# Discontinuous Galerkin Methods for Extended Hydrodynamics

by

Yoshifumi Suzuki

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering and Scientific Computing)
in The University of Michigan
2008

Doctoral Committee:

Professor Bram van Leer, Chairperson
Professor Edward W. Larsen
Professor Kenneth G. Powell
Professor Philip L. Roe
Hung T. Huynh, NASA Glenn Research Center

To my parents and friends, who have been my support
throughout the ups and downs as a doctoral student.

# ACKNOWLEDGEMENTS

First and foremost, I would like to thank my chairperson, Professor Bram van Leer. He gave me an opportunity to explore the new frontier of CFD, and has been instructing me not only from his vast knowledge and abundant experience in the discipline, but also taught me how to think and approach problems. His energy, passion, and joy in research have impressed and motivated me since I started working with him on a class project almost six years ago. I will never forget his lesson: "Think outside the box!"

I would also like to thank Professor Philip Roe for his helpful guidance and insightful comments. He has been giving me observations from perspectives of which I had never thought. His physically-motivated explanations have always been intuitive and helpful.

For fruitful discussions and being on my dissertation committee, I would like to express my gratitude to Professor Kenneth Powell and Professor Edward Larsen. They shed light on current research in new directions. I am also grateful to Dr. Hung Huynh for being on my dissertation committee, and for his kindness and patience in answering the many questions I have asked about discontinuous Galerkin methods. I sill remember that his kind words made me relax at my very first conference presentation. His deep understanding of numerical methods and tangible explanations have helped me ahead.

Besides my dissertation committee there are two key persons who helped this dissertation to be completed. I am indebted to them for their ceaseless efforts to pave the way in the field ahead of me. Firstly, I would like to acknowledge Dr. Jeffrey Hittinger, who gave me an opportunity to visit Lawrence Livermore National Laboratory in the summer of 2004, and introduced me to the secrets of hyperbolic-relaxation equations. It is no exaggeration to say that this research had never taken off without Jeff's help. Those intensive two days of Fourier analysis with Jeff and Professor van Leer are a precious experience and memory. Secondly, I would like to thank Dr. Hiroaki Nishikawa, who has been giving me instrumental advice and generous support not only in research but also in private life from the beginning of my days in Ann Arbor. His commitment to research has been impressed and motivated me ever since I met him. For their frank and open discussion, and friendship, I would also like to acknowledge Dr. Farzad Ismail, Dr. Marc van Raalte, Loc Khieu, and Marcus Lo.

Outside of my life as a researcher, I have come across some great people. They, too, have been supportive throughout the ups and downs of my life as a doctoral student. Especially, I am grateful to Dr. Soshi Kawai, Sachiko Kawai, Dr. Hirotaka Saito, Kumiko Saito, and Dr. Hiroaki Fukuzawa for their support and friendship for many years. I would also like to show my gratitude to Catherine Alter, in whose home I have lived for more than three years. She has been treating me as a part of the family, and given me opportunities to experience real American life. I will never forget the taste of her special chocolate cake.

My doctoral study would never have been initiated without the opportunity to come to the University of Michigan as an exchange student in my first year. I thank Professor Tsutomu Nomizu, Yuko Ito, and staff of the Student Exchange Program

at Nagoya University for their effort and their financial support. I also acknowledge that this research has been funded by the U.S. Air Force Office of Scientific Research.

Last but not least, I gratefully acknowledge my parents with deepest appreciation for their unstinting support and understanding over the course of years.

# TABLE OF CONTENTS

CHAPTER

# LIST OF FIGURES

**Figure**

# LIST OF TABLES

**Table**

# LIST OF APPENDICES

**Appendix**

# ABSTRACT

This dissertation presents a step towards high-order methods for continuum-transition flows. In order to achieve maximum accuracy and efficiency for numerical methods on a distorted mesh, it is desirable that both governing equations and corresponding numerical methods are in some sense compact. We argue our preference for a physical model described solely by first-order partial differential equations called hyperbolic-relaxation equations, and, among various numerical methods, for the discontinuous Galerkin method. Hyperbolic-relaxation equations can be generated as moments of the Boltzmann equation and can describe continuum-transition flows.

Two challenging properties of hyperbolic-relaxation equations are the presence of a stiff source term, which drives the system towards equilibrium, and the accompanying change of eigenstructure. The first issue can be solved by an implicit treatment of the source term. To cope with the second difficulty, we develop a space-time discontinuous Galerkin method, based on Huynh's "upwind moment scheme." It is called the DG(1)–Hancock method.

The DG(1)–Hancock method for one- and two-dimensional meshes is described, and Fourier analyses for both linear advection and linear hyperbolic-relaxation equations are conducted. The analyses show that the DG(1)–Hancock method is not only accurate but efficient in terms of turnaround time in comparison to other semi- and fully discrete finite-volume and discontinuous Galerkin methods. Numerical tests confirm the analyses, and also show the properties are preserved for nonlinear equations; the efficiency is superior by an order of magnitude.

Subsequently, discontinuous Galerkin and finite-volume spatial discretizations are applied to more practical equations, in particular, to the set of 10-moment equations, which are gas dynamics equations that include a full pressure/temperature tensor among the flow variables. Results for flow around a micro-airfoil are compared to experimental data and to solutions obtained with a Navier–Stokes code, and with particle-based methods. While numerical solutions in the continuum regime for both the 10-moment and Navier–Stokes equations are similar, clear differences are found in the continuum-transition regime, especially near the stagnation point, where the Navier–Stokes code, even when implemented with wall-slip, overestimates the density.

## CHAPTER I

# INTRODUCTION

## 1.1 Motivation

In the design process, engineers need the resultant performance of devices instantaneously to optimize and finalize the design. This is necessary, especially in industry, to shorten the design process. In general, one can conduct a theoretical or experimental analysis to understand the physics and the sources of loss in desired performance. A theoretical analysis is a strong method owing to its universality, however, it can not be applied to real engineering problems because of the generally simple assumptions made. Conversely, experimental analysis is case dependent, yet it allows testing of a reasonably complex system. The drawbacks of the experiment are that it is often expensive and time consuming, especially if a parameter study is necessary to optimize a design.

To overcome the limitations of both theoretical and experimental analysis, the approach of computational simulation was introduced. Originally the complement of theoretical analysis, it is now been recognized as a third mode of science. In this context, computational simulation is referred to as scientific computing, or computational science. Figure 1.1 shows the relations among these three approaches; each approach has its own strength and weakness.

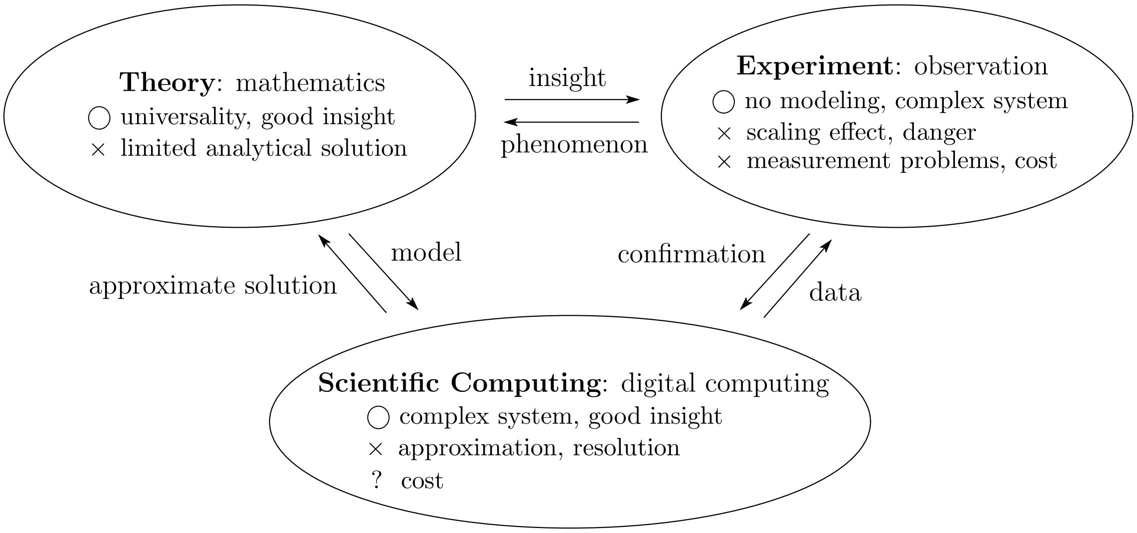

**Theory**: mathematics
○ universality, good insight
× limited analytical solution
insight
phenomenon
**Experiment**: observation
○ no modeling, complex system
× scaling effect, danger
× measurement problems, cost
model
approximate solution
confirmation
data
**Scientific Computing**: digital computing
○ complex system, good insight
× approximation, resolution
? cost

Figure 1.1: Three pillars of the scientific method and their relations are shown. The scientific computing approach is relatively new, complementing both theoretical and experimental approaches.

In a computational simulation, complex mathematical descriptions of physics, mainly by differential, integral, or integro-differential equations, are solved numerically by approximation methods, instead of by deducing an analytical ‘exact’ solution. Because the computational domain is discretized, the solution can be obtained for a fairly complex geometry. However, since in this approach the original governing equations are approximated numerically, extra care is necessary to ensure that an obtained numerical result is in some sense consistent with the original equations. More precisely, one needs to know how much numerical error is introduced in the approximated solution.

Thirty-five years ago, it had been thought that computational simulation would be both cost- and time-effective compared to experiments; this may indeed be true for a simulation on simple geometries, yet it is still arguable for real engineering simulations where geometries are complex. In these cases, numerical results obtained by currently available solvers are highly dependent on the quality of computational grids. Expertise of grid generation is necessary to generate grids ‘smooth’ enough to accommodate deficiencies of solvers. This process may result in a duration of a few weeks or even a month to generate a computational grid.

It is now recognized that the experiment can never be neglected; it may serve, for instance, to validate the design of a few final candidates based on numerical simulations.

The generality, versatility, and manageable cost of computational simulation have lead to its heavy use in the past three decades as an analysis tool to assist engineers with design.

‘Efficiency’ is an important concept in computational simulation. Here, we exclude the pre-process (grid generation) and post-process (visualization) from consideration. Under this assumption, efficiency can be decomposed into two major factors: speed, i.e., CPU time to complete the calculation up to a given evolution time $t_{\text{end}}$ on a given grid, and the numerical accuracy of the resultant numerical solution. Therefore, efficiency can be defined as:

> the total CPU time needed to yield a given solution accuracy.

This is a useful index, especially when comparing methods with different orders of convergence. Here, the order of convergence is defined in regard to the local truncation error of the method. If

$$\text{truncation error} \sim O(\Delta x^r, \Delta t^s)\,, \tag{1.1}$$

where $\Delta x$ and $\Delta t$ are the size of discretization intervals in space and time, the method is said to be $r$-th order in space and $s$-th order in time. A low-order method tends to be computationally less expensive, requiring less CPU time, than a high-order method per computational grid or time step. However, it requires finer grids to reduce the truncation error to a desirable level. Thus, to achieve a pre-set accuracy, the overall run-time of a low-order method might be longer than that of a high-order method. This becomes more significant once a multidimensional problem is considered. One caution we need to observe is that the order of accuracy as defined about is not everything. Indeed, the actual truncation error is expressed by

$$\text{truncation error} = C \times (\Delta x^r, \Delta t^s) + \text{higher-order error}, \tag{1.2}$$

where $C$ is the coefficient of the lowest-order numerical error. Thus, the magnitude of the coefficient, $C$, is as important as the order of a numerical method. An interesting discussion of this issue can be found in [LeV02, p. 150].

The efficiency index is based on the user's point of view; if one wants to obtain a numerical result within a particular error margin, then what method provides the result in the shortest run-time? Here, the emphasis is not on the order of convergence itself, but on efficiency, which indeed demonstrates how 'good' a method is. Nevertheless, it has been observed that there is a high correlation between efficiency and the order of convergence: a higher-order method tends to be more efficient [VAKJ03, Bon99]. Besides the efficiency of a method in terms of turnaround time, it is critical, as mentioned previously, to consider the method's capability of handling complex geometries. If a method is incapable of producing accurate results on the mediocre-quality grids provided by grid-generation software packages, a vast amount of time needs to be spent on improving the grid properties even before starting a calculation.

In the aerospace engineering community, the compressible Navier–Stokes (NS) equations have been adopted as the model equations to understand and analyze flow phenomena theoretically. Due to the nonlinearity and complexity of the equations, analytical solutions are only available in special cases [Whi91]. This limitation of theoretical analysis, and the advent of the modern digital computer have motivated scientists and engineers to solve the NS equations numerically. As a result, numerous methods have been proposed for almost half a century, and great successes have been achieved. Nowadays, computational fluid dynamics (CFD) has become one of the most important design tools for aerospace engineers besides wind-tunnel experiments. Some historical perspectives of the development of CFD in the aerospace community, and the current status, can be bound in [Jam01, Fuj05]. Unfortunately, there is a general consensus in the community that CFD is mature/solved, and not much space is left for the development of an innovative numerical method. This is somewhat understandable: the currently available methods are fairly accurate and robust in engineering applications, and if their efficiency is not sufficient, then one can always utilize adaptive mesh refinement to increase resolution where needed, and implement the multigrid methods or brute-force parallelization to reduce run-time. The ceaseless advancement of computer architectures also discourages researchers to invest their time in developing a new numerical method. Frankly, it becomes more and more difficult and risky to devote one's career to inventing a method that would appeal to other researchers and engineers, but would force them to rewrite their in-house Euler or Navier–Stokes codes. A further discussion regarding the stagnation of method development in the '90s, and some unsolved problems of current methods are presented by Roe [Roe05a].

Even though the currently available methods provide reasonably accurate results, these methods are not necessary efficient. Also, it is well-known that, on greatly distorted grids, these methods show at most second-order convergence in space for practical applications. The demand of solving realistic/practical problems translates into the use of unstructured grids, and efficient implementation on parallel computers. The local-preconditioning [vLLR91, Tur99, RNK02] and multigrid-relaxation techniques [Jam91] have been developed to accelerate the convergence to steady solutions; however, the benefits are still limited, and, furthermore, applying these techniques to parallel computers is not trivial. Recently, a success in computing steady Euler solutions with $N$ unknowns in $O(N)$ operations was achieved [vLD99, NvL03]; for the Navier–Stokes equations such progress is still far away [DvL03, KvLW05].

Because of the lack of efforts to develop new, efficient high-order methods, CFD users have remained using classical methods, and heavily rely on parallel computing.

After a recession in method development lasting almost a decade, the need for efficient and robust discretizations for high-fidelity CFD on unstructured grids has become widely recognized in recent years [DH05, Wan07, Eka05]. In keeping with this insight, we propose in this dissertation a combination of two approaches toward efficient and robust schemes for advection-dominated flows on unstructured grids, one at the partial-differential-equation (PDE) level and the other at the discretization level. Specifically, we aim to develop a unified numerical method for simulating continuum and transitional flow. This can be achieved by simultaneously taking the following two approaches: the use of first-order PDEs and the use of compact high-order discretizations. These will be highlighted in the next two sections.

## 1.2 First Approach: First-Order PDEs

The first approach is replacing everybody's favorite Navier–Stokes equations by a larger set of first-order hyperbolic-relaxation PDEs, which contains the NS equations. (N.B.: here 'first-order' refers to the order of the PDEs.) This is a rather radical approach. First-order hyperbolic-relaxation equations for transitional flow can be derived by taking moments of the Boltzmann equation. From a numerical point of view, the loss of accuracy inherent in adopting the NS equations as the model equations is linked to the second-order derivative modeling molecular diffusion. The elliptic nature of this term yields global data dependence on the discretized domain, and causes a loss of accuracy on nonsmooth adaptively-refined grids. In comparison, a first-order PDE model offers many numerical advantages, including the following:

1. it can replace global stiffness from diffusion terms with local stiffness from source terms, and yield the best accuracy on nonsmooth, adaptively refined grids [CP95];
2. it requires smaller discrete stencils, reducing communications in parallel processing;
3. it has the form of the moment closures of the Boltzmann equation, where the source term describes departure from local thermodynamic equilibrium.

