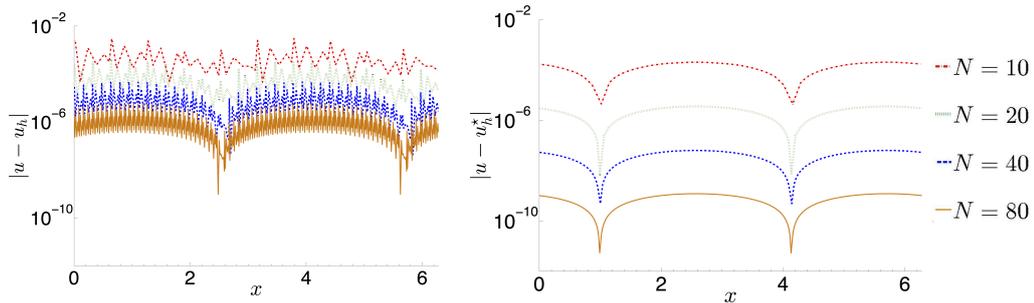
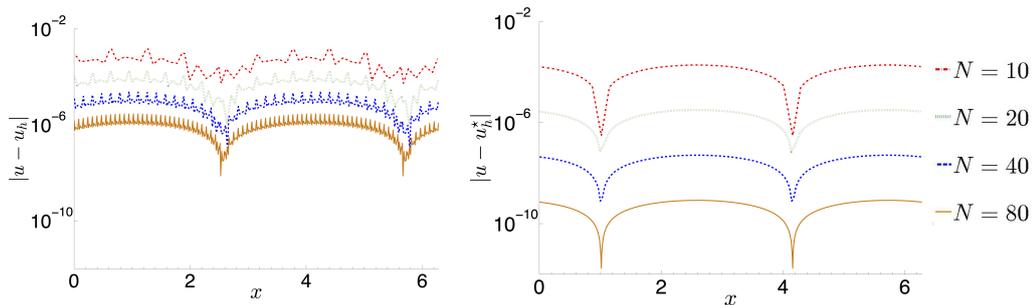
solved by the DG scheme with upwind-biased flux paired with the three-stage third-order Strong Stability Preserving Runge-Kutta timestepping method described in [30]. The CFL is taken so that spatial errors dominate.

Tables 2.3 and 2.4 illustrate the $\mathcal{O}(h^{p+1})$ accuracy of the DG solution in the \mathcal{L}^2 - and \mathcal{L}^∞ -norms. After post-processing by the SIAC filter, we observe the $\mathcal{O}(h^{2p+1})$ accuracy in the \mathcal{L}^2 -norm described in previous section and we also see $\mathcal{O}(h^{2p+1})$ accuracy in the \mathcal{L}^∞ -norm. For odd p , convergence to the expected orders is slower for lower values of θ but is eventually achieved. Furthermore, if one compares the same degrees of mesh refinement for decreasing values of θ , one observes increasing errors for odd p and reducing errors for even p . For the post-processed solution, this is due in large part to the contribution of θ to the constant attached to the order term in the error estimate of Theorem 2.2.1.

The highly oscillatory nature of the DG solution, indicating the existence of the hidden superconvergent points, can be seen in Figures 2.2.2 and 2.2.3 alongside the post-processed solutions which have increased smoothness and improved

Table 2.3: \mathcal{L}^2 - and \mathcal{L}^∞ -norms of errors before and after post-processing for case $p = 2$.

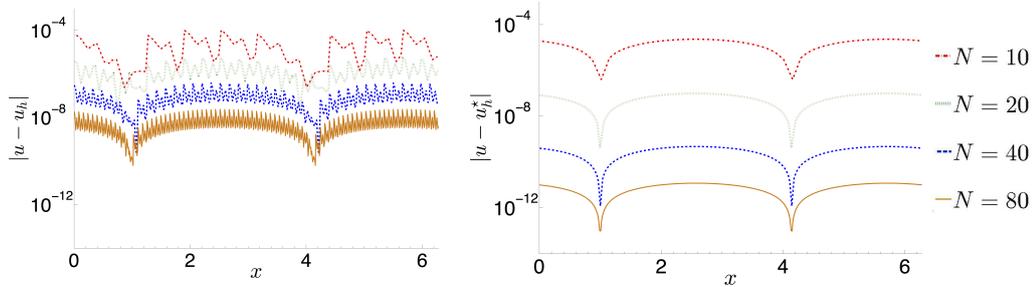
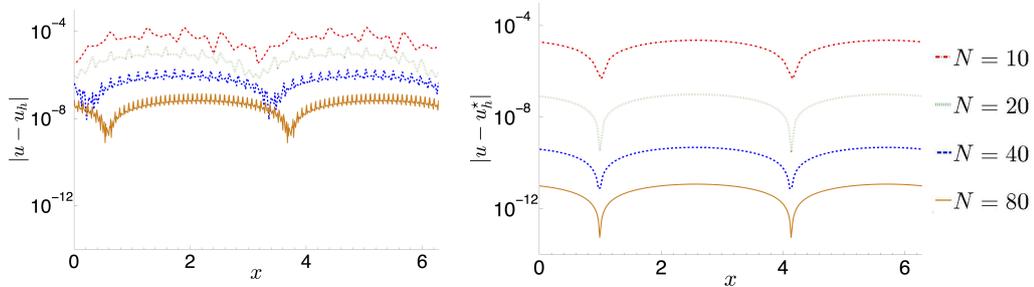
mesh	\mathcal{P}^2 : Before filter				\mathcal{P}^2 : After filter			
	\mathcal{L}^2 error	order	\mathcal{L}^∞ error	order	\mathcal{L}^2 error	order	\mathcal{L}^∞ error	order
$\theta = 1$								
10	8.59E-04	-	3.02E-03	-	1.43E-04	-	2.04E-04	-
20	1.06E-04	3.00	3.66E-03	3.04	2.52E-06	5.83	3.85E-06	5.83
40	1.33E-05	2.99	4.62E-05	2.98	4.46E-08	5.81	6.34E-08	5.82
$\theta = 0.85$								
10	7.35E-04	-	2.61E-03	-	1.41E-04	-	2.01E-04	-
20	9.03E-05	3.02	3.10E-04	3.07	2.44E-06	5.86	3.47E-06	5.86
40	1.12E-05	3.00	3.85E-05	3.00	4.19E-08	5.86	5.95E-08	5.86
$\theta = 0.55$								
10	5.66E-04	-	1.46E-03	-	1.36E-03	-	1.93E-04	-
20	6.97E-05	3.01	1.86E-04	2.97	2.26E-06	5.91	3.20E-06	5.91
40	8.70E-06	3.00	2.31E-05	3.00	3.63E-08	5.95	5.15E-08	5.96

Figure 2.2.2: DG and filtered errors for $p = 2$ at time $T = 1$.(a) Before and after post-processing for $\theta = 1$.(b) Before and after post-processing for $\theta = 0.55$.

accuracy. The reduced numerical viscosity enforced by the upwind-biased flux is evident when comparing plots for $\theta = 1$ and $\theta = 0.55$.

Table 2.4: \mathcal{L}^2 - and \mathcal{L}^∞ -norms of errors before and after post-processing for case $p = 3$.

mesh	\mathcal{P}^3 : Before filter				\mathcal{P}^3 : After filter			
	\mathcal{L}^2 error	order	\mathcal{L}^∞ error	order	\mathcal{L}^2 error	order	\mathcal{L}^∞ error	order
$\theta = 1$								
10	2.35E-04	-	1.91E-04	-	1.61E-05	-	2.28E-05	-
20	1.30E-05	4.16	1.06E-05	4.16	6.97E-08	7.86	9.81E-08	7.86
40	8.67E-07	3.91	7.33E-07	3.86	3.34E-10	7.69	4.72E-10	7.69
$\theta = 0.85$								
10	2.74E-04	-	2.18E-04	-	1.61E-05	-	2.28E-05	-
20	1.63E-05	4.06	1.31E-05	4.06	6.94E-08	7.86	9.82E-08	7.86
40	1.07E-06	3.92	8.81E-07	3.89	3.34E-10	7.69	4.73E-10	7.69
$\theta = 0.55$								
10	4.04E-04	-	2.65E-04	-	1.61E-05	-	2.28E-05	-
20	4.99E-05	3.01	3.22E-05	3.04	6.96E-08	7.85	9.85E-08	7.85
40	4.72E-06	3.40	2.97E-06	3.43	3.39E-10	7.68	4.80E-10	7.68

Figure 2.2.3: DG and filtered errors for $p = 3$ at time $T = 1$.(a) Before and after post-processing for $\theta = 1$.(b) Before and after post-processing for $\theta = 0.55$.

3

Timestepping

“The future is but the present a little further on.”

– Jules Verne

Having focussed in Chapter 2 on the DG spatial discretisation in isolation, we now turn our attention to methods for time integration. A considerable downfall of the current methodologies is that the time-stepping method ultimately destroys the superconvergent accuracy obtained by the DG spatial discretisation. While we do not address this issue directly in this thesis, by better understanding how the effects of the numerical flux – including superconvergence – are carried forwards in time, we lay some of the foundations for further work and help to articulate the current state of affairs from which we may progress.

Consider a discretisation of the time interval $[0, T]$ into points t_n of, for the sake of simplicity, uniform separation Δt so that $t_{n+1} = t_n + \Delta t$. Denote by $u^n = u(x, t_n)$ the value of a function u at time $t = t_n$. In this chapter, we discuss methods for evolving in time the solution to a hyperbolic conservation law. We initially consider two opposing approaches to the temporal discretisation. Firstly, we describe an example of a fully-decoupled Method of Lines that was the prescribed treatment when DG schemes were originally developed and which remains, arguably, the most popular approach. Secondly, we describe an approach that couples the two discretisations and that has

increased in popularity more recently. In §3.3 and §3.4, we consider a new example of a scheme that lies halfway between these two approaches.

3.1 Strong Stability Preserving Runge-Kutta Methods

Consider an ODE

$$\frac{d}{dt}u = \mathcal{D}(u).$$

The linear operator \mathcal{D} could arise, for example, from a DG discretisation of the partial derivative $f(u)_x$ per §1.3.2. A particular class of RK methods named Strong Stability Preserving (SSP) RK methods *assume* that, with a suitable time-step restriction $\Delta t \leq \Delta t_0$ which often depends on the spatial discretisation, the first-order Euler time discretisation (1.3.9) is stable under some norm or semi-norm:

$$\|u + \Delta t \mathcal{D}(u)\| \leq \|u\|. \quad (3.1.1)$$

Under this assumption, Shu and Osher [54] showed that a Runge-Kutta method of the form

$$\begin{aligned} u^{(0)} &= u^n \\ u^{(i)} &= \sum_{j=0}^{i-1} \left(\alpha_{ij} u^{(j)} + \Delta t \beta_{ij} \mathcal{D}(u^{(j)}) \right), \quad 1 \leq i \leq s \\ u^{n+1} &= u^{(s)}, \end{aligned}$$

where $\alpha_{ij}, \beta_{ij} \geq 0$, is, often under the same time-step restriction, strongly stable:

$$\|u_h^{n+1}\| \leq \|u_h^n\|.$$

The CFL number ν , which is related to the maximal ratio of temporal to spatial step sizes and is often computed via a von Neumann analysis, can be a very limiting constraint on the step size Δt . Incidentally, it is extremely convenient

that this class of RK methods can be written as convex combinations of forward Euler stages since the SSP properties of the forward Euler method are maintained.

Perhaps the most popular of such methods is the three-stage third-order method *SSPRK(3,3)*:

$$u^{(1)} = u^n + \Delta t \mathcal{D}(u^n) \quad (3.1.2a)$$

$$u^{(2)} = \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t \mathcal{D}(u^{(1)}) \quad (3.1.2b)$$

$$u^{n+1} = \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t \mathcal{D}(u^{(2)}). \quad (3.1.2c)$$

To illustrate a link between RK methods and the multiderivative methods described later in this chapter, we rewrite the stages of *SSPRK(3,3)* in terms of the solution at the previous time-step:

$$u^{(1)} = u^n + \Delta t \mathcal{D}(u^n) \quad (3.1.3a)$$

$$u^{(2)} = u^n + \Delta t \mathcal{D}\left(\frac{1}{2}u^n + \frac{1}{4}\Delta t \mathcal{D}u^n\right) \quad (3.1.3b)$$

$$u^{n+1} = u^n + \Delta t \mathcal{D}\left(u^n + \frac{1}{2}\Delta t \mathcal{D}u^n + \frac{1}{6}\Delta t^2 \mathcal{D}^2 u^n\right). \quad (3.1.3c)$$

Note that the final stage has a similar form to a Taylor expansion.

3.2 Multiderivative Methods: Lax-Wendroff DG

An alternative to the MoL seeks to intertwine the two discretisations and make use of higher order derivatives of the solution. In the context of hyperbolic equations, the original LW-DG scheme was proposed by Qiu et al. in [46]. Beginning with an order- m truncated Taylor expansion in time of the solution u :

$$u(x, t + \Delta t) \approx u(x, t) + \sum_{i=1}^m \frac{\Delta t^i}{i!} \frac{\partial^i}{\partial t^i} u(x, t), \quad (3.2.1)$$

the Cauchy-Kowalewski procedure is employed to convert, using the original equation, the temporal derivatives into spatial derivatives. We rewrite the

derivatives in the following recursive manner in which we aim to factor out a spatial derivative:

$$u_t = -f(u)_x \quad (3.2.2a)$$

$$u_{tt} = (-f(u)_x)_t = -(f(u)_t)_x = -(f'(u)u_t)_x = (f'(u)f(u)_x)_x \quad (3.2.2b)$$

$$\begin{aligned} u_{ttt} &= (f'(u)f(u)_x)_{xt} = (f'(u)f(u)_x)_{tx} = (f''(u)u_t f(u)_x + f'(u)(f'(u)u_t)_x)_x \\ &= -\left(f''(u)(f(u)_x)^2 + f'(u)(f'(u)f(u)_x)_x\right)_x. \end{aligned} \quad (3.2.2c)$$

The third order ($m = 3$) method sets a LW-type flux equal to

$$\tilde{f} = f(u) - \frac{\Delta t}{2} f'(u)f(u)_x + \frac{\Delta t^2}{6} \left(f''(u)(f(u)_x)^2 + f'(u)(f'(u)f(u)_x)_x \right) \quad (3.2.3)$$

so that the Taylor expansion (3.2.1) becomes, upon substitution of the derivatives (3.2.2a)-(3.2.2c),

$$u(x, t + \Delta t) \approx u(x, t) - \Delta t \tilde{f}(u, \Delta t)_x. \quad (3.2.4)$$

The DGM can now be applied locally to discretise the derivative term:

$$\int_{I_j} u_h(x, t + \Delta t) v \, dx = \int_{I_j} u_h(x, t) v \, dx - \Delta t \mathcal{B}_j \left(\tilde{f}(u_h, \Delta t), v \right). \quad (3.2.5)$$

It remains to describe how to treat the derivative terms such as $f(u)_x$ and $(f'(u)f(u)_x)_x$ appearing in the LW-type flux (3.2.3). Indeed, there is some leeway in how to do so and the choice informs the construction of the numerical flux \tilde{f} in the DG term $\mathcal{B}(\tilde{f}(u_h, \Delta t), v)$ in equation (3.2.5). We discuss, in the context of the scheme in the next section, two ways of computing the derivative $f(u)_x$: by direct differentiation and by introducing an auxiliary variable. These treatments within the Lax-Wendroff framework result in the original and new methods presented in [46] and [31] respectively and we refer the reader to these papers for an explicit discussion of these methods.

Note that, although equation (3.2.5) results in, after substitution of the

Galerkin expansion of u_h and the inversion of a mass matrix, a similar form to the RK method described in the previous section, these methods are not the same since the spatial discretisations usually employ different choices of numerical flux function. Whilst the same DG operator is applied to each stage value of a RK method, the derivative terms comprising a LW-type flux require application of different numerical fluxes in order to respect their differing physical properties with regards to the wind direction.

In the remainder of this chapter, we discuss a method that, in some sense, lies halfway between Runge-Kutta and Lax-Wendroff methods.

3.3 DG-TDRK4

We introduce the class of multi-stage multiderivative methods discussed in [51] by means of the same case study with two stages and two derivatives. We begin with a review of this work in which higher-order derivatives are computed directly from the basis functions.

Let $u^n := u(x, t^n)$. Recalling the LW and RK forms (3.2.4) and (3.1.3), we consider two-stage methods where we apply a DG discretisation to the derivative on the right-hand side of the general stage equation

$$q = u^n - \Delta t \tilde{f}_x, \quad (3.3.1)$$

where

$$\tilde{f} = \alpha f(u^n) + \alpha^* f(u^*) + \beta \Delta t f'(u^n) f(u^n)_x + \beta^* \Delta t f'(u^*) f(u^*)_x \quad (3.3.2)$$

is called a *modified flux* (that is, it is a modification of the LW-type flux function (3.2.3)). In this way, by multiplying equation (3.3.1) by a test function v and

integrating by parts over a cell I_j , we obtain the weak formulation

$$\int_{I_j} u_h v \, dx = \int_{I_j} u_h^n v \, dx + \Delta t \mathcal{B}_j(\tilde{f}, v), \quad (3.3.3)$$

where \mathcal{B}_j is the DG operator defined in equation (1.3.16). The quantity u^* is the value of the solution at the intermediate stage. Note that this process can include both Runge-Kutta methods (when, for each stage, $\beta = 0 = \beta^*$) and Lax-Wendroff methods (when there is just one stage). Further stages can be added by including more stage quantities (e.g. u^{**}) and more derivatives can be inserted by using further terms derived from the Cauchy-Kowalewski procedure (3.2.2a)-(3.2.2c).

The two-stage explicit method that we discuss here has in equation (3.3.2)

$$(\alpha, \alpha^*, \beta, \beta^*) = \left(\frac{1}{2}, 0, -\frac{1}{8}, 0 \right)$$

for the intermediate stage and

$$(\alpha, \alpha^*, \beta, \beta^*) = \left(1, 0, -\frac{1}{6}, -\frac{1}{3} \right)$$

for the full update. These values are unique and are determined by an analysis of the order conditions that can be found in [18].

We now formally define the DG implementation of the two-derivative Runge Kutta method. We refer to this as *TDRK4-DG*. Given the TDRK4-DG numerical solution $u_h^n := u_h(x, t_n)$ at time $t = t_n$, the solution at time $t = t_n + \Delta t$ is determined by the following process: find $u_h^*, u_h^{n+1} \in V_h^k$ such that, for all $w, v \in V_h^k$ and for all $j = 1, \dots, N$, there holds on each element I_j

$$\int_{I_j} u_h^* w \, dx = \int_{I_j} u_h^n w \, dx + \Delta t \mathcal{B}_j \left(\frac{1}{2} f(u_h^n) + \frac{\Delta t}{8} f'(u_h^n) f(u_h^n)_x, w \right) \quad (3.3.4a)$$

$$\begin{aligned} \int_{I_j} u_h^{n+1} v \, dx &= \int_{I_j} u_h^n v \, dx \\ &+ \Delta t \mathcal{B}_j \left(f(u_h^n) + \frac{\Delta t}{6} f'(u_h^n) f(u_h^n)_x + \frac{\Delta t}{3} f'(u_h^*) f(u_h^*)_x, v \right). \end{aligned} \quad (3.3.4b)$$

It remains to define the approximation $f(u_h^n)_x$ to the derivative and to choose corresponding numerical flux functions $\widehat{f}, \widehat{f'f_x}$ in the DG terms in equations (3.3.4a) and (3.3.4b).

The most convenient treatment of the derivative $f(u)_x$ on a cell \mathcal{I}_j is to directly differentiate the Galerkin expansion:

$$\partial_x f_j(\xi) := \frac{2}{h} \sum_{\ell=0}^k F_j^{(\ell)}(t) \frac{\partial \varphi^{(\ell)}}{\partial \xi}(\xi).$$

The Galerkin expansion of $g_{h,j} = \partial_x f_j$ is then found by projection:

$$G_j^{(m)}(t) = \frac{1}{2} \int_{\mathcal{I}_j} \partial_x f_j(\xi) \varphi^{(m)}(\xi) \, d\xi, \quad m = 0, \dots, p.$$

The original LW-DG scheme of Qiu et al. [46] and the TDRK4-DG scheme described previously in [51] employed this approach. In this case, the modified flux \tilde{f} can be treated as a single quantity and one can prescribe in equations (3.3.4a) and (3.3.4b) the Rusanov flux function

$$\widehat{f}_{j+\frac{1}{2}} = \frac{1}{2} \left(\tilde{f}_{j+\frac{1}{2}}^+ + \tilde{f}_{j+\frac{1}{2}}^- - \gamma (u_{j+\frac{1}{2}}^+ - u_{j+\frac{1}{2}}^-) \right), \quad \gamma = \max |f'(u_j), f'(u_{j+1})|, \quad (3.3.5)$$

where f is the flux function in the hyperbolic conservation law $u_t + f(u)_x = 0$, \tilde{f} is the first input of the DG operator \mathcal{B}_j in equation (3.3.4a) or (3.3.4b), the superscript $+$ and $-$ denote evaluation from the right and left of cell boundary points respectively, and the hat notation denotes a single-valued numerical flux defined at each cell boundary. This function may be seen as a central flux with a dissipative correction term to ensure stability. Note that for this scheme, we take the correction terms $u_{j+\frac{1}{2}}^\pm$ in equation (3.3.5) at time $t = t_n$ in both the intermediate stage and the full update.

It turns out, as demonstrated in §3.5, that the TDRK4-DG method described above does not benefit from superconvergence properties similar to those seen in the SSPRK-DG methods. In the next section, we define a new scheme that does

exhibit superconvergence. Such a property benefits the method's ability to accurately propagate waves.

3.4 A New DG-TDRK4 Method

We follow Guo et al.'s incorporation of LDG technologies into the LW-DG method to describe a new class of multi-stage multiderivative methods which are superconvergent. As an alternative to the direct treatment of higher-order derivatives in §3.3, we propose treating $g = f(u)_x$ as an auxiliary variable and instead approximating derivatives by a DG scheme. In effect, this results in an extra two equations compared to the formulation (3.3.4a)-(3.3.4b). Furthermore, in order to regain superconvergence, the numerical flux within the scheme must carefully balance the form of the higher-order derivatives. We describe the process through a modification of the TDRK4-DG method.

Given the numerical solution $u_h^n := u_h(x, t^n)$ at time $t = t^n$, we determine the *modified TDRK4-DG* solution at time $t = t^n + \Delta t$ by the following process: find $g_h^n, g_h^*, u_h^*, u_h^{n+1} \in V_h^p$ such that, for all $\varphi, \phi, \zeta, \eta \in V_h^p$ and for all $j = 1, \dots, N$, there holds on each element I_j

$$\int_{I_j} g_h^n \varphi \, dx = -\mathcal{B}_j^- (f(u_h^n), \varphi) \quad (3.4.1a)$$

$$\int_{I_j} u_h^* \phi \, dx = \int_{I_j} u_h^n \phi \, dx + \Delta t \mathcal{B}_j \left(\frac{1}{2} f(u_h^n) - \frac{\Delta t}{8} f'(u_h^n) g_h^n, \phi \right) \quad (3.4.1b)$$

$$\int_{I_j} g_h^* \zeta \, dx = -\mathcal{B}_j^- (f(u_h^*), \zeta) \quad (3.4.1c)$$

$$\int_{I_j} u_h^{n+1} \eta \, dx = \int_{I_j} u_h^n \eta \, dx + \Delta t \mathcal{B}_j \left(f(u_h^n) - \frac{\Delta t}{6} f'(u_h^n) g_h^n - \frac{\Delta t}{3} f'(u_h^*) g_h^*, \eta \right). \quad (3.4.1d)$$

The notation \mathcal{B}_j^- in the equations (3.4.1a) and (3.4.1c) for the derivative denotes use of the upwind numerical flux f^- . This must be counter-balanced within equations (3.4.1b) and (3.4.1d), both for stability reasons and in order to observe

superconvergence. Moreover, the modified flux functions

$$\tilde{f}_h = \frac{1}{2}f(u_h^n) - \frac{\Delta t}{8}f'(u_h^n)g_h^n$$

in equation (3.4.1b) and

$$\tilde{f}_h = f(u_h^n) - \frac{\Delta t}{6}f'(u_h^n)g_h^n - \frac{\Delta t}{3}f'(u_h^*)g_h^*$$

in equation (3.4.1d), in contrast to the scheme with direct differentiation, include information from the left-neighbour cell (in light of the definition of g in equations (3.4.1a) and (3.4.1c)) and must be treated component-wise when applying the numerical flux within the scheme in order to respect the alternating principle. To achieve this, we propose using the same fluxes as [32] within the stage equations (3.4.1b) and (3.4.1d): we choose $\widehat{f}_{j+\frac{1}{2}}$ to be a standard upwind monotone flux $\widehat{f} = f^-$, and set

$$\widehat{f}g_{j+\frac{1}{2}} = \frac{\llbracket f \rrbracket_{j+\frac{1}{2}}}{\llbracket u \rrbracket_{j+\frac{1}{2}}}g_{j+\frac{1}{2}}^+,$$

where the “jump” is defined by $\llbracket w \rrbracket_{j+\frac{1}{2}} := w_{j+\frac{1}{2}}^+ - w_{j+\frac{1}{2}}^-$. Note that for linear advection when $f(u) = u$, this reduces to $\widehat{g}_{j+\frac{1}{2}} = g_{j+\frac{1}{2}}^+$. The downwinding on g is chosen to provide a symmetrical approximation when combined with the upwinding used in its computation. Further guidance on how to make such choices in accordance with the alternating principle can be procured from [59].