The NS equations are only valid in the continuum-fluid regime where the macroscopic representation of the gas is sufficient. First-order PDEs may overcome this physical limitation. The dimensionless number that indicates whether the continuum assumption is valid or not, is the Knudsen number, denoted by $Kn$. The Knudsen number is defined as the ration of molecular mean free path to characteristic length scale, thus

$$Kn := \frac{\text{molecular mean free path}}{\text{characteristic length scale}} = \frac{\lambda}{L}. \tag{1.3}$$

Introducing the Mach number, $Ma$, and the Reynolds number, $Re$, and using kinetic theory, the Knudsen number is found to have the following relation to these [GeH99]:

$$Kn \sim \frac{Ma}{Re}. \tag{1.4}$$

Hence, high Mach number or low Reynolds number lead to high Knudsen number, resulting in a regime where the continuum assumption is no longer valid. For instance, flow in or around micro-electro-mechanical systems (MEMS) or a reentry vehicle are typically in the so-called transition regime; the flow is in between continuum and free-molecular flow, with Knudsen numbers in the range $0.1 \leq Kn \leq 10$. In this regime the NS equations, even allowing for slip at a solid boundary, do not describe the flow with sufficient accuracy. Table 1.1 summarizes the properties of the simplest models available for a reliable description in different ranges of Knudsen numbers [AYB01]. Figure 1.2 is a schematic of physical regimes of hypersonic flow. A typical Space Shuttle flight trajectory shows that a vehicle experiences nonequilibrium flow in the large part of its flight path [Sal07].

As pointed out by Vincenti and Kruger, there may be a tendency to regard the Boltzmann equation as the last mathematical model in the microscopic description of gases, and its limitations are often overlooked [VK86, p. 333]. The limitations of the Boltzmann equation become clear through its derivation from an even more fundamental equation of motion, the Liouville equation. The Liouville equation is a continuity equation, describing the time evolution of the $N$-particle distribution function in a $6N$-dimensional phase space; the Boltzmann equation deals only with a single-particle distribution function. The Boltzmann equation can be derived from the Liouville equation under the assumptions of binary (two-body) collisions and molecular chaos (no correlation of initial velocities between two molecules before a

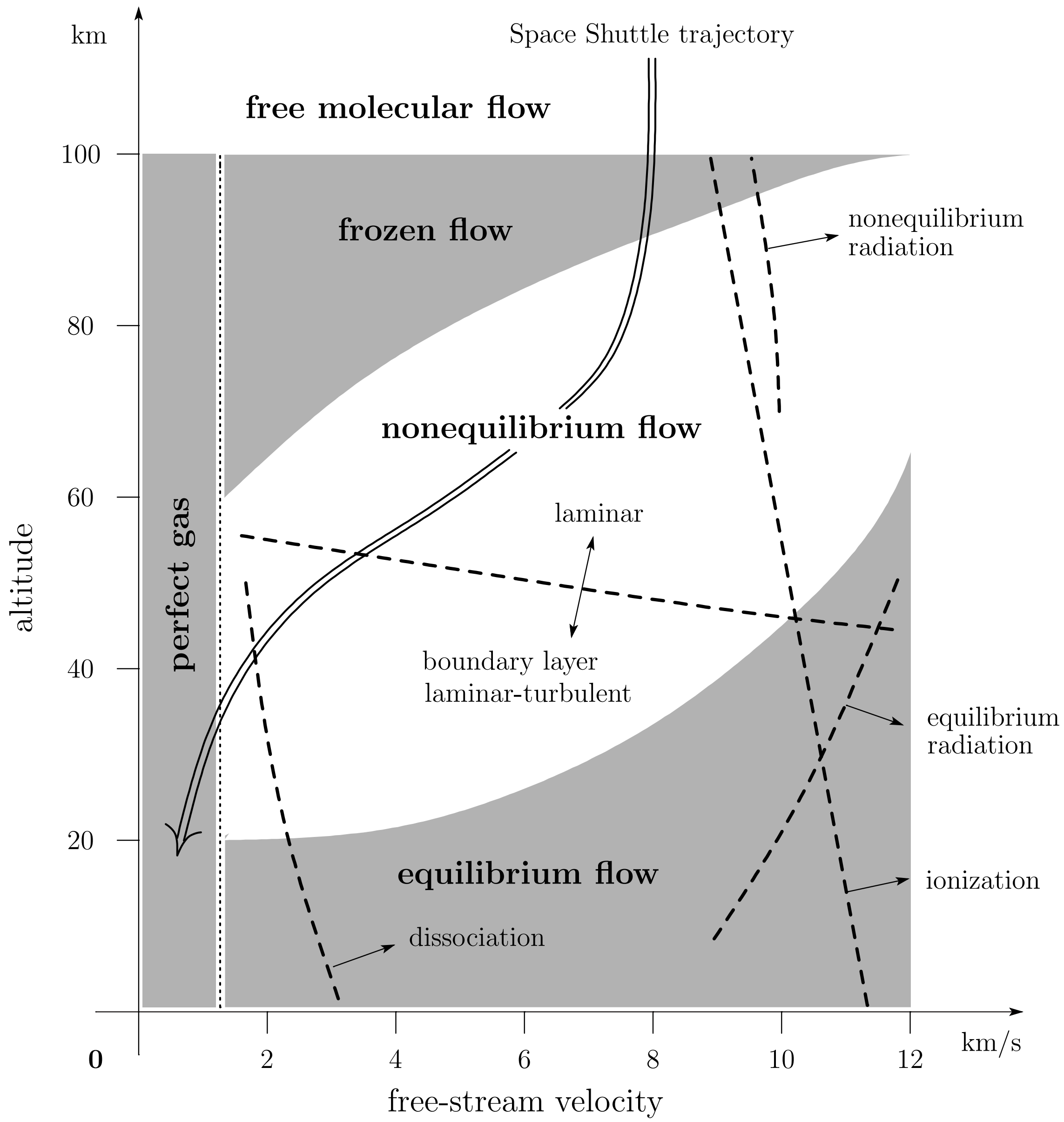


Figure 1.2: Different flow regimes in hypersonic flow are shown with respect to vehicle's speed and flight altitude. A typical flight path of the Space Shuttle is indicated by arrow. This Figure is duplicated [Sal07, p. 13].

| Knudsen number | Assumption | Mathematical model |
|---|---|---|
| $Kn \to 0$ | continuum flow<br>(no molecular diffusion) | Euler equations |
| $Kn \leq 10^{-3}$ | continuum flow<br>(with molecular diffusion) | Navier–Stokes equations<br>(no-slip B.C.) |
| $10^{-3} \leq Kn \leq 10^{-1}$ | continuum-transition regime | Navier–Stokes equations<br>(1st-order slip B.C.) |
| | | moment equations |
| | | Burnett equations<br>(1st-order slip B.C.) |
| $10^{-1} \leq Kn \leq 10$ | transition regime | moment equations |
| | | Burnett equations<br>(2nd-order slip B.C.) |
| $Kn \gg 10$ | free-molecular flow | Vlasov equation<br>(collisionless Boltzmann equation) |

Table 1.1: Simplest mathematical model needed in different flow regimes categorized by the Knudsen number. The full Boltzmann equation (including collisions) is the most complete model among these models and valid in all Knudsen regimes.

collision). Similarly to Table 1.1, relations among yet other mathematical models are shown in Table 1.2. The arrows indicate the direction of derivations; also indicated are the necessary assumptions.

If binary collision and molecular chaos are valid assumptions, then the Boltzmann equation is the most competent and complete model equation since it describes microscopic/particle physics. However, it is an integro-differential equation, for which it is even more cumbersome to obtain analytical solutions than for the NS equations. Its numerical approximation is not an easy task either, because a seven-dimensional phase space must be discretized. Nevertheless, some progress has been presented recently for the direct discretization of the Boltzmann equation [Ari01, KAA$^{+}$07, Tch06, Mor06].

| Viewpoint of description | Mathematical model | Solution method |
|---|---|---|
| deterministic molecular | Newton's Law<br>$f = ma$ | molecular dynamics |
|  | $\Updownarrow$ |  |
| probabilistic molecular | Liouville equation<br>$\mathcal{F}(\boldsymbol{x}_i,\, \boldsymbol{v}_i, t),\ i \in [1, N_p]$ | Monte Carlo methods<br>direct simulation Monte Carlo |
|  | $\Downarrow$ *molecular chaos binary collisions* |  |
|  | Boltzmann equation<br>$\mathcal{F}(\boldsymbol{x},\, \boldsymbol{v},\, t)$ | direct solution |
|  | $\Downarrow$ *thermodynamics* |  |
| hydrodynamic continuum | extended hydrodynamics | direct solution |
|  | *method of moments* $\Downarrow$ $\Downarrow$ *Chapman–Enskog expansion* |  |
|  | moment equations Burnett, super-Burnett equations | direct solution |
|  | $\searrow$ $\swarrow$ *small deviation from LTE* |  |
|  | Navier–Stokes–Fourier equations<br>$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F} = \nabla \cdot (\mathbf{D} \nabla \mathbf{U})$ | direct solution |
|  | $\Downarrow$ *local equilibrium* |  |
|  | Euler equations<br>$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F} = \mathbf{0}$ | direct solution |
|  | $\Downarrow$ *irrotational* |  |
|  | nonlinear potential equation | direct solution |
|  | $\Downarrow$ *small dusturbance* |  |
|  | transonic small-disturbance equation | (〃) |
| *incompressible* | $\Downarrow$ *linearize* |  |
|  | Prandtl–Glauert equation<br>$(1 - M_\infty^2)\phi_{xx} + \phi_{yy} + \phi_{zz} = 0$ | (〃) |
|  | $\Downarrow$ *incompressible* |  |
|  | $\Longrightarrow$ Laplace equation: $\Delta\phi = 0$ | (〃) |

Table 1.2: The various mathematical models describing the motion of gases, and their relations among each other. The hierarchical assumptions lead from the Liouville equation, through the Euler equations, to the Laplace equation [Myo01, OOC98, Jam04].

To circumvent the numerical and mathematical adversities of the Boltzmann equation, mainly two approaches have been proposed: the direct-simulation Monte Carlo (DSMC) method [Bir63, Bir94] and extended-hydrodynamics (generalized hydrodynamics) methods [CC70, Str05, MR98, Eu92].

The DSMC method, a particle-based method, introduces computational particles representing the bulk of actual molecules to model the translational and collisional phenomena. Thus, this method does not literally solve the Liouville/Boltzmann equation numerically, yet under the assumption of molecular chaos and binary collisions, it has been proved that the DSMC method converges to the solution of the Boltzmann equation as the number of particles tends to infinity [Wag92]. The method is extremely accurate, especially in the high Knudsen regime. The DSMC method is required for the highest Knudsen numbers, i.e., rarefied flow, however, in the transition regime there is competition with extended-hydrodynamics methods. The DSMC method produces statistical scatter in the solutions, and it requires a cell size of the order of the molecular mean free path. These properties lead to a computational penalty, especially in the transition (low Knudsen number) regime. Conversely, the extended-hydrodynamics methods are PDE-based, thus they do not have such statistical issues.

Extended-hydrodynamics methods assume the shape of the velocity-distribution function (VDF) in the Boltzmann equation, then transform from the microscopic to the macroscopic representation through taking moments over the velocity spaces. Reducing the dimension of the equation by defining macroscopic quantities provides the mathematical simplicity and computational efficiency. Actually, there are two essential approaches to deriving macroscopic governing equations: the Chapman–Enskog expansion and Grad's method of moments.

The Chapman–Enskog expansion adopts a perturbed Maxwellian distribution function, then the macroscopic variables are expanded with respect to the Knudsen number as the small parameter. The advantage of the Chapman–Enskog expansion is that the number of state variables stays the same as in the NS equations, i.e., $(\rho, \rho\mathbf{u}, \rho E)$ for the conserved quantities. However, higher-order derivatives are introduced to describe the non-equilibrium phenomena. The resulting equations are called the Burnett [Bur36, CC70] and super-Burnett equations [WC48, Foc73, Sha93] corresponding to the second- and third-order expansion with respect to the Knudsen number. The Burnett equations, for instance, contain third-order derivatives; these undesirable higher-order terms cause discretization issues on a nonsmooth grid. Furthermore, the Burnett and super-Burnett equations are known to be linearly unstable [Bob82, UVGC00]. In the augmented Burnett equations [ZMC93], some super-Burnett terms are added to stabilize the equations. Another direction is simplifying the collision integral via the Bhatnagar–Gross–Krook (BGK) model; the resulting system is called the BGK–Burnett equations [AYB01, Bal04]. Despite the efforts to improve the original Burnett equations, the higher-order terms remain highly undesirable with regard to discretization on nonsmooth grids. For this reason, we have eliminated the choice of the Burnett equations or the extended-hydrodynamics equations derived by the Chapman–Enskog expansion as the governing equations.

Grad's method of moments utilizes a distribution function in a Hilbert space, and takes moments over the phase space [Gra49]. The resulting equations are called 'moment equations.' The advantage of Grad's method of moments is that the resulting equations contain only first-order derivatives. However, now the number of state variables, hence the number of governing equations, is increased. This would seem to be a computational penalty, but these quantities actually have their benefits. For instance, the heat fluxes, which form a vector in the NS equations, now fill up a tensor, and evolution equations for these higher-order quantities exist and are coupled to the mass and momentum equations. Similarly, all elements of the shear stress tensor have their own evolution equation. In comparison, the NS equations employ algebraic constitutive laws for stress (Stokes) and heat flux (Fourier); these quantities are proportional to the gradients of velocity and temperature, respectively. Having the stress and heat-flux tensors evolve together with the other conserved quantities makes one expect to obtain a more accurate prediction of these physical quantities. Combining the representation of non-equilibrium gas dynamics with our vision of numerical efficiency, the Grad-type method of moments appears the most suitable approach to constructing the governing equations from the Boltzmann equation. Recall that describing physics solely by first derivatives is the key to developing efficient, highly parallelizable schemes on nonsmooth grids.

Despite the promising properties of the moment equations, the level of maturity of this approach is far from affording it to replace or complement the NS equations and the DSMC method. Mainly, two obstacles need to be overcome to make the approach applicable to a practical engineering problem; again one is at the PDE level, and the other is at the discretization level.

The first issue is that, particularly for steady supersonic flow, the moment equations produce a discontinuity inside a smooth shock structure [Gra52, Hol64]. This is due to the nature of hyperbolic equations, which allow the physical quantities to propagate only at finite characteristic speeds. In reality, owing to the significant effect of molecular diffusion, a smooth shock profile connects the upstream and downstream states. This smooth profile is not realized by the moment equations once a flow is above a critical Mach number. For Grad's 13-moment equations, the critical Mach number is approximately 1.65. Some improvements in the model have been shown to increase the critical Mach number, e.g., by taking further higher-moment equations [Wei95], or introducing second-order dissipation terms in the system [TS04]. Even though making the moment approach more suitable for super/hypersonic flow is a critical issue, the derivation of a new set of equations is beyond the scope of this study. Here, we will only adopt the robust and physically reasonable '10-moment equations' [Lev96, Bro96] as a representative set of model equations.

The second issue is the lack of an efficient numerical method. The moment equations have the form of hyperbolic equations with a relaxation source term. Since the source term is parameterized by the 'relaxation time', which is of the order of the mean collision time, any standard explicit method has a severe time-step restriction with regard to both stability and accuracy, especially if one is only interested in evolution at the macroscopic temporal and spatial scale. A standard implicit treatment for the source term overcomes the stability restriction; however, taking the large time step does not necessary guarantee the accuracy of solutions. This study mainly focuses on this issue, i.e., the development of efficient and accurate methods for hyperbolic-relaxation equations with stiff relaxation source terms.

## 1.3 Second Approach: Compact High-Order Method

The second approach is to adopt a high-order discretization method that can preserve compactness in both space and time. Here, a compact method refers to one satisfying the following property:

> the update to a given cell should only be a function of the $t^n$-solutions

> of the cell itself, and its immediate neighbors [Low96, p. 4].

Since the methodologies to achieve high-order convergence in space and in time differ considerably, a discretization method in space is discussed first.

### 1.3.1 Spatial Discretization

In a standard finite-volume method (FVM), solutions are defined as cell-averaged quantities over the computational domain, and the higher-order accuracy in space relies on local piecewise-polynomial reconstruction, which requires extended stencils. Representative of the great successes achieved by higher-order FVMs are MUSCL[1] (second-order in space) by Van Leer [vL79] and PPM[2] (third-order in space) by Colella and Woodward [CW84]. Later, the $k$-exact reconstruction was proposed by Barth and Jespersen [BJ89]. Among these reconstruction methods, where reconstruction stencils are fixed, a total-variation-diminishing (TVD) limiter is necessary to ensure solution monotonicity near a discontinuity. Two defects of this approach are the clipping of local extrema and the difficulty of extending the TVD philosophy to multiple dimensions. (For a multidimensional solutions, sensing monotonicity by the total variation is unsuited.) In practice, limiting is done dimension by dimension using a one-dimensional TVD condition. The clipping of extrema can be avoided by replacing the TVD condition by the total-variation-bounded (TVB) condition [Shu87, CS89].

To overcome the difficulty of extending the TVD principle to multidimensional problems, Harten et al. proposed the Essentially Non-Oscillatory (ENO) scheme, which is TVB and retains high-order accuracy in smooth regions [HOEC86, HO87,

[1] The acronym for Monotone Upstream-centered Scheme for Conservation Laws.

[2] The acronym for Piecewise Parabolic Method.

HEOC87]. In brief, an ENO scheme uses adaptive stencils in order to choose the stencil on which the solution is smoothest; this way the reconstructed polynomial never spans a discontinuity. However, choosing the best stencil may vary erratically, causing anomalies in the reconstructed solution. Later, a more robust reconstruction process, based on a convex combination of interpolants on all possible stencils, called weighted ENO (WENO) was proposed by Liu et al. [LOC94], and Jiang and Shu [JS96]. The extension of the WENO scheme to nonsmooth grids was proposed by Friedrichs [Fri98], and Hu and Shu [HS99]. These reconstruction techniques have allowed higher-order spatial discretization in the finite volume framework; however, an issue is that the higher a method's order, the larger the reconstruction stencils are. For instance, stencils for the quadratic reconstruction (for third-order accuracy) on tetrahedral grids require 50 to 70 cells [DL99].

The discontinuous Galerkin (DG) method overcomes the issue of growing stencils by increasing a solution representation in each element; a solution in a cell/element is no longer piecewise constant, but polynomial of degree $k$. Obviously, when $k = 0$, a DG method is equivalent to a first-order finite volume method. A DG method was first introduced by Reed and Hill at Los Alamos National Laboratory to solve the steady linear neutron transport equation [RH73]. Soon after, LaSaint and Raviart presented error analyses, and showed that a DG($k$) method for a steady one-dimensional problem is of order $2k+1$, and for a two-dimensional problem with strictly rectangular elements it is of order $k+1$ [LR74]. The analysis was later extended to general triangular elements by Johnson and Pitkäranta, who showed that the formal order of accuracy of DG($k$) is $k+\frac{1}{2}$ [JP86, Joh87]. The result was confirmed numerically by Peterson [Pet91]. For a comprehensive literature review of almost three decades of DG research, see Cockburn and Shu [CS01], and Zienkiewicz et al. [ZTSP03]. Surprisingly, attention to DG methods in the aerospace community came quite recently. Part of the reason was the robustness of second-order finite-volume methods, and the advent of massively parallel computers in the early '90s. These circumstances made CFD practitioners and code developers complacent.

Besides the DG spatial discretization, there are other methods that also preserve the compactness while achieving high-order convergence in space on nonsmooth grids, in particular, the spectral difference (SD) [LVW06] and spectral volume (SV) [Wan02] methods. A comparisons between the SV and DG methods was presented in [ZS05, SW04]. The authors conclude that the DG method is more accurate but requires more memory and has a more restrictive stability condition than the SV method. Shu also compared high-order finite-difference, finite volume WENO, and DG methods [Shu03]. He concludes that DG methods are the most flexible in terms of arbitrary triangulation and boundary conditions, but the development of more robust and high-order-preserving limiters is necessary. In this thesis, we purposely exclude high-order finite-difference methods and spectral methods, as their applicability is limited to structured grids. Among the exceptions is the spectral method developed by Kopriva, which can be applied to unstructured quadrilateral staggered grids. Carpenter and Gottlieb also extended spectral methods to unstructured grids [CG96].

Another class of high-order methods called the spectral element method, originally developed by Patera [Pat84], uses high-order polynomials to achieve spectral-like convergence. The further development of the spectral element method in the context of a DG method was done by Karniadakis and Sherwin [KS05].

Related to the Galerkin formulation, Hermitian methods achieve high-order accuracy by defining not only solutions at nodes, but their derivatives at the same points. This type of element is called a Hermitian element, compared to the Lagrangean elements, which define only the solution itself, but at multiple places in an element. Since the method adopts the Hermitian polynomials for the solution approximation, we could consider the Hermitian methods a Galerkin method, yet it does not utilize the weighted-residual formulation completely. Due to the continuity requirement of a certain order at each node, the method is currently restricted to linear equations, and a main drawback is that Hermitian methods are non-conservative [Roe05b, pp. 240–244]. Recent developments and applications of Hermitian methods to computational acoustics are presented by Capdeville [Cap05, Cap06, Cap07].

### 1.3.2 Temporal Discretization

So far, only spatial discretization has been discussed. As to the temporal discretization, the semi-discrete method is currently the most successful approach. A semi-discrete method incorporating the method-of-lines (MOL) [Sch91] decomposes the spatial and temporal discretizations. This simplifies the development/formulation of a method and its coding significantly. Once the spatial discretization is constructed, one's favorite ODE solver can be employed for the time discretization. Details of methods for ODE's can be found in [Lam91, HNW93, HW96]. For hyperbolic conservation laws, a TVD Runge–Kutta ODE solver is the common choice [SO88]. More recently it is referred to as the strong stability-preserving (SSP) method [GST01]. The methods are one-step multi-stage and assure nonlinear temporal stability. One of the drawbacks of the semi-discrete approach is that the stability condition becomes increasingly restrictive as the spatial discretization method goes higher-order. This has been observed for the DG method [CS01, p. 191] and the SV method [Wan02, p. 249]. Increasing the order of the RK solver slightly relaxes the stability restriction; however, this introduces extra function evaluations, making the method more expensive. Another defect of RK methods is that, if the fifth or higher order in time is desired, the required number of stages is greater than the order of a method; this is called the Butcher barrier [Lan98, p. 182].

Another class of ODE solvers are multi-step methods: Adams–Bashforth, Adams–Moulton, and the backward-difference formula (BDF) et al.. A multi-step method achieves high-order accuracy by utilizing the prior solutions, whereas a multi-stage method does it by adding function evaluations. Thus, a multi-step method is generally less expensive, but more memory is required to store the prior solutions. Furthermore, the size of the time-step is more restricted due to the data-dependence over multiple times.

### 1.3.3 Space-Time Discretization

To overcome the stability restriction and lesser efficiency of the semi-discrete MOL approach, the fully discrete method is considered. In this approach, spatial and temporal operators are discretized in similar manner. The classical one is the Lax–Wendroff method, second-order in both space and time [LW64]. The method takes advantage of the original PDEs, replacing the temporal derivative by spatial derivatives. To our knowledge, the first attempt to include a DG spatial discretization in the space-time approach was done by Bar-Yoseph [BY89], and Bar-Yoseph and Elata [BYE90]. They apply the space-time DG method to the 1-D Euler equations. Due to their choice of space-time mesh, their method is fully implicit. Choe and Holsapple combine the idea of the temporal Taylor–Galerkin method, which is a finite-element extension of the Lax–Wendroff idea, with the discontinuous Galerkin method in space (they denote this as TG–DFEM: Taylor–Galerkin discontinuous finite-element method). The resulting method is one step, explicit second-order in both space and time, and stable up to a Courant number of 0.4. The method is applied to 1-D scalar linear and nonlinear equations. Later, Lowrie et al. develop an explicit space-time DG method, and present a superconvergence result, $O(2k+1)$ convergence, in both Fourier analyses and numerical tests [LRvL95, Low96, LRvL98]. However, the resulting method for 2-D problems requires staggered grids, and could suffer from substantial numerical dissipation.

More recently, the idea of high-order space-time discretizations has returned to the finite volume framework. Toro et al. developed an arbitrarily high order (ADER) method: a one step method with modified generalized Riemann problems solved at each cell interface [TMN01]. Later, Dumbser et al. applied the ADER approach to the DG framework [DM05, QDS05]. Again, the method is one step, requiring the same memory space as the forward-Euler time integration. The results show that their DG–ADER or DG–LW method is more efficient than a DG–RK method due to the reduced usage of a limiter at intermediate stages. However, a Fourier analysis shows that these fully discrete methods still have a similar stability restriction as DG–RK methods [DM06, p. 224][QDS05, p. 4533].

The approach by Van der Vegt and Van der Ven is more direct; they treat space and time variables in the same manner, and construct a four-dimensional discontinuous functional space to represent solutions. They denote this approach as the space-time discontinuous Galerkin method [vdVvdV02, KvdVvdV06]. Unlike semi-discrete methods, the method is inherently implicit, and pseudo-time stepping is introduced to solve the implicit equations in each time interval.

Our approach is also to develop a method with complete coupling, thus discretizing in both space and time simultaneously. Yet, the current method still stays in between semi-discrete and fully discrete methods since trial (basis) functions for the solution representation solely depend on space, not on time. The method proposed here is based on the 'upwind moment scheme' recently developed by Huynh for hyperbolic conservation laws [Huy06a], and based in turn on Van Leer's Scheme III in [vL77, p. 281]. The solution representation is only piecewise linear. The two key characteristics of this method are:

1. cell variables are updated over a half time step without any interactions with neighboring cells (Hancock's observation [vAvLR82]);
2. the gradient of each flow variable evolves by an independent equation (DG representation).

The story behind the invention of Hancock's scheme and its relation to MUSCL and PPM are described in [vL06, Hol96]. The resulting scheme looks promising in comparison to the popular MOL approach. The upwind moment scheme is a fully discrete, one-step method with one intermediate update step needed for computing the volume integral of the fluxes. It requires solving a Riemann problem twice at each cell interface but achieves third-order accuracy in space and time.

In this thesis, the upwind moment method, originally developed for hyperbolic conservation laws, is extended to hyperbolic-relaxation equations. Here, we will call the method in a more generic manner, 'DG($k$)–Hancock method.' The method preserves compactness through a DG spatial discretization, and attains a high efficiency by the Hancock discretization in time interlaced by a Gauss–Radau quadrature for the source term.

## 1.4 Current State of Hyperbolic–Relaxation Equations

### 1.4.1 Mathematical Background

Our target equations, the moment equations, have the form of hyperbolic-relaxation equations. More precisely, these are systems of hyperbolic equations with *stiff relaxation* source terms such that

$$\partial_t \mathbf{u}(\boldsymbol{x},t) + \nabla \cdot \mathbf{f}(\mathbf{u}) = \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}); \quad \boldsymbol{x} \in \Omega,\ t > 0, \tag{1.5}$$

where $\mathbf{u}, \mathbf{s} \in \mathbb{R}^m, \mathbf{f} \in \mathbb{R}^{m\times 3}$, and $\epsilon > 0$ is the relaxation time. In the above equations there exist two time scales: one is the advective time scale denoted by $T$ and the other is the relaxation time $\epsilon$. We need to keep in mind that these two time scales are genuinely disparate. Also, the above form is somewhat more restrictive than the general form of hyperbolic equations with *stiff* source terms. For instance, advection-reaction equations have the same form as the above equations, but the source term is not the *relaxation* source term. In this case, the system possesses multiple equilibrium states, whereas the stiff relaxation source term always leads the system to the unique equilibrium state. Here, we only consider such stiff-relaxation source terms.

Let $a$ be the characteristic wave speed, hence of the order of the eigenvalues of the flux Jacobian $\partial \mathbf{f}_n / \partial \mathbf{u}$, and $L$ be the characteristic length, then we have

$$T = \frac{L}{a}. \tag{1.6}$$

The behavior of hyperbolic-relaxation equations dramatically changes as the ratio of two time scales (stiffness parameter), $\dfrac{T}{\epsilon}$, changes; the stiffness parameter is inversely proportional to the Knudsen number. When the advection is dominant, and the relaxation towards the equilibrium state is slow, hence $T \ll \epsilon$, then the systems is called to be in the *frozen* limit. In this case, the original equations become pure advection equations,

$$\partial_t \mathbf{u}(\boldsymbol{x}, t) + \nabla \cdot \mathbf{f}(\mathbf{u}) = \mathbf{0}, \tag{1.7}$$

which correspond to the collisionless Boltzmann equation. When wave propagation and relaxation processes are equally important, hence $T \sim \epsilon$, then the state is in the transition regime. See Table 1.1 on page 10 for the corresponding Knudsen range.

When the source-term effect dominates that of the wave propagation, $T \gg \epsilon$, the system is said to be in the near-equilibrium limit. In such a regime, the original hyperbolic-relaxation equations reduce to a system of $n < m$ second-order equations

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \nabla \cdot (\mathbf{D}(\mathbf{U}) \nabla \mathbf{U}), \tag{1.8}$$

where $\mathbf{U} \in \mathbb{R}^n, \mathbf{F} \in \mathbb{R}^{n \times 3}$, and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a tensor of diffusion coefficients, with eigenvalues proportional to $\epsilon$. This reduced system is derived through a Chapman–Enskog expansion [CLL94, Liu87], and the corresponding physical equations are the Navier–Stokes equations. Subsequently, when $\epsilon \to 0$, the system reaches the equilibrium limit:

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{0}, \tag{1.9}$$

and this is the form of the Euler equations. In the context of gas kinetics, the equilibrium limit means that wherever a velocity distribution function is away from the equilibrium (Maxwellian) distribution, the source term drives the state to the local thermodynamic equilibrium instantaneously.

When we consider a numerical method to solve the hyperbolic-relaxation equations (1.5) in the near-equilibrium or equilibrium limit, a common explicit integration in time will result in the following restrictive time step:

$$\Delta t \sim \epsilon. \tag{1.10}$$

In contrast, a typical explicit method for the Euler equations (1.9) leads to the time step:

$$\Delta t \sim \frac{\Delta x}{a} \gg \epsilon, \tag{1.11}$$

where $\Delta x$ is the typical size of a computational cell. Hence, one might expect that even though we solve the hyperbolic-relaxation equations (1.5), in the equilibrium limit, we could construct a method with a time step $\Delta t \propto \Delta x$. This means that a method does not need to resolve the fast relaxation process described by the source term, yet provides the correct equilibrium limit numerically. In general, this property is called *underresolved* or *asymptotic preserving*. A similar argument can be made for the Navier–Stokes equations, but the well-known restriction due to the explicit treatment of the diffusion term leads to $\Delta t \propto \Delta x^2$. If the diffusion term is treated implicitly, then we recover the advective time-step constraint: $\Delta t \propto \Delta x$.