3.4.1 Stability analysis

In this section, we review the process for proving a stability energy estimate for Runge-Kutta and modified Lax-Wendroff discontinuous Galerkin schemes in preparation for approaching the modified TDRK4 scheme. We outline the approaches taken to prove stability for RK-DG schemes in [66] and LW-DG schemes in [55]. We work towards preparation of the machinery necessary for stability analysis of our proposed TDRK4 scheme.

As well as the results included in §1.3.1, we require the following useful results.

Lemma 3.4.1. *For any $u, v, w \in \mathbb{R}$, we have*

$$vw = \frac{1}{2}v^2 + \frac{1}{2}w^2 - \frac{1}{2}(v-w)^2 \quad (3.4.2)$$

$$(u-v \pm w)^2 = (u-v)^2 \pm 2(u-v \pm w)w - w^2. \quad (3.4.3)$$

Proof: The proof follows from a trivial algebraic manipulation. \square

Consider the linear advection equation in one spatial dimension. The *SSPRK(3,3)* time-stepping method paired with upwind DG spatial discretisation results in the scheme

$$\left(u_h^{(1)}, \varphi_h\right)_\Omega = \left(u_h^n, \varphi_h\right)_\Omega + \Delta t \mathcal{B}^-(u_h^n; \varphi_h) \quad (3.4.4a)$$

$$\left(u_h^{(2)}, \psi_h\right)_\Omega = \frac{3}{4} \left(u_h^n, \psi_h\right)_\Omega + \frac{1}{4} \left(u_h^{(1)}, \psi_h\right)_\Omega + \frac{1}{4} \Delta t \mathcal{B}^-(u_h^{(1)}; \psi_h) \quad (3.4.4b)$$

$$\left(u_h^{n+1}, \eta_h\right)_\Omega = \frac{1}{3} \left(u_h^n, \eta_h\right)_\Omega + \frac{2}{3} \left(u_h^{(2)}, \eta_h\right)_\Omega + \frac{2}{3} \Delta t \mathcal{B}^-(u_h^{(2)}; \eta_h). \quad (3.4.4c)$$

We outline the process as follows:

- The goal is to prove that $\|u^{n+1}\|^2 - \|u^n\|^2 \leq 0$.
- Test each stage of the global form (3.4.4) against the solution at the previous stage (where we take $u_h^{(0)} = u_h^n$) to obtain inner products of different stage values, \mathcal{L}^2 -norms and, via the definition (1.3.31) of the global DG operator \mathcal{B}^\pm , jump semi-norms.
- Form a linear combination of the tested stage equations. Possibly using Lemma 3.4.1, rewrite the inner products as inner products on “difference operators” \mathbb{D}_i that satisfy

$$\left(\mathbb{D}_i, v\right)_\Omega = \frac{\Delta t}{i!} \mathcal{B}^-(\mathbb{D}_{i-1}, v) \quad \text{for } i = 1, 2, \dots, s \quad \text{with } \mathbb{D}_0 = u^n. \quad (3.4.5)$$

Note that $\mathbb{D}_1 = u_h^{(1)} - u_h^n$ can be determined by rearranging the first stage equation (3.4.4a).

- This process results in the equation

$$\|u^{n+1}\|^2 - \|u^n\|^2 = -\Pi_1 + \Pi_2,$$

where Π_1 comprises the jump-semi-norms that arose from the DG terms and Π_2 comprises the remaining inner products and any other positive terms on the right-hand side. To obtain the stability result, seek to bound Π_2 by Π_1 .

For the modified LW-DG method, it is convenient to introduce an additional auxiliary variable and write the update only in terms of inner products:

$$(u^{n+1}, \varphi)_\Omega = (u^n, \varphi)_\Omega - \Delta t (p^n, \varphi)_\Omega + \frac{\Delta t^2}{2} (q^n, \varphi)_\Omega - \frac{\Delta t^3}{6} (r^n, \varphi)_\Omega \quad (3.4.6a)$$

$$(p^n, \psi)_\Omega = -\mathcal{B}^-(u^n, \psi) \quad (3.4.6b)$$

$$(q^n, \eta)_\Omega = -\mathcal{B}^+(p^n, \eta) \quad (3.4.6c)$$

$$(r^n, \zeta)_\Omega = -\mathcal{B}^+(r^n, \zeta). \quad (3.4.6d)$$

While the approach for LW-DG is similar to that for RK-DG, there are some important differences resulting from the presence of the derivative terms.

- The definitions (3.4.6b-3.4.6d) of the derivatives can be used as a mechanism for transforming inner products to jumps and norms via the DG terms. One advantage of writing the update in the form of (3.4.6a) is that it implies that $u^{n+1} = u^n - \Delta t p^n + \frac{\Delta t^2}{2} q^n - \frac{\Delta t^3}{6} r^n$.
- Test the scheme against the solution u^n at the previous time-step. Use the algebraic identities in Lemma 3.4.1 to rewrite the resulting norm terms to incorporate a difference operator $\mathbb{B}_1 = u_h^{n+1} - u_h^n + \Delta t p_h^n = \frac{\Delta t^2}{2} q^n - \frac{\Delta t^3}{6} r^n$ that now includes the derivative p^n . Products of jump terms can be treated by the Schwarz inequality.
- Ultimately, to deal with a positive jump semi-norm in p_h^n , extra terms must be added to balance the inequality so that the final expression takes the

form

$$(\|u^{n+1}\|^2 + \Delta t^2 \|p^{n+1}\|^2) - (\|u^n\|^2 + \Delta t \|p^n\|^2) \leq \Pi \leq 0.$$

To bound these extra terms, one can work with the quantity $\mathbb{B}_2 = p_h^{n+1} - p_h^n + \Delta t q_h^n$, employing a similar approach to the one already developed.

Note that for the second order methods (RK2 and LW2 time-stepping), there is less machinery available, in particular for dealing with positive jump semi-norm terms, and one needs to use derivative properties of the piecewise linear basis; the proofs are actually more challenging than those for the third-order schemes.

The two-stage two-derivative TDRK4 scheme for the linear advection equation can be written as

$$\begin{aligned} (u^*, \varphi)_\Omega &= (u^n, \varphi)_\Omega - \frac{1}{2} \Delta t (p^n, \varphi)_\Omega + \frac{\Delta t^2}{8} (q^n, \varphi)_\Omega \\ (p^n, \psi)_\Omega &= -\mathcal{B}^-(u^n, \psi) \\ (q^n, \eta)_\Omega &= -\mathcal{B}^+(p^n, \eta) \\ (u^{n+1}, \varphi)_\Omega &= (u^n, \varphi)_\Omega - \Delta t (p^n, \varphi)_\Omega + \frac{\Delta t^2}{6} (q^n, \varphi)_\Omega + \frac{\Delta t^2}{3} (q^*, \varphi)_\Omega \\ (p^*, \psi)_\Omega &= -\mathcal{B}^-(u^*, \psi) \\ (q^*, \eta)_\Omega &= -\mathcal{B}^+(p^*, \eta) \end{aligned}$$

The stability analysis for this scheme is the topic of ongoing work.

3.5 Numerical Experiments

We present results for the linear advection equation that demonstrate that, after a post-processing by the SIAC filter, superconvergence is regained by our new TDRK4-DG method. Numerical results for nonlinear equations is left to future work.

3.5.1 Linear advection

Consider the linear advection equation

$$u_t + u_x = 0, \quad x \in [0, 2\pi], \quad u_0(x) = \sin(x). \quad (3.5.1)$$

Table 3.1 shows \mathcal{L}^2 - and \mathcal{L}^∞ -errors and orders of accuracy for the DG solution and the post-processed DG solution evolved in time by our new TDRK4 method, the original TDRK4-DG scheme investigated in [51] and the ten-stage fourth-order SSP Runge-Kutta scheme. For the Method of Lines RK-DG scheme, we employ three different values of the DG flux parameter θ : the purely upwind flux when $\theta = 1$ and one value either side of 1. According to the observations of Chapter 2, we expect $\theta = 1.5 > 1$ to offer favourable results for this odd ($p = 3$) polynomial order compared to $\theta = 0.75 < 1$ and even $\theta = 1$. The CFL number $\nu = \frac{\Delta t}{\Delta x}$ is taken to be small enough for the spatial errors to dominate so that post-processed superconvergence can be observed. A full numerical investigation of the maximal CFL for the TDRK4 methods and for RK-DG schemes with $\theta \neq 1$ is left to further work. As expected, when compared to the upwind scheme, the solution obtained with $\theta = 1.5 > 1$ sees a reduction in both the \mathcal{L}^2 - and \mathcal{L}^∞ -errors while the solution obtained with $\theta = 0.75 < 1$ is inferior and has slower convergence the pre-processed $\mathcal{O}(h^{p+1})$ order of accuracy. The disparity between the post-processed SSPRK-DG for different values of θ is lessened compared to the pre-processed solutions, suggesting that the power of the flux parameter to reduce the constants in the SIAC error estimate is minimal.

As claimed, the modified TDRK4-DG scheme that uses techniques borrowed from LDG to compute higher-order derivatives displays superconvergent $\mathcal{O}(h^{2p+2})$ post-processed orders of accuracy whilst the initial inflation of the orders in the post-processed original TDRK4-DG solution dies off and the order converges to the expected $\mathcal{O}(h^{p+1})$. In fact, the numerical results in this case for the new TDRK4-DG method are practically identical to those for upwind RK-DG.

mesh	\mathcal{P}^3 : Pre-processed				\mathcal{P}^3 : Post-processed			
	\mathcal{L}^2 error	order	\mathcal{L}^∞ error	order	\mathcal{L}^2 error	order	\mathcal{L}^∞ error	order
Modified TDRK4: $\nu = 0.01$								
16	1.29E-03	-	3.56E-03	-	5.85E-03	-	1.87E-02	-
32	7.99E-05	4.02	2.20E-04	4.01	3.99E-05	7.19	1.27E-04	7.19
64	5.05E-06	3.98	1.45E-05	3.92	1.86E-07	7.74	5.74E-07	7.79
128	3.15E-07	4.00	9.27E-07	3.97	7.56E-10	8.05	2.33E-09	8.05
256	1.97E-08	3.99	5.80E-08	3.99	2.99E-12	8.03	9.33E-12	8.02
Original TDRK4: $\nu = 0.01$								
16	1.27E-03	-	3.16E-03	-	5.88E-03	-	1.88E-02	-
32	8.03E-05	3.98	2.06E-04	3.93	4.18E-05	7.13	1.31E-04	7.16
64	5.13E-06	3.96	1.39E-05	3.89	3.03E-07	7.10	8.64E-07	7.25
128	3.21E-07	3.99	8.85E-07	3.97	1.08E-08	4.87	3.32E-08	4.76
256	2.00E-08	3.99	5.53E-08	3.99	6.48E-10	4.09	2.05E-09	4.04
SSPRK(10,4)($\theta = 1$): $\nu = 0.01$								
16	1.28E-03	-	3.53E-03	-	5.84E-03	-	1.87E-02	-
32	7.97E-05	4.01	2.20E-04	4.00	3.99E-05	7.19	1.27E-04	7.19
64	5.05E-06	3.98	1.45E-05	3.92	1.86E-07	7.74	5.74E-07	7.79
128	3.15E-07	4.00	9.27E-07	3.97	7.56E-10	8.05	2.33E-09	8.05
256	1.97E-08	3.99	5.80E-08	3.99	2.99E-12	8.03	9.33E-12	8.02
SSPRK(10,4)($\theta = 1.5$): $\nu = 0.01$								
16	1.30E-03	-	3.47E-03	-	5.85E-03	-	1.87E-02	-
32	6.49E-05	4.33	1.72E-04	4.33	3.99E-05	7.19	1.27E-04	7.20
64	3.92E-06	4.04	1.04E-05	4.03	1.85E-07	7.74	5.72E-07	7.79
128	2.40E-07	4.02	6.54E-07	3.97	7.51E-10	8.06	2.32E-09	8.05
256	1.49E-08	4.00	4.07E-08	4.00	2.95E-12	8.04	9.21E-12	8.03
SSPRK(10,4)($\theta = 0.75$): $\nu = 0.01$								
16	1.29E-03	-	2.82E-03	-	5.84E-03	-	1.87E-02	-
32	1.05E-04	3.62	2.94E-04	3.25	3.99E-05	7.19	1.27E-04	7.19
64	7.73E-06	3.76	2.20E-05	3.74	1.87E-07	7.73	5.77E-07	7.78
128	5.07E-07	3.92	1.45E-06	3.92	7.65E-10	8.04	2.36E-09	8.04
256	3.21E-08	3.98	9.22E-08	3.97	3.07E-12	8.01	9.58E-12	8.00

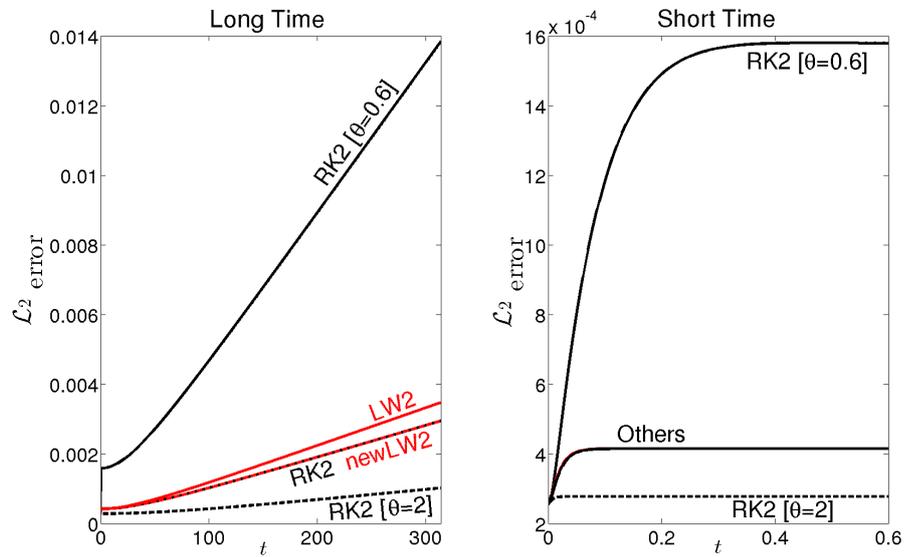
Table 3.1: \mathcal{L}^2 and \mathcal{L}^∞ errors and order of accuracy for fourth-order DG solutions to the linear advection equation at time $T = 1$

Time history

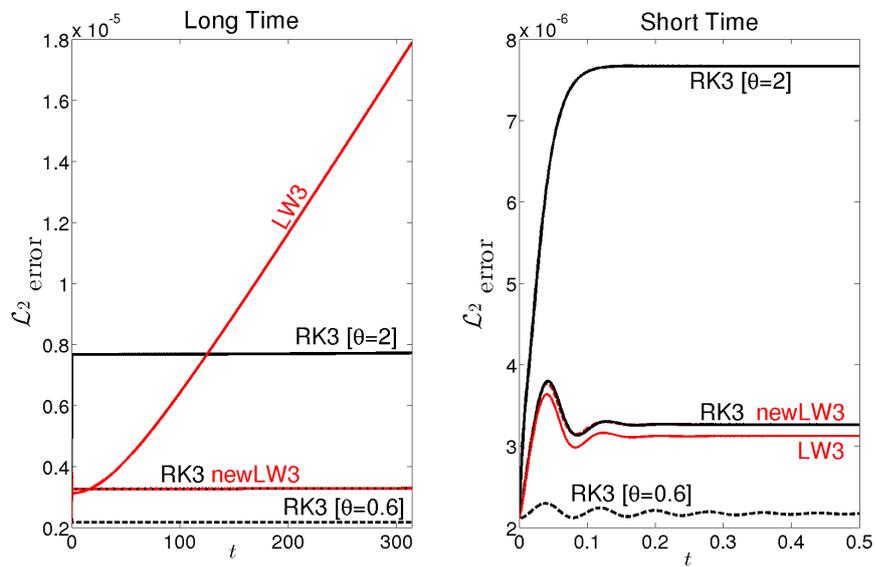
We also investigate the behaviour of the solutions over time. Figures (3.5.1a) and (3.5.1b) show, for $p = 1$ and $p = 2$ respectively, the evolving \mathcal{L}^2 -errors of $(p + 1)^{\text{th}}$ -order RK-DG schemes with three values of $\theta : 0.6, 1, 2$. For comparison, we include the appropriate order of original and new (superconvergent) LW-DG schemes described in [46] and [31] respectively. Over short time, some fluctuations are seen in the errors. These are smaller in magnitude, but prolonged in duration, for the upwind-biased solutions compared to the upwind or the LW schemes. The original LW-DG errors, which does not benefit from superconvergence, can be seen to grow with time while, in general, the other methods, which are superconvergent, do not grow with time. This has significant and clear implications for accurate wave propagation over long time periods. The experiment with $p = 1$ does not see the schemes become approximately constant but we speculate that we would see this over longer time. A fuller investigation of these time regimes is left to further work.

We emphasise the favourable results obtained by the RK-DG scheme when θ is chosen sensitively with respect to the parity of the polynomial degree p ($\theta > 1$ when p is odd; $\theta < 1$ when p is even). The solutions for RK2-DG and RK(10,4)-DG with $\theta = 2$ and the solution for RK3-DG with $\theta = 0.6$ sit visibly below the other schemes over all time regimes.

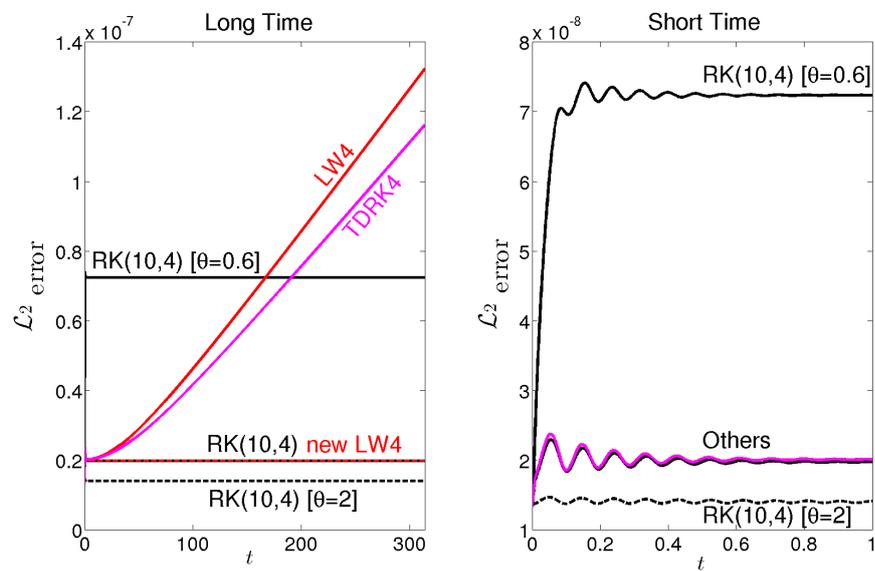
In Figure (3.5.1c), which shows fourth-order methods with $p = 3$, we include the results for the original and our new TDRK4-DG schemes. Over all time regimes, the upwind MoL and the LW and TDRK schemes that use LDG offer practically identical results. The lack of long-term accurate wave propagation properties arising from superconvergence is evident in the original TDRK4 method, which grows with time. Note that even for a poor choice of θ (e.g. $\theta = 0.6$ when $p = 3$), over a long enough time period, the method outperforms those that do not benefit from superconvergence.



(a) Time history of errors for $p = 1$. Long time $t \in [0, 100\pi]$ (left); Short time $t \in [0, 0.6]$ (right).



(b) Time history of errors for $p = 2$. Long time $t \in [0, 100\pi]$ (left); Short time $t \in [0, 0.5]$ (right).



(c) Time history of errors for $p = 3$. Long time $t \in [0, 100\pi]$ (left); Short time $t \in [0, 1]$ (right).

Figure 3.5.1: Time history of \mathcal{L}^2 -errors of DG solutions to the linear advection equation

3.5.2 Linear advection with discontinuous coefficient

Consider the linear advection equation with discontinuous coefficient

$$u_t + cu_x = 0, \quad x \in [0, 2\pi],$$

where the coefficient c is given by

$$c = \begin{cases} 1, & \text{if } x \in [0, 2\pi] \setminus [\frac{\pi}{2}, \frac{3\pi}{2}] \\ \frac{1}{2}, & \text{if } x \in [\frac{\pi}{2}, \frac{3\pi}{2}], \end{cases}$$

and where the initial condition is

$$u_0(x) = \begin{cases} 2 + \cos(2x), & \text{if } x \in [0, 2\pi] \setminus [\frac{\pi}{2}, \frac{3\pi}{2}] \\ 4 - 2\cos(x), & \text{if } x \in [\frac{\pi}{2}, \frac{3\pi}{2}]. \end{cases}$$

In Figure 3.5.2 and Table 3.2, we give results at final time $T = 157 \approx 50\pi$ for (from top to bottom) our new and the original TDRK4 schemes and the RK(10,4)-DG scheme with upwind and upwind-biased ($\theta = 1.5$) fluxes. Note that here, as opposed to the previous example, there is a marked difference between the superconvergent methods; our new TDRK4-DG scheme and the RK-DG scheme with $\theta = 1.5$ clearly outperform the upwind RK-DG scheme. A full investigation of the maximal CFL number (which is responsible for the eventual suboptimal order as the mesh is refined) and the computational efficiency of each method would be an interesting topic for further study.