In order to develop such a method, there are two numerical issues to overcome: maintaining stability in the stiff (near-equilibrium/equilibrium) regime, and obtaining the accurate reduced system numerically. The stability issue can be solved quite easily; implicit treatment of the source terms, particularly an $L$-stable method, guaranties the unconditional stability even in the stiff regime. However, a stable numerical method for hyperbolic-relaxation equations does not always ensure that the method simultaneously reduces to a consistent discretization of the reduced equations. This is simply because the method does not know anything about the reduced equations. As shown by Hittinger [Hit00], and Hittinger and Roe [HR04], the resulting equilibrium flux $\mathbf{F}(\mathbf{U})$ is the resultant of tight coupling between the frozen flux $\mathbf{f}(\mathbf{u})$ and the relaxation source term $\mathbf{s}(\mathbf{u})$. This becomes more clear when the following 1-D linear hyperbolic-relaxation equations,

$$\partial_t \mathbf{u} + \mathbf{A}\partial_x \mathbf{u} = \mathbf{Q}\mathbf{u}, \tag{1.12}$$

are considered. By introducing new variables $\mathbf{w}$ such that $\mathbf{u} = e^{\mathbf{Q}\,t}\mathbf{w}$, they show that the resulting finite-volume method has the form [HR99, RH01]:

$$\bar{\mathbf{u}}_j^{n+1} = e^{\mathbf{Q}\,\Delta t}\bar{\mathbf{u}}_j^{n} - \frac{1}{\Delta x}\int\limits_{t^n}^{t^{n+1}} e^{\mathbf{Q}(t^{n+1}-t)}\left(\mathbf{f}_{j+1/2}(t) - \mathbf{f}_{j-1/2}(t)\right)\,dt. \tag{1.13}$$

The above equations show that when the relaxation is fast, thus $\mathbf{Q}$ is large, cell-interface fluxes are a combination of the frozen flux and exponential damping. Hence, separating the flux and source evaluations as with operator splitting fails to capture the coupling, and results in a reduction of the order of accuracy due to excessive numerical dissipation [LMM87, LM89, Pem93b, JL96, AR97].

The above formulation also reveals that a straightforward frozen Riemann solver based on $\mathbf{f}(\mathbf{u})$ becomes inadequate in the near-equilibrium/equilibrium limit, where a Riemann flux should be constructed based on the equilibrium flux $\mathbf{F}(\mathbf{u})$. Using a 'frozen' Riemann solver even in the near-equilibrium limit means that the numerical method tries to push the system back to the frozen limit; this is completely opposite to how the source term acts in the system. There are basically two approaches to achieve the coupling. The first approach, the ideal one but definitely uphill, is constructing a Riemann solver with built-in source-term effect. In this way, the eigenstructure of the Riemann solver automatically adapts from frozen to equilibrium limit based on the parameter $\dfrac{\Delta t}{\epsilon}$. Substantial work along this path has been conducted by Pember [Pem93b, Pem93a] and Hittinger [Hit00].

The second approach is more straightforward. It was shown that the simple operator splitting results in reducing the order of accuracy, due to insufficient coupling. But we can introduce more coupling, for instance, by adding extra stages in method-of-lines. Recently, Pareschi and Russo extended implicit-explicit (IMEX) Runge–Kutta methods, originally developed by Ascher et al. for advection-diffusion equations [ARW95, ARS97], to hyperbolic-relaxation equations [PR05]. The IMEX Runge–Kutta methods together with the WENO spatial reconstruction result in second- and third-order methods, and a numerical test confirms analyses.

Our approach based on a compact high-order method stays in between the above two approaches. Due to the usage of a frozen Riemann solver, this approach does not solve the fundamental issue described previously. However, it is well-known that, as a method becomes higher-order, the choice of Riemann solver becomes less important to accuracy; in this way, we circumvent the lack of an ideal Riemann solver in the near-equilibrium/equilibrium limit by adopting a high-order method with enough coupling between the frozen flux and the sources. A high-order method also prevents the physical dissipation to be smeared by the numerical dissipation on unresolved coarse grids.

In this section, we have briefly overviewed the properties of hyperbolic-relaxation equations. The more detail mathematical descriptions and proofs can be found in the review paper by Natalini [Nat98] and references therein. A concise description is also provided in [LeV02, pp. 410–415].

### 1.4.2 Previous Work

In [HSvL05], various numerical methods for hyperbolic-relaxation equations are categorized into the following four types:

- semi-discrete method [LMM87, LM89, JL96, LM99b, CJR97, PR05],
- characteristic method [AR97, AR98, Par98],
- central-difference finite-volume method [BS97, LRR00],
- modified coupled-space-time Godunov method [Pem93b, Hit00].

Recently, Lowrie and Morel developed the DG(1) spatial discretization method together with an implicit time integrator for linear hyperbolic-relaxation equations [LM99b, LM99a, LM02]. Based on Fourier analyses and numerical tests, they show that the DG(1) method is asymptotic preserving in their diffusion scaling. Hence, second-order accuracy is uniformly achieved without damaged by the numerical dissipation even though the relaxation process is not resolved at all. Their promising results interested us in the potential use of DG methods for discretizing first-order systems for compressible, viscous flow. However, the application of their results to our problem must be done with care because their choices of scaling and limit-taking are not the same as ours. A detailed discussion is formed in [HSvL05], while further investigation for fully discrete methods based on Fourier analyses and numerical tests are presented in this thesis in Chapter IV.

## 1.5 Outline of Thesis

This thesis is organized as follows. In Chapter 2, at first the DG(1)–Hancock method for systems of one-dimensional hyperbolic-relaxation equations is described. The detailed mathematical finite-element formulations are omitted for simplicity. Once the 1-D DG(1)–Hancock method is described, the multidimensional extension is provided in the following section together with more rigorous formulations. Besides the DG(1)–Hancock method, the original Hancock method and semi-discrete methods (ODE solvers) are also described for later comparisons. The original upwind moment scheme by Huynh for systems of hyperbolic conservation laws is also described.

In Chapter 3, instead of jumping straight to the analysis for hyperbolic-relaxation equations, the analysis of DG(1)–Hancock method for the scalar one- and two-dimensional linear advection equations are performed. In these cases, the DG(1)–Hancock method is equivalent to both Van Leer's scheme III and Huynh's upwind moment scheme. The dissipation/dispersion errors, the order of accuracy, and the stability condition are compared to those of various semi-discrete and fully discrete methods. The numerical results presented later confirm the analyses. It is shown that the DG(1)–Hancock method preserves the prominent properties of both finite-volume and finite-element methods, and the stability constraint can be more relaxed than that of semi-discrete DG–RK methods.

In Chapter 4, after understanding the basic properties of the DG(1)–Hancock method for hyperbolic conservation laws, the analyses of the DG(1)–Hancock method and other currently available methods for one- and two-dimensional linear hyperbolic-relaxation equations are conducted. Again, dissipation/dispersion error, and the order of accuracy are compared to those of various methods. The Fourier analysis shows high accuracy with minimum number of Riemann solvers. The following numerical tests confirm the analyses, and also the efficiency of the DG(1)–Hancock method compared to several semi-discrete methods.

In Chapter 5, the methods are applied to more practical equations, in particular, the set of 10-moment equations which are gas dynamics equations that include a pressure/temperature tensor. The external flow around a NACA0012 airfoil is computed and compared with flow results from alternative methods: the NS equations, Information Preservation (IP) method, and DSMC; these solutions are also compared to experimental flow measurements.

The conclusions and suggestions for future work are following in Chapter 6.

CHAPTER II

# NUMERICAL METHODS FOR HYPERBOLIC EQUATIONS WITH RELAXATION SOURCE TERM

## 2.1 Introduction

The method proposed here to solve the hyperbolic-relaxation equations,

$$\partial_t \mathbf{u}(\boldsymbol{x}, t) + \partial_x \mathbf{f}(\mathbf{u}) = \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}); \quad \boldsymbol{x} \in \mathbb{R},\ t > 0, \tag{2.1}$$

where $\mathbf{u} \in \mathbb{R}^m$, is based on the 'upwind moment scheme' [Huy06a] recently developed by Huynh for hyperbolic conservation laws:

$$\partial_t \mathbf{U}(\boldsymbol{x}, t) + \partial_x \mathbf{F}(\mathbf{U}) = 0; \quad \boldsymbol{x} \in \mathbb{R},\ t > 0, \tag{2.2}$$

where $\mathbf{U} \in \mathbb{R}^n, n < m$. Recall that the hyperbolic-relaxation equations we are interested in have a *unique* equilibrium limit; when $\epsilon \to 0$, (2.1) reduces to the hyperbolic conservation laws (2.2). The solution representation of the upwind moment scheme is only piecewise linear. The two key characteristics of this method are:

1. After initial gradient is established at each cell, cell variables are updated over a half time step without any interactions with neighboring cells (Hancock's observation [vAvLR82, vL06]);

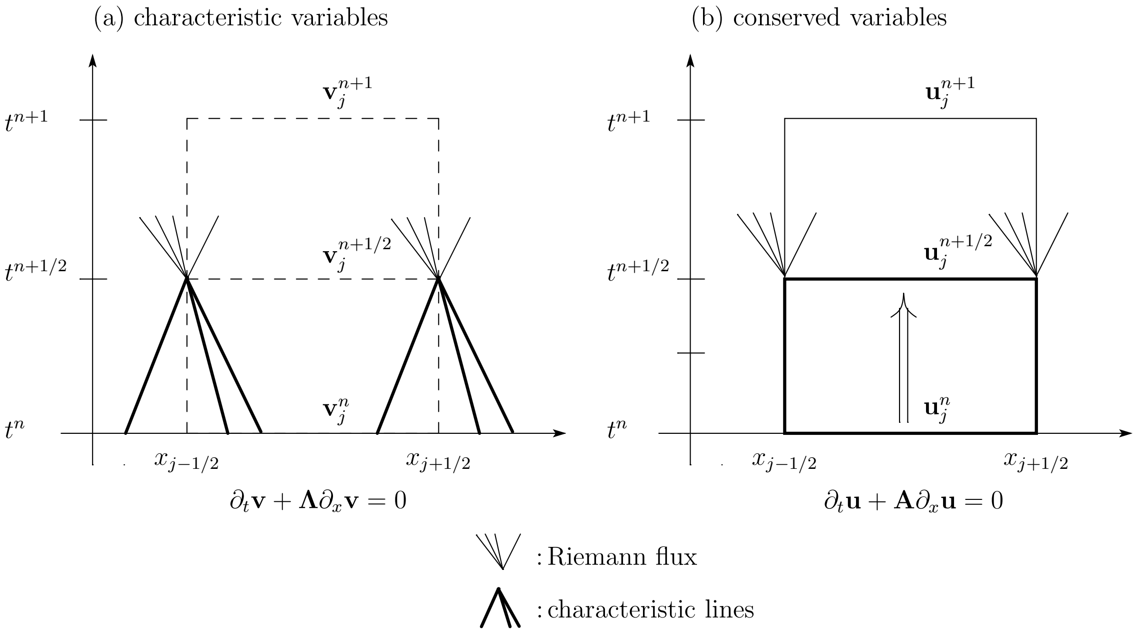


Figure 2.1: The development of the Hancock time integration is motivated by the fact that when a Riemann flux is evaluated at the half time step $t^{n+1/2}$, then there is no wave interaction at both interfaces along $(x_{j\pm1/2}, [t^n, t^{n+1/2}])$. Hence, predicted values associated to the half-time can be obtained by any form of equations, e.g., the conservation, or primitive form, expressed in the cell's interior. These equations then produce the input values for the Riemann problems at $(x_{j\pm1/2}, t^{n+1/2})$.

2. the gradient of each flow variable evolves by an independent equation (DG representation).

Hancock's observation is illustrated in Figure 2.1 by using the 1-D Euler equations as an example. In order to evaluate fluxes at the half-time $t^{n+1/2}$, three characteristic lines can be drawn for each interface. When we pay attention in the domain $[x_{j-1/2}, x_{j+1/2}] \times [t^n, t^{n+1/2}]$, it is realized that we can neglect wave interaction to adjacent cells until flow quantities evolve to $t^{n+1/2}$. This observation leads to the conclusion that any form of the Euler equations besides the characteristic form can also accurately evolve the flow quantities from time $t^n$ to $t^{n+1/2}$. Here, this observation is extended to hyperbolic-relaxation equations.

As to the DG discretization, the number of discretized update equations is twice that for the original PDEs. We denote the upwind moment scheme by “DG(1)–Hancock method”; it looks promising in comparison to the popular method-of-lines (MOL) approaches such as FV/DG–MOL. The MOL decouples the discretizations in space and time: first a semi-discrete form in space is evaluated, then integrated over time by a suitable ODE solver, e.g., RK$s$. The upwind moment scheme $\big($DG(1)–Hancock$\big)$ is a fully discrete, one-step method with one intermediate update step needed for computing the volume integral of the flux. It requires solving a Riemann problem twice at each cell interface. Based on a Fourier analysis, the method achieves third-order accuracy in space and time. By design, the upwind moment scheme for a linear advection equation reduces to Van Leer’s ‘scheme III’ [vL77], which is a DG spatial discretization with an exact shift operator for the time evolution. It was shown that the method is linearly stable up to Courant number 1 with an upwind flux. Conversely, DG spatial discretizations combined with MOL typically have a more strict stability condition: for DG(1)–RK2 (second-order) the limit is $\dfrac{1}{3}$, and for the DG(2)–RK3 (third-order) it is $\dfrac{1}{5}$ [CS01, p. 190].

When discretizing hyperbolic-relaxation equations (2.1), the source term has to be treated implicitly to ensure the stability in the stiff regime ($\epsilon \ll 1$). In contrast, the advection term is treated explicitly due to the complexity of the flux evaluation. These methods are ‘semi-implicit.’ It is expected that the stability of a method is solely constrained by the explicit discretization, that is, by an advective CFL condition: $\Delta t \propto \Delta x$. This can indeed be realized; however, the simple implicit treatment of source terms, e.g., the backward Euler method (1st-order), does not always guarantee high-order accuracy in the stiff regime, where the source-term effect is important.

The unique feature of the upwind moment scheme is the evaluation of the space-time volume integral in the update equation of the solution gradient. Indeed, this volume integral of the flux makes the method third-order accurate. When applying the upwind moment scheme to hyperbolic-relaxation equations, a difficulty arises in computing the volume integral of the source term in the update equation of the cell-average. Since the vector of cell-average variables at the next time level, $\bar{\mathbf{u}}^{n+1}$, is still unknown, no simple quadrature rule in time can be used. Thus, instead of computing the volume integral by a quadrature, we use an idea from stiff ODE solvers to solve stiff source terms accurately. In our method, the time integral of the source term is discretized by the two-point Gauss–Radau quadrature, which can be regarded as an $L$-stable ODE solver if advection terms are omitted. The same quadrature points are used for the volume integral of the flux in the gradient-update equation, whereas Huynh's original upwind moment method for conservation laws adopts the three-point Gauss–Lobatto quadrature.

Because the source term does not contain spatial derivatives, the method is point-implicit, that is, the implicitness is local, requiring no information from neighboring cells. Previously a method for the case of a linear flux and source was presented [SvL06]; here we extend the method to a nonlinear flux and source. The extension to multiple dimensions is also described.

## 2.2 DG(1)–Hancock Method for One-Dimensional Equations

### 2.2.1 DG Formulation

For brevity, we only consider a one-dimensional case with a uniform grid in this section; the multidimensional extension is presented in the next section. Let $\Delta x := x_{j+1/2} - x_{j-1/2}$ be the cell width and $I_j := [x_{j-1/2}, x_{j+1/2}]$ be the domain of cell $j$. The general DG method is obtained by converting a differential equation to a weak formulation (weighted-residual method). Here, since a DG discretization is adopted only in space, a test function $v(x)\colon \mathbb{R} \to \mathbb{R}$ is just a function of space. The Hancock method is adopted for time discretization. Multiplying (2.1) by a test function, $v(x)$, and integrating over the interval $I_j$ leads to the semi-discrete form of the weak formulation:

$$\frac{\partial}{\partial t} \int\limits_{I_j} \mathbf{u}(x,t)v(x)\,dx = -\int\limits_{I_j} \partial_x \mathbf{f}(\mathbf{u}(x,t))v(x)\,dx + \int\limits_{I_j} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}(x,t))v(x)\,dx. \tag{2.3}$$

Applying integration by parts on the flux term transfers the spatial differential operator acting on the flux $\mathbf{f}(\mathbf{u})$ to the test function $v(x)$,

$$\begin{aligned}\frac{\partial}{\partial t} \int\limits_{I_j} \mathbf{u}(x,t)v(x)\,dx = -\mathbf{f}(\mathbf{u}(x,t))v(x)\Big|_{x_{j-1/2}}^{x_{j+1/2}} \\ + \int\limits_{I_j} \mathbf{f}(\mathbf{u}(x,t))\partial_x v(x)\,dx + \int\limits_{I_j} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}(x,t))v(x)\,dx.\end{aligned} \tag{2.4}$$

To derive the fully discrete method, we integrate again in time over $T^n := [t^n, t^{n+1}]$,

$$\begin{aligned}\underbrace{\int\limits_{I_j} \mathbf{u}(x,t)v(x)\,dx\Big|_{t^n}^{t^{n+1}}}_{\text{Time evolution}} = \underbrace{-\int\limits_{T^n} \mathbf{f}(\mathbf{u}(x,t))v(x)\Big|_{x_{j-1/2}}^{x_{j+1/2}} dt}_{\text{Boundary integral}} \\ + \underbrace{\iint\limits_{I_j\times T^n} \mathbf{f}(\mathbf{u}(x,t))\partial_x v(x)\,dxdt}_{\text{Volume integral}} + \underbrace{\iint\limits_{I_j\times T^n} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}(x,t))v(x)\,dxdt}_{\text{Volume integral}}.\end{aligned} \tag{2.5}$$

Note that (2.5) is still exact in the weak formulation. To discretize the weak formulation, we now approximate the exact solution $\mathbf{u}(x,t)$ by piecewise linear polynomials, $\mathbf{u}_h(x,t)|_{I_j} \in P^1$, and the test function $v(x)$ by $v_h(x)|_{I_j} \in P^1$, where the subscript $h$ represents the approximate solution in polynomial space. Figure 2.2 shows the solution distribution of DG-$P^1$ method (a), compared to the first- (solid line) and

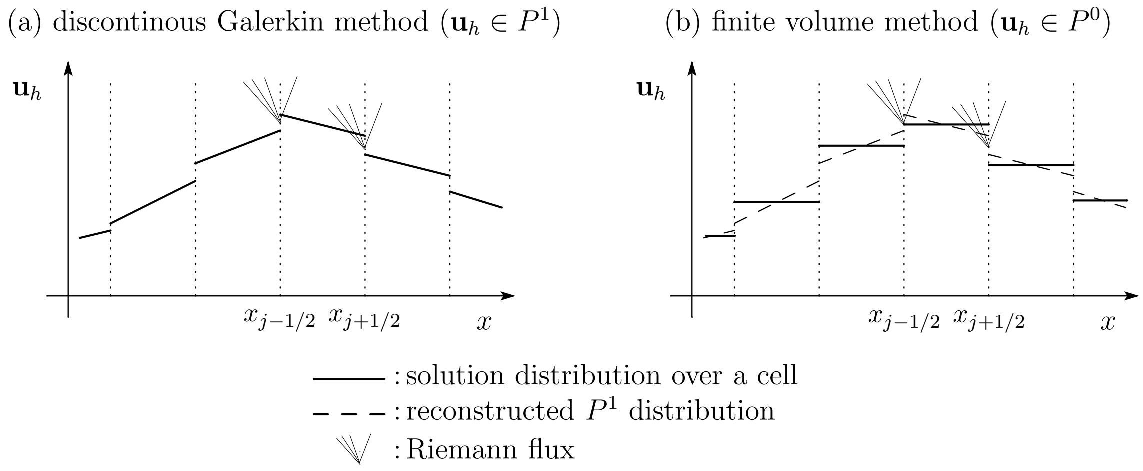

(a) discontinous Galerkin method ($\mathbf{u}_h \in P^1$)
(b) finite volume method ($\mathbf{u}_h \in P^0$)
$\mathbf{u}_h$
$x_{j-1/2}$
$x_{j+1/2}$
$x$
$\mathbf{u}_h$
$x_{j-1/2}$
$x_{j+1/2}$
$x$
: solution distribution over a cell
: reconstructed $P^1$ distribution
: Riemann flux

Figure 2.2: Solution distribution of a DG-$P^1$ method (a) and the first-order finite volume method (b) are compared. Dashed lines in (b) are reconstructed piecewise linear distributions of a second-order finite volume method.

second-order (dashed line) finite-volume method (b). The Legendre polynomials up to degree 1 are adopted for both basis (trial) and test functions, thus

$$\mathbf{u}_h(x,t) = \bar{\mathbf{u}}_j(t)\phi_0(x) + \overline{\Delta\mathbf{u}}_j(t)\phi_1(x), \tag{2.6a}$$

$$v_h(x) \in \operatorname{span}\{\phi_0(x), \phi_1(x)\}, \tag{2.6b}$$

where

$$\phi_0(x) = 1, \quad \phi_1(x) = \frac{x - x_j}{\Delta x}. \tag{2.7}$$

Here, the cell-average and the undivided gradient of $\mathbf{u}(x,t)$ in space are defined by

$$\bar{\mathbf{u}}_j(t) := \frac{1}{\Delta x} \int\limits_{I_j} \mathbf{u}(x,t)\, dx, \tag{2.8a}$$

$$\overline{\Delta\mathbf{u}}_j(t) := \frac{12}{\Delta x^2} \int\limits_{I_j} (x - x_j)\mathbf{u}(x,t)\, dx. \tag{2.8b}$$

Note that $\mathbf{u}(x,t) = \mathbf{u}_h(x,t) + O(\Delta x^2)$ in $x \in I_j$; however, the distributions of the true solution $\mathbf{u}(x,t)$ and the approximated polynomial function $\mathbf{u}_h(x,t)$ over the domain $I_j$ are equivalent in the weak sense due to the orthogonality of the Legendre polynomials,

$$\int\limits_{I_j} \mathbf{u}(x,t)\phi_k(x)\,dx \equiv \int\limits_{I_j} \mathbf{u}_h(x,t)\phi_k(x)\,dx, \quad k=0,1. \tag{2.9}$$

Once the basis and test functions are chosen, approximate governing equations are derived by adopting the basis functions $\phi_0(x)$ and $\phi_1(x)$ as the test function $v_h(x)$. Inserting $\phi_i(x)$ into $v_h(x)$ leads to two independent update equations:

- $v(x) = 1:$

$$\int\limits_{I_j} \mathbf{u}_h(x,t)\,dx\Big|_{t^n}^{t^{n+1}} = -\int\limits_{T^n} \mathbf{f}(\mathbf{u}_h(x,t))\Big|_{x_{j-1/2}}^{x_{j+1/2}} dt + \iint\limits_{I_j\times T^n} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(x,t))\,dxdt, \tag{2.10a}$$

- $v(x) = \dfrac{x-x_j}{\Delta x}:$

$$\begin{aligned}\int\limits_{I_j} \mathbf{u}_h(x,t)\frac{x-x_j}{\Delta x}\,dx\Big|_{t^n}^{t^{n+1}} &= -\int\limits_{T^n} \mathbf{f}(\mathbf{u}_h(x,t))\frac{x-x_j}{\Delta x}\Big|_{x_{j-1/2}}^{x_{j+1/2}} dt \\ &\quad + \frac{1}{\Delta x}\iint\limits_{I_j\times T^n} \mathbf{f}(\mathbf{u}_h(x,t))\,dxdt + \iint\limits_{I_j\times T^n} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(x,t))\frac{x-x_j}{\Delta x}\,dxdt. \end{aligned} \tag{2.10b}$$

The time-evolution term and boundary integral can be further simplified by inserting (2.6a) into $\mathbf{u}_h(x,t)$, then

$$\Delta x\left[\bar{\mathbf{u}}_j^{n+1} - \bar{\mathbf{u}}_j^n\right] = \underbrace{-\int\limits_{T^n}\left[\mathbf{f}_{j+1/2}(t) - \mathbf{f}_{j-1/2}(t)\right]dt}_{\text{Boundary integral}} + \underbrace{\iint\limits_{I_j\times T^n}\frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(x,t))\,dxdt}_{\text{Volume integral}}, \tag{2.11a}$$

$$\begin{aligned}\frac{\Delta x}{12}\left[\overline{\Delta\mathbf{u}}_j^{n+1} - \overline{\Delta\mathbf{u}}_j^{n}\right] &= \underbrace{-\frac{1}{2}\int\limits_{T^n}\left[\mathbf{f}_{j+1/2}(t) + \mathbf{f}_{j-1/2}(t)\right]dt}_{\text{Boundary integral}} \\ &\quad + \underbrace{\frac{1}{\Delta x}\iint\limits_{I_j\times T^n}\mathbf{f}(\mathbf{u}_h(x,t))\,dxdt}_{\text{Volume integral}} + \underbrace{\iint\limits_{I_j\times T^n}\frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(x,t))\frac{x-x_j}{\Delta x}\,dxdt}_{\text{Volume integral}}, \end{aligned} \tag{2.11b}$$

where the flux at the interface $x_{j\pm1/2}$ is $\mathbf{f}_{j\pm1/2}(t) = \mathbf{f}(\mathbf{u}_h(x_{j\pm1/2}, t))$. The approximation made so far is in the solution representation $\mathbf{u}_h \in P^1$, which is polynomial of degree 1, and in using the same polynomial basis for basis and test functions. The next step is approximating the boundary integral, $\int\limits_{T^n} (\cdot)\, dt$, for the interface flux, and the volume integral, $\iint\limits_{I_j\times T^n} (\cdot)\, dxdt$, for both source term and flux by quadrature. Previously, the method for a linear flux and source term was introduced [SvL06]. Also, a Fourier analysis of both semi-discrete second-order high-resolution Godunov (HR2) and DG(1) methods was presented [HSvL05]. Here, we extend the linear method to nonlinear flux and source terms. Fourier analyses of a fully discrete method are presented in Chapter IV.

### 2.2.2 Boundary Integral of the Flux

At the cell-interfaces, $x_{j\pm1/2}$, the time integral of a flux is approximated by the midpoint rule. Thus when a flux is integrated over the time interval in $[t^n, t^{n+k}]$, the flux at $t^{n+k/2}$ is considered as the averaged flux:

$$\int\limits_{t^n}^{t^{n+k}} \mathbf{f}_{j\pm1/2}(t)\, dt \approx (k\Delta t)\ \mathbf{f}_{j\pm1/2}^{n+k/2}, \tag{2.12}$$

where $\Delta t := t^{n+1} - t^n$ and $k \in [0, 1]$. Since the approximated solution at the cell-interface, $\mathbf{u}_h(x_{j\pm1/2}, t)$, is discontinuous, and not uniquely determined, a Riemann problem is solved exactly or approximately to compute the interface-flux. Let $\hat{\mathbf{f}}$ be the solution of a Riemann problem, then

$$\mathbf{f}_{j\pm1/2}^{n+k/2} \approx \hat{\mathbf{f}}_{j\pm1/2}\big(\mathbf{u}_{j\pm1/2,L}^{n+k/2}, \mathbf{u}_{j\pm1/2,R}^{n+k/2}\big). \tag{2.13}$$

For a linear flux, $\mathbf{f}(\mathbf{u}) = \mathbf{A}\mathbf{u}$ where $\mathbf{A} \in \mathbb{R}^{m\times m}$, the upwind flux is given by

$$\hat{\mathbf{f}}(\mathbf{u}_L, \mathbf{u}_R) = \mathbf{A}^+\mathbf{u}_L + \mathbf{A}^-\mathbf{u}_R, \tag{2.14}$$

with $\mathbf{A}^{\pm} = \mathbf{R}\boldsymbol{\Lambda}^{\pm}\mathbf{L}$, where $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ is the diagonal matrix of eigenvalues of $\mathbf{A}$. For a nonlinear flux, common approximate Riemann solvers are given by the following form:

$$\hat{\mathbf{f}}(\mathbf{u}_L, \mathbf{u}_R) = \frac{1}{2}\left[\mathbf{f}(\mathbf{u}_R) + \mathbf{f}(\mathbf{u}_L)\right] - \frac{1}{2}\mathbf{Q}\left[\mathbf{u}_R - \mathbf{u}_L\right], \tag{2.15}$$

with

$$\mathbf{Q} = \begin{cases} \mathbf{R}|\boldsymbol{\Lambda}|\mathbf{L} & \text{Roe (upwind), all waves [Roe81],} \\ \dfrac{(1-\alpha)\lambda^+\lambda^- + \alpha(\lambda^- V^+ + \lambda^+ V^-)}{\lambda^+ - \lambda^-} & \text{HLLL or HLL2, three waves} \\ \qquad\qquad\qquad -\dfrac{1}{2}\dfrac{\lambda^+ + \lambda^-}{\lambda^+ - \lambda^-}\dfrac{\Delta\mathbf{f}}{\Delta\mathbf{u}} & \qquad\qquad\text{[Lin02, HLvL83],} \\ \dfrac{\lambda^+\lambda^-}{\lambda^+ - \lambda^-} - \dfrac{1}{2}\dfrac{\lambda^+ + \lambda^-}{\lambda^+ - \lambda^-}\dfrac{\Delta\mathbf{f}}{\Delta\mathbf{u}} & \text{HLL1, two waves [HLvL83],} \\ |\lambda_i|_{\max}\, \mathbf{I} & \text{Rusanov, one wave [Rus62],} \\ \dfrac{\Delta x}{\Delta t}\, \mathbf{I} & \text{Lax–Friedrichs, zero wave [Lax54],} \end{cases}$$

where $\mathbf{I} \in \mathbb{R}^{m\times m}$ is the identity matrix. The cell-interface values at the half-time, $\mathbf{u}_{j+1/2,L/R}^{n+k/2}$, are obtained by a Taylor-series expansion of $\mathbf{u}(x,t)$ in space and time using the Cauchy–Kovalevskaya (or Lax–Wendroff) procedure [Lan98, p. 317]: replacing the time derivative by the spatial derivative,

$$\begin{aligned} \mathbf{u}(x,t) &= \mathbf{u}(x_j,t^n) + (x-x_j)\partial_x\mathbf{u}(x_j,t^n) + (t-t^n)\partial_t\mathbf{u}(x_j,t^n) + O\big(\Delta x^2, \Delta t^2, \Delta x\Delta t\big) \\ &\approx \mathbf{u}_j^n + (x-x_j)\partial_x\mathbf{u}_j^n + (t-t^n)\left[-\partial_x\mathbf{f}(\mathbf{u}_j^n) + \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_j^n)\right] \\ &= \mathbf{u}_j^n + \big[(x-x_j)\mathbf{I} - (t-t^n)\mathbf{A}(\mathbf{u}_j^n)\big]\,\partial_x\mathbf{u}_j^n + \frac{t-t^n}{\epsilon}\mathbf{s}(\mathbf{u}_j^n), \quad x \in I_j,\ t \in T^n, \end{aligned} \tag{2.16}$$

where the flux Jacobian $\mathbf{A}(\mathbf{u}) := \dfrac{\partial\mathbf{f}}{\partial\mathbf{u}}$ with $\mathbf{A}(\mathbf{u})\colon \mathbb{R}^m \to \mathbb{R}^{m\times m}$. Replacing point values, $\mathbf{u}_j^n$ and $\partial_x\mathbf{u}_j^n$, by cell-averages and undivided gradient values preserves the second-order accuracy since $\mathbf{u}_j^n = \bar{\mathbf{u}}_j^n + O(\Delta x^2)$ and $\Delta x\partial_x\mathbf{u}_j^n = \overline{\Delta\mathbf{u}}_j^n + O(\Delta x^3)$.