The errors and orders displayed in Table 3.2 are computed away from the two (stationary) discontinuities in order to see the superconvergent orders of accuracy. Details of what proportion of the domain need be omitted may be procured from the seminal monograph of David Gottlieb and Steven Orszag ([29]). It is important to note that we apply the symmetric SIAC filter. This results in an approximation on the order of 1 around the discontinuity since the SIAC

filter assumes the solution to be analytic. It is possible to obtain a $\mathcal{O}(h^{2p+1})$ solution by employing the one-sided filter (although this does not necessarily give a better approximation away from the discontinuities). An investigation of one-sided filters in this context is left to further work. It is also worth noting that the grid is aligned such that the discontinuities, which are stationary, lie on element boundaries. Finally, we speculate that the undesirable oscillations in the SIAC filtered errors on the finest mesh are a result of numerical round-off error.

mesh	\mathcal{P}^3 : Pre-processed				\mathcal{P}^3 : Post-processed			
	\mathcal{L}^2 error	order	\mathcal{L}^∞ error	order	\mathcal{L}^2 error	order	\mathcal{L}^∞ error	order
Modified TDRK4: $\nu = 0.01$								
16	4.05E-03	-	9.57E-03	-	1.10E-02	-	4.42E-02	-
32	1.17E-04	5.10	5.13E-04	4.22	7.57E-05	7.19	3.05E-04	7.17
64	7.13E-06	4.04	3.01E-05	4.09	3.19E-07	7.59	1.49E-06	7.67
128	4.46E-07	4.00	1.85E-06	4.01	2.05E-09	7.68	9.90E-09	7.33
256	2.78E-08	3.99	1.16E-07	3.99	1.73E-11	6.93	1.58E-10	6.01
Original TDRK4: $\nu = 0.01$								
16	7.16E-03	-	1.52E-02	-	3.64E-02	-	4.98E-02	-
32	3.00E-04	4.57	6.97E-04	4.44	3.92E-04	4.12	6.66E-04	3.92
64	1.83E-05	4.03	4.19E-05	4.05	1.12E-05	5.13	2.52E-05	4.72
128	1.15E-06	3.99	2.61E-06	4.00	6.12E-07	4.19	1.47E-06	4.09
RKDG(10,4) with $\theta = 1$: $\nu = 0.01$								
16	4.03E-03	-	9.53E-03	-	3.17E-02	-	4.42E-02	-
32	1.17E-04	5.09	5.13E-04	4.21	1.94E-04	4.63	3.05E-04	4.52
64	7.13E-06	4.04	3.01E-05	4.09	1.00E-07	7.60	1.49E-06	7.67
128	4.45E-07	3.99	1.85E-06	4.01	5.47E-09	7.51	9.90E-09	7.23
RKDG(10,4) with $\theta = 1.5$: $\nu = 0.01$								
16	4.12E-03	-	1.23E-02	-	1.10E-02	-	4.39E-02	-
32	9.74E-05	5.40	4.77E-04	3.85	7.06E-05	7.29	2.84E-04	7.27
64	5.61E-06	4.11	2.93E-05	3.98	3.40E-07	7.69	1.31E-06	7.76
128	3.43E-07	4.02	1.81E-06	3.99	1.63E-09	7.80	8.82E-09	7.31
256	1.79E-08	4.01	1.13E-07	4.00	1.60E-11	6.72	1.61E-10	5.81

Table 3.2: \mathcal{L}^2 - and \mathcal{L}^∞ -errors and order of accuracy for fourth-order DG solutions to the linear advection equation at time $T = 157$

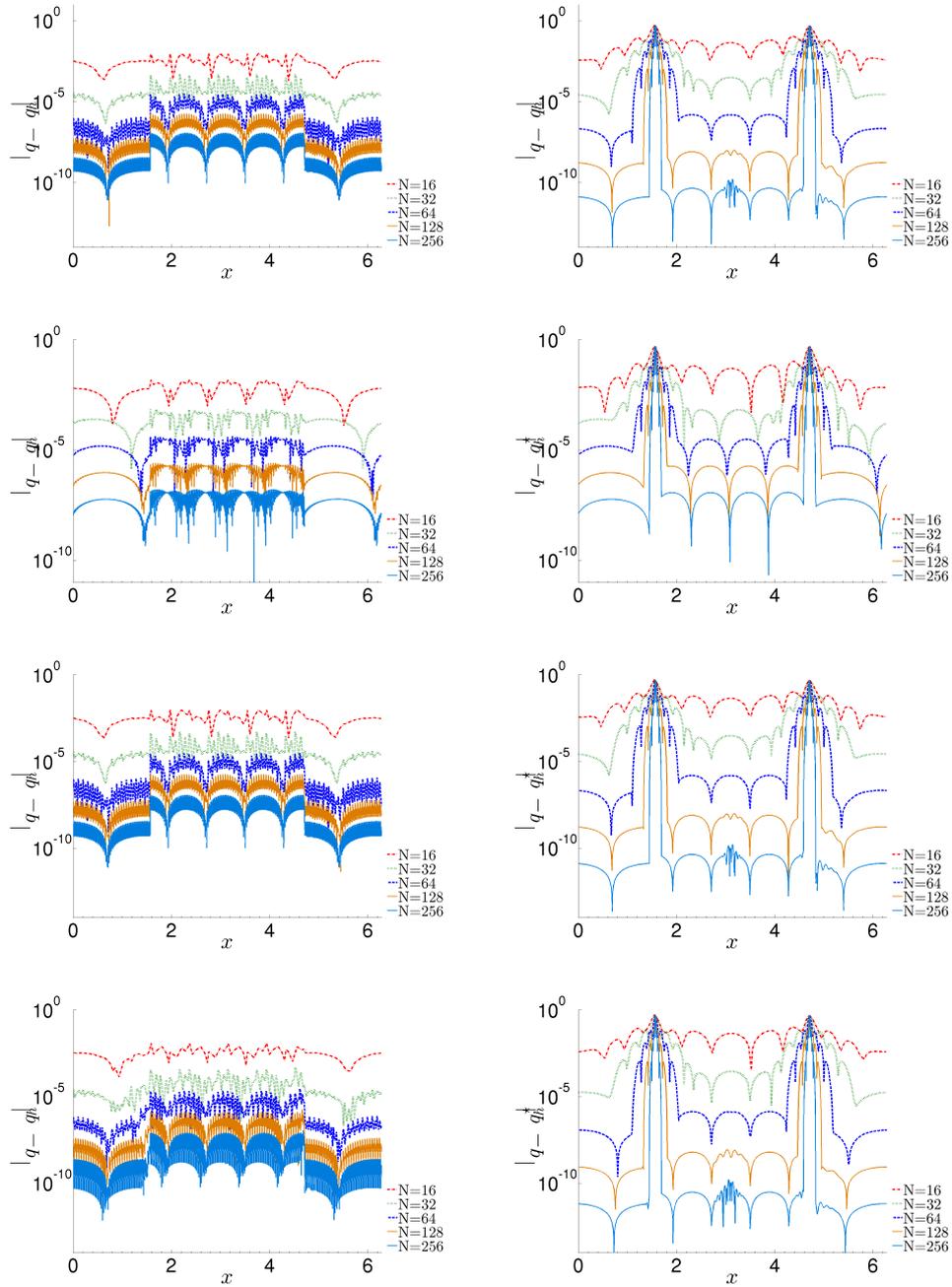


Figure 3.5.2: Linear Advection with discontinuous coefficient. \mathcal{L}^2 -errors (left) and post-processed errors (right) of Modified TDRK4, TDRK4, RK(10,4)-DG ($\theta = 1, 1.5$) (from top to bottom). \mathcal{P}^3 solutions at $T = 157$, all with $\nu = 0.01$.

Dispersion and Dissipation of DG Schemes

“There was a fantastic universal sense that whatever we were doing was right, that we were winning...We had all the momentum; we were riding the crest of a high and beautiful wave...So now, less than five years later, you can go up on a steep hill in Las Vegas and look West, and with the right kind of eyes you can almost see the high water mark — that place where the wave finally broke, and rolled back.”

– Hunter S. Thompson, *Fear and Loathing in Las Vegas*

Numerical methods for solving problems that involve the propagation of waves can give rise to solutions of a qualitatively different nature to the exact solution. For example, the travelling plane wave described by the one-dimensional linear advection equation

$$u_t + cu_x = 0, \quad u(x, 0) = u_0(x) = e^{ikx}, \quad x \in [0, 2\pi], \quad c > 0, \quad (4.0.1)$$

with periodic boundary conditions, where k is the wavenumber and $i = \sqrt{-1}$, displays simple transport of the initial data, which is also a Fourier mode, with constant speed from left to right. The exact solution is known analytically:

$$u(x, t) = u_0(x - ct) = e^{i(kx - \omega t)}, \quad (4.0.2)$$

where ω is the (angular) wave frequency. The relationship between the wavenumber and the frequency is described by the *dispersion relation*; for equation (4.0.1), this is $\omega(k) = ck$. Thus, we see that the wavespeed is $c = \frac{\omega}{k}$; all waves travel at the same speed. It is desirable that the numerical solution mimic this behaviour.

Numerical methods that involve a discretisation of the spatial domain and sampling of the solution at discrete points in time can lead to a complex-valued *numerical* angular frequency, which we denote by $\tilde{\omega} = \tilde{\omega}_{\text{Re}} + i\tilde{\omega}_{\text{Im}}$, where $\tilde{\omega}_{\text{Re}}, \tilde{\omega}_{\text{Im}} \in \mathbb{R}$. For example, consider a first-order parametrised finite difference approximation to the derivative

$$u'(x) \approx \frac{1}{h} [(1 - \theta)u_{j+1} + (2\theta - 1)u_j - \theta u_{j-1}]. \quad (4.0.3)$$

Note that when $\theta = 1$, this is the backward difference; $\theta = 0$ corresponds to the forward difference. Let $\tau = \Delta t$ denote the time-step size. Solving equation (4.0.1) using a uniform mesh, the finite difference operator (4.0.3) to discretise the spatial derivative and Euler's method to advance the solution in time, we get

$$u_j^{n+1} = u_j^n - c \frac{\tau}{h} [(1 - \theta)u_{j+1}^n + (2\theta - 1)u_j^n - \theta u_{j-1}^n]. \quad (4.0.4)$$

By utilising the Taylor expansion of u_{j+1}^n and u_{j-1}^n about $x = x_j$, we see that

$$u_j^{n+1} = u_j^n - c\tau (u_j^n)_x + \frac{c(2\theta - 1)}{2} \tau h (u_j^n)_{xx} - \frac{1}{6} \tau h^2 (u_j^n)_{xxx} + \mathcal{O}(\tau h^4) \quad (4.0.5)$$

or, rather,

$$u_t \approx \frac{u_j^{n+1} - u_j^n}{\tau} = -c (u_j^n)_x + \frac{c(2\theta - 1)}{2} h (u_j^n)_{xx} + \mathcal{O}(h^2). \quad (4.0.6)$$

Thus, we are in effect actually solving the so-called *modified equation*

$$u_t + cu_x = \frac{c(2\theta - 1)}{2} hu_{xx}. \quad (4.0.7)$$

The second-derivative term stabilises the method but it introduces diffusion which has a dissipative effect on the numerical solution. The strength of this effect can be controlled by the flux parameter θ which, by the way, only contributes to the constants attached to the derivative terms of even order. Note that the central difference scheme arising from choosing $\theta = \frac{1}{2}$ is not stable.

To see this dissipative effect, suppose the numerical solution u_h is of the same form as the exact solution (4.0.2) and assume that its wavenumber is the exact wavenumber k . That is, we assume that the numerical solution satisfies

$$u_h(x, t) = \mu e^{i(kx - \tilde{\omega}_{\text{Re}} t)}, \quad (4.0.8)$$

where $\mu = e^{\tilde{\omega}_{\text{Im}} t}$ is the amplitude. It is clear to see that if $\tilde{\omega}_{\text{Re}} \neq \omega$, the numerical solution will suffer from phase errors (dispersion). Similarly, if $\tilde{\omega}_{\text{Im}} < 0$, the resulting smearing of the amplitude is described as dissipation errors.

Figure 4.0.1 shows a travelling square wave approximated by the first-order finite difference method (4.0.4) with $N = 100$ and $T = 32$. As the wave propagates, the numerical solution experiences a decrease in amplitude, a phase lag and smoothing of the peaks. For the finite difference method above, let

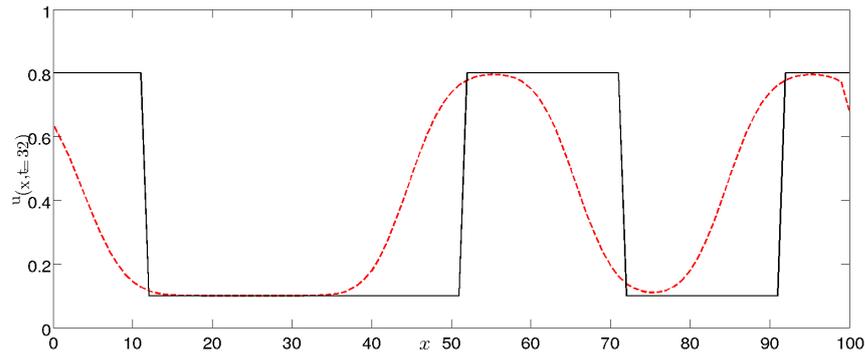


Figure 4.0.1: A propagating square wave (black) approximated by first order finite difference (red) at $t = 32$ with $\theta = 1$

$$u_j^n = e^{i(kjh - \tilde{\omega} n \tau)}$$

in the scheme. Then, by truncating the Taylor series for $e^{-i\tilde{\omega}\tau}$, it is easy to show that the dispersion relation is approximated by

$$\begin{aligned}\omega \approx \tilde{\omega} &= c \frac{\sin(kh)}{h} - i(2\theta - 1) \frac{c(1 - \cos(kh))}{h} \\ &= ck - i \frac{(2\theta - 1)}{2} ck^2 h - \frac{1}{6} ck^3 h^2 + i \frac{(2\theta - 1)}{24} ck^4 h^3 + \frac{1}{120} ck^5 h^4 + \mathcal{O}(k^6 h^5).\end{aligned}\tag{4.0.9}$$

Note that $\text{Im}(\tilde{\omega}) = -(2\theta - 1)c \frac{1 - \cos(kh)}{h} < 0$ increases in magnitude with (small) kh so that the method becomes less accurate. This term represents dissipative errors. When $kh \ll 1$ (that is, when there are a large number of mesh points (or elements) per wavelength), we have $\tilde{\omega} \approx ck = \omega$. On the other hand, if the spatial discretisation is too coarse and $\tilde{\omega}_{\text{Re}}$ does not provide a good approximation to ω , then the wave will suffer substantial spurious damping. Note that the magnitude of the error can be reduced by choosing $\frac{1}{2} < \theta < 1$. The spectral properties of the scheme discussed above are very similar to those for a piecewise constant DG method paired with Forward Euler time-stepping. The problem of how many points per wavelength are required for the error for RK-DG methods to fall within a given tolerance was addressed by Zhong and Shu ([68]).

4.1 Fourier Analysis of Amplification Matrices

In what follows, we analyse for the first time the eigenvalues of the amplification matrix of the upwind-biased DG operator and the TDRK4 schemes, paying particular attention to the effect of the flux function on the ability of the method to accurately propagate a wave. We exclusively consider the linear advection equation (4.0.1) and strongly require a uniform mesh (due to the infinite nature of the Fourier sum; we consider only a single mode), periodic boundary conditions (to avoid a Gibbs-type effect) and a smooth initial condition (so that the solution is analytic; we expand the solution as a power series). The choice of initial condition and basis functions can be crucial in obtaining optimal results. Recent work that demonstrates the importance of

these choices includes [19, 61] and [16].

Consider the local DG solution

$$u_h(x(\xi), t)|_{I_j} = \sum_{\ell=0}^P u_j^{(\ell)}(t) \phi_j^\ell(\xi), \quad \phi_j^\ell(\xi) \in V_h^p,$$

to the linear advection equation (4.0.1) with periodic boundary conditions, a uniform mesh and an initial condition

$$u(x, 0) = u_0(x) = e^{ikx}, \quad (4.1.1)$$

where k is the wavenumber. Note that here, the initial condition $u(x, 0)$ is also a Fourier mode. The spatial derivative of the exact solution (4.0.2) gives $\frac{\partial}{\partial x} u(x, t) = iku(x, t)$ so we expect one of the eigenvalues of the semi-discrete DG operator, which approximates $-cu_x$, to satisfy $\lambda \approx -ick$.

We analyse both the semi-discrete and fully-discrete schemes. Recall that the DG semi-discretisation with upwind-biased flux can be written as the following semi-discrete system of ODEs:

$$\frac{d}{dt} \mathbf{U}_j = \frac{c}{h} [(\theta A_1 + (1 - \theta) A_2) \mathbf{U}_j + \theta B \mathbf{U}_{j-1} + (1 - \theta) C \mathbf{U}_{j+1}], \quad (4.1.2)$$

where A_1, A_2, B and C are $(p+1) \times (p+1)$ matrices and \mathbf{U}_j is the vector of Galerkin coefficients of u_h . In order to inspect the semi-discrete numerical dispersion relation for the DG scheme, we substitute a wave-like solution

$$\mathbf{U}_j(t) = e^{ikx_j} \hat{\mathbf{u}}_k(t) \quad (4.1.3)$$

to rewrite the ODE (4.1.2) in terms of a global coefficient vector $\hat{\mathbf{u}}_k$. This results in a statement about the stage amplification:

$$\frac{d}{dt} \hat{\mathbf{u}}_k = \frac{c}{h} \mathcal{G} \hat{\mathbf{u}}_k, \quad (4.1.4)$$

where

$$\mathcal{G} = \theta A_1 + (1 - \theta)A_2 + \theta B e^{-ik\Delta x} - (1 - \theta)C e^{ik\Delta x} \quad (4.1.5)$$

is known as the *semi-discrete amplification matrix*. If \mathcal{G} is diagonalisable (this is often observed numerically) then it has a full set of eigenvalues, say $\bar{\lambda}_1, \dots, \bar{\lambda}_{p+1}$, and corresponding eigenvectors $\Lambda_1, \dots, \Lambda_{p+1}$. One of the eigenvalues, say $\bar{\lambda}_1$, has physical relevance, approximating the semi-discrete dispersion relation $\omega = -ick$, while the others are associated to spurious modes. Guo et al. [32] showed that, over time, the contribution of the spurious modes to the approximation error is exponentially damped with mesh refinement whilst the dispersion and dissipation errors of the physically relevant eigenvalue grow linearly.

Similarly, suppose the fully-discrete numerical solution at time $t = t_{n+1}$ has the form

$$\mathbf{U}_j^{n+1} = \sum_{m=-s}^s A_m \mathbf{U}_{j+m}^n, \quad (4.1.6)$$

where s is the size of the stencil and A_m are $(p+1) \times (p+1)$ matrices with entries that are polynomials in the CFL number $\nu = \frac{\Delta x}{\tau} = \frac{h}{\tau}$, and transform the Galerkin coefficient vectors to Fourier space via the assumption

$$\mathbf{U}_j(t^n) = e^{ikx_j - i\tilde{\omega}t} \hat{\mathbf{u}}_{k,\omega} \quad (4.1.7)$$

to obtain a *fully-discrete amplification matrix*

$$\mathcal{G} = \sum_{m=-s}^s A_m e^{imkh} \quad (4.1.8)$$

that satisfies

$$e^{-i\tilde{\omega}\tau} \hat{\mathbf{u}}_{k,\omega} = \mathcal{G} \hat{\mathbf{u}}_{k,\omega}. \quad (4.1.9)$$

The fully discrete amplification matrix \mathcal{G} contains information about how properties of the spatial discretisation are propagated forwards in time. The physically relevant eigenvalue, say λ_1 , is associated to a frequency whose real

part approximates the physical quantity k . The remaining modes, as with the semi-discrete case, are spurious.

4.1.1 Non-dimensionalised presentation of eigenvalues

He Yang et al. [60] considered refinement of the mesh relative to the wavenumber. Consider an eigenvalue λ of the amplification matrix \mathcal{G} with argument $\alpha = \arctan \frac{\text{Im}(\lambda)}{\text{Re}(\lambda)}$ and modulus $r = e^a = \sqrt{\text{Re}(\lambda)^2 + \text{Im}(\lambda)^2}$ so that $a = \frac{1}{2}(\text{Re}(\lambda)^2 + \text{Im}(\lambda)^2)$. Accordingly,

$$\lambda = e^a e^{i\alpha} = e^{\frac{1}{2}(\text{Re}(\lambda)^2 + \text{Im}(\lambda)^2)} e^{i \arctan \frac{\text{Im}(\lambda)}{\text{Re}(\lambda)}}.$$

Recalling equation (4.1.9), we also have

$$e^{-i\tilde{\omega}\tau} = e^{\omega_{Im}\tau} e^{i\omega_{Re}\tau} = \lambda$$

so, comparing the two preceding equations,

$$\begin{aligned} \omega_{Im}\tau &= \frac{1}{2}(\text{Re}(\lambda)^2 + \text{Im}(\lambda)^2) \\ -i\omega_{Re}\tau &= i \arctan \frac{\text{Im}(\lambda)}{\text{Re}(\lambda)}. \end{aligned}$$

Now perform the change of variables

$$\begin{aligned} K &= kh \\ \Omega &= \Omega_{Re} + i\Omega_{Im} = \omega_{Re}h + i\omega_{Im}h \end{aligned}$$

so that

$$\begin{aligned} \omega_{Im}\tau &= \nu h \omega_{Im} = \nu \Omega_{Im} \\ \omega_{Re}\tau &= \nu h \omega_{Re} = \nu \Omega_{Re}. \end{aligned}$$

Finally, we define the non-dimensionalised quantities

$$\Omega_{\text{Re}} = \omega_{\text{Re}} h = -\frac{h}{\tau} \arctan\left(\frac{\text{Im}(\lambda_1)}{\text{Re}(\lambda_1)}\right) \quad (4.1.10a)$$

$$\Omega_{\text{Im}} = \omega_{\text{Im}} h = \frac{1}{2} \frac{h}{\tau} \ln\left((\text{Re}(\lambda_1))^2 + (\text{Im}(\lambda_1))^2\right) \quad (4.1.10b)$$

in the asymptotic regime $K = kh \ll 1$, where λ_1 is the physically relevant eigenvalue.

Following [60], we expand equations (4.1.10a) and (4.1.10b) as Taylor series and present the error in the numerical dispersion relation:

$$\begin{cases} \Omega_{\text{Re}} = K + \text{dispersion error} = \mathcal{O}(K^{N_1}) \\ \Omega_{\text{Im}} = \text{dissipation error} = \mathcal{O}(K^{N_2}). \end{cases} \quad (4.1.11)$$

Note that dispersion and dissipation errors $K - \Omega_{\text{Re}}$ and Ω_{Im} on the order of K^{r+1} equates to corresponding errors $k - \omega_{\text{Re}}$ and ω_{Im} on the order of h^r (when $k = \mathcal{O}(1)$ and $h \ll 1$). We find that presentation of the errors as in equation (4.1.11) more readily facilitates comparison between different methods than directly presenting the Taylor expansions of the eigenvalues λ .

Suppose we make two consecutive approximations by changing the element size h (and thus K) while fixing the wavenumber k . Noting that $|u_h| = e^{\Omega_{\text{Im}} \frac{T}{h}}$, where T is the final time, we get

$$\Omega_{\text{Im}, K_1} = \frac{h_1}{T} \ln |u_{h_1}| \quad \text{and} \quad \Omega_{\text{Im}, K_2} = \frac{h_2}{T} \ln |u_{h_2}|.$$

The order of the dissipation error is then given by

$$N_2 = \frac{\ln\left(\frac{\Omega_{\text{Im}, K_1}}{\Omega_{\text{Im}, K_2}}\right)}{\ln\left(\frac{K_1}{K_2}\right)} = 1 + \frac{\ln\left(\frac{\ln |u_{h_1}|}{\ln |u_{h_2}|}\right)}{\ln\left(\frac{h_1}{h_2}\right)}. \quad (4.1.12)$$

4.2 Numerical Process

Computation and the meaningful presentation of the eigenstructure of these amplification matrices is all but impossible to perform by hand. The process involves finding roots of higher order polynomials, matrix inversion, numerical interpolation, numerical simplification of vastly complicated expressions in multiple variables and series expansions of long rational forms involving fractional exponents. In order to make these tasks more manageable, we employ a symbolic computation computer program: our choice is Mathematica. Even still, the process can rarely be left to run in its entirety without input from the user. Expressions, as they arise, often require interpretation before embarking on the next stage and, occasionally, numerical identities must be formed, or manipulations performed, manually. Even with numerical aids, the computations quickly become prohibitively complicated and slow. Increased computing power is not, in itself, a solution to these problems; the manual element also becomes increasingly difficult and there are several numerical bottlenecks such as root-finding for higher-order polynomials. Moreover, a tidy closed form is usually desired for the final truncated series expressions and, as the basis polynomial degree p increases, it quickly becomes the case that certain quantities must be evaluated numerically as opposed to symbolically. The difficulties described above become a serious problem as early as $p = 3$.

We outline the process undertaken in Mathematica to obtain the results presented in the next section.

- Construct the component matrices for the amplification matrix G . To obtain the eigenvalues, the inbuilt Mathematica command suffices for $p \leq 1$; for larger p , it may be necessary to manually solve the characteristic equation of G

$$|G - \lambda I| = 0. \quad (4.2.1)$$

For $p = 3$, when the characteristic equation of G is of order 4, we use Ferrari's method.

- This results in large expressions in h (and, for the upwind-biased flux, θ) featuring fractional exponents. In order to obtain a more insightful form of the eigenvalues, the next step is to expand the expressions in a Maclaurin series in kh .
- For higher order polynomial degrees, the Maclaurin expansion can be evaluated for several values of θ and the final result interpolated to write the coefficients as rational functions in θ . The interpolation may involve “guessing” the denominator. This is often a quantity that arises in other analysis of the method, for example $(2\theta - 1)$.
- The Maclaurin expansion or the solution to the characteristic equation of G may include large numerical expressions with fractional components that are not easily simplified. Dealing with these terms often includes identifying a value which is equal to zero. For example, when $p = 2$, the result relies on identifying the equality

$$\sqrt[3]{\frac{1}{2}(5\sqrt{17}) + \frac{13}{2}} - 4 \left(\sqrt[3]{\frac{5\sqrt{17}}{2} + \frac{13}{2}} \right)^{-1} - 1 = 0. \quad (4.2.2)$$

- To find the eigenvectors of G , observe that

$$\hat{\mathbf{u}}_k = \sum_{i=1}^{p+1} C_i e^{\bar{\lambda}_i t} \Lambda_i. \quad (4.2.3)$$

We solve the systems

$$G\Lambda = \lambda\Lambda \quad (4.2.4)$$

and

$$\sum_{i=1}^{p+1} C_i \Lambda_i = \hat{\mathbf{u}}_k(0), \quad (4.2.5)$$

where the second equation follows from equations (4.2.3) and (4.1.7).

In the next section, we treat the DG discretisation alone. Here, the focus is on the upwind-biased flux and how the flux parameter θ appears in the coefficients of Taylor series expansions of the eigenvalues of the semi-discrete amplification matrix \mathcal{G} . We analyse the fully-discrete schemes in §4.5.