Also, the source term is evaluated from the approximated solution, $\mathbf{u}(x,t)$, instead of the known solution, $\bar{\mathbf{u}}_j^n$, to make the source term implicit. This again does not affect the order of approximation. Finally, the approximation of the state variable $\mathbf{u}(x,t)$ in domain $I_j \times T^n$ is given by

$$\mathbf{u}(x,t) \approx \bar{\mathbf{u}}_j^n + \big[(x-x_j)\mathbf{I} - (t-t^n)\mathbf{A}(\bar{\mathbf{u}}_j^n)\big]\frac{\overline{\Delta\mathbf{u}}_j^n}{\Delta x} + \frac{t-t^n}{\epsilon}\mathbf{s}\big(\mathbf{u}(x,t)\big). \tag{2.17}$$

Inserting $x = x_j + \dfrac{\Delta x}{2}, x_{j+1} - \dfrac{\Delta x}{2}$ and $t = t^n + \dfrac{k\Delta t}{2}$ leads to the cell-interface values for a Riemann solver at $(x_{j+1/2}, t^{n+k/2})$, where $k \in [0,1]$ will be used below to define quadrature points:

$$\begin{aligned}\mathbf{u}_{j+1/2,L}^{n+k/2} &= \mathbf{u}_j(x_j + \Delta x/2,\ t^n + k\Delta t/2)\\ &= \bar{\mathbf{u}}_j^n + \frac{1}{2}\left[\mathbf{I} - \frac{k\Delta t}{\Delta x}\mathbf{A}(\bar{\mathbf{u}}_j^n)\right]\overline{\Delta\mathbf{u}}_j^n + \frac{k\Delta t}{2\epsilon}\ \mathbf{s}\big(\mathbf{u}_{j+1/2,L}^{n+k/2}\big),\end{aligned} \tag{2.18a}$$

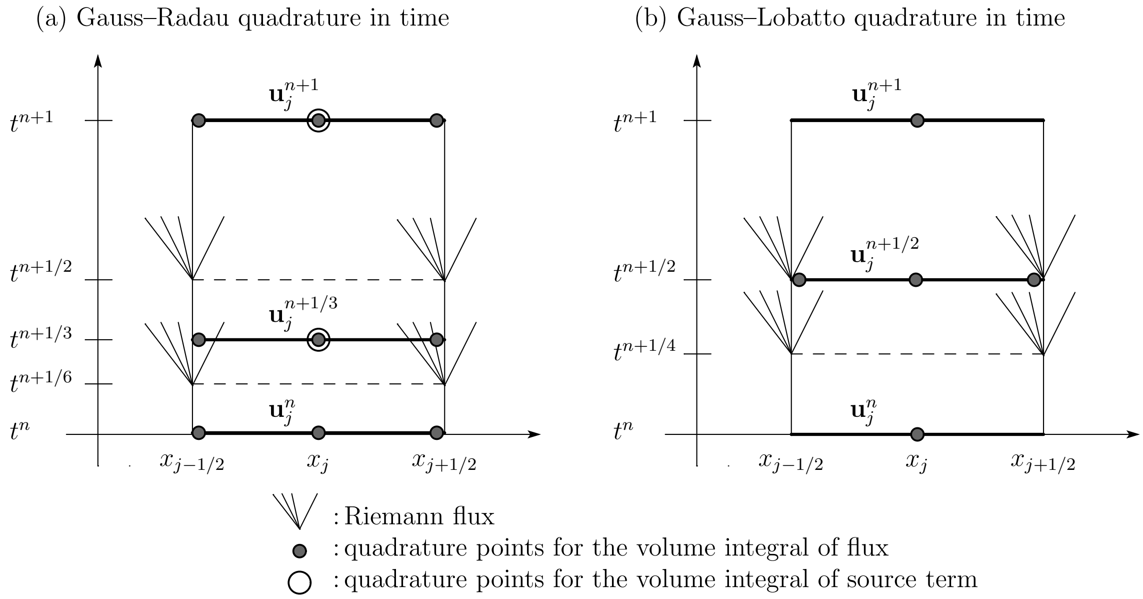
$$\begin{aligned}\mathbf{u}_{j+1/2,R}^{n+k/2} &= \mathbf{u}_{j+1}(x_{j+1} - \Delta x/2,\ t^n + k\Delta t/2)\\ &= \bar{\mathbf{u}}_{j+1}^n - \frac{1}{2}\left[\mathbf{I} + \frac{k\Delta t}{\Delta x}\mathbf{A}(\bar{\mathbf{u}}_{j+1}^n)\right]\overline{\Delta\mathbf{u}}_{j+1}^n + \frac{k\Delta t}{2\epsilon}\ \mathbf{s}\big(\mathbf{u}_{j+1/2,R}^{n+k/2}\big).\end{aligned} \tag{2.18b}$$

Note that the implicit character is caused by the source term. In practice, for fluid dynamics equations, this predictor step can be simplified by using a different form of governing equations. Typically, the source term is a homogeneous function of degree one with respect to $\mathbf{w}$, hence it satisfies

$$\frac{1}{\epsilon(\mathbf{w})}\mathbf{s}(\mathbf{w}) = \mathbf{Q}_{\mathrm{w}}\mathbf{w}, \tag{2.19}$$

where $\mathbf{w} \in \mathbb{R}^m$ is the vector of primitive variables, and $\mathbf{Q}_{\mathrm{w}} := \dfrac{\partial\mathbf{s}(\mathbf{w})}{\partial\mathbf{w}} \in \mathbb{R}^{m\times m}$. The constant coefficient matrix $\mathbf{Q}_{\mathrm{w}}$ can be inverted analytically, thus the predictor step is evaluated explicitly as follows:

$$\mathbf{w}(x,t) = \left[\mathbf{I} - (t-t^n)\,\mathbf{Q}_{\mathrm{w}}\right]^{-1}\left[\mathbf{w}_j^n + \Big((x-x_j)\,\mathbf{I} - (t-t^n)\,\mathbf{A}(\mathbf{w}_j^n)\Big)\frac{\Delta\mathbf{w}_j^n}{\Delta x}\right]. \tag{2.20}$$


(a) Gauss–Radau quadrature in time

(b) Gauss–Lobatto quadrature in time

Figure 2.3: The locations of quadrature points for the volume integral of the flux and source terms are shown in the space-time domain, $[x_{j-1/2}, x_{j+1/2}] \times [t^n, t^{n+1}]$. Corresponding locations where wave interactions occur by a Riemann solver are also presented. Both Gauss–Radau and Gauss–Lobatto quadratures require two Riemann fluxes at each interface, and one intermediate stage at either $t^{n+1/3}$ or $t^{n+1/2}$. In this thesis, the Gauss–Lobatto quadrature is only used for hyperbolic conservation laws, therefore, Figure (b), does not contain quadrature points (○) for the source term.

Once primitive variables at the half-time step are obtained, these values are used as inputs for a Riemann solver (2.15) to compute the interface flux. The locations where a Riemann solver is applied over the space-time domain are shown in Figure 2.3.

### 2.2.3 Volume Integral of the Source Term

When a DG spatial discretization with a piecewise linear solution representation is applied to hyperbolic-relaxation equations, three volume integrals appear in (2.11): one is in the update equation of $\bar{\mathbf{u}}_j$, and the other two are in the update equation of $\overline{\Delta \mathbf{u}}_j$. The same strategy as in the upwind moment method can be applied to the latter two volume integrals, assuming state variables in all quadrature points in time are already known. This is true as long as the update equation for $\bar{\mathbf{u}}_j$ is solved first. A difficulty arises when a quadrature rule is applied to the volume integral of the source term in (2.11a). Since this is the update equation of the cell-average variables $\bar{\mathbf{u}}_j$, the updated state variables at a quadrature point, $\bar{\mathbf{u}}_j^{n+1}$, are still unknown. Yet, we can update $\bar{\mathbf{u}}_j$ by iterating with a quadrature for the volume integral of the source term; however, the quadrature rule or points have to be chosen carefully when solving systems of stiff ODEs. Again, see Figure 2.3 for quadrature points.

Here, we focus on constructing a third-order discretization in time for the source term, while accepting second-order accuracy in space, so as to circumvent the quadrature in space, more specifically, removing the $\overline{\Delta\mathbf{u}}_j$ dependence in the volume integral of $\mathbf{s}(\mathbf{u}_h)$ in (2.11a). Thus, the following source-term expansion in space is adopted:

$$\frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(x,t)) = \frac{1}{\epsilon(\bar{\mathbf{u}}_j(t))}\mathbf{s}(\bar{\mathbf{u}}_j(t)) + \frac{x-x_j}{\Delta x}\mathbf{Q}(\bar{\mathbf{u}}_j(t))\,\overline{\Delta\mathbf{u}}_j(t) + O\big(\Delta x^2\big)\,, \tag{2.21}$$

where $\mathbf{Q}(\mathbf{u}) := \dfrac{\partial(\mathbf{s}/\epsilon)}{\partial\mathbf{u}}$ with $\mathbf{Q}(\mathbf{u})\colon \mathbb{R}^m \to \mathbb{R}^{m\times m}$. Inserting (2.21) into the volume integral of the source term in (2.11a) leads to

$$\iint\limits_{I_j\times T^n} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(x,t))\,dxdt = \Delta x\int\limits_{T^n}\frac{1}{\epsilon(\bar{\mathbf{u}}_j(t))}\mathbf{s}(\bar{\mathbf{u}}_j(t))\,dt + O\big(\Delta x^3\big)\,. \tag{2.22}$$

This approximation removes the coupling between (2.11a) and (2.11b), allowing independent updates of the two equations.

In order to integrate the above equation in time, quadrature points have to be chosen carefully in view of its stiffness. Also, to ensure stability in the stiff regime ($\epsilon \ll 1$), the time-integration method for the source term needs to be implicit. Previously, the backward Euler method, which is only first-order accurate, was used to

| $s$-stage RK method | order $p$ | stage order $\tilde{p}$ | linear stability | nonlinear stability |
|---|---|---|---|---|
| Gauss | $2s$ | $s$ | $A$-stability | algebraic stability |
| Radau IA | $2s-1$ | $s-1$ | $L$-stability | algebraic stability |
| Radau IIA | $2s-1$ | $s$ | $L$-stability | algebraic stability |
| Lobatto IIIA | $2s-2$ | $s$ | $A$-stability | No algebraic stability |
| Lobatto IIIB | $2s-2$ | $s-2$ | $A$-stability | No algebraic stability |
| Lobatto IIIC | $2s-2$ | $s-1$ | $L$-stability | algebraic stability |

Table 2.1: The properties of the classes of implicit Runge–Kutta methods are tabulated [Lam91]. The order $p$ is based on the linear theory, and the stage order $\tilde{p}$ is the lower bound obtained by the nonlinear theory. Thus, in general, the order of a method is within $[\tilde{p}, p]$.

integrate the source term and obtain the intermediate stage [SvL06]. Unfortunately, linear analysis shows that the source-term discretization is overall only second-order accurate due to the first-order temporal discretization in the intermediate step. In order to achieve high-order accuracy and circumvent the stiffness, a fully-implicit method is preferable. The properties of the classes of implicit Runge–Kutta methods are tabulated in Table 2.1 [Lam91, p. 250, 282]. Based on these properties, the Radau IA, Radau IIA, and Lobatto IIIC methods, which possess both $L$-stability and nonlinear stability, are candidates for time integration of the source term. In order to achieve third-order accuracy in time, the Radau IA/IIA methods require only two stages ($s = 2, p = 3$), whereas the Lobatto IIIC method requires three stages ($s = 3, p = 4$). To minimize the computational cost of the fully-implicit procedure for the source term in a scheme, we chose the former, particularly the Radau IIA method, for the source-term integral. (This is an original contribution.) Hence, the volume integral of the source term is approximated as

$$\iint\limits_{I_j \times T^n} \frac{1}{\epsilon} \mathbf{s}(\mathbf{u}_h(x,t))\, dxdt \approx \Delta x \Delta t \left[ \frac{3}{4} \frac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1/3})}{\epsilon(\bar{\mathbf{u}}_j^{n+1/3})} + \frac{1}{4} \frac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1})}{\epsilon(\bar{\mathbf{u}}_j^{n+1})} \right], \tag{2.23}$$

where a new intermediate stage at time level $n + \dfrac{1}{3}$ is introduced. Figure 2.3(a) shows the quadrature points of the Radau IIA method for the source term by the circle symbol (◯). The overall update equations are given by

$$\bar{\mathbf{u}}_j^{n+1/3} = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{3}\frac{1}{\Delta x}\underbrace{\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/6} - \hat{\mathbf{f}}_{j-1/2}^{n+1/6}\right]}_{\text{explicit}} + \frac{\Delta t}{3}\underbrace{\left[\frac{5}{4}\frac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1/3})}{\epsilon(\bar{\mathbf{u}}_j^{n+1/3})} - \frac{1}{4}\frac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1})}{\epsilon(\bar{\mathbf{u}}_j^{n+1})}\right]}_{\text{implicit}}, \tag{2.24a}$$

$$\bar{\mathbf{u}}_j^{n+1} = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{\Delta x}\underbrace{\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/2} - \hat{\mathbf{f}}_{j-1/2}^{n+1/2}\right]}_{\text{explicit}} + \Delta t\underbrace{\left[\frac{3}{4}\frac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1/3})}{\epsilon(\bar{\mathbf{u}}_j^{n+1/3})} + \frac{1}{4}\frac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1})}{\epsilon(\bar{\mathbf{u}}_j^{n+1})}\right]}_{\text{implicit}}. \tag{2.24b}$$

To solve this system numerically, first the interface fluxes are computed explicitly. Then, the problem reduces to finding the solutions of systems of nonlinear algebraic equations of the following form:

$$\mathbf{u}_{\mathrm{A}} = \mathbf{C}_{\mathrm{f}} + \mathbf{C}_{\mathrm{s}}\ \mathbf{s}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}}), \tag{2.25}$$

where

$$\mathbf{u}_{\mathrm{A}} = \begin{pmatrix} \bar{\mathbf{u}}_j^{n+1/3} \\ \bar{\mathbf{u}}_j^{n+1} \end{pmatrix}, \qquad \mathbf{s}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}}) = \begin{pmatrix} \dfrac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1/3})}{\epsilon(\bar{\mathbf{u}}_j^{n+1/3})} \\ \dfrac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1})}{\epsilon(\bar{\mathbf{u}}_j^{n+1})} \end{pmatrix}, \tag{2.26a}$$

$$\mathbf{C}_{\mathrm{f}} = \begin{pmatrix} \bar{\mathbf{u}}_j^n - \dfrac{\Delta t}{3}\dfrac{1}{\Delta x}\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/6} - \hat{\mathbf{f}}_{j-1/2}^{n+1/6}\right] \\ \bar{\mathbf{u}}_j^n - \dfrac{\Delta t}{\Delta x}\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/2} - \hat{\mathbf{f}}_{j-1/2}^{n+1/2}\right] \end{pmatrix}, \qquad \mathbf{C}_{\mathrm{s}} = \Delta t \begin{pmatrix} \dfrac{5}{12}\mathbf{I} & -\dfrac{1}{12}\mathbf{I} \\ \dfrac{3}{4}\mathbf{I} & \dfrac{1}{4}\mathbf{I} \end{pmatrix}, \tag{2.26b}$$

with $\mathbf{u}_{\mathrm{A}}, \mathbf{C}_{\mathrm{f}} \in \mathbb{R}^{2m}, \mathbf{C}_{\mathrm{s}} \in \mathbb{R}^{2m\times 2m}$, and $\mathbf{s}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}})\colon \mathbb{R}^{2m} \to \mathbb{R}^{2m}$. Here, the Newton–Raphson method is employed to find the solution, thus the iteration process at step $p$ is given by

$$\mathbf{u}_{\mathrm{A}}^{p+1} = \mathbf{u}_{\mathrm{A}}^{p} - \left[\mathbf{I}_{\mathrm{A}} - \mathbf{C}_{\mathrm{s}}\mathbf{Q}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}}^{p})\right]^{-1}\left[\mathbf{u}_{\mathrm{A}}^{p} - \mathbf{C}_{\mathrm{f}} - \mathbf{C}_{\mathrm{s}}\ \mathbf{s}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}}^{p})\right], \quad p = 0, 1, 2\ldots, \tag{2.27}$$

where

$$\mathbf{Q}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}}) := \frac{\partial \mathbf{s}_{\mathrm{A}}}{\partial \mathbf{u}_{\mathrm{A}}} = \begin{pmatrix} \mathbf{Q}(\bar{\mathbf{u}}_j^{n+1/3}) & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}(\bar{\mathbf{u}}_j^{n+1}) \end{pmatrix}, \quad \mathbf{Q}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}})\colon \mathbb{R}^{2m} \to \mathbb{R}^{2m\times 2m}, \tag{2.28}$$

and $\mathbf{I}_{\mathrm{A}} \in \mathbb{R}^{2m\times 2m}$. To start up the iteration, the initial guess at the time level $n$, $\mathbf{u}_{\mathrm{A}}^{0} = \left[\bar{\mathbf{u}}_j^n; \bar{\mathbf{u}}_j^n\right]$, is used. The iteration on the system of $2m$ equations can be reduced to $2(m-l)$ equations where $l < n$, since the first $l$ entries of the source term are zero.

In general, when the Newton–Raphson method is implemented, it is more efficient to adopt $LU$-decomposition to the matrix $[\mathbf{I}_{\mathrm{A}} - \mathbf{C}_{\mathrm{s}}\mathbf{Q}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}}^{p})]$, and solve the system of linear algebraic equations instead of inverting it. However, the structure of the source term is typically simple, and the inverse matrix, $[\mathbf{I}_{\mathrm{A}} - \mathbf{C}_{\mathrm{s}}\mathbf{Q}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}}^{p})]^{-1}$, can be obtained analytically as a function of $\mathbf{u}_{\mathrm{A}}$. The advantage of the choice of hyperbolic-relaxation equations over the NS equations is clear here: since the method is point-implicit due to the source term, the inverse of the matrix is still local, whereas the implicit treatment of the diffusion term in the NS equations makes the domain of dependence global.

### 2.2.4 Volume Integral of the Flux

#### Gauss–Lobatto Points (Original Upwind Moment Scheme)

First, we review Huynh's original upwind moment scheme [Huy06a]. The method utilizes the three-point Gauss–Lobatto quadrature (see Figure 2.3(b)) for both space and time integration of the flux, thus the volume integral of the flux is approximated by

$$\begin{aligned}
\iint\limits_{I_j\times T^n} \mathbf{f}(\mathbf{u}_h(x,t))\,dxdt &\approx \Delta t \int\limits_{I_j} \frac{1}{6}\left[\mathbf{f}(\mathbf{u}(x,t^n)) + 4\mathbf{f}(\mathbf{u}(x,t^{n+1/2})) + \mathbf{f}(\mathbf{u}(x,t^{n+1}))\right] dx \\
&\approx \Delta t \int\limits_{I_j} \mathbf{f}(\hat{\mathbf{u}}(x))\,dx \\
&\approx \Delta t \frac{\Delta x}{6}\left[\mathbf{f}(\hat{\mathbf{u}}(x_{j-1/2})) + 4\mathbf{f}(\hat{\mathbf{u}}(x_j)) + \mathbf{f}(\hat{\mathbf{u}}(x_{j+1/2}))\right], \qquad (2.29)
\end{aligned}$$

where

$$\hat{\mathbf{u}}(x) = \hat{\bar{\mathbf{u}}}_j + \widehat{\overline{\Delta \mathbf{u}}}_j \frac{x - x_j}{\Delta x}, \tag{2.30a}$$

$$\hat{\bar{\mathbf{u}}}_j = \frac{1}{6}\left(\bar{\mathbf{u}}_j^n + 4\bar{\mathbf{u}}_j^{n+1/2} + \bar{\mathbf{u}}_j^{n+1}\right), \tag{2.30b}$$

$$\widehat{\overline{\Delta \mathbf{u}}}_j = \frac{1}{2}\left(\overline{\Delta \mathbf{u}}_j^n + \overline{\Delta \mathbf{u}}_j^{n+1}\right). \tag{2.30c}$$

Here, $\hat{(\cdot)}$ denotes a time-averaged value. When the discretization of conservation laws, (2.11) with $\mathbf{s} = 0$, is considered, the volume integral appears only in the second equation (2.11b). Since the cell-average variables are updated in the first equation, $\bar{\mathbf{u}}_j^{n+1}$ is already known when the volume integral in the second equation is evaluated. Thus $\hat{\bar{\mathbf{u}}}_j$ is evaluated explicitly whereas $\overline{\Delta \mathbf{u}}_j^{n+1}$ in (2.30c) is still unknown, and an iterative process is required. To start the iteration, the slope at the time level $n$ is used as the initial guess; however, Huynh reported that no improvement was observed by iterations, and suggested to replace (2.30c) by

$$\widehat{\overline{\Delta \mathbf{u}}}_j = \overline{\Delta \mathbf{u}}_j^n. \tag{2.31}$$

When the flux is linear, $\mathbf{f}(\mathbf{u}) = \mathbf{A}\mathbf{u}$, the volume integral (2.29) is evaluated exactly in space and approximately in time, thus

$$\begin{aligned} \iint\limits_{I_j \times T^n} \mathbf{f}(\mathbf{u}_h(x,t))\, dxdt &= \Delta x \mathbf{A} \int\limits_{T^n} \bar{\mathbf{u}}_j(t)\, dt \\ &\approx \Delta x \Delta t\ \mathbf{A} \hat{\bar{\mathbf{u}}}_j, \end{aligned} \tag{2.32}$$

where $\hat{\bar{\mathbf{u}}}_j$ is given by (2.30b). In (2.30b), the new intermediate stage, $\bar{\mathbf{u}}_j^{n+1/2}$, is introduced, and computed in advance by updating over a half time step:

$$\begin{aligned} \bar{\mathbf{u}}_j^{n+1/2} &= \bar{\mathbf{u}}_j^n - \frac{1}{\Delta x} \int\limits_{t^n}^{t^{n+1/2}} \left[\mathbf{f}_{j+1/2}(t) - \mathbf{f}_{j-1/2}(t)\right] dt \\ &= \bar{\mathbf{u}}_j^n - \frac{\Delta t}{2} \frac{1}{\Delta x} \left[\hat{\mathbf{f}}_{j+1/2}^{n+1/4} - \hat{\mathbf{f}}_{j-1/2}^{n+1/4}\right]. \end{aligned} \tag{2.33}$$

Note that the method only requires the cell-average value at the half-time step, not the undivided gradient, since the slope at the time level $n$ is used in the entire space-time domain according to (2.31). Once the intermediate state is obtained, from (2.11), the final update equations become

$$\bar{\mathbf{u}}_j^{n+1} = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{\Delta x}\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/2} - \hat{\mathbf{f}}_{j-1/2}^{n+1/2}\right], \tag{2.34a}$$

$$\overline{\Delta\mathbf{u}}_j^{n+1} = \overline{\Delta\mathbf{u}}_j^n - \frac{\Delta t}{\Delta x}6\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/2} + \hat{\mathbf{f}}_{j-1/2}^{n+1/2} - \frac{2}{\Delta x\Delta t}\overline{\mathbf{f}(\hat{\mathbf{u}}_j)}\right], \tag{2.34b}$$

where $\overline{\mathbf{f}(\hat{\mathbf{u}}_j)}$ is given by (2.29). In summary, the original upwind moment scheme is a one-step method with one intermediate stage for the volume integral, requiring two Riemann solvers at each cell-interface.

### Gauss–Radau Points

When the hyperbolic-relaxation equations are considered, quadrature points for the flux in (2.11b) need to be modified based on the quadrature for the source term. Since we adopt the two-point Radau IIA method (2.23) as the time integrator for the source term, the same Radau points are employed for the volume integral of the flux. The bullet symbols (●) in Figure 2.3(a) on page 40 represent the location of quadrature points for the flux over the space-time domain. Consequently, the Gauss–Radau quadrature in time and the Gauss–Lobatto quadrature in space are applied:

$$\begin{aligned}\iint\limits_{I_j\times T^n} \mathbf{f}\big(\mathbf{u}_h(x,t)\big)\,dxdt &= \Delta t\int\limits_{I_j}\left[\frac{3}{4}\mathbf{f}\big(\mathbf{u}(x,t^{n+1/3})\big) + \frac{1}{4}\mathbf{f}\big(\mathbf{u}(x,t^{n+1})\big)\right]dx + O\big(\Delta t^4\big)\\ &\approx \Delta x\Delta t\left(\frac{3}{4}\bar{\mathbf{f}}^{n+1/3} + \frac{1}{4}\bar{\mathbf{f}}^{n+1}\right),\end{aligned} \tag{2.35}$$

where

$$\bar{\mathbf{f}}^k = \frac{1}{6}\left[\mathbf{f}\big(\tilde{\mathbf{u}}^k_{j-1/2}\big) + 4\mathbf{f}\big(\tilde{\mathbf{u}}^k_j\big) + \mathbf{f}\big(\tilde{\mathbf{u}}^k_{j+1/2}\big)\right], \tag{2.36a}$$

$$\tilde{\mathbf{u}}^k(x) = \bar{\mathbf{u}}^k_j + \overline{\Delta\mathbf{u}}^n_j\,\frac{x - x_j}{\Delta x}. \tag{2.36b}$$

Here, the undivided gradient is frozen at the time level $n$ in order to keep the treatment explicit. In the update equation of the undivided gradient, the intermediate-stage value, $\overline{\Delta\mathbf{u}}^{n+1/3}_j$, is required, and another volume integral of the flux over the time domain $T^{n'} = [t^n, t^{n+1/3}]$ is needed. Unlike in the method for hyperbolic conservation laws, the intermediate slope quantities are necessary due to the fully implicit treatment of the source term. Since the flux is not a stiff term and the cell-averaged variables at the quadrature points $k = 0, \dfrac{1}{3}$ are known, the trapezoidal rule is applied in time while the Gauss–Lobatto quadrature is applied in space:

$$\begin{aligned}\iint\limits_{I_j\times T^{n'}} \mathbf{f}\big(\mathbf{u}_h(x,t)\big)\,dxdt &= \frac{\Delta t}{3}\int\limits_{I_j}\frac{1}{2}\left[\mathbf{f}\big(\mathbf{u}(x,t^n)\big) + \mathbf{f}\big(\mathbf{u}(x,t^{n+1/3})\big)\right]\,dx + O\big(\Delta t^3\big)\\ &\approx \frac{\Delta x\Delta t}{6}\left(\bar{\mathbf{f}}^n + \bar{\mathbf{f}}^{n+1/3}\right),\end{aligned} \tag{2.37}$$

where spatial averaged fluxes are obtained by (2.36).

### 2.2.5 Integral of the Moment of the Source Term

The second-order approximation of the source term, (2.21), is inserted into the volume integral of the moment of the source term in (2.11b), and the Gauss–Radau quadrature is used in time:

$$\begin{aligned}\iint\limits_{I_j\times T^n} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(x,t))\frac{x-x_j}{\Delta x}\,dxdt &= \frac{\Delta x}{12}\int\limits_{T^n}\mathbf{Q}(\bar{\mathbf{u}}_j(t))\overline{\Delta\mathbf{u}}_j(t)\,dt + O\big(\Delta x^3\big)\\ &\approx \frac{\Delta x\Delta t}{12}\left[\frac{3}{4}\mathbf{Q}(\bar{\mathbf{u}}^{n+1/3}_j)\overline{\Delta\mathbf{u}}^{n+1/3}_j + \frac{1}{4}\mathbf{Q}(\bar{\mathbf{u}}^{n+1}_j)\overline{\Delta\mathbf{u}}^{n+1}_j\right].\end{aligned} \tag{2.38}$$

Following the same procedure as the cell-average update, the Radau IIA method is adopted for the source term, and the final update formulas for the undivided gradient are given by

$$
\begin{aligned}
\overline{\Delta\mathbf{u}}_j^{n+1/3} &= \overline{\Delta\mathbf{u}}_j^{n} - \frac{\Delta t}{3}\frac{6}{\Delta x}\underbrace{\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/6} + \hat{\mathbf{f}}_{j-1/2}^{n+1/6} - \frac{2}{\Delta x\Delta t}\overline{\mathbf{f}(\mathbf{u}_j^{n+1/6})}\right]}_{\text{explicit}} \\
&\quad + \frac{\Delta t}{3}\frac{1}{\epsilon}\underbrace{\left[\frac{5}{4}\mathbf{Q}(\bar{\mathbf{u}}_j^{n+1/3})\overline{\Delta\mathbf{u}}_j^{n+1/3} - \frac{1}{4}\mathbf{Q}(\bar{\mathbf{u}}_j^{n+1})\overline{\Delta\mathbf{u}}_j^{n+1}\right]}_{\text{implicit}},
\end{aligned} \tag{2.39a}
$$

$$
\begin{aligned}
\overline{\Delta\mathbf{u}}_j^{n+1} &= \overline{\Delta\mathbf{u}}_j^{n} - \frac{\Delta t}{\Delta x}6\underbrace{\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/2} + \hat{\mathbf{f}}_{j-1/2}^{n+1/2} - \frac{2}{\Delta x\Delta t}\overline{\mathbf{f}(\mathbf{u}_j^{n+1/2})}\right]}_{\text{explicit}} \\
&\quad + \frac{\Delta t}{\epsilon}\underbrace{\left[\frac{3}{4}\mathbf{Q}(\bar{\mathbf{u}}_j^{n+1/3})\overline{\Delta\mathbf{u}}_j^{n+1/3} + \frac{1}{4}\mathbf{Q}(\bar{\mathbf{u}}_j^{n+1})\overline{\Delta\mathbf{u}}_j^{n+1}\right]}_{\text{implicit}},
\end{aligned} \tag{2.39b}
$$

where $\overline{\mathbf{f}(\mathbf{u}_j^{n+1/6})}$ and $\overline{\mathbf{f}(\mathbf{u}_j^{n+1/2})}$ are obtained by (2.37) and (2.35) respectively. Once the interface fluxes and volume integrals of fluxes are computed explicitly, the problem is reduced to solving a system of linear algebraic equations:

$$
\Delta\mathbf{u}_{\mathrm{A}} = \Delta\mathbf{C}_{\mathrm{f}} + \mathbf{C}_{\mathrm{s}}\mathbf{Q}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}})\Delta\mathbf{u}_{\mathrm{A}}, \tag{2.40}
$$

where

$$
\Delta\mathbf{u}_{\mathrm{A}} = \begin{pmatrix} \overline{\Delta\mathbf{u}}_j^{n+1/3} \\ \overline{\Delta\mathbf{u}}_j^{n+1} \end{pmatrix}, \tag{2.41a}
$$

$$
\Delta\mathbf{C}_{\mathrm{f}} = \begin{pmatrix} \overline{\Delta\mathbf{u}}_j^{n} - \dfrac{\Delta t}{3}\dfrac{6}{\Delta x}\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/6} + \hat{\mathbf{f}}_{j-1/2}^{n+1/6} - \dfrac{2}{\Delta x\Delta t}\overline{\mathbf{f}(\mathbf{u}_j^{n+1/6})}\right] \\ \overline{\Delta\mathbf{u}}_j^{n} - \dfrac{\Delta t}{\Delta x}6\left[\mathbf{f}_{j+1/2}^{n+1/2} + \mathbf{f}_{j-1/2}^{n+1/2} - \dfrac{2}{\Delta x\Delta t}\overline{\mathbf{f}(\mathbf{u}_j^{n+1/2})}\right] \end{pmatrix}, \tag{2.41b}
$$

with $\Delta\mathbf{u}_{\mathrm{A}}, \Delta\mathbf{C}_{\mathrm{f}} \in \mathbb{R}^{2m}$. Since $\mathbf{u}_{\mathrm{A}}$ is already known by (2.27), no iteration is required, and $\Delta\mathbf{u}_{\mathrm{A}}$ is obtained by

$$
\Delta\mathbf{u}_{\mathrm{A}} = \left[\mathbf{I}_{\mathrm{A}} - \mathbf{C}_{\mathrm{s}}\mathbf{Q}_{\mathrm{A}}(\mathbf{u}_{\mathrm{A}})\right]^{-1}\Delta\mathbf{C}_{\mathrm{f}}. \tag{2.42}
$$

As mentioned previously, the structure of the source term is typically simple; the inverse of the above matrix can be obtained analytically.