4.3 Semi-discrete DG Scheme: Results

Using Mathematica to computationally perform an asymptotic analysis on $kh = 0$, we obtain the following sets of eigenvalues $\bar{\lambda}_j$ of the amplification matrix \mathcal{G} :

$$\begin{aligned}
 p = 0 : \quad & \bar{\lambda}_1 = -ik - \frac{1}{2}(2\theta - 1)k^2h + \mathcal{O}(h^2); \\
 p = 1 : \quad & \left\{ \begin{aligned} \bar{\lambda}_1 &= -ik - \frac{1}{72} \frac{1}{2\theta-1} k^4 h^3 - \frac{i}{270} \frac{1+6\theta-6\theta^2}{(1-2\theta)^2} k^5 h^4 + \mathcal{O}(h^5), \\ \bar{\lambda}_2 &= -\frac{6(2\theta-1)}{h} + 3ik + (2\theta - 1)k^2h + \mathcal{O}(h^2); \end{aligned} \right. \\
 p = 2 : \quad & \left\{ \begin{aligned} \bar{\lambda}_1 &= -ik - \frac{2\theta-1}{7200} k^6 h^5 + \frac{i}{3000} [\theta^2 - \theta + \frac{1}{14}] k^7 h^6 + \mathcal{O}(h^7), \\ \bar{\lambda}_{2,3} &= -\frac{3(2\theta-1)}{h} \pm i\sqrt{51 + 36\theta - 36\theta^2}k + \mathcal{O}(h); \end{aligned} \right. \\
 p = 3 : \quad & \bar{\lambda}_1 = -ik - \frac{3.125 \times 10^{-4}}{441(2\theta - 1)} k^8 h^7 \\
 & - \frac{1.25 \times 10^{-3}}{27783} \frac{19 - 48\theta + 28\theta^2}{(1 - 2\theta)^2} k^9 h^8 + \mathcal{O}(h^9).
 \end{aligned}$$

For each value of p , the physically-relevant eigenvalue $\bar{\lambda}_1$ approximates $-ik$ with dispersion error on the order of h^{2p+1} and dissipation error on the order of h^{2p+2} . This is consistent with the previous findings of [9, 52, 33, 60, 68] and [32].

Contributions to understanding the role of the flux function

We comment for the first time on how changing the flux parameter alters the constants attached to the superconvergent error terms in expansions such as those above. The coefficient of the leading order real term on the order of h^{2p+1} of the physically relevant eigenvalues $\bar{\lambda}_1$ is negative for all values of $\theta > \frac{1}{2}$. For even polynomial order p , the coefficient of the leading order term vanishes in the limit $\theta \rightarrow \frac{1}{2}$. For odd polynomial order p , due to the factor $(2\theta - 1)^{-1}$, one can reduce the magnitude of the both the dispersion and dissipation errors by taking a larger value of θ . On the other hand, one should avoid taking θ close to $\frac{1}{2}$ when p is odd since the coefficients grow without bound as $\theta \rightarrow \frac{1}{2}$. The importance of the magnitude of these constants should not be undervalued: they have a direct impact on the approximation errors and, crucially, on the accuracy with which waves are propagated.

Note that the order of the dissipation error can (theoretically) be increased by choosing θ carefully. For example, when $p = 1$ and $\theta = \frac{1}{6}(3 + \sqrt{15}) \approx 1.145$ or when $p = 2$ and $\theta = \frac{1}{14}(7 + \sqrt{35}) \approx 0.923$, the leading order dissipation term vanishes. In the $p = 1$ case, the physically relevant eigenvalue becomes

$$\lambda_1 = -ik - \frac{1}{24\sqrt{15}}k^4h^3 - \frac{1}{360\sqrt{15}}k^6h^5 - \frac{13i}{75600}k^7h^6. \quad (4.3.1)$$

However, since the leading order error term is dispersive, such choices do not improve the overall error in comparison to more extreme choices of θ .

Remark. The traditional choice of flux for linear advection with windspeed $c > 0$ is the monotone purely upwind flux function $\hat{u} = u^-$. This is an intuitive selection; initial data moves from left to right so we set the solution value $u_j(x_{j-\frac{1}{2}})$ at the left boundary point to be the value $u_{j-1}(x_{j-\frac{1}{2}})$ from the neighbouring cell, allowing information to propagate only in the desired direction. Our previous remarks suggest that a wave is propagated by DG more accurately if we also include information from the right of cell boundaries. With regards to the design

of flux functions, this is a counter-intuitive situation. We speculate that the increased accuracy of the upwind-biased flux may be explained by the dependence on the numerical flux of the constant attached to the higher order derivative terms in the modified equation (for a finite difference method, recall equation (4.0.7)). Moreover, when $\theta > 1$ (a choice we advocate for odd polynomial degree p), the upwind-biased flux no longer satisfies the monotonicity property.

The remaining eigenvalues are non-physically relevant but have negative real part on the order of $\frac{1}{h}$. Thus the corresponding eigenvectors in the solution are damped over time. This occurs more slowly for lower values of θ .

For the case $p = 3$, the findings are consistent with the other cases but the algebra involved in the computation becomes prohibitively substantial and the need to evaluate components numerically makes it particularly difficult to obtain tidy expressions for the coefficients.

Eigenvectors for the piecewise linear approximation

We perform a short study of the effect of the flux parameter θ on the eigenvectors. Recall that one eigenvector, say Λ_1 , a constant multiple of which approximates $\hat{u}(0)$ with order $p + 2$ at Radau points and order $2p + 1$ at the downwind ends per the original result in [32], is associated with the physically relevant mode while the others are associated with spurious modes and are on the order of $p + 2$ at Radau points. We discuss only the case $p = 1$ since it is not clear how to directly compare in a closed form the outcomes of different choices of θ .

While the eigenvalues are independent of the choice of basis functions, one must make an appropriate choice of interpolating initial condition and basis functions in order to extract superconvergent accuracy in the eigenvectors. If one uses a Lagrange-Radau basis on roots of $R_{p+1}^+(\xi)$, the appropriate choice when $\theta = 1$,

in the case $p = 1$ the physically relevant eigenvector satisfies

$$C_1\Lambda_1 - \hat{\mathbf{u}}(0) = \begin{bmatrix} -\frac{1-\theta}{18(1-2\theta)}k^2h^2 + \frac{11-11\theta+2\theta^2}{324(1-2\theta)^2}ik^3h^3 + \mathcal{O}(h^4) \\ \frac{1-\theta}{6(1-2\theta)}k^2h^2 + \frac{1-25\theta+22\theta^2}{108(1-2\theta)^2}ik^3h^3 + \mathcal{O}(h^4) \end{bmatrix}, \quad (4.3.2)$$

while similarly the non-physically relevant eigenvector satisfies

$$C_2\Lambda_2 = \begin{bmatrix} \frac{1-\theta}{18(1-2\theta)}k^2h^2 - \frac{11-11\theta+2\theta^2}{324(1-2\theta)^2}ik^3h^3 + \mathcal{O}(h^4) \\ -\frac{1-\theta}{6(1-2\theta)}k^2h^2 - \frac{1-25\theta+22\theta^2}{108(1-2\theta)^2}ik^3h^3 + \mathcal{O}(h^4) \end{bmatrix}. \quad (4.3.3)$$

The leading order terms vanish only when $\theta = 1$ when the interpolation points of the initial condition coincide with the superconvergent points of the scheme.

Guo et al.'s [32] analysis of the DG scheme with upwind flux ($\theta = 1$) gave

$$\begin{aligned} C_1\Lambda_1 - \hat{\mathbf{u}}(0) &= \begin{bmatrix} \frac{i}{162}(kh)^3 - \frac{5}{3888}(kh)^4 + \mathcal{O}((kh)^5) \\ -\frac{i}{54}(kh)^3 + \frac{7}{432}(kh)^4 + \mathcal{O}((kh)^5) \end{bmatrix} = -C_2\Lambda_2 \quad (4.3.4) \\ &\approx \begin{bmatrix} 0.00617i(kh)^3 - 0.00129(kh)^4 \\ -0.0185i(kh)^3 + 0.0162(kh)^4 \end{bmatrix}. \end{aligned}$$

A poor choice of θ (here $\theta < 1$ since the polynomial degree p is odd) results in expressions on the same order as those for the upwind flux (4.3.4) but with larger coefficients. For example, when $\theta = \frac{3}{5}$, choosing the basis functions to be Lagrangian polynomials based on the corresponding superconvergent points (roots of R^* as discussed in Chapter 2)

$$x = x_j - \frac{5}{6} \left(1 \mp \frac{2\sqrt{7}}{5} \right) h,$$

we obtain

$$\begin{aligned}
C_1\Lambda_1 - \hat{\mathbf{u}}(0) &= \begin{bmatrix} \frac{61(2\sqrt{7}-5)}{162}i(kh)^3 - \frac{11450\sqrt{7}-29357}{3888}(kh)^4 + \mathcal{O}((kh)^5) \\ -\frac{183\sqrt{7}}{54(42-15\sqrt{7})}i(kh)^3 + \frac{7(2190\sqrt{7}-5231)}{432(5\sqrt{7}-14)^2}(kh)^4 + \mathcal{O}((kh)^5) \end{bmatrix} \\
&\approx \begin{bmatrix} 0.110i(kh)^3 - 0.241(kh)^4 \\ -3.88i(kh)^3 + 15.3(kh)^4 \end{bmatrix}.
\end{aligned} \tag{4.3.5}$$

Note that the constants attached to the error terms are two orders of magnitude greater than those in the $\theta = 1$ case (4.3.4). On the other hand, the constants can be reduced by choosing $\theta > 1$. Moreover, based on the observations from the analysis of the eigenvalues in §4.3, we might choose $\theta = \frac{1}{6}(3 + \sqrt{15})$ when the superconvergent points are

$$\xi = \frac{\sqrt{6} \mp 1}{\sqrt{15}} \in (-1, 1).$$

In this case, we gain an extra order of accuracy in the eigenvectors as well as reducing the magnitude of the constants:

$$\begin{aligned}
C_1\Lambda_1 - \hat{\mathbf{u}}(0) &= \begin{bmatrix} \frac{6-\sqrt{6}}{3600\sqrt{6}}(kh)^4 + \frac{108\sqrt{15}-79\sqrt{10}}{180000\sqrt{6}}i(kh)^5 + \mathcal{O}((kh)^6) \\ -\frac{12-7\sqrt{6}}{720(54-19\sqrt{6})}(kh)^4 + \frac{283\sqrt{10}-126\sqrt{15}}{36000(54-19\sqrt{6})}i(kh)^5 + \mathcal{O}((kh)^6) \end{bmatrix} \\
&\approx \begin{bmatrix} 0.000403(kh)^4 + 0.000382i(kh)^5 \\ -0.000958(kh)^4 - 0.00152i(kh)^5 \end{bmatrix}.
\end{aligned} \tag{4.3.6}$$

4.4 Fully-discrete DG Schemes: Formulation

Whilst much of the dispersion analysis of DG methods has been performed on the semi-discrete scheme alone, we must consider the effect of the coupling between the spatial and temporal discretisations to get a fuller picture of the wave propagation abilities. For the MoL, where the discretisations are completely decoupled, much of the information about the dispersion properties

of the fully discrete-scheme is contained in the semi-discrete amplification matrix \mathcal{G} discussed in §4.3. For much of the rest of this chapter, we discuss coupled schemes but first we summarise the role of the RK time discretisation.

4.4.1 Fully decoupled Runge-Kutta methods

As we suggested in Chapter 1, a RK scheme can be written in the form

$$\mathbf{U}_j^{n+1} = \left(I + \tau \mathcal{D} + \cdots + \frac{\tau^p}{p!} \mathcal{D}^p \right) \mathbf{U}_j^n, \quad (4.4.1)$$

where \mathcal{D} is the DG spatial discretisation operator comprising matrix multiplications and \mathcal{D}^p refers to p repeated applications of this operator. If we make the same assumption (4.1.7) as in the semi-discrete case, we see that the fully discrete amplification matrix can be written as a linear combination of the one for the semi-discrete scheme:

$$\hat{\mathbf{U}}_k^{n+1} = \mathcal{G} \hat{\mathbf{U}}_k^n,$$

where

$$\mathcal{G} = \left(I + \tau \mathcal{G} + \cdots + \frac{\tau^p}{p!} \mathcal{G}^p \right). \quad (4.4.2)$$

Hence the eigenvalues for the fully discretised solution resulting from a RK-DG discretisation are just linear combinations of the eigenvalues for the semi-discrete DG scheme. Before presenting results on the eigenvalues of the TDRK4 methods, we perform some necessary calculations for our analysis. We seek to write the schemes in a fully resolved form in terms of Galerkin coefficients of the solution u_h .

4.4.2 Fully resolved forms: DG-TDRK4 with direct differentiation

We consider a general stage equation

$$\int_{\mathcal{I}_j} u_h v \, dx = \int_{\mathcal{I}_j} u_h^n v \, dx + \tau \int_{\mathcal{I}_j} \tilde{f} v_x \, dx - \tau \widehat{f}_{j+\frac{1}{2}}^- v_{j+\frac{1}{2}}^- + \tau \widehat{f}_{j-\frac{1}{2}}^+ v_{j-\frac{1}{2}}^+, \quad (4.4.3)$$

where

$$\tilde{f} = \alpha u_h^n + \alpha^* u_h^* + \beta \tau g_h^n + \beta^* \tau g_h^* \quad (4.4.4)$$

is the modified flux with $g_h = \frac{\partial}{\partial x} u_h$ and where the numerical flux is

$$\widehat{f}_{j+\frac{1}{2}} = \frac{1}{2} \left(\tilde{f}_{j+\frac{1}{2}}^+ + \tilde{f}_{j+\frac{1}{2}}^- - (u_h)_{j+\frac{1}{2}}^+(t_n) + (u_h)_{j+\frac{1}{2}}^-(t_n) \right). \quad (4.4.5)$$

The solution u_h on the right-hand side of equation (4.4.3) can be either the solution at the intermediate stage or the full update. The coefficients in equation (4.4.4) take different values depending on the stage. After scaling, the stage equation (4.4.3) representing the TDRK4 method can be written in terms of the discretisation operators as

$$\mathbf{U}_j(t) = \mathbf{U}_j(t_n) + \frac{1}{2} \frac{\tau}{h} (\mathcal{D}^- + \mathcal{D}^+) \tilde{\mathbf{F}}_j + \frac{1}{2} \frac{\tau}{h} (\mathcal{D}^- - \mathcal{D}^+) \mathbf{U}_j(t_n), \quad (4.4.6)$$

where \mathcal{D}^- and \mathcal{D}^+ are defined by equation (1.3.33) and where $\tilde{\mathbf{F}}_j$ is the vector of Galerkin coefficients of the modified flux (4.4.4).

Some of the numerical manipulations in the analysis that follows are eased by considering the mathematically equivalent scheme arising from an extra integration by parts. We are motivated to do so partially by the analogous alternative LW-DG formulation analysed in [60]. We consider instead the stage

equation

$$\begin{aligned} \int_{\mathcal{I}_j} u_h v \, dx &= \int_{\mathcal{I}_j} u_h^n v \, dx - \tau \int_{\mathcal{I}_j} \tilde{f}_x v \, dx \\ &\quad - \tau \left(\widehat{f} - \tilde{f}^- \right)_{j+\frac{1}{2}} v_{j+\frac{1}{2}}^- + \tau \left(\widehat{f} - \tilde{f}^+ \right)_{j-\frac{1}{2}} v_{j-\frac{1}{2}}^+ \end{aligned} \quad (4.4.7)$$

obtained by integrating by parts the stiffness integral in equation (4.4.3). Using the notation for discretisation operators in equations (1.3.37) and (1.3.38), this alternative formulation can be written as a scheme for Galerkin coefficients:

$$\mathbf{U}_j(t) = \mathbf{U}_j(t_n) + \frac{1}{2} \frac{\tau}{h} (\bar{\mathcal{D}}^- + \bar{\mathcal{D}}^+) \tilde{\mathbf{F}}_j + \frac{1}{2} \frac{\tau}{h} (\mathcal{D}^- - \mathcal{D}^+) \mathbf{U}_j(t_n), \quad (4.4.8)$$

We have a local expression for the vector of Galerkin coefficients of the derivative g_h in terms of that for the solution u_h :

$$\mathbf{G}_j = \frac{1}{h} S^T \mathbf{U}_j, \quad (4.4.9)$$

where S^T is the transpose of the stiffness matrix S defined in equation (1.3.35) and represents differentiation. Using this expression (4.4.9), recast the vector of coefficients for the modified flux \tilde{f} in terms of those for the solution u_h :

$$\tilde{\mathbf{F}}_j = (\alpha I + \beta \frac{\tau}{h} S^T) \mathbf{U}_j(t_n) + \left(\alpha^* I + \beta^* \frac{\tau}{h} S^T \right) \mathbf{U}_j^*, \quad (4.4.10)$$

where I is the $(p+1)$ -identity matrix. Next, we rework the stage equation (4.4.6) using the expression (4.4.10):

$$\begin{aligned} \mathbf{U}_j(t) &= \left[I + \frac{1}{2} \frac{\tau}{h} ((\alpha+1)\bar{A}_1 + (\alpha-1)\bar{A}_2) + \frac{\beta}{2} \frac{\tau^2}{h^2} (\bar{A}_1 + \bar{A}_2) S^T \right] \mathbf{U}_j^n \\ &\quad + \left[\frac{1}{2} (\alpha+1) \frac{\tau}{h} B + \frac{\beta}{2} \frac{\tau^2}{h^2} B S^T \right] \mathbf{U}_{j-1}^n \\ &\quad + \left[\frac{1}{2} (1-\alpha) \frac{\tau}{h} C - \frac{\beta}{2} \frac{\tau^2}{h^2} C S^T \right] \mathbf{U}_{j+1}^n \\ &\quad + \left[\frac{\alpha^*}{2} \frac{\tau}{h} (\bar{A}_1 + \bar{A}_2) + \frac{\beta^*}{2} \frac{\tau^2}{h^2} (\bar{A}_1 + \bar{A}_2) S^T \right] \mathbf{U}_j^* \\ &\quad + \left[\frac{\alpha^*}{2} \frac{\tau}{h} B + \frac{\beta^*}{2} \frac{\tau^2}{h^2} B S^T \right] \mathbf{U}_{j-1}^* - \left[\frac{\alpha^*}{2} \frac{\tau}{h} C + \frac{\beta^*}{2} \frac{\tau^2}{h^2} C S^T \right] \mathbf{U}_{j+1}^*. \end{aligned} \quad (4.4.11)$$

The coefficients $\alpha, \alpha^*, \beta, \beta^*$ take different values depending on whether the left-hand side is the intermediate stage $\mathbf{U}_j(t) = \mathbf{U}_j^*$ or the full update $\mathbf{U}_j(t) = \mathbf{U}_j^{n+1}$. Note that since the TDRK4 method is explicit, $\alpha^* = 0 = \beta^*$ for the intermediate stage. In this case, equation (4.4.11) is of the form

$$\mathbf{U}_j^* = \sum_{m=-1}^1 T_m \mathbf{U}_{j+m}^n, \quad (4.4.12)$$

where T_m are $(p+1) \times (p+1)$ matrices with entries on the order of $(\frac{\tau}{h})^2$. The full update in turn includes information from two neighbouring cells on either side:

$$\mathbf{U}_j^{n+1} = \sum_{m=-2}^2 W_m \mathbf{U}_{i+m}^{n+1} \quad (4.4.13)$$

where W_m are $(p+1) \times (p+1)$ matrices with entries on the order of $(\frac{\tau}{h})^4$.

Denote by $\nu = \frac{\tau}{h}$. For $p = 1$, the fully-discrete scheme matrices are given by

$$\begin{aligned} W_0 &= \begin{pmatrix} -\nu^3 - \nu + 1 & -\frac{\sqrt{3}}{2}(\nu^3 + 2\nu) \\ \frac{\sqrt{3}}{2}(-3\nu^3 + 2\nu) & \frac{9\nu^4}{4} + 3\nu^3 - 3\nu^2 - 3\nu + 1 \end{pmatrix}; \\ W_1 &= \begin{pmatrix} -\frac{\nu^3}{2} + \nu & \frac{\sqrt{3}}{4}\nu(\nu^3 + 4\nu^2 - 2\nu + 4) \\ \sqrt{3}\nu(2\nu^2 - 1) & -\frac{3}{2}\nu(\nu^3 - \nu^2 - \nu + 2) \end{pmatrix} \\ W_2 &= \begin{pmatrix} \frac{\nu^3}{2} & -\frac{\sqrt{3}}{4}\nu^2(\nu^2 + 4\nu - 2) \\ 0 & -\frac{3}{2}\nu^2(\nu^2 + \nu - 1) \end{pmatrix} \\ W_3 &= \begin{pmatrix} \frac{3}{4}\nu^3 & -\frac{\sqrt{3}}{8}\nu^3(\nu - 6) \\ -\frac{3\sqrt{3}}{4}\nu^3 & \frac{3}{8}\nu^3(\nu - 6) \end{pmatrix}; \quad W_4 = \begin{pmatrix} \frac{\nu^3}{4} & \frac{\sqrt{3}}{8}\nu^3(\nu - 2) \\ \frac{\sqrt{3}}{4}\nu^3 & \frac{3}{8}\nu^3(\nu - 2) \end{pmatrix}. \end{aligned} \quad (4.4.14)$$

Note that some of the entries are on an order in ν lower than 4. This is caused by the sparseness of the discretisation matrix S^T . While the temporal order of the TDRK4 schemes is higher than necessary for DG schemes with $p \leq 2$, computation of the eigenvalues (more specifically, root-finding of polynomials) severely inhibits progress with higher order bases.

4.4.3 Fully resolved forms: DG-TDRK4 with differentiation by DG

For the linear advection equation, we reformulate the modified TDRK4-DG scheme in terms of Galerkin coefficients. When $f(u) = u$, the standard monotone fluxes in equations (3.4.1b) and (3.4.1d) are $\widehat{f}_{j+\frac{1}{2}} = \widehat{u}_{j+\frac{1}{2}} = u_{j+\frac{1}{2}}^-$ and $\widehat{f}'g_{j+\frac{1}{2}} = \widehat{g}_{j+\frac{1}{2}} = g_{j+\frac{1}{2}}^+$. The derivative and general stage update can be written as

$$\mathbf{G}_j = \frac{1}{h} \mathcal{D}^- \mathbf{U}_j \quad (4.4.15a)$$

$$\begin{aligned} \mathbf{U}_j &= \mathbf{U}_j^n + \frac{\tau}{h} [\alpha \mathcal{D}^- \mathbf{U}_j^n + \beta \tau \mathcal{D}^+ \mathbf{G}_j^n + \beta^* \tau \mathcal{D}^+ \mathbf{G}_j^*] \\ &= \left[I + \alpha \frac{\tau}{h} \mathcal{D}^- - \beta \frac{\tau^2}{h^2} \mathcal{D}^+ \mathcal{D}^- \right] \mathbf{U}_j^n - \beta^* \frac{\tau^2}{h^2} \mathcal{D}^+ \mathcal{D}^- \mathbf{U}_j^*, \end{aligned} \quad (4.4.15b)$$

For clarity, we note that

$$\mathcal{D}^+ \mathcal{D}^- \mathbf{U}_j = (A_2 A_1 - CB) \mathbf{U}_j + A_2 B \mathbf{U}_{j-1} - C A_1 \mathbf{U}_{j+1}.$$

Using equation (4.4.15b) with the coefficient values $\alpha, \alpha^*, \beta, \beta^*$ given in §3.3, we successively write expressions for the Galerkin coefficient vectors \mathbf{U}_j^* and \mathbf{U}_j^{n+1} to obtain a final update of the form

$$\mathbf{U}_j^{n+1} = \sum_{m=-2}^2 W_m \mathbf{U}_{j+m}^n. \quad (4.4.16)$$

For $p = 1$, the matrices are given by

$$\begin{aligned}
W_0 &= \begin{pmatrix} 7\nu^4 + 2\nu^3 - 4\nu^2 - \nu + 1 & \frac{\sqrt{3}}{2}\nu(5\nu^3 + 8\nu^2 - 2\nu - 2) \\ \frac{\sqrt{3}}{2}\nu(5\nu^3 - 4\nu^2 - 2\nu + 2) & 30\nu^4 + 18\nu^3 - 12\nu^2 - 3\nu + 1 \end{pmatrix} \\
W_{-1} &= \begin{pmatrix} -\frac{19}{6}\nu^4 + \nu^3 + 2\nu^2 + \nu & \frac{\nu(-31\nu^3 - 18\nu^2 + 12\nu + 6)}{2\sqrt{3}} \\ \frac{\nu(17\nu^3 + 18\nu^2 - 6\nu - 6)}{2\sqrt{3}} & \frac{1}{2}\nu(23\nu^3 + 30\nu^2 - 6\nu - 6) \end{pmatrix} \\
W_1 &= \begin{pmatrix} -\frac{1}{6}\nu^2(19\nu^2 + 10\nu - 12) & \frac{\nu^2(17\nu^2 + 2\nu - 6)}{2\sqrt{3}} \\ -\frac{\nu^2(31\nu^2 + 10\nu - 12)}{2\sqrt{3}} & \frac{23\nu^4}{2} + \nu^3 - 3\nu^2 \end{pmatrix} \\
W_{-2} &= \begin{pmatrix} -\frac{1}{3}\nu^3(\nu + 4) & -\frac{\nu^3(\nu + 4)}{\sqrt{3}} \\ \frac{\nu^3(\nu + 4)}{2\sqrt{3}} & \frac{1}{2}\nu^3(\nu + 4) \end{pmatrix}; \quad W_2 = \begin{pmatrix} -\frac{\nu^4}{3} & \frac{\nu^4}{2\sqrt{3}} \\ -\frac{\nu^4}{\sqrt{3}} & \frac{\nu^4}{2} \end{pmatrix}.
\end{aligned} \tag{4.4.17}$$

Note that, in contrast to the matrices (??) for the original scheme, each entry here has a term on the order of ν^4 .

4.5 Fully-discrete DG Schemes: Results

In the remainder of this section, we compare the numerical dispersion and dissipation errors for various fully-discrete schemes for basis polynomials of maximal order $p = 0, 1, 2$. Even for these low orders, we provide a context for the temporally fourth-order TDRK4-DG methods by including results for RK4-DG with an upwind-biased flux and the original and new LW-DG method of fourth-order. The computations for the case $p = 3$ become prohibitively slow and cumbersome whilst for $p \geq 4$, the characteristic equation itself becomes more difficult to solve.

Remark. We find that the eigenvalues associated to methods that utilise the LDG technologies to compute higher-order derivatives exhibit terms of a kind not seen in eigenvalues for, say, RK-DG. Let $\zeta := k\tau$. For a fourth-order method, equation (4.1.9) indicates that the physically relevant eigenvalue λ_1 satisfies

$$\lambda_1 = 1 - i\zeta - \frac{1}{2}\zeta^2 + \frac{1}{6}i\zeta^3 + \frac{1}{24}\zeta^4 + \mathcal{O}(\zeta K^{\sigma_1} + \zeta^2 K^{\sigma_2}). \tag{4.5.1}$$

For the fourth order RK4-DG, we have $\sigma_1 = 2p + 1 = \sigma_2$. On the other hand, in addition to such terms, the eigenvalues for the modified TDRK4-DG scheme (which is also superconvergent) display purely spatial error terms and terms on the order of $\frac{K}{\zeta} = \frac{h}{\tau} = \frac{1}{\nu}$. We refer to such terms- those with τ on the denominator- as *reciprocal* terms. We speculate that these reciprocal terms, which amount to a fundamental difference from the original schemes and from RK-DG, are a direct consequence of the integration of LDG into the schemes; in some sense, part of the error is not stepped forward in time.

4.5.1 Piecewise constant basis

When $p = 0$, the schemes comprise only of information from the boundaries. Furthermore, the original LW-DG and TDRK4-DG methods are reduced to the first-order Forward-Euler (FE) DG method with upwind flux ($\theta = 1$) since, by differentiating directly, there are no higher-order derivative terms due to the constant basis.

Forward Euler

The eigenvalue λ of the amplification matrix \mathcal{G} for FE-DG is given by

$$\lambda = 1 + \nu \left((\theta - 1)e^{ikh} - (2\theta - 1) + \theta e^{-ikh} \right), \quad (4.5.2)$$

where $\nu = \frac{\tau}{h}$ is the CFL number. Let $K = kh$. Then, for FE-DG,

$$\begin{aligned} \lambda = 1 & - iK\nu - \frac{1}{2}K^2\nu(2\theta - 1) + \frac{i}{6}K^3\nu + \frac{1}{24}K^4\nu(2\theta - 1) \\ & - \frac{i}{120}K^5\nu - \frac{1}{720}K^6\nu(2\theta - 1) + \mathcal{O}(K^7\nu). \end{aligned} \quad (4.5.3)$$

Note that $K\nu = k\tau$ and, with reference to equation (4.1.9), that

$$e^{-ik\tau} = e^{-iK\nu} = 1 - iK\nu - \frac{1}{2}(K\nu)^2 + \frac{i}{6}(K\nu)^3 + \frac{1}{24}(K\nu)^4 - \frac{i}{120}(K\nu)^5 - \frac{1}{720}(K\nu)^6 + \mathcal{O}((K\nu)^7). \quad (4.5.4)$$

Hence, given that this first-order (in time) method will not reproduce higher order purely temporal terms, the error

$$e^{-ik\tau} - \lambda = \frac{1}{2}K^2\nu(2\theta - 1) - \frac{i}{6}K^3\nu + \mathcal{O}(K^2\nu^2 + K^4\nu)$$

is characterised by *cross-terms*— that is, terms that have unbalanced powers of K and ν . These result in products of τ and h (for example, $K^2\nu = k^2\tau h$) that can lead to ambiguity in which to call the leading order error term. For this reason, we instead focus on the non-dimensionalised quantities described in equations (4.1.10) when discussing higher order bases in §4.5.2 and §4.5.3.

The new TDRK4-DG scheme

The structure of our new TDRK4-DG scheme is more complicated than the schemes equivalent, when $p = 0$, to FE-DG. The associated eigenvalue is

$$\lambda = \frac{1}{24} \left(e^{2iK}\nu^4 - 4e^{iK}(\nu^2 + \nu - 3)\nu^2 + 6(\nu^4 + 2\nu^3 - 4\nu^2 - 4\nu + 4) - 4e^{-iK}(\nu^3 + 3\nu^2 - 3\nu - 6)\nu + e^{-2iK}(\nu + 4)\nu^3 \right), \quad (4.5.5)$$

which, after a Taylor expansion about $K = kh$, can also be written as

$$\lambda = 1 - iK\nu - \frac{1}{2}(K\nu)^2 + \frac{i}{6}(K\nu)^3 + \frac{1}{24}(K\nu)^4 - \frac{1}{2}K^2\nu + \frac{i}{6}K^3\nu + \frac{1}{24}K^4\nu(2\nu^2 + \nu + 1) + K^5 \left(-\frac{i\nu^3}{24} - \frac{i\nu}{120} \right) + \mathcal{O}(K^6\nu). \quad (4.5.6)$$

Note that the terms in equation (4.5.4) up to and including fourth order in time are reproduced. This is in contrast to the schemes that use direct differentiation

to compute the terms arising from the Cauchy Kowalewski procedure.

Non-dimensionalised presentation

We now present Taylor series expansions of the dispersion $\Omega_{\text{Re}} - K$ and dissipation Ω_{Im} errors using the quantities defined in equations (4.1.10).

$p = 0 : \Omega_{\text{Re}} + i\Omega_{\text{Im}} = K + X_2 K^2 + X_3 K^3 + \mathcal{O}(K^4)$		
	Dissipation error X_2	Dispersion error X_3
Forward Euler	$-\frac{2\theta-1}{2} + \frac{1}{2}\nu$	$-\frac{1}{6} + \frac{2\theta-1}{2}\nu - \frac{1}{3}\nu^2$
RK4	$-\frac{2\theta-1}{2}$	$-\frac{1}{6}$
Modified TDRK4/LW4	$-\frac{1}{2}$	$-\frac{1}{6} + \frac{1}{2}\nu$

Table 4.1: Leading dispersion and dissipation errors for $p = 0$

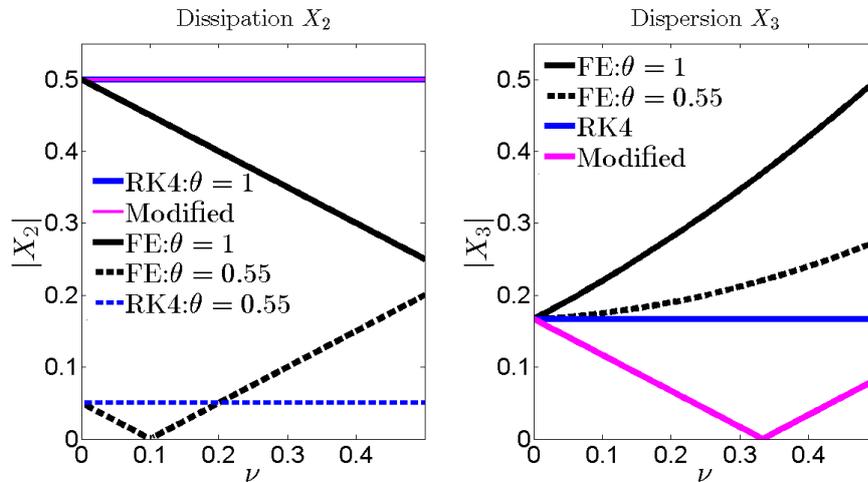


Figure 4.5.1: $p = 0$: Magnitude of coefficients of leading order error terms $X_2 K^2 + iX_3 K^3$

The coefficients, which are polynomials in ν , of the leading order dispersion and dissipation terms are shown in Table 4.1 respectively. Each of the schemes we consider has dispersion error $K - \Omega_{\text{Re}}$ on the order of K^3 and dissipation error Ω_{Im} on the order of K^2 . Note that the leading-order terms for the Forward-Euler methods are reduced when one employs an upwind-biased flux. Furthermore, the

leading order dissipation term is zero when $\nu = 2\theta - 1$ for $\theta > \frac{1}{2}$.

Naturally, employing higher-order Runge-Kutta schemes (we consider RK4) reduces the contribution of the CFL number ν to the leading order terms. For RK4-DG, while θ does contribute to the trailing dispersion error terms, in the unstable limit $\theta = \frac{1}{2}$, the scheme is purely dispersive; the constants attached to the dissipation error terms can be reduced by taking a smaller value of $\theta \in (\frac{1}{2}, 1]$.

For the modified methods (with $p = 0$, the modified TDRK4-DG and LW4-DG reduce to the same scheme), the reduced number of stages in comparison to RK4 leads to a higher dependence on the CFL number ν . Note that for $\nu > \frac{1}{3}$, the sign of the leading order dispersion term changes. However, it is important to note that Figure 4.5.1 includes some data for unstable schemes. A full investigation of maximal CFL numbers is left to further work.

4.5.2 Piecewise linear basis

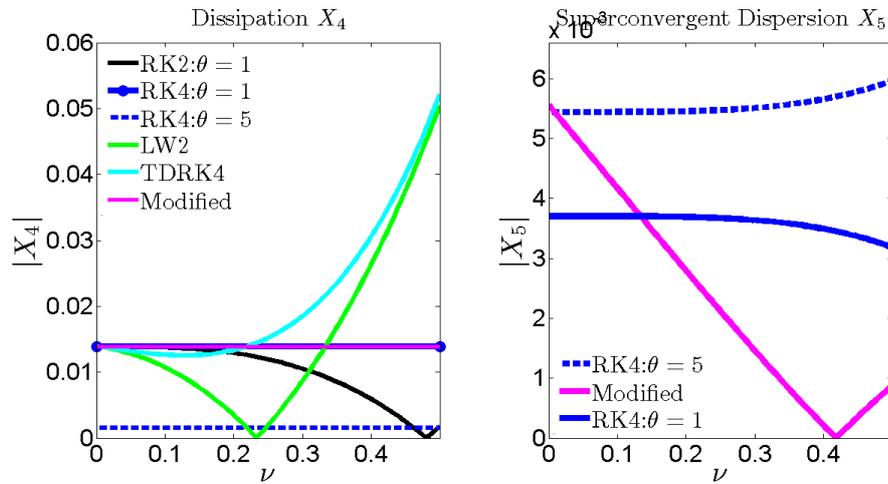
As with the piecewise constant case, we give results for the lowest necessary order of RK method (RK2), for RK4 and for both the original and modified LW-DG methods in order to provide a context for the TDRK4-DG results.

Tables 4.3 and 4.2 give the leading order coefficients for the dispersion and dissipation errors when $p = 1$. The RK4-DG method as well as the modified

$p = 1 : \Omega_{\text{Re}} + i\Omega_{\text{Im}} = K + X_3K^3 + X_4K^4 + \mathcal{O}(K^5)$		
	Leading dispersion X_3	Leading dissipation X_4
RK2	$\frac{1}{6}\nu^2$	$-\left(\frac{1}{72(2\theta-1)} - \frac{1}{8}\nu^3\right)$
LW2	$\left(\frac{1}{6}\nu^2 + \frac{1}{12}\nu\right)$	$\left(\frac{1}{8}\nu^3 + \frac{1}{6}\nu^2 + \frac{1}{72}\nu - \frac{1}{72}\right)$
TDRK4	$-\left(\frac{1}{6}\nu^2 - \frac{1}{12}\nu\right)$	$\left(-\frac{5}{12}\nu^3 + \frac{1}{36}\nu^2 + \frac{1}{72}\nu - \frac{1}{72}\right)$

Table 4.2: Leading dispersion and dissipation errors for $p = 1$

$p = 1 : \Omega_{\text{Re}} + i\Omega_{\text{Im}} = K + X_4 K^4 + X_5 K^5 + \mathcal{O}(K^6)$		
	Leading dissipation X_4	Leading dispersion X_5
RK4	$-\frac{1}{72(2\theta-1)}$	$-\left(\frac{1}{120}\nu^4 - \frac{1+6\theta-6\theta^2}{270(1-2\theta)^2}\right)$
Modified LW4	$-\frac{1}{72}$	$-\left(\frac{1}{120}\nu^4 - \frac{1}{72}\nu + \frac{1}{180}\right)$
Modified TDRK4	$-\frac{1}{72}$	$-\left(\frac{1}{120}\nu^4 - \frac{1}{72}\nu + \frac{1}{180}\right)$

Table 4.3: Leading dispersion and dissipation errors for $p = 1$ Figure 4.5.2: $p = 1$: Magnitude of coefficients of leading order error terms $iX_4 K^4 + X_5 K^5$

methods have superconvergent dispersion errors on the order of K^5 . In contrast, the second-order schemes and the original TDRK4-DG have dispersion errors on the order of K^3 . That is, the order of the RK dispersion error can be increased by adding more stages but, while the original TDRK4 has excessive temporal order for this problem, its order is the same as for RK2 and LW2. In terms of magnitude, for small ν , the original TDRK4 has better dispersion than LW2 but worse than RK2. The dissipation error for RK4 can be reduced by taking a larger value of θ ; Figure 4.5.2 includes a curve for $\theta = 5$. The price for this choice is an increase in dispersion error.

When $\theta = 1$, the expressions for the second-order schemes agree with equation (3.9) in [60]. In that paper, the authors point out that the CFL restriction

(here, it is $\nu \leq \frac{1}{3}$) prevents the careful selection which would cancel the leading order dissipation term. With the flux parameter θ , the required CFL number $\nu = \frac{1}{\sqrt[3]{9(2\theta-1)}}$ can be achieved for $\theta \geq 2$. Note that in contrast to the $p = 0$ case, the coefficients grow as $\theta \rightarrow \frac{1}{2}$.

While the leading order error term for RK2 is dispersive, for RK4, it is dissipative and the dispersion error is two orders higher. As with the RK methods, for LWDG2, we see reduced dispersion errors but increased dissipation with a reduced CFL number ν . For a full discussion of how LW2 compares with RK2, see [60].

As can be seen from the tables, the dispersion error $K - \Omega_{\text{Re}}$ for the original TDRK4 scheme, which is on the order of K^3 , is two orders lower than RK4 and the scheme does not exhibit superconvergence. On the other hand, the dissipation error Ω_{Im} , which is $\mathcal{O}(K^4)$, is of the same order. As with the original LWDG2 method, it is not possible to improve the dispersion error (although here the constant attached to the leading order dispersion term is smaller in comparison) by choosing $\nu = \frac{1}{2}$ since this contradicts the (numerically) observed CFL restriction. However, the original LW-DG has better dissipation error than the original TDRK4-DG. We now turn to the modified methods which use LDG to compute derivatives.

With the carefully complimented numerical flux and derivative approximation, the superconvergent dispersion/dissipation orders of error seen with RK4-DG are recovered. For $\nu \lesssim 0.134$, the constant attached to the leading order dispersion error term is greater than for RK4.

4.5.3 Piecewise quadratic basis

As the order (here it is 3) of the DG spatial discretisation gets closer to the temporal order of the TDRK4 method, the leading order dispersion and dissipation error terms come to agree with those of the methods we use for comparison: the modified LW4-DG method of Qiu et al. [31] and the RK4-DG

method.

Non-trivial identities

Key to the realisation of an estimate for our new TDRK4 method is the identification of what the literature has referred to as a *non-trivial identity*. The radical expression for the eigenvalues obtained via Mathematica includes, after a small amount of manual simplification, the term

$$\Psi = \left(13 + 90\nu - 75\nu^2 - 1450\nu^3 - 750\nu^4 + 7500\nu^5 + 1875\nu^6 - 18750\nu^7 + 31250\nu^9 + 5(1 + 3\nu - 10\nu^2 - 15\nu^3) \sqrt{17 + 30\nu - 275\nu^2 - 150\nu^3 + 750\nu^4 - 1875\nu^6} \right)^{\frac{1}{3}}.$$

The fractional exponent causes this expression to halt the preprogrammed simplification process. To deal with this term, we set

$$\Upsilon = 13 + 90\nu - 75\nu^2 - 1450\nu^3 - 750\nu^4 + 7500\nu^5 + 1875\nu^6 - 18750\nu^7 + 31250\nu^9$$

$$\Phi = 5(1 + 3\nu - 10\nu^2 - 15\nu^3)$$

$$\Gamma = 17 + 30\nu - 275\nu^2 - 150\nu^3 + 750\nu^4 - 1875\nu^6$$

so that $\Psi = \left(\Upsilon + \Phi\sqrt{\Gamma} \right)^{\frac{1}{3}}$ and seek an expression of the form $a + b\sqrt{\Gamma}$ equivalent to Ψ . We solve for a and b , using Mathematica, the simultaneous equations

$$a^3 + 3ab^2\Gamma = \Upsilon$$

$$3a^2b + b^3\Gamma = \Phi,$$

requiring that the solutions be real. This results in the identity

$$\Psi = \frac{1 + 5\nu - 25\nu^3}{2^{2/3}} + \frac{1}{2^{2/3}}\sqrt{\Gamma}. \quad (4.5.7)$$

By replacing in the expression for the eigenvalues integer multiples of the problematic term Ψ by integer multiples of our simplified expression, we rid the

expression of most of its components with fractional exponents (we retain some square roots of Γ of course). The Taylor series expansions of the eigenvalues are now more amenable to manipulation and simplification and, in inspecting the individual coefficients with the goal of presenting them in a succinct manner, it remains only to identify, usually with the assistance of the mathematical software, algebraic identities such as

$$0 = -\nu - 5\nu^2 + 25\nu^4 + \nu\sqrt{17 + 30\nu - 275\nu^2 - 150\nu^3 + 750\nu^4 - 1875\nu^6} \\ - \frac{16\nu + 20\nu^2 - 300\nu^3 + 100\nu^4 + 1000\nu^5 - 2500\nu^7}{1 + 5\nu - 25\nu^3 + \sqrt{17 + 30\nu - 275\nu^2 - 150\nu^3 + 750\nu^4 - 1875\nu^6}}.$$

Such further identities are usually related to the quantities discussed above. Indeed, “guessing” that one of these quantities (for example, $1 + 5\nu - 25\nu^3$) is a factor of some expression one desires to present in a form more amenable to interpretation can be the *crux move* in the process. This, for example, can motivate the identification of the denominator in the rational functions in θ seen our presentation of the Taylor series expansions of the DG scheme with upwind-biased flux.

Superconvergent methods

We categorise the methods in our study into two sets: the superconvergent DG methods (RK4-DG, modified LW4-DG and our new TDRK4-DG) and non-superconvergent DG methods (the original TDRK4-DG scheme studied by Seal et al. and the original LW-DG scheme). The physically relevant eigenvalue for each of the superconvergent methods satisfies

$$\Omega_{\text{Re}} + i\Omega_{\text{Im}} = K - \frac{1}{120}\nu^4 K^5 - \left(\frac{\nu^5}{144} + \frac{2\theta - 1}{7200}\right) K^6 + \mathcal{O}(K^7), \quad (4.5.8)$$

where we take $\theta = 1$ for the new TDRK4-DG and modified LW4-DG methods.

	Trailing dispersion X_7
RK4	$\frac{\nu^6}{336} + \frac{1}{300} (\theta^2 - \theta + \frac{1}{14})$
NewTD4	$\frac{\nu^6}{336} + \frac{\nu}{7200} - \frac{1}{31500} + \frac{1}{45000\nu} + \frac{1}{168750\nu^2}$
NewLW4	$\frac{\nu^6}{336} + \frac{\nu}{7200} - \frac{1}{31500} - \frac{13}{2250000\nu^3} - \frac{49}{16875000\nu^4}$

Table 4.4: $p = 2$; Coefficient of trailing dispersion error term $X_7 K^7$ in equation (4.5.8)

	Trailing dissipation X_8
RK4	$\frac{\nu^7}{1152} + \frac{48\theta^3 - 72\theta^2 + 26\theta - 1}{120000}$
NewTD4	$\frac{\nu^7}{1152} + \frac{\nu^2}{10800} - \frac{\nu}{36000} + \frac{1}{40000} - \frac{1}{675000\nu} + \frac{1}{2250000\nu^2}$
NewLW4	$\frac{\nu^7}{1152} + \frac{\nu^2}{21600} - \frac{\nu}{36000} + \frac{1}{360000} - \frac{1}{84375\nu} - \frac{17}{2250000\nu^2}$

Table 4.5: $p = 2$; Coefficient of trailing dissipation error term $X_8 K^8$ in equation (4.5.8)

Non-superconvergent methods

The physically relevant eigenvalue for the original TDRK4-DG scheme with $p = 2$ satisfies

$$\lambda = 1 - iK\nu - \frac{(K\nu)^2}{2} + \frac{i}{6}(K\nu)^3 + K^4 \nu \frac{(102500\nu^3 - 108690\nu^2 + 36625\nu - 3792)}{15000(15\nu^3 + 10\nu^2 - 3\nu - 1)} - iK^5 \nu \frac{(8757500\nu^3 - 10247480\nu^2 + 3698215\nu - 399928)}{625000(15\nu^3 + 10\nu^2 - 3\nu - 1)^2} + \mathcal{O}(K^6). \quad (4.5.9)$$

Note that the fourth order temporal term $\frac{1}{12}(K\nu)^4 = \frac{1}{12}k^4\tau^4$ is not reproduced by this method. This is due to the direct method of differentiation of the piecewise quadratic basis functions. The above expression (4.5.9) yields

$$\Omega_{\text{Re}} = K + K^5 \left(\frac{\nu^4}{30} - \frac{(102500\nu^3 - 108690\nu^2 + 36625\nu - 3792)\nu}{15000(15\nu^3 + 10\nu^2 - 3\nu - 1)} + \frac{8757500\nu^3 - 10247480\nu^2 + 3698215\nu - 399928}{625000(-15\nu^3 - 10\nu^2 + 3\nu + 1)^2} \right)$$

$$\Omega_{\text{Im}} = K^4 \left(-\frac{\nu^3}{24} + \frac{102500\nu^3 - 108690\nu^2 + 36625\nu - 3792}{15000(15\nu^3 + 10\nu^2 - 3\nu - 1)} \right).$$

4.6 Numerical Experiments

4.6.1 Linear advection: dispersion and dissipation

For the linear advection equation, we seek to numerically verify the observations made in the Fourier analysis section and, more importantly, predict the situation for $p = 3$ which we were not able to perform analysis for, with an investigation into the values related to dissipation defined in equation (4.1.12). The analogous values for dispersion are considerably more difficult to track. While the results are promising, a complete investigation into these quantities is left to further work.

mesh	RKDG2, $\theta = 1, \nu = \frac{1}{3}$		RKDG2, $\theta = 2, \nu = \frac{1}{10}$	
	$\ln(\max u_h)$	N2	$\ln(\max u_h)$	N2
50	-5.76E-00	-	-3.07E-00	-
100	-7.28E-01	3.98	-3.60E-01	4.09
200	-8.99E-02	4.01	-4.43E-02	4.02
400	-1.13E-02	3.98	-5.28E-03	4.06
mesh	RKDG3, $\theta = 1, \nu = \frac{1}{5}$		RKDG3, $\theta = 0.55, \nu = \frac{1}{5}$	
	$\ln(\max u_h)$	N2	$\ln(\max u_h)$	N2
50	-2.34E-01	-	-2.14E-01	-
100	-2.72E-02	4.10	-2.66E-02	4.00
200	-3.37E-03	4.01	-3.35E-03	3.99
400	-4.16E-02	4.00	-4.16E-04	4.00

Table 4.6: Linear advection with $u_0(x) = \cos(4x)$ at $T = 400\pi$. Dissipation errors of RKDG methods for $p = 1$ and $p = 2$ with different values of θ .

mesh	RK(10,4)-DG, $\nu = 0.4$		TDRK4, $\nu = 0.08$		newTDRK4, $\nu = 0.1$	
	$\ln(\max u_h)$	N2	$\ln(\max u_h)$	N2	$\ln(\max u_h)$	N2
50	-2.85E-04	-	-3.43E-05	-	-2.73E-03	-
100	-9.82E-06	5.86	-1.66E-06	5.36	-8.85E-05	5.94
200	-2.91E-05	-0.56	-2.87E-05	-3.11	-3.17E-05	2.48
400	-9.05E-07	6.00	-8.99E-07	6.00	-9.84E-07	6.01

Table 4.7: Linear advection with $u_0(x) = \cos(4x)$ at $T = 400\pi$. Dissipation errors of methods for $p = 3$.

Conclusions

“Eventually, all things merge into one, and a river runs through it. The river was cut by the world’s great flood and runs over rocks from the basement of time. On some of those rocks are timeless raindrops. Under the rocks are the words, and some of the words are theirs. I am haunted by waters.”

– Norman Maclean, *A River Runs Through It*

Three types of superconvergence that reside in physical space, in a space governed by the negative-order norm, and in Fourier space are deeply connected to each other. The special, super-accurate points of the discontinuous Galerkin semi-discretisation are created by the numerical flux function and are obfuscated by a sea of points around them that are of the expected order of accuracy. Their riches, far from localised in certain non-physical spaces, can be mined and globally disseminated via a SIAC post-processing at the final time, leading to highly accurate long-term propagation of waves. We showed that the upwind-biased flux offers opportunities to reduce the errors in RK-DG approximations by choosing the value of its parameter sensitively relative to the parity of the polynomial degree p . This is done by reducing the magnitude of error constants that depend on the flux. No such additional complications were encountered in obtaining superconvergent orders of accuracy in the SIAC filtered error. This suggests interesting further work related to simplifying the pointwise and spectral arguments with a properly defined initial interpolation.

Further work also includes stability analysis for the new two-derivative Runge-Kutta DG method we introduced in Chapter 3, nonlinear numerical experiments and numerical verification of the expected outcome of the highly complicated Fourier analysis of multi-stage multi-derivative schemes for piecewise cubic polynomials. In the longer term, it would be interesting to investigate in more depth potential benefits of varying the parameters in a Rusanov-type flux. This could bring the benefits seen for the upwind-biased flux to multidervative methods and to nonlinear equations. One direction to investigate with regards to improving the pairing between the two discretisations is a mechanism for switching between time scheme and/or flux depending on the time regime, or more precisely, the component of the error dominant at a particular time. The superconvergence properties of the DG method will be key to developing improved time-stepping schemes that offer more to ever evolving computer architectures; yesterday's weirdness is tomorrow's reason why ¹.

¹Hunter S. Thompson, *The Curse of Lono*

A

Mathematica Code

A.1 Semi-discrete example

Appendix 1: Example *Mathematica* code: semi-discrete scheme

Some “tricks” worth mentioning: 1. For schemes with intertwined discretisations, it can be useful to perform an extra integration by parts

2. Changes of variables to reduce the number of unknowns can improve computational efficiency

3. Keep an eye out for factors that pop up elsewhere e.g. $(2\theta-1)$ in the P^1 DG eigenvalues. These are key to finding tidy closed forms.

4. A potentially useful trick could involve restricting the value of the CFL number ν (e.g. $\nu < \frac{1}{3}$). It seems that some expressions may simplify to the expected value only for small enough ν .

5.

■ Runge-Kutta discontinuous Galerkin: P^1 polynomial basis

```
Clear["Global`*"]
```

General Setup

Orthonormal Legendre polynomials

```
f1[x_] = LegendreP[0, x]
```

```
f2[x_] = Sqrt[3] * LegendreP[1, x]
```

1

$\sqrt{3} x$

M is mass matrix, DM is stiffness matrix, DMt is transpose stiffness matrix for direct differentiation of approximation (for Lax-Wendroff type methods), DMt2 for second derivative. The other matrices represent boundary contributions from the flux (e.g. LLM=Left boundary from the left). All matrices scaled to canonical element [-1,1].

```

F[x_] = {f1[x], f2[x]} // Simplify;
M = Table[a, {2}, {2}];
DM = Table[a, {2}, {2}];
DMt = Table[a, {2}, {2}];
D2Mt = Table[a, {2}, {2}];
RLM = Table[a, {2}, {2}];
RRM = Table[a, {2}, {2}];
LRM = Table[a, {2}, {2}];
LLM = Table[a, {2}, {2}];

GG = Module[
  { k, j},
  For[ k = 1, k ≤ 2, k++,
  For[ j = 1, j ≤ 2, j++,
  M[[k, j]] =  $\frac{1}{2}$  * Integrate[F[x][[k]] * F[x][[j]], {x, -1, 1}]]];
  For[ k = 1, k ≤ 2, k++,
  For[ j = 1, j ≤ 2, j++,
  DM[[k, j]] = Integrate[D[F[x][[k]], x] * F[x][[j]], {x, -1, 1}]]];
  (*For[ k=1,k≤2,k++,
  For[ j=1,j≤2,j++,
  DMt[[k, j]] = Integrate[D[F[x][[j]], x] * F[x][[k]], {x, -1, 1}]]]; *)
  (*For[ k=1,k≤2,k++,
  For[ j=1,j≤2,j++,
  D2Mt[[k, j]] = Integrate[D[D[F[x][[j]], x], x] * F[x][[k]], {x, -1, 1}]]]; *)
  For[ k = 1, k ≤ 2, k++,
  For[ j = 1, j ≤ 2, j++, LLM[[k, j]] = F[-1][[k]] * F[1][[j]]];
  For[ k = 1, k ≤ 2, k++,
  For[ j = 1, j ≤ 2, j++, LRM[[k, j]] = F[-1][[k]] * F[-1][[j]]];
  For[ k = 1, k ≤ 2, k++,
  For[ j = 1, j ≤ 2, j++, RLM[[k, j]] = F[1][[k]] * F[1][[j]]];
  For[ k = 1, k ≤ 2, k++,
  For[ j = 1, j ≤ 2, j++, RRM[[k, j]] = F[1][[k]] * F[-1][[j]]];
  ]

```

A1 is contribution to cell I_j resulting from an upwind flux u^- ; A2 is contribution to cell I_j resulting from a downwind flux u^+ ; B applies to cell I_{j-1} and results from u^- flux; C to I_{j+1} due to u^+ flux.

```

M
A1 = Simplify[Expand[(DM - RLM)]]
A2 = Simplify[Expand[(DM + LRM)]]
A = A1 + A2
Cp = Simplify[Expand[RRM]]
Bm = Simplify[Expand[LLM]]
Simplify[Expand[DM]]
Simplify[Expand[DMt]]
{{1, 0}, {0, 1}}
{{-1, -√3}, {√3, -3}}
{{1, -√3}, {√3, 3}}
{{0, -2√3}, {2√3, 0}}
{{1, -√3}, {√3, -3}}
{{1, √3}, {-√3, -3}}
{{0, 0}, {2√3, 0}}
{{a, a}, {a, a}}

Ath = θ * A1 + (1 - θ) * A2 // Simplify
{{1 - 2θ, -√3}, {√3, 3 - 6θ}}

```

Digression: Differentiation operators

My thesis shows how the schemes can be written as linear combinations of operators D^- and D^+ . For RK-DG schemes, the choice of flux is always the same. Hence, one only needs information about the semi-discrete scheme (in turn, these operators) to be able to derive all information desired about the fully discrete scheme.

$$D_p = A2 - C_p * \text{Exp}[z]$$

$$D_m = A1 + B_m * \text{Exp}[-z]$$

$$D_{pp} = A2.A2 - (A2.C_p + C_p.A2) * \text{Exp}[z] + C_p.C_p * \text{Exp}[z] * \text{Exp}[z]$$

$$D_{pm} = A2.A1 - C_p.B_m + A2.B_m * \text{Exp}[-z] - C_p.A1 * \text{Exp}[z]$$

$$\left\{ \left\{ 1 - e^z, -\sqrt{3} + \sqrt{3} e^z \right\}, \left\{ \sqrt{3} - \sqrt{3} e^z, 3 + 3 e^z \right\} \right\}$$

$$\left\{ \left\{ -1 + e^{-z}, -\sqrt{3} + \sqrt{3} e^{-z} \right\}, \left\{ \sqrt{3} - \sqrt{3} e^{-z}, -3 - 3 e^{-z} \right\} \right\}$$

$$\left\{ \left\{ -2 + 4 e^z - 2 e^{2z}, -4 \sqrt{3} + 2 \sqrt{3} e^z + 2 \sqrt{3} e^{2z} \right\}, \right. \\ \left. \left\{ 4 \sqrt{3} - 2 \sqrt{3} e^z - 2 \sqrt{3} e^{2z}, 6 + 24 e^z + 6 e^{2z} \right\} \right\}$$

$$\left\{ \left\{ -8 + 4 e^{-z} + 4 e^z, -2 \sqrt{3} + 4 \sqrt{3} e^{-z} - 2 \sqrt{3} e^z \right\}, \right. \\ \left. \left\{ -2 \sqrt{3} - 2 \sqrt{3} e^{-z} + 4 \sqrt{3} e^z, -24 - 6 e^{-z} - 6 e^z \right\} \right\}$$

Eigenvalues[Dp]

$$\text{lamb1}[z_] = 2 + e^z - \sqrt{-2 + 10 e^z + e^{2z}} // \text{Simplify};$$

$$\text{temp1}[z_] = \text{Simplify}[\text{Expand}[\text{Normal}[\text{Series}[\text{lamb1}[z], \{z, 0, 9\}]]]];]$$

$$\text{lamb1Dp}[z_] = \text{temp1}[z];$$

$$\text{lamb1Dp}[z] // \text{Expand}$$

$$\text{EvaluateDp} = \frac{1}{dx} \% /. z \rightarrow i * \omega * dx // \text{Expand}$$

$$\left\{ 2 + e^z - \sqrt{-2 + 10 e^z + e^{2z}}, 2 + e^z + \sqrt{-2 + 10 e^z + e^{2z}} \right\}$$

$$-z + \frac{z^4}{72} - \frac{z^5}{270} + \frac{z^6}{648} - \frac{25 z^7}{27216} + \frac{103 z^8}{233280} - \frac{79 z^9}{349920}$$

$$-i \omega + \frac{dx^3 \omega^4}{72} - \frac{1}{270} i dx^4 \omega^5 - \frac{dx^5 \omega^6}{648} + \frac{25 i dx^6 \omega^7}{27216} + \frac{103 dx^7 \omega^8}{233280} - \frac{79 i dx^8 \omega^9}{349920}$$

Eigenvalues[Dm]

```

lamb1[z_] = e-z  $\left(-1 - 2 e^z + \sqrt{1 + 10 e^z - 2 e^{2z}}\right)$  // Simplify;
temp1[z_] = Simplify[Expand[Normal[Series[lamb1[z], {z, 0, 9}]]]];
lamb1Dm[z_] = temp1[z];
lamb1Dm[z] // Expand
Evalue1Dm =  $\frac{1}{dx}$  % /. z → i * ω * dx // Expand
{e-z  $\left(-1 - 2 e^z - \sqrt{1 + 10 e^z - 2 e^{2z}}\right)$ , e-z  $\left(-1 - 2 e^z + \sqrt{1 + 10 e^z - 2 e^{2z}}\right)$ }
-z -  $\frac{z^4}{72} - \frac{z^5}{270} - \frac{z^6}{648} - \frac{25 z^7}{27216} - \frac{103 z^8}{233280} - \frac{79 z^9}{349920}$ 
-i ω -  $\frac{dx^3 \omega^4}{72} - \frac{1}{270} i dx^4 \omega^5 + \frac{dx^5 \omega^6}{648} + \frac{25 i dx^6 \omega^7}{27216} - \frac{103 dx^7 \omega^8}{233280} - \frac{79 i dx^8 \omega^9}{349920}$ 

```

Eigenvalues[Dpp]

```

lamb1[z_] = 2  $\left(1 + 7 e^z + e^{2z} + \sqrt{-8 + 32 e^z + 42 e^{2z} + 14 e^{3z} + e^{4z}}\right)$  // Simplify;
temp1[z_] = Simplify[Expand[Normal[Series[lamb1[z], {z, 0, 5}]]]];
lamb1Dpp[z_] = temp1[z];
lamb1Dpp[z] // Expand
% /. z → i * ω * dx
{2  $\left(1 + 7 e^z + e^{2z} - \sqrt{-8 + 32 e^z + 42 e^{2z} + 14 e^{3z} + e^{4z}}\right)$ ,
2  $\left(1 + 7 e^z + e^{2z} + \sqrt{-8 + 32 e^z + 42 e^{2z} + 14 e^{3z} + e^{4z}}\right)$ }
36 + 36 z + 21 z2 + 10 z3 +  $\frac{23 z^4}{6} + \frac{239 z^5}{180}$ 
36 + 36 i dx ω - 21 dx2 ω2 - 10 i dx3 ω3 +  $\frac{23 dx^4 \omega^4}{6} + \frac{239}{180} i dx^5 \omega^5$ 

```

Construction of semi-discrete amplification matrix G for upwind-biased scheme

```

G = Expand[Ath + θ * Bm * Exp[-I * k * dx] - (1 - θ) * Cp * Exp[I * k * dx]] // Simplify
{{1 + ei dx k (-1 + θ) - 2 θ + e-i dx k θ, -√3 e-i dx k (-1 + ei dx k) (ei dx k (-1 + θ) + θ)},
{√3 e-i dx k (-1 + ei dx k) (ei dx k (-1 + θ) + θ), -3 e-i dx k (1 + ei dx k) (ei dx k (-1 + θ) + θ)}}

```

For this simple case, the inbuilt *Mathematica* function manages just fine

Eigenvalues[G] // Simplify

```

{e-2 i dx k
(e2 i dx k (2 - 4 θ) - e3 i dx k (-1 + θ) - ei dx k θ - √(e2 i dx k (e4 i dx k (-1 + θ)2 + θ2 + 2 ei dx k θ (1 + 4 θ) + 2 e3 i dx k (5 - 9 θ + 4 θ2) + 2 e2 i dx k (-1 - 9 θ + 9 θ2)))},
e-2 i dx k (e2 i dx k (2 - 4 θ) - e3 i dx k (-1 + θ) - ei dx k θ + √(e2 i dx k (e4 i dx k (-1 + θ)2 + θ2 + 2 ei dx k θ (1 + 4 θ) + 2 e3 i dx k (5 - 9 θ + 4 θ2) + 2 e2 i dx k (-1 - 9 θ + 9 θ2)))})}

```

```

lamb1[dx_] = e-2 i dx k (e2 i dx k (2 - 4 θ) - e3 i dx k (-1 + θ) -
  ei dx k θ - √(e2 i dx k (e4 i dx k (-1 + θ)2 + θ2 + 2 ei dx k θ (1 + 4 θ) +
    2 e3 i dx k (5 - 9 θ + 4 θ2) + 2 e2 i dx k (-1 - 9 θ + 9 θ2))) // Simplify;
lamb2[dx_] = e-2 i dx k (e2 i dx k (2 - 4 θ) - e3 i dx k (-1 + θ) - ei dx k θ +
  √(e2 i dx k (e4 i dx k (-1 + θ)2 + θ2 + 2 ei dx k θ (1 + 4 θ) +
    2 e3 i dx k (5 - 9 θ + 4 θ2) + 2 e2 i dx k (-1 - 9 θ + 9 θ2))) // Simplify;

```

For the upwind scheme, these are

```

lamb1th1[dx_] = e-2 i dx k (e2 i dx k (2 - 4 θ) - e3 i dx k (-1 + θ) -
  ei dx k θ - √(e2 i dx k (e4 i dx k (-1 + θ)2 + θ2 + 2 ei dx k θ (1 + 4 θ) +
    2 e3 i dx k (5 - 9 θ + 4 θ2) + 2 e2 i dx k (-1 - 9 θ + 9 θ2))) /. θ → 1 // Simplify
lamb2th1[dx_] = e-2 i dx k (e2 i dx k (2 - 4 θ) - e3 i dx k (-1 + θ) - ei dx k θ +
  √(e2 i dx k (e4 i dx k (-1 + θ)2 + θ2 + 2 ei dx k θ (1 + 4 θ) + 2 e3 i dx k (5 - 9 θ + 4 θ2) +
    2 e2 i dx k (-1 - 9 θ + 9 θ2))) /. θ → 1 // Simplify
-2 - e-i dx k - e-2 i dx k √(e2 i dx k + 10 e3 i dx k - 2 e4 i dx k)
-2 - e-i dx k + e-2 i dx k √(e2 i dx k + 10 e3 i dx k - 2 e4 i dx k)

```

Taylor expansions of eigenvalues for different values of θ (so that we can interpolate to find a closed form for the coefficients)

```

temp1[dx_] = Simplify[Expand[Normal[Series[lamb1th1[dx], {dx, 0, 2}]]]];
temp2[dx_] = Simplify[Expand[Normal[Series[lamb2th1[dx], {dx, 0, 7}]]]];

lamb1th1[dx_] = temp1[dx];
lamb2th1[dx_] = temp2[dx];

temp1[dx_] = Simplify[Expand[Normal[Series[lamb1[dx] /. θ → 9/10, {dx, 0, 2}]]]];
temp2[dx_] = Simplify[Expand[Normal[Series[lamb2[dx] /. θ → 9/10, {dx, 0, 7}]]]];

lamb1th9[dx_] = temp1[dx];
lamb2th9[dx_] = temp2[dx];

temp1[dx_] = Simplify[Expand[Normal[Series[lamb1[dx] /. θ → 4/5, {dx, 0, 2}]]]];
temp2[dx_] = Simplify[Expand[Normal[Series[lamb2[dx] /. θ → 4/5, {dx, 0, 7}]]]];

lamb1th8[dx_] = temp1[dx];
lamb2th8[dx_] = temp2[dx];

temp1[dx_] = Simplify[Expand[Normal[Series[lamb1[dx] /. θ → 7/10, {dx, 0, 2}]]]];
temp2[dx_] = Simplify[Expand[Normal[Series[lamb2[dx] /. θ → 7/10, {dx, 0, 7}]]]];

```

```

lamb1th7[dx_] = temp1[dx];
lamb2th7[dx_] = temp2[dx];

temp1[dx_] = Simplify[Expand[Normal[Series[lamb1[dx] /.  $\theta \rightarrow \frac{3}{5}$ , {dx, 0, 2}]]]];
temp2[dx_] = Simplify[Expand[Normal[Series[lamb2[dx] /.  $\theta \rightarrow \frac{3}{5}$ , {dx, 0, 7}]]]];

lamb1th6[dx_] = temp1[dx];
lamb2th6[dx_] = temp2[dx];

```

Now the tricky part: find tidy closed form

Closed form for semi-discrete eigenvalues

Leading order term of non-physically relevant eigenvalue is given by interpolation:

```

InterpolatingPolynomial[
  {Coefficient[lamb1th1[dx], dx, 0], Coefficient[lamb1th9[dx], dx, 0],
   Coefficient[lamb1th8[dx], dx, 0], Coefficient[lamb1th7[dx], dx, 0],
   Coefficient[lamb1th6[dx], dx, 0]}, x] /. x -> 11 - 10  $\theta$  // Simplify
6 - 12  $\theta$ 

```

For the physically relevant eigenvalue:

```

lamb2th1[dx]
lamb2th6[dx]
- i dx k -  $\frac{dx^4 k^4}{72} - \frac{1}{270} i dx^5 k^5 + \frac{dx^6 k^6}{648} + \frac{25 i dx^7 k^7}{27 216}$ 
- i dx k -  $\frac{5 dx^4 k^4}{72} - \frac{61}{270} i dx^5 k^5 + \frac{485 dx^6 k^6}{648} + \frac{67 393 i dx^7 k^7}{27 216}$ 

```

Term on the order dx^4 : (Educated guess factor of $\frac{1}{2\theta-1}$)

```

InterpolatingPolynomial[{Coefficient[(2 * 1 - 1) * lamb2th1[dx], dx, 4],
  (2 *  $\frac{9}{10}$  - 1) * Coefficient[lamb2th9[dx], dx, 4],
  (2 *  $\frac{8}{10}$  - 1) * Coefficient[lamb2th8[dx], dx, 4],
  (2 *  $\frac{7}{10}$  - 1) * Coefficient[lamb2th7[dx], dx, 4],
  (2 *  $\frac{6}{10}$  - 1) * Coefficient[lamb2th6[dx], dx, 4]}, x] /. x -> 11 - 10  $\theta$  // Simplify
-  $\frac{k^4}{72}$ 

```

Term on the order dx^4 : (Guess factor of $\frac{1}{(2\theta-1)^2}$)

```

InterpolatingPolynomial[{Coefficient[(2 * 1 - 1)^2 * lamb2th1[dx], dx, 5],
  (2 *  $\frac{9}{10}$  - 1)^2 * Coefficient[lamb2th9[dx], dx, 5],
  (2 *  $\frac{8}{10}$  - 1)^2 * Coefficient[lamb2th8[dx], dx, 5],
  (2 *  $\frac{7}{10}$  - 1)^2 * Coefficient[lamb2th7[dx], dx, 5],
  (2 *  $\frac{6}{10}$  - 1)^2 * Coefficient[lamb2th6[dx], dx, 5]}, x] /. x -> 11 - 10  $\theta$  // Simplify
 $\frac{1}{270} i k^5 (-1 - 6 \theta + 6 \theta^2)$ 

```

$$\text{physrel4}[\theta] = -\frac{1}{72} \frac{1}{2\theta - 1};$$

$$\text{physrel5}[\theta] = -\frac{1}{270} \frac{1 + 6\theta - 6\theta^2}{(1 - 2\theta)^2};$$

So phys.rel. eigenvalue given by $\lambda = -i h k - \frac{k^4}{72} + \frac{1}{270} i k^5 (-1 - 6\theta + 6\theta^2)$

Aside: Fully discrete RK-DG upwind scheme

Note that information on the fully-discrete (completely decoupled) RK-DG scheme is held entirely within the semi-discrete form in the sense that one can rewrite RK schemes in a similar form to Taylor series, as briefly discussed in my thesis. For the SSPRK(3,3) scheme then, the physically relevant eigenvalue is a combination of eigenvalue of semi-discrete operator D^- :

```

Normal[Series[1 + dt * Evaluate1Dm +  $\frac{dt^2}{2}$  * Evaluate1Dm^2 +  $\frac{dt^3}{6}$  * Evaluate1Dm^3 // Expand,
  {dx, 0, 4}]] // Expand

```

$$1 - i dt \omega - \frac{dt^2 \omega^2}{2} + \frac{1}{6} i dt^3 \omega^3 - \frac{1}{72} dt dx^3 \omega^4 + \frac{1}{72} i dt^2 dx^3 \omega^5 - \frac{1}{270} i dt dx^4 \omega^5 + \frac{1}{144} dt^3 dx^3 \omega^6 - \frac{1}{270} dt^2 dx^4 \omega^6 + \frac{1}{540} i dt^3 dx^4 \omega^7$$

```

Normal[Series[Exp[-i *  $\omega$  * dt], {dt, 0, 3}]]

```

$$1 - i dt \omega - \frac{dt^2 \omega^2}{2} + \frac{1}{6} i dt^3 \omega^3$$

Error terms:

```
%116 - %119
```

$$-\frac{1}{72} dt dx^3 \omega^4 + \frac{1}{72} i dt^2 dx^3 \omega^5 - \frac{1}{270} i dt dx^4 \omega^5 + \frac{1}{144} dt^3 dx^3 \omega^6 - \frac{1}{270} dt^2 dx^4 \omega^6 + \frac{1}{540} i dt^3 dx^4 \omega^7$$

A.2 Fully-discrete example

Appendix 2: Example *Mathematica* code: fully-discrete scheme

- Original Lax-Wendroff discontinuous Galerkin:
 P^2 polynomial basis

```
In[3]:= Clear["Global`*"]
```

General Setup

Orthonormal Legendre polynomials

```
In[121]:= f1[x_] = Sqrt[1/2] * LegendreP[0, x]
```

```
f2[x_] = Sqrt[3/2] * LegendreP[1, x]
```

```
f3[x_] = Sqrt[5/2] * LegendreP[2, x]
```

```
Out[121]= 1/√2
```

```
Out[122]= √(3/2) x
```

```
Out[123]= 1/2 √(5/2) (-1 + 3 x^2)
```

M is mass matrix, DM is stiffness matrix, DMt is transpose stiffness matrix for direct differentiation of approximation (for Lax-Wendroff type methods), DMt2 for second derivative. The other matrices represent boundary contributions from the flux (e.g. LLM=Left boundary from the left). All matrices scaled to canonical element [-1,1].

```
In[124]:= F[x_] = {f1[x], f2[x], f3[x]} // Simplify;
M = Table[a, {3}, {3}];
DM = Table[a, {3}, {3}];
DMt = Table[a, {3}, {3}];
D2Mt = Table[a, {3}, {3}];
RLM = Table[a, {3}, {3}];
RRM = Table[a, {3}, {3}];
LRM = Table[a, {3}, {3}];
LLM = Table[a, {3}, {3}];
R1LM = Table[a, {3}, {3}];
R1RM = Table[a, {3}, {3}];
L1RM = Table[a, {3}, {3}];
L1LM = Table[a, {3}, {3}];
R2LM = Table[a, {3}, {3}];
R2RM = Table[a, {3}, {3}];
L2RM = Table[a, {3}, {3}];
L2LM = Table[a, {3}, {3}];

In[131]:= GG = Module[
  { k, j },
  For[ k = 1, k ≤ 3, k++,
    For[ j = 1, j ≤ 3, j++, M[[k, j]] = Integrate[F[x][[k]] * F[x][[j]], {x, -1, 1}]]];

  For[ k = 1, k ≤ 3, k++,
    For[ j = 1, j ≤ 3, j++,
      DM[[k, j]] = Integrate[D[F[x][[k]], x] * F[x][[j]], {x, -1, 1}]]];
  For[ k = 1, k ≤ 3, k++,
    For[ j = 1, j ≤ 3, j++,
      DMt[[k, j]] = Integrate[D[F[x][[j]], x] * F[x][[k]], {x, -1, 1}]]];
  For[ k = 1, k ≤ 3, k++,
    For[ j = 1, j ≤ 3, j++,
      D2Mt[[k, j]] = Integrate[D[D[F[x][[j]], x], x] * F[x][[k]], {x, -1, 1}]]];

  For[ k = 1, k ≤ 3, k++,
    For[ j = 1, j ≤ 3, j++, LLM[[k, j]] = F[-1][[k]] * F[1][[j]]];

  For[ k = 1, k ≤ 3, k++,
    For[ j = 1, j ≤ 3, j++, LRM[[k, j]] = F[-1][[k]] * F[-1][[j]]];
```

```

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++, RLM[[k, j]] = F[1][[k]] * F[1][[j]]];

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++, RRM[[k, j]] = F[1][[k]] * F[-1][[j]]];

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++, L1LM[[k, j]] = F[-1][[k]] * (D[F[x][[j]], x] /. x → 1)];

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++, L1RM[[k, j]] = F[-1][[k]] * (D[F[x][[j]], x] /. x → -1)];

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++, R1LM[[k, j]] = F[1][[k]] * (D[F[x][[j]], x] /. x → 1)];

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++, R1RM[[k, j]] = F[1][[k]] * (D[F[x][[j]], x] /. x → -1)];

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++,
    L2LM[[k, j]] = F[-1][[k]] * (D[D[F[x][[j]], x], x] /. x → 1)];

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++,
    L2RM[[k, j]] = F[-1][[k]] * (D[D[F[x][[j]], x], x] /. x → -1)];

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++,
    R2LM[[k, j]] = F[1][[k]] * (D[D[F[x][[j]], x], x] /. x → 1)];

For[ k = 1, k ≤ 3, k++,
  For[ j = 1, j ≤ 3, j++,
    R2RM[[k, j]] = F[1][[k]] * (D[D[F[x][[j]], x], x] /. x → -1)];
]

```

A1 is contribution to cell I_j resulting from an upwind flux u^- ; A2 is contribution to cell I_j resulting from a downwind flux u^+ ; B applies to cell I_{j-1} and results from u^- flux; C to I_{j+1} due to u^+ flux.

```

In[132]= Minv = Inverse[M];
A1 = Simplify[Expand[(DM - RLM)]];
A2 = Simplify[Expand[(DM + LRM)]];
A = A1 + A2;
Cp = Simplify[Expand[RRM]];
Bm = Simplify[Expand[LLM]];
Simplify[Expand[DM]];
Simplify[Expand[DMt]];

```

Construction of fully-discrete amplification matrix G. Derive by hand

```

In[148]= Dmat = Simplify[Expand[DMt - v * D2Mt]];
Gplus = Simplify[Expand[LRM - 1/2 * (v * L1RM - 2/3 v^2 * L2RM)]];
Gminus = Simplify[Expand[-LLM + 1/2 * (v * L1LM - 2/3 v^2 * L2LM)]];
Eplus = Simplify[Expand[1/2 * (v * R1RM - 2/3 v^2 * R2RM)]];
Eminus = Simplify[Expand[-1/2 * (v * R1LM - 2/3 v^2 * R2LM)]];

```

```

In[145]= A0 = Simplify[Expand[Dmat + Gplus - Eminus]]
Ap1 = -Eplus
Am1 = Gminus

```

```

Out[145]= {{1/2, sqrt(3)/2, 1/2 sqrt(5) (1 - 3 v)},
{-sqrt(3)/2, 3(1+v)/2, 1/2 sqrt(15) (1 - 2 v^2)}, {sqrt(5)/2, -sqrt(15)/2, 5/2 (1 + 3 v)}}

```

```

Out[146]= {{0, -sqrt(3)v/4, 1/4 sqrt(5)v (3 + 2 v)},
{0, -3v/4, 1/4 sqrt(15)v (3 + 2 v)}, {0, -sqrt(15)v/4, 5/4 v (3 + 2 v)}}

```

```

Out[147]= {{-1/2, 1/4 sqrt(3) (-2 + v), -1/4 sqrt(5) (2 - 3 v + 2 v^2)},
{sqrt(3)/2, -3/4 (-2 + v), 1/4 sqrt(15) (2 - 3 v + 2 v^2)},
{-sqrt(5)/2, 1/4 sqrt(15) (-2 + v), -5/4 (2 - 3 v + 2 v^2)}}

```

The following is the amplification matrix

```
In[157]:= Gold = Expand[A0 + Am1 * Exp[-I * k * dx] + Ap1 * Exp[I * k * dx]] // Simplify;
eigLWo = Eigenvalues[Gold] // Simplify
eigLWoR = ToRadicals[eigLWo];
```

$$\text{Out[158]} = \left\{ \frac{1}{4} e^{-i dx k} \text{Root}\left[480 e^{2 i dx k} - 480 e^{3 i dx k} + 720 e^{2 i dx k} \nu - 720 e^{4 i dx k} \nu + 240 e^{2 i dx k} \nu^2 - 480 e^{3 i dx k} \nu^2 + 240 e^{4 i dx k} \nu^2 + \left(96 e^{i dx k} + 144 e^{2 i dx k} - 72 e^{i dx k} \nu + 624 e^{2 i dx k} \nu + 168 e^{3 i dx k} \nu - 120 e^{i dx k} \nu^2 + 120 e^{3 i dx k} \nu^2\right) \#1 + \left(6 - 18 e^{i dx k} - 12 \nu - 36 e^{i dx k} \nu - 12 e^{2 i dx k} \nu + 10 \nu^2 - 10 e^{2 i dx k} \nu^2\right) \#1^2 + \#1^3 \&, 1\right], \frac{1}{4} e^{-i dx k} \text{Root}\left[480 e^{2 i dx k} - 480 e^{3 i dx k} + 720 e^{2 i dx k} \nu - 720 e^{4 i dx k} \nu + 240 e^{2 i dx k} \nu^2 - 480 e^{3 i dx k} \nu^2 + 240 e^{4 i dx k} \nu^2 + \left(96 e^{i dx k} + 144 e^{2 i dx k} - 72 e^{i dx k} \nu + 624 e^{2 i dx k} \nu + 168 e^{3 i dx k} \nu - 120 e^{i dx k} \nu^2 + 120 e^{3 i dx k} \nu^2\right) \#1 + \left(6 - 18 e^{i dx k} - 12 \nu - 36 e^{i dx k} \nu - 12 e^{2 i dx k} \nu + 10 \nu^2 - 10 e^{2 i dx k} \nu^2\right) \#1^2 + \#1^3 \&, 2\right], \frac{1}{4} e^{-i dx k} \text{Root}\left[480 e^{2 i dx k} - 480 e^{3 i dx k} + 720 e^{2 i dx k} \nu - 720 e^{4 i dx k} \nu + 240 e^{2 i dx k} \nu^2 - 480 e^{3 i dx k} \nu^2 + 240 e^{4 i dx k} \nu^2 + \left(96 e^{i dx k} + 144 e^{2 i dx k} - 72 e^{i dx k} \nu + 624 e^{2 i dx k} \nu + 168 e^{3 i dx k} \nu - 120 e^{i dx k} \nu^2 + 120 e^{3 i dx k} \nu^2\right) \#1 + \left(6 - 18 e^{i dx k} - 12 \nu - 36 e^{i dx k} \nu - 12 e^{2 i dx k} \nu + 10 \nu^2 - 10 e^{2 i dx k} \nu^2\right) \#1^2 + \#1^3 \&, 3\right] \right\}$$

The following may take a while to run

```
eigLWoDS = eigLWoR[[1]] /. \nu \to \frac{dt}{dx} /. k \to \frac{\kappa}{dx} /. dt \to \nu * dx // Simplify;
eigLWoDS2 = eigLWoR[[2]] /. \beta \to \frac{dt}{dx} /. k \to \frac{\kappa}{dx} /. dt \to \nu * dx // Simplify;
eigLWoDS3 = eigLWoR[[3]] /. \beta \to \frac{dt}{dx} /. k \to \frac{\kappa}{dx} /. dt \to \nu * dx // Simplify;
```

It is good to save output not too large to save recomputing it:

$$\begin{aligned}
\text{In[160]:= eigLWoDS} &= \frac{1}{6} e^{-ix} \left(-3 + 6v - 5v^2 + 9e^{ix} (1 + 2v) + e^{2ix} v (6 + 5v) + \right. \\
&\left. \left(e^{4ix} v^2 (6 + 5v)^2 + (3 - 6v + 5v^2)^2 + 18e^{3ix} v (-1 + 12v + 10v^2) - \right. \right. \\
&\quad \left. \left. 18e^{ix} (7 - 3v - 12v^2 + 10v^3) - e^{2ix} (27 + 180v - 366v^2 + 50v^4) \right) \right) / \\
&\left(-27 + 567e^{ix} - 2025e^{2ix} + 81e^{3ix} + 162v - 1377e^{ix}v - 2592e^{2ix}v - 5913e^{3ix}v - \right. \\
&\quad 459v^2 + 459e^{ix}v^2 + 54e^{2ix}v^2 - 4671e^{3ix}v^2 - 3321e^{4ix}v^2 - 162e^{5ix}v^2 + \\
&\quad 756v^3 + 2349e^{ix}v^3 + 7560e^{2ix}v^3 + 9450e^{3ix}v^3 + 4860e^{4ix}v^3 + 1809e^{5ix}v^3 + \\
&\quad 216e^{6ix}v^3 - 765v^4 - 3240e^{ix}v^4 - 4950e^{2ix}v^4 + 5175e^{4ix}v^4 + 3240e^{5ix}v^4 + \\
&\quad 540e^{6ix}v^4 + 450v^5 + 1350e^{ix}v^5 - 450e^{2ix}v^5 - 2700e^{3ix}v^5 - 450e^{4ix}v^5 + \\
&\quad 1350e^{5ix}v^5 + 450e^{6ix}v^5 - 125v^6 + 375e^{2ix}v^6 - 375e^{4ix}v^6 + 125e^{6ix}v^6 + \\
&\quad \frac{1}{8} \sqrt{\left(64 \left(e^{6ix}v^3 (6 + 5v)^3 - (3 - 6v + 5v^2)^3 + 27e^{5ix}v^2 (-6 + 67v + 120v^2 + \right. \right. \\
&\quad \quad \left. \left. 50v^3) - 3e^{4ix}v^2 (1107 - 1620v - 1725v^2 + 150v^3 + 125v^4) + \right. \right. \\
&\quad \quad \left. \left. 27e^{ix} (21 - 51v + 17v^2 + 87v^3 - 120v^4 + 50v^5) - \right. \right. \\
&\quad \quad \left. \left. 27e^{3ix} (-3 + 219v + 173v^2 - 350v^3 + 100v^5) + \right. \right. \\
&\quad \quad \left. \left. 3e^{2ix} (-675 - 864v + 18v^2 + 2520v^3 - 1650v^4 - 150v^5 + 125v^6) \right)^2 + \right. \\
&\quad \left. \left(-4 (-3 + 6v - 5v^2 + 9e^{ix} (1 + 2v) + e^{2ix} v (6 + 5v))^2 + \right. \right. \\
&\quad \quad \left. \left. 72e^{ix} (4 - 3v - 5v^2 + e^{2ix} v (7 + 5v) + e^{ix} (6 + 26v)) \right)^3 \right)^{1/3} + \\
&\left(-27 + 567e^{ix} - 2025e^{2ix} + 81e^{3ix} + 162v - 1377e^{ix}v - 2592e^{2ix}v - \right. \\
&\quad 5913e^{3ix}v - 459v^2 + 459e^{ix}v^2 + 54e^{2ix}v^2 - 4671e^{3ix}v^2 - 3321e^{4ix}v^2 - \\
&\quad 162e^{5ix}v^2 + 756v^3 + 2349e^{ix}v^3 + 7560e^{2ix}v^3 + 9450e^{3ix}v^3 + \\
&\quad 4860e^{4ix}v^3 + 1809e^{5ix}v^3 + 216e^{6ix}v^3 - 765v^4 - 3240e^{ix}v^4 - \\
&\quad 4950e^{2ix}v^4 + 5175e^{4ix}v^4 + 3240e^{5ix}v^4 + 540e^{6ix}v^4 + 450v^5 + \\
&\quad 1350e^{ix}v^5 - 450e^{2ix}v^5 - 2700e^{3ix}v^5 - 450e^{4ix}v^5 + 1350e^{5ix}v^5 + \\
&\quad 450e^{6ix}v^5 - 125v^6 + 375e^{2ix}v^6 - 375e^{4ix}v^6 + 125e^{6ix}v^6 + \\
&\quad \frac{1}{8} \sqrt{\left(64 \left(e^{6ix}v^3 (6 + 5v)^3 - (3 - 6v + 5v^2)^3 + 27e^{5ix}v^2 (-6 + 67v + 120v^2 + \right. \right. \\
&\quad \quad \left. \left. 50v^3) - 3e^{4ix}v^2 (1107 - 1620v - 1725v^2 + 150v^3 + 125v^4) + \right. \right. \\
&\quad \quad \left. \left. 27e^{ix} (21 - 51v + 17v^2 + 87v^3 - 120v^4 + 50v^5) - \right. \right. \\
&\quad \quad \left. \left. 27e^{3ix} (-3 + 219v + 173v^2 - 350v^3 + 100v^5) + \right. \right. \\
&\quad \quad \left. \left. 3e^{2ix} (-675 - 864v + 18v^2 + 2520v^3 - 1650v^4 - 150v^5 + 125v^6) \right)^2 + \right. \\
&\quad \left. \left(-4 (-3 + 6v - 5v^2 + 9e^{ix} (1 + 2v) + e^{2ix} v (6 + 5v))^2 + \right. \right. \\
&\quad \quad \left. \left. 72e^{ix} (4 - 3v - 5v^2 + e^{2ix} v (7 + 5v) + e^{ix} (6 + 26v)) \right)^3 \right)^{1/3} \Big);
\end{aligned}$$

```
In[164]:= temp1[κ_] = Expand[Normal[Series[eigLWoDS, {κ, 0, 8}]]]
```

Out[164]=

$$\begin{aligned}
 & 1 + \frac{i\kappa}{2} + \frac{\kappa^2}{4} - \frac{i\kappa^3}{12} - \frac{\kappa^4}{48} + \frac{i\kappa^5}{240} + \dots 26061 \dots + \\
 & \frac{1}{36} i \kappa^3 \left(-1404 - 9720 \nu - 8100 \nu^2 + 27000 \nu^3 + 540 \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)} \right)^{1/3} + \\
 & \frac{1}{144} \kappa^4 \left(-1404 - 9720 \nu - 8100 \nu^2 + 27000 \nu^3 + 540 \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)} \right)^{1/3} - \\
 & \frac{1}{720} i \kappa^5 \\
 & \left(-1404 - 9720 \nu - 8100 \nu^2 + 27000 \nu^3 + 540 \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)} \right)^{1/3} - \\
 & \frac{\kappa^6 \left(-1404 - 9720 \nu - 8100 \nu^2 + 27000 \nu^3 + 540 \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)} \right)^{1/3}}{4320} + \\
 & \frac{i \kappa^7 \left(-1404 - 9720 \nu - 8100 \nu^2 + 27000 \nu^3 + 540 \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)} \right)^{1/3}}{30240} + \\
 & \frac{\kappa^8 \left(-1404 - 9720 \nu - 8100 \nu^2 + 27000 \nu^3 + 540 \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)} \right)^{1/3}}{241920}
 \end{aligned}$$

large output show less show more show all set size limit...

Non-trivial identity

This term features heavily in the Taylor expansion of the eigenvalues. It needs to be simplified manually.

$$\begin{aligned}
 & \left(-1404 - 9720 \nu - 8100 \nu^2 + 27000 \nu^3 + 540 \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)} \right)^{1/3} // \text{Simplify} \\
 & 3 \times 2^{2/3} \left(-13 - 90 \nu - 75 \nu^2 + 250 \nu^3 + 5 \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)} \right)^{1/3}
 \end{aligned}$$

Define magic quantity (found by wine, trial and error!)

```
In[162]:= iden = 3 (-(5 ν + 1) + √(17 + 30 ν - 75 ν²));
```

Check it equals original quantity:

```
iden³ // Expand
```

$$-1404 - 9720 \nu - 8100 \nu^2 + 27000 \nu^3 + 540 \sqrt{17 + 30 \nu - 75 \nu^2} + 1620 \nu \sqrt{17 + 30 \nu - 75 \nu^2}$$

```
(iden / (3 × 2^{2/3}))³ // Expand
```

$$-13 - 90 \nu - 75 \nu^2 + 250 \nu^3 + 5 \sqrt{17 + 30 \nu - 75 \nu^2} + 15 \nu \sqrt{17 + 30 \nu - 75 \nu^2}$$

Messy simplification processes:

I had some difficulties automating this

$$\text{lamb11wo} = \text{temp1}[\kappa] /. \left(-1404 - 9720 \nu - 8100 \nu^2 + 27000 \nu^3 + 540 \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)} \right)^{1/3} \rightarrow$$

$$\begin{aligned}
& \text{idem} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{1/3} \rightarrow \\
& \frac{1}{\text{idem}} /. \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{2/3} \rightarrow \\
& \text{idem}^2 /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 \right. \\
& \quad \left. v^3 + 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{2/3} \rightarrow \\
& \frac{1}{\text{idem}^2} /. \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{4/3} \rightarrow \text{idem}^4 /. \\
& 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + 540 \right. \\
& \quad \left. \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{4/3} \rightarrow \\
& \frac{1}{\text{idem}^4} /. \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{5/3} \rightarrow \text{idem}^5 /. \\
& 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + 540 \right. \\
& \quad \left. \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{5/3} \rightarrow \\
& \frac{1}{\text{idem}^5} /. \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{7/3} \rightarrow \\
& \text{idem}^7 /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{7/3} \rightarrow \\
& \frac{1}{\text{idem}^7} /. \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{8/3} \rightarrow \\
& \text{idem}^8 /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{8/3} \rightarrow \\
& \frac{1}{\text{idem}^8} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{10/3} \rightarrow \\
& \frac{1}{\text{idem}^{10}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
& \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{11/3} \rightarrow
\end{aligned}$$

$$\begin{aligned}
 & \frac{1}{\text{idn}^{11}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{13/3} \rightarrow \\
 & \frac{1}{\text{idn}^{13}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{14/3} \rightarrow \\
 & \frac{1}{\text{idn}^{14}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{16/3} \rightarrow \\
 & \frac{1}{\text{idn}^{16}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{17/3} \rightarrow \\
 & \frac{1}{\text{idn}^{17}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{18/3} \rightarrow \\
 & \frac{1}{\text{idn}^{18}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{19/3} \rightarrow \\
 & \frac{1}{\text{idn}^{19}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{20/3} \rightarrow \\
 & \frac{1}{\text{idn}^{20}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + 540 \right. \\
 & \quad \left. \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{22/3} \rightarrow \\
 & \frac{1}{\text{idn}^{22}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{23/3} \rightarrow \\
 & \frac{1}{\text{idn}^{23}} /. 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{25/3} \rightarrow \frac{1}{\text{idn}^{25}} /. \\
 & 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{26/3} \rightarrow \\
 & \frac{1}{\text{idn}^{26}} /. \\
 & 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + \right. \\
 & \quad \left. 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{28/3} \rightarrow \frac{1}{\text{idn}^{28}} /. \\
 & 1 / \left(-1404 - 9720 v - 8100 v^2 + 27000 v^3 + 540 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right)^{29/3} \rightarrow
 \end{aligned}$$

```

1
---- //
iden29
Expand;

```

Out[163]=

$$\frac{\frac{1}{2} + i \kappa + \dots 26079 \dots + \frac{6103515625 i \kappa^7 \nu^{15} \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)}}{36 \times 2^{1/3} (1+3\nu) (-17-30\nu+75\nu^2)^3 \left(-13-90\nu-75\nu^2+250\nu^3+5\sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)}\right)^{8/3}}{1220703125 \kappa^8 \nu^{15} \sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)}}{36 \times 2^{1/3} (1+3\nu) (-17-30\nu+75\nu^2)^3 \left(-13-90\nu-75\nu^2+250\nu^3+5\sqrt{-(1+3\nu)^2 (-17-30\nu+75\nu^2)}\right)^{8/3}}$$

large output | show less | show more | show all | set size limit...

```

lamb11w0 =

```

$$\begin{aligned}
 & \text{lamb11w0} /. \frac{1}{\left(-13 - 90 \nu - 75 \nu^2 + 250 \nu^3 + 5 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)}\right)^{8/3}} \rightarrow \\
 & \frac{1}{\left(\frac{\text{iden}}{3 \times 2^{2/3}}\right)^8} /. \\
 & \frac{1}{\left(-13 - 90 \nu - 75 \nu^2 + 250 \nu^3 + 5 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)}\right)^{10/3}} \rightarrow \\
 & \frac{1}{\left(\frac{\text{iden}}{3 \times 2^{2/3}}\right)^{10}} /. \\
 & \frac{1}{\left(-13 - 90 \nu - 75 \nu^2 + 250 \nu^3 + 5 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)}\right)^{11/3}} \rightarrow \\
 & \frac{1}{\left(\frac{\text{iden}}{3 \times 2^{2/3}}\right)^{11}} /. \\
 & \frac{1}{\left(-13 - 90 \nu - 75 \nu^2 + 250 \nu^3 + 5 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)}\right)^{13/3}} \rightarrow \\
 & \frac{1}{\left(\frac{\text{iden}}{3 \times 2^{2/3}}\right)^{13}} /. \\
 & \frac{1}{\left(-13 - 90 \nu - 75 \nu^2 + 250 \nu^3 + 5 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)}\right)^{14/3}} \rightarrow \\
 & \frac{1}{\left(\frac{\text{iden}}{3 \times 2^{2/3}}\right)^{14}} /. \\
 & \frac{1}{\left(-13 - 90 \nu - 75 \nu^2 + 250 \nu^3 + 5 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)}\right)^{16/3}} \rightarrow
 \end{aligned}$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{16}} / .$$

$$\frac{1}{\left(-13 - 90 v - 75 v^2 + 250 v^3 + 5 \sqrt{-(1+3v)^2 (-17 - 30v + 75v^2)}\right)^{17/3}} \rightarrow$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{17}} / .$$

$$\frac{1}{\left(-13 - 90 v - 75 v^2 + 250 v^3 + 5 \sqrt{-(1+3v)^2 (-17 - 30v + 75v^2)}\right)^{19/3}} \rightarrow$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{19}} / .$$

$$\frac{1}{\left(-13 - 90 v - 75 v^2 + 250 v^3 + 5 \sqrt{-(1+3v)^2 (-17 - 30v + 75v^2)}\right)^{20/3}} \rightarrow$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{20}} / .$$

$$\frac{1}{\left(-13 - 90 v - 75 v^2 + 250 v^3 + 5 \sqrt{-(1+3v)^2 (-17 - 30v + 75v^2)}\right)^{22/3}} \rightarrow$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{22}} / .$$

$$\frac{1}{\left(-13 - 90 v - 75 v^2 + 250 v^3 + 5 \sqrt{-(1+3v)^2 (-17 - 30v + 75v^2)}\right)^{23/3}} \rightarrow$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{23}} / .$$

$$\frac{1}{\left(-13 - 90 v - 75 v^2 + 250 v^3 + 5 \sqrt{-(1+3v)^2 (-17 - 30v + 75v^2)}\right)^{25/3}} \rightarrow$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{25}} / .$$

$$\frac{1}{\left(-13 - 90 v - 75 v^2 + 250 v^3 + 5 \sqrt{-(1+3v)^2 (-17 - 30v + 75v^2)}\right)^{26/3}} \rightarrow$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{26}} / .$$

$$\frac{1}{\left(-13 - 90 v - 75 v^2 + 250 v^3 + 5 \sqrt{-(1+3v)^2 (-17 - 30v + 75v^2)}\right)^{28/3}} \rightarrow$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{28}} / .$$

$$\frac{1}{\left(-13 - 90 \nu - 75 \nu^2 + 250 \nu^3 + 5 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)}\right)^{29/3}} \rightarrow$$

$$\frac{1}{\left(\frac{\text{idem}}{3 \times 2^{2/3}}\right)^{29}} // \text{Expand};$$

In[165]=

```
lwo2 = Collect[lamb1lwo, κ];
```

Now inspect coefficients and find closed forms

Various methods for isolating quantities help to realise major simplifications

Coeff constants

```
Coefficient[lwo2, κ, 0]
```

```
Coefficient[lwo2, κ, 0] // Simplify
```

$$\frac{1}{2} + \frac{5 \nu}{2} + \frac{1}{2} \sqrt{17 + 30 \nu - 75 \nu^2} - \frac{8}{-1 - 5 \nu + \sqrt{17 + 30 \nu - 75 \nu^2}} -$$

$$\frac{10 \nu}{-1 - 5 \nu + \sqrt{17 + 30 \nu - 75 \nu^2}} + \frac{50 \nu^2}{-1 - 5 \nu + \sqrt{17 + 30 \nu - 75 \nu^2}}$$

0

Coeff κ

`simpnu = Coefficient[lwo2, κ, 1] // Simplify`

$$\begin{aligned} & \left(4 i \left(-5457 - 843750 v^8 + 3417 \sqrt{17 + 30 v - 75 v^2} - 2958 \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} + \right. \right. \\ & \quad 174 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} + \\ & \quad 168750 v^7 \left(-25 + \sqrt{17 + 30 v - 75 v^2} \right) + \\ & \quad 1875 v^6 \left(77 + 432 \sqrt{17 + 30 v - 75 v^2} - 30 \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} \right) - \\ & \quad 375 v^5 \left(-8334 + 779 \sqrt{17 + 30 v - 75 v^2} + 595 \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} + \right. \\ & \quad \quad \left. 10 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} \right) - \\ & \quad 5 v^3 \left(94562 + 5048 \sqrt{17 + 30 v - 75 v^2} - 23300 \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} + \right. \\ & \quad \quad \left. 95 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} \right) + \\ & \quad 3 v \left(-24222 + 12397 \sqrt{17 + 30 v - 75 v^2} - 7605 \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} + \right. \\ & \quad \quad \left. 635 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} \right) - \\ & \quad 25 v^4 \left(-40043 + 22935 \sqrt{17 + 30 v - 75 v^2} - 6120 \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} + \right. \\ & \quad \quad \left. 685 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} \right) + \\ & \quad 5 v^2 \left(-67991 + 23198 \sqrt{17 + 30 v - 75 v^2} - 5504 \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} + \right. \\ & \quad \quad \left. 1147 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1 + 3 v)^2 (-17 - 30 v + 75 v^2)} \right) \Big) \Big) \Big) \Big) / \\ & \left(3 (1 + 3 v) (-17 - 30 v + 75 v^2) (-1 - 5 v + \sqrt{17 + 30 v - 75 v^2})^4 \right) \end{aligned}$$

Simplify by numerical interpolation:

```
InterpolatingPolynomial[{Rationalize[N[simpnu /. v -> 1/100, 64]],
  Rationalize[N[simpnu /. v -> 2/100, 64]],
  Rationalize[N[simpnu /. v -> 3/100, 64]],
  Rationalize[N[simpnu /. v -> 4/100, 64]],
  Rationalize[N[simpnu /. v -> 5/100, 64]]}, x] /. x -> 100 v // Simplify
```

$$\frac{i}{2}$$

```
-i v κ /. v -> dt/dx /. κ -> k dx // Simplify
```

$$-i dt k$$

Coeff κ²

`simpnu2 = Coefficient[lwo2, κ, 2] // Simplify`

$$\begin{aligned}
& \left(64 \left(-5\,932\,617\,187\,500 v^{17} + 131\,835\,937\,500 v^{16} \left(-256 + 3 \sqrt{17 + 30 v - 75 v^2} \right) + \right. \right. \\
& \quad 2197\,265\,625 v^{15} \left(-27\,305 + 5172 \sqrt{17 + 30 v - 75 v^2} - \right. \\
& \quad \quad \left. 60 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) - 1\,318\,359\,375 v^{14} \\
& \quad \left(-39\,660 - 11\,399 \sqrt{17 + 30 v - 75 v^2} + 2690 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \right. \\
& \quad \quad \left. 20 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) - 29\,296\,875 v^{13} \\
& \quad \left(-5\,266\,345 + 1\,588\,629 \sqrt{17 + 30 v - 75 v^2} + 145\,935 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \right. \\
& \quad \quad \left. 4730 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) - 5\,859\,375 v^{12} \\
& \quad \left(922\,392 + 4\,122\,254 \sqrt{17 + 30 v - 75 v^2} - 2\,847\,740 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \right. \\
& \quad \quad \left. 45\,255 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) + \\
& \quad 1\,171\,875 v^{11} \left(-115\,387\,769 + 35\,804\,836 \sqrt{17 + 30 v - 75 v^2} + \right. \\
& \quad \quad 2\,662\,090 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \\
& \quad \quad \left. 81\,645 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) - 90 \left(-3\,650\,155 + \right. \\
& \quad \quad 1\,080\,248 \sqrt{17 + 30 v - 75 v^2} - 1\,080\,248 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \\
& \quad \quad \left. 214\,715 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) + \\
& \quad 15\,625 v^9 \left(3\,323\,904\,805 - 891\,935\,859 \sqrt{17 + 30 v - 75 v^2} - \right. \\
& \quad \quad 222\,724\,935 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \\
& \quad \quad \left. 6\,349\,165 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) + \\
& \quad 78\,125 v^{10} \left(-486\,944\,610 + 308\,984\,791 \sqrt{17 + 30 v - 75 v^2} - \right. \\
& \quad \quad 188\,410\,630 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \\
& \quad \quad \left. 7\,166\,085 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) - \\
& \quad 9 v \left(-873\,874\,991 + 246\,797\,801 \sqrt{17 + 30 v - 75 v^2} - \right. \\
& \quad \quad 212\,697\,399 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \\
& \quad \quad \left. 40\,757\,407 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) - \\
& \quad 6250 v^8 \left(-4\,676\,099\,575 + 1\,953\,305\,692 \sqrt{17 + 30 v - 75 v^2} - \right. \\
& \quad \quad 891\,121\,120 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \\
& \quad \quad \left. 56\,960\,210 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) + \\
& \quad 25 v^4 \left(23\,203\,302\,468 - 2\,807\,242\,378 \sqrt{17 + 30 v - 75 v^2} - \right. \\
& \quad \quad 2\,388\,686\,420 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \\
& \quad \quad \left. 102\,608\,835 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) - \\
& \quad 625 v^7 \left(9\,128\,736\,047 - 327\,305\,316 \sqrt{17 + 30 v - 75 v^2} - \right. \\
& \quad \quad 3\,623\,388\,240 \sqrt{-(1+3v)^2 (-17-30v+75v^2)} + \\
& \quad \quad \left. 235\,152\,230 \sqrt{17 + 30 v - 75 v^2} \sqrt{-(1+3v)^2 (-17-30v+75v^2)} \right) +
\end{aligned}$$

$$\begin{aligned}
 & 125 \nu^5 \left(-12\,729\,719\,335 + 5\,921\,889\,375 \sqrt{17 + 30 \nu - 75 \nu^2} - \right. \\
 & \quad 4\,663\,733\,319 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)} + \\
 & \quad \left. 478\,760\,720 \sqrt{17 + 30 \nu - 75 \nu^2} \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)} \right) - \\
 & 15 \nu^3 \left(-23\,557\,348\,261 + 5\,619\,964\,604 \sqrt{17 + 30 \nu - 75 \nu^2} - \right. \\
 & \quad 2\,579\,435\,570 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)} + \\
 & \quad \left. 495\,611\,135 \sqrt{17 + 30 \nu - 75 \nu^2} \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)} \right) + \\
 & 125 \nu^6 \left(-65\,420\,598\,320 + 20\,321\,790\,667 \sqrt{17 + 30 \nu - 75 \nu^2} - \right. \\
 & \quad 5\,932\,176\,010 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)} + \\
 & \quad \left. 644\,025\,610 \sqrt{17 + 30 \nu - 75 \nu^2} \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)} \right) - \\
 & 3 \nu^2 \left(-25\,311\,595\,154 + 6\,747\,221\,489 \sqrt{17 + 30 \nu - 75 \nu^2} - \right. \\
 & \quad 4\,717\,859\,930 \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)} + \\
 & \quad \left. 871\,605\,325 \sqrt{17 + 30 \nu - 75 \nu^2} \sqrt{-(1 + 3 \nu)^2 (-17 - 30 \nu + 75 \nu^2)} \right) \Big) \Big) \Big) / \\
 & \left(9 (1 + 3 \nu) (17 + 30 \nu - 75 \nu^2)^2 (-1 - 5 \nu + \sqrt{17 + 30 \nu - 75 \nu^2})^{10} \right)
 \end{aligned}$$

```

InterpolatingPolynomial[ {Rationalize[N[simpnu2 /. ν → 1/10, 64]],
Rationalize[N[simpnu2 /. ν → 2/10, 64]],
Rationalize[N[simpnu2 /. ν → 3/10, 64]],
Rationalize[N[simpnu2 /. ν → 4/10, 64]],
Rationalize[N[simpnu2 /. ν → 5/10, 64]],
Rationalize[N[simpnu2 /. ν → 6/10, 64]]}, x] /. x → 10 ν // Simplify
ν
4

```

Coeff κ³

```

simpnu3 = Coefficient[lwo2, κ, 3] // Simplify;

```

```

InterpolatingPolynomial[{Rationalize[N[simpnu3 /. v -> 1/10, 64]],
  Rationalize[N[simpnu3 /. v -> 2/10, 64]],
  Rationalize[N[simpnu3 /. v -> 3/10, 64]],
  Rationalize[N[simpnu3 /. v -> 4/10, 64]],
  Rationalize[N[simpnu3 /. v -> 5/10, 64]],
  Rationalize[N[simpnu3 /. v -> 6/10, 64]]}, x] /. x -> 10 v // Simplify
- i v^2
  12
- i v^2 /. v -> dt/dx /. x -> k dx // Simplify
- i dt^2
  12 dx^2

```

Coeff κ^4

Recall:

```
simpnu4 = Coefficient[lwo2, x, 4];
```

Seek to simplify by numerical interpolation again:

Extensive numerical investigations lead to discovery of rational factors.

```
cof4 =
```

```

InterpolatingPolynomial[{Rationalize[N[240 (1 + 3 v)/v simpnu4 /. v -> 1/5, 64]],
  Rationalize[N[240 (1 + 3 v)/v simpnu4 /. v -> 2/5, 64]],
  Rationalize[N[240 (1 + 3 v)/v simpnu4 /. v -> 3/5, 64]],
  Rationalize[N[240 (1 + 3 v)/v simpnu4 /. v -> 4/5, 64]]}, x] /. x -> 5 v // Simplify

```

```
1 + 2 v - 5 v^2 - 20 v^3
```

Coeff κ^5

```
simpnu5 = Coefficient[lwo2, x, 5] // Simplify;
```

cof5 =

```
InterpolatingPolynomial[{Rationalize[N[ $\frac{3600 (1 + 3 v)^2}{v}$  simpnu5 /. v ->  $\frac{1}{10}$ , 64]],
  Rationalize[N[ $\frac{3600 (1 + 3 v)^2}{v}$  simpnu5 /. v ->  $\frac{2}{10}$ , 64]],
  Rationalize[N[ $\frac{3600 (1 + 3 v)^2}{v}$  simpnu5 /. v ->  $\frac{3}{10}$ , 64]],
  Rationalize[N[ $\frac{3600 (1 + 3 v)^2}{v}$  simpnu5 /. v ->  $\frac{4}{10}$ , 64]],
  Rationalize[N[ $\frac{3600 (1 + 3 v)^2}{v}$  simpnu5 /. v ->  $\frac{5}{10}$ , 64]],
  Rationalize[N[ $\frac{3600 (1 + 3 v)^2}{v}$  simpnu5 /. v ->  $\frac{6}{10}$ , 64]]], x] /. x -> 10 v //
```

Simplify

i $(9 - 7 v - 75 v^2 - 5 v^3 + 150 v^4 + 300 v^5)$

Bibliography

- [1] M. Abramowitz, I.A. Stegun, et al. *Handbook of mathematical functions*, volume 1. Dover New York, 1972.
- [2] S. Adjerid, M. Aiffa, and J. Flaherty. High-order finite element methods for singularly-perturbed elliptic and parabolic problems. *SIAM Journal on Applied Mathematics*, 55:520–543, 1995.
- [3] S. Adjerid and M. Baccouch. The discontinuous Galerkin method for two-dimensional hyperbolic problems. Part I: Superconvergence error analysis. *Journal of Scientific Computing*, 33(1):75–113, 2007.
- [4] S. Adjerid and M. Baccouch. The discontinuous Galerkin method for two-dimensional hyperbolic problems part ii: A posteriori error estimation. *Journal of Scientific Computing*, 38(1):15–49, 2009.
- [5] S. Adjerid and M. Baccouch. A superconvergent local discontinuous Galerkin method for elliptic problems. *Journal of Scientific Computing*, 52(1):113–152, 2012.
- [6] S. Adjerid, K.D. Devine, J.E. Flaherty, and L. Krivodonova. A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems. *Computer methods in applied mechanics and engineering*, 191(11):1097–1112, 2002.
- [7] S. Adjerid and A. Klausner. Superconvergence of discontinuous finite element

- solutions for transient convection-diffusion problems. *Journal of Scientific Computing*, 22–23:5–24, 2005.
- [8] S. Adjerid and T.C. Massey. Superconvergence of discontinuous Galerkin solutions for a nonlinear scalar hyperbolic problem. *Computer Methods in Applied Mechanics and Engineering*, 195:3331–3346, 2006.
- [9] M. Ainsworth. Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. *Journal of Computational Physics*, 198(1):106–130, 2004.
- [10] K. Asthana and A. Jameson. High-order flux reconstruction schemes with minimal dispersion and dissipation. *Journal of Scientific Computing*, pages 1–32, 2014.
- [11] M. Baccouch. A superconvergent local discontinuous Galerkin method for the second-order wave equation on cartesian grids. *Computers and Mathematics with Applications*, 68:1250–1278, 2014.
- [12] R. Biswas, K.D. Devine, and J.E. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14(1):255–283, 1994.
- [13] J.H. Bramble and A.H. Schatz. Higher order local accuracy by averaging in the finite element method. *Mathematics of Computation*, 31(137):94–111, 1977.
- [14] S.C. Brenner and L.R. Scott. *The mathematical theory of finite element methods*. Springer-Verlag, New York, 1994.
- [15] W. Cao, D. Li, Y. Yang, and Q. Zou. Superconvergence of discontinuous Galerkin methods based on upwind-biased fluxes for 1D linear hyperbolic equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 51(2):467–486, 2017.
- [16] W. Cao, Z. Zhang, and Q. Zou. Superconvergence of discontinuous Galerkin

- methods for linear hyperbolic equations. *SIAM Journal on Numerical Analysis*, 52(5):2555–2573, 2014.
- [17] N. Chalmers, L. Krivodonova, and R. Qin. Relaxing the CFL number of the discontinuous Galerkin method. *SIAM Journal on Scientific Computing*, 36(4):A2047–A2075, 2014.
- [18] R.P.K. Chan and A.Y.J. Tsai. On explicit two-derivative Runge–Kutta methods. *Numerical Algorithms*, 53(2-3):171–194, 2010.
- [19] Y. Cheng, C.-S. Chou, F. Li, and Y. Xing. \mathcal{L}^2 stable discontinuous Galerkin methods for one-dimensional two-way wave equations. *Mathematics of Computation*, 86:121–155, 2017.
- [20] Y. Cheng, F. Li, J. Qiu, and L. Xu. Positivity-preserving DG and central DG methods for ideal MHD equations. *Journal of Computational Physics*, 238:255–280, 2013.
- [21] Y. Cheng and C.-W. Shu. Superconvergence and time evolution of discontinuous Galerkin finite element solutions. *Journal of Computational Physics*, 227(22):9612–9627, 2008.
- [22] P.G. Ciarlet. *The finite element method for elliptic problems*. Elsevier, 1978.
- [23] B. Cockburn. *An introduction to the discontinuous Galerkin method for convection-dominated problems*. Springer, 1998.
- [24] B. Cockburn, G.E. Karniadakis, and C.-W. Shu. *The development of discontinuous Galerkin methods*. Springer, 2000.
- [25] B. Cockburn, M. Luskin, C.-W. Shu, and E. Süli. Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. *Mathematics of Computation*, 72(242):577–606, 2003.
- [26] B. Cockburn and C.-W. Shu. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. II: General framework. *Mathematics of Computation*, 52(186):411–435, 1989.

- [27] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.
- [28] B. Cockburn and C.-W. Shu. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2002.
- [29] D. Gottlieb and Orszag S.A. *Numerical Analysis of Spectral Methods: Theory and Applications*. SIAM, 1977.
- [30] S. Gottlieb, D.I. Ketcheson, and C.-W. Shu. *Strong stability preserving Runge–Kutta and multistep time discretizations*. World Scientific, 2011.
- [31] W. Guo, J.-M. Qiu, and J. Qiu. A new Lax–Wendroff discontinuous Galerkin method with superconvergence. *Journal of Scientific Computing*, 65(1):299–326, 2015.
- [32] W. Guo, X. Zhong, and J.-M. Qiu. Superconvergence of discontinuous Galerkin and local discontinuous Galerkin methods: Eigen-structure analysis based on Fourier approach. *Journal of Computational Physics*, 235:458–485, 2013.
- [33] F.Q. Hu and H.L. Atkins. Eigensolution analysis of the discontinuous Galerkin method with nonuniform grids: I. One space dimension. *Journal of Computational Physics*, 182(2):516–545, 2002.
- [34] M.Y. Hu F.Q, Hussaini and P. Rasetarinera. An analysis of the discontinuous Galerkin method for wave propagation problems. *Journal of Computational Physics*, 151(2):921–946, 1999.
- [35] L. Ji and J. Ryan. Smoothness-Increasing Accuracy-Conserving (SIAC) filters in Fourier space. In: Kirby, R., Berzins, M., Hesthaven, J. (eds) *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014. Lecture Notes in Computational Science and Engineering*, 106(289):2239–2262, 2015.

-
- [36] L. Ji, P. van Slingerland, J. Ryan, and K. Vuik. Superconvergent error estimates for position-dependent Smoothness-Increasing Accuracy-Conserving (SIAC) post-processing of discontinuous Galerkin solutions. *Mathematics of Computation*, 83(289):2239–2262, 2014.
- [37] L. Ji, Y. Xu, and J.K. Ryan. Accuracy-enhancement of discontinuous Galerkin solutions for convection-diffusion equations in multiple-dimensions. *Mathematics of Computation*, 81(280):1929–1950, 2012.
- [38] L. Ji, Y. Xu, and J.K. Ryan. Negative-order norm estimates for nonlinear hyperbolic conservation laws. *Journal of Scientific Computing*, 54(2-3):531–548, 2013.
- [39] L. Krivodonova and R. Qin. An analysis of the spectrum of the discontinuous Galerkin method. *Applied Numerical Mathematics*, 64:1–18, 2013.
- [40] P. Lesaint and P.-A. Raviart. On a finite element method for solving the neutron transport equation. *Mathematical aspects of finite elements in partial differential equations*, (33):89–123, 1974.
- [41] R.J. LeVeque. *Numerical methods for conservation laws*, volume 132. Springer, 1992.
- [42] Y. Liu, C.-W. Shu, E. Tadmor, and M. Zhang. \mathcal{L}^2 stability analysis of the central discontinuous Galerkin method and a comparison between the central and regular discontinuous Galerkin methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(04):593–607, 2008.
- [43] R.B. Lowrie. *Compact higher-order numerical methods for hyperbolic conservation laws*. PhD thesis, University of Michigan, 1996.
- [44] X. Meng, C.-W. Shu, and B. Wu. Optimal error estimates for discontinuous Galerkin methods based on upwind-biased fluxes for linear hyperbolic equations. *Mathematics of Computation*, 85:1225–1261.
- [45] X. Meng, C.-W. Shu, Q. Zhang, and B. Wu. Superconvergence of discontinuous Galerkin methods for scalar nonlinear conservation laws in

- one space dimension. *SIAM Journal on Numerical Analysis*, 50(5):2336–2356, 2012.
- [46] J. Qiu, M. Dumbser, and C.-W. Shu. The discontinuous Galerkin method with Lax–Wendroff type time discretizations. *Computer methods in applied mechanics and engineering*, 194(42):4528–4543, 2005.
- [47] J. Qiu, M. Dumbser, and C.-W. Shu. The discontinuous Galerkin method with Lax–Wendroff type time discretizations. *Computer Methods in Applied Mechanics and Engineering*, 194(42):4528–4543, 2005.
- [48] W.H. Reed and T.R. Hill. Triangular-mesh methods for the neutron transport equation. *Los Alamos Report LA-UR-73-479*, 1973.
- [49] G.R. Richter. An optimal-order error estimate for the discontinuous Galerkin method. *Mathematics of Computation*, 50(181):75–88, 1988.
- [50] J.K. Ryan, C.-W. Shu, and H. Atkins. Extension of a post processing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem. *SIAM Journal on Scientific Computing*, 26(3):821–843, 2005.
- [51] D.C. Seal, Y. Güçlü, and A.J. Christlieb. High-order multiderivative time integrators for hyperbolic conservation laws. *Journal of Scientific Computing*, 60(1):101–140, 2014.
- [52] S. Sherwin. Dispersion analysis of the continuous and discontinuous Galerkin formulations. In *Discontinuous Galerkin Methods*, pages 425–431. Springer, 2000.
- [53] C.-W. Shu. A survey of strong stability preserving high order time discretizations. *Collected lectures on the preservation of stability under discretization*, 109:51–65, 2002.
- [54] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

-
- [55] Z. Sun and C.-W. Shu. Stability analysis and error estimates of Lax–Wendroff discontinuous Galerkin methods for linear conservation laws. *ESAIM: Mathematical Modelling and Numerical Analysis*, 51(3):1063–1087, 2017.
- [56] P. van Slingerland, J.K. Ryan, and C. Vuik. Position-dependent Smoothness-Increasing Accuracy-Conserving (SIAC) filtering for improving discontinuous Galerkin solutions. *SIAM Journal on Scientific Computing*, 33(2):802–825, 2011.
- [57] Z.J. Wang. Spectral finite volume method for conservation laws on unstructured grids: Basic formulation. *Journal of Computational Physics*, 178(1):210–251, 2002.
- [58] Y. Xia, Y. Xu, and C.-W. Shu. Efficient time discretization for local discontinuous Galerkin methods. *Discrete and Continuous Dynamical Systems Series B*, 8(3):677, 2007.
- [59] J. Yan and C.-W. Shu. Local discontinuous Galerkin methods for partial differential equations with higher order derivatives. *Journal of Scientific Computing*, 17(1-4):27–47, 2002.
- [60] H. Yang, F. Li, and J. Qiu. Dispersion and dissipation errors of two fully discrete discontinuous Galerkin methods. *Journal of Scientific Computing*, 55(3):552–574, 2013.
- [61] Y. Yang and C.-W. Shu. Analysis of optimal superconvergence of discontinuous Galerkin method for linear hyperbolic equations. *SIAM Journal on Numerical Analysis*, 50(6):3110–3133, 2012.
- [62] Y. Yang and C.-W. Shu. Discontinuous Galerkin method for hyperbolic equations involving δ -singularities: negative-order norm error estimates and applications. *Numerische Mathematik*, 124(4):753–781, 2013.
- [63] M. Zhang and C.-W. Shu. An analysis of three different formulations of the

- discontinuous Galerkin method for diffusion equations. *Mathematical Models and Methods in Applied Sciences*, 13(03):395–413, 2003.
- [64] M. Zhang and C.-W. Shu. An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods. *Computers & fluids*, 34(4):581–592, 2005.
- [65] Q. Zhang and C.-W. Shu. Error estimates to smooth solutions of Runge–Kutta discontinuous Galerkin methods for scalar conservation laws. *SIAM Journal on Numerical Analysis*, 42(2):641–666, 2004.
- [66] Q. Zhang and C.-W. Shu. Stability analysis and a priori error estimates of the third order explicit Runge–Kutta discontinuous Galerkin method for scalar conservation laws. *SIAM Journal of Numerical Analysis*, 48(3):1038–1063, 2010.
- [67] X. Zhang and C.-W. Shu. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918–8934, 2010.
- [68] X. Zhong and C.-W. Shu. Numerical resolution of discontinuous Galerkin methods for time dependent wave equations. *Computer Methods in Applied Mechanics and Engineering*, 200(41):2814–2827, 2011.