## 2.3 Extension to Multidimensional Equations

In this section, we extend the 1-D DG(1)–Hancock method described in the previous section to multidimensional problems. Here, we closely follow the notations conventionally adopted in the finite-element community. Nevertheless, sufficient explanations of notations and terminologies are provided for those who are more accustomed to finite-volume or finite-difference methodologies. Interested readers are referred to the following books for the rigorous mathematical foundation of the finite-element methodology: [ZTZ05, FFS03, BS02].

As to the semi-discrete discontinuous Galerkin methods combined with Runge–Kutta time-marching (DG($k$)–RK methods), Cockburn and Shu extended their one-dimensional formulation [CLS89] to triangular grids [CHS90]; later, Bey and Oden extended such methods to quadrilateral elements of arbitrary polynomial degree [BO91]. Prior to the development of multidimensional DG($k$)–RK methods, Allmaras and Giles indeed developed the semi-discrete $P^1$ discontinuous Galerkin method combined with the third-order Runge–Kutta method, DG(1)–RK3, for quadrilateral elements [AG87, All89]. However, they did not denote their method as a discontinuous Galerkin method, and this might be a reason why their seminal work is little known in the community nowadays. Their method is the direct extension of Van Leer's scheme III (for a scalar linear equation) to the two-dimensional Euler/NS equations.

Subsequently, Halt and Agarwal extended the Allmaras–Giles method to higher order and triangular elements, and they denote the method as the "moment method"

[HA92, AH99]. Borrel and Berde also adopted the moment approach, and presented a few numerical results for the two-dimensional Euler and Navier–Stokes equations [BB95, BB98]. The specific aspect of the moment method is that, even though the formulation is now recognized as a semi-discrete discontinuous Galerkin method, the method is discretized directly on the physical domain except for the volume integral; a typical DG method is discretized in a local $(\xi, \eta)$ coordinate system, to which the original governing equations, expressed in $(x, y)$, are transformed. An advantage of the formulation of the moment method is that the update equation for the cell-averaged quantities is always independent of the other update equations for slopes. In this section, the spatial approach of the moment method is taken, yet time integration is done by the Hancock method.

### 2.3.1 Ritz–Galerkin Method

The system of multidimensional hyperbolic-relaxation equations is given by

$$\partial_t \mathbf{u}(\boldsymbol{x}, t) + \nabla \cdot \mathbf{f}(\mathbf{u}(\boldsymbol{x}, t)) = \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}(\boldsymbol{x}, t)); \quad \boldsymbol{x} \in \Omega,\ t > 0, \tag{2.43}$$

where $\mathbf{u} \in \mathbb{R}^m$ is the vector of conserved quantities, $\mathbf{f} \in \mathbb{R}^{m\times 3}$ is the 3-D flux tensor, $\mathbf{s} \in \mathbb{R}^m$ is the source vector for the conservation form, and $\Omega$ is the domain of interest. Let $T$ be a time interval, then the solution $\mathbf{u}(\boldsymbol{x}, t)$ in (2.43) is defined over the entire spatial domain $\Omega$ and temporal domain, i.e. $\mathbf{u}(\boldsymbol{x}, t) \in \Omega \times T$. Figure 2.4(a) shows a schematic of notations introduced in the above equation. By specifying the initial condition, $\mathbf{u}(\boldsymbol{x}, 0) = \mathbf{u}_0(\boldsymbol{x})$, and sufficient boundary conditions along boundary $\partial\Omega$, the problem statement of the differential equations (2.43) is:

> Find $\mathbf{u}(\boldsymbol{x}, t)$ such that the solution $\mathbf{u}(\boldsymbol{x}, t)$ satisfies (2.43) for any $\boldsymbol{x} \in \Omega$ and any $t \in T$.

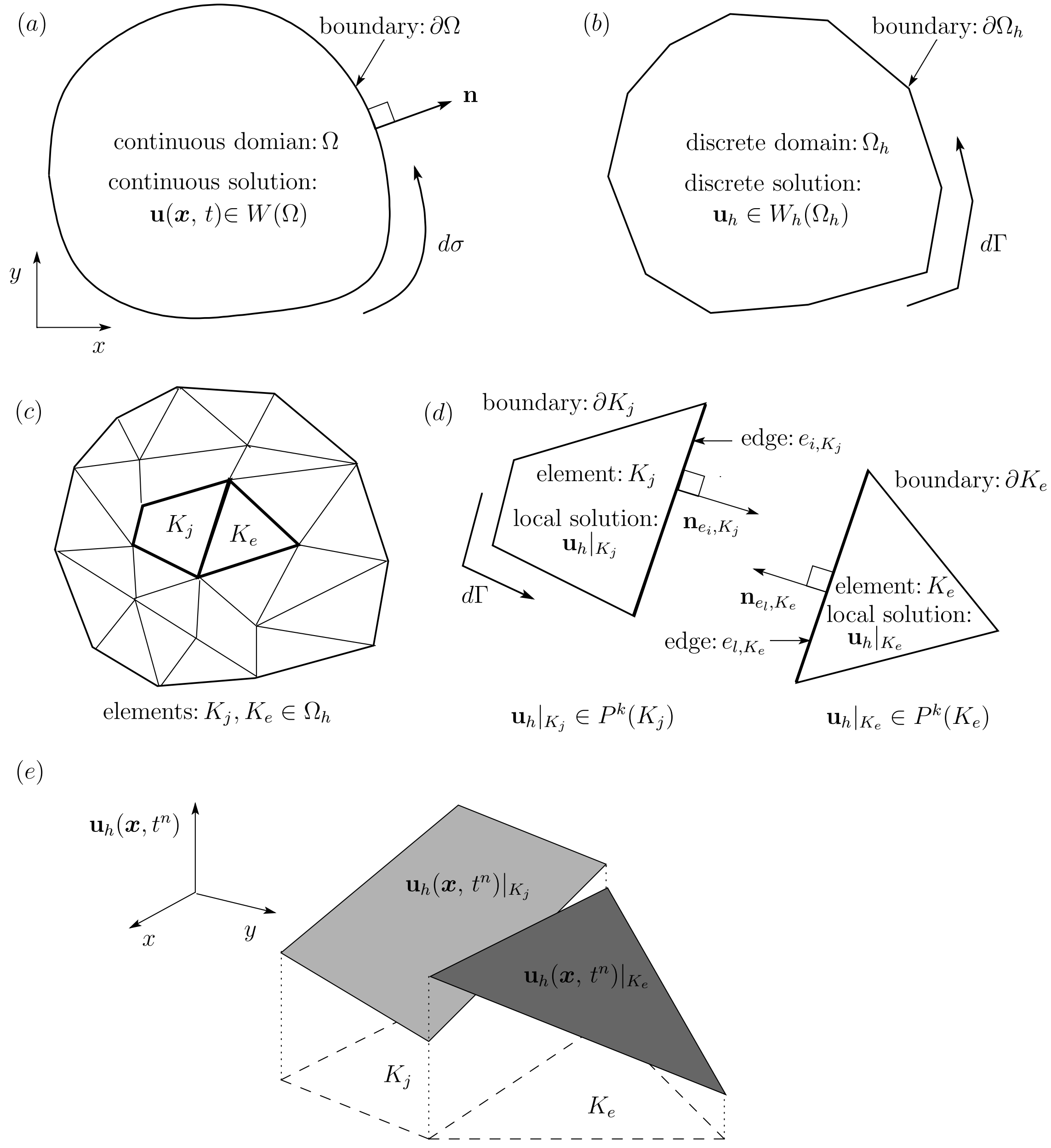

(a)
boundary: $\partial\Omega$
$\mathbf{n}$
continuous domian: $\Omega$
continuous solution:
$\mathbf{u}(\boldsymbol{x}, t) \in W(\Omega)$
$d\sigma$
$y$
$x$
(b)
boundary: $\partial\Omega_h$
discrete domain: $\Omega_h$
discrete solution:
$\mathbf{u}_h \in W_h(\Omega_h)$
$d\Gamma$
(c)
$K_j$
$K_e$
elements: $K_j, K_e \in \Omega_h$
(d)
boundary: $\partial K_j$
edge: $e_{i,K_j}$
element: $K_j$
local solution:
$\mathbf{u}_h|_{K_j}$
$\mathbf{n}_{e_i,K_j}$
boundary: $\partial K_e$
$d\Gamma$
$\mathbf{n}_{e_l,K_e}$
element: $K_e$
local solution:
$\mathbf{u}_h|_{K_e}$
edge: $e_{l,K_e}$
$\mathbf{u}_h|_{K_j} \in P^k(K_j)$
$\mathbf{u}_h|_{K_e} \in P^k(K_e)$
(e)
$\mathbf{u}_h(\boldsymbol{x}, t^n)$
$x$
$y$
$\mathbf{u}_h(\boldsymbol{x}, t^n)|_{K_j}$
$\mathbf{u}_h(\boldsymbol{x}, t^n)|_{K_e}$
$K_j$
$K_e$

Figure 2.4: Schematic of the sequence of the discontinuous Galerkin approximation to solve an original partial differential equation. The analytical solution $\mathbf{u}(\boldsymbol{x}, t)$ is projected to the finite dimensional subspace $W_h(\Omega_h)$, then further decomposed to local elements. Defining the space of polynomial functions $P^k(K)$, where $K \in \Omega_h$, independently in each element with possible discontinuity along edges, allows us to recast the global problem as the union of local problems.

Since the solution $\mathbf{u}(\boldsymbol{x}, t)$ is defined over the continuous space-time domain, it is impossible to represent the true solution $\mathbf{u}(\boldsymbol{x}, t)$ on a digital computer with a finite resource of memory or storage. Thus we make an assumption that the true solution can be accurately approximated by a finite number of functions such that

$$\mathbf{u}(\boldsymbol{x}, t) \approx \mathbf{u}_h(\boldsymbol{x}, t) = \sum_i \mathbf{c}_i(\boldsymbol{x}, t)\phi_i(\boldsymbol{x}, t), \tag{2.44}$$

where the coefficients $\mathbf{c}_i(\boldsymbol{x}, t)$ are called the *degrees of freedom*, and $\phi_i(\boldsymbol{x}, t)$ the *trial (basis) functions*. Our goal is to construct the evolution equations of the degrees of freedom $\mathbf{c}_i(\boldsymbol{x}, t)$ to compute an approximate solution of the original differential equations (2.43). In general, the procedure of projecting a solution defined in an infinite-dimensional space $W(\Omega)$ to an approximating solution like (2.44) in a finite dimensional space $W_h(\Omega_h)$ is called the Ritz–Galerkin method. More specifically, the Ritz–Galerkin method can be subdivided to the Ritz method, which employs a variational formulation, and the weighted-residual method, where a weak formulation is applied. See Figure 2.5 for their hierarchical relations [KS05, FFS03, DH03] A brief summary of these methods are as follows:

- Ritz method, or also called Rayleigh–Ritz method (variational formulation): An variational principle equivalent to the original differential equations is formulated, and the original problem is recast as the minimization problem for the variational formulation. For instance, the solution of physical problem expressed in differential form is restated as the extremum of a function. A drawback of this approach is that constructing a variational formulation might not be possible for complex physical systems such as the Navier–Stokes equations.

- Weighted residual method (weak formulation):

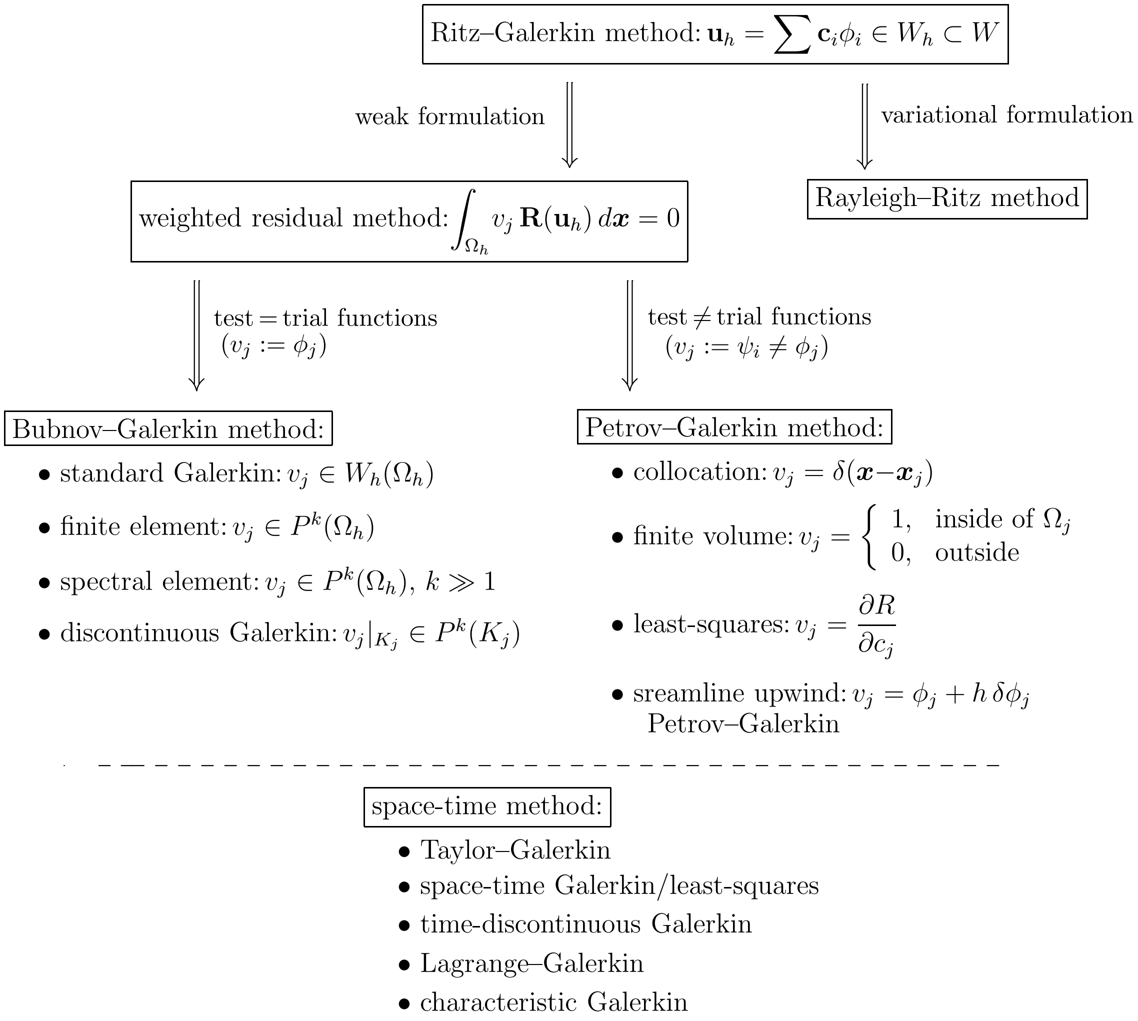

Ritz–Galerkin method: $\mathbf{u}_h = \sum \mathbf{c}_i \phi_i \in W_h \subset W$
weak formulation
variational formulation
weighted residual method: $\int_{\Omega_h} v_j \, \mathbf{R}(\mathbf{u}_h) \, d\boldsymbol{x} = 0$
Rayleigh–Ritz method
test = trial functions
$(v_j := \phi_j)$
test ≠ trial functions
$(v_j := \psi_i \neq \phi_j)$
Bubnov–Galerkin method:
• standard Galerkin: $v_j \in W_h(\Omega_h)$
• finite element: $v_j \in P^k(\Omega_h)$
• spectral element: $v_j \in P^k(\Omega_h),\ k \gg 1$
• discontinuous Galerkin: $v_j|_{K_j} \in P^k(K_j)$
Petrov–Galerkin method:
• collocation: $v_j = \delta(\boldsymbol{x} - \boldsymbol{x}_j)$
• finite volume: $v_j = \begin{cases} 1, & \text{inside of } \Omega_j \\ 0, & \text{outside} \end{cases}$
• least-squares: $v_j = \dfrac{\partial R}{\partial c_j}$
• sreamline upwind: $v_j = \phi_j + h\,\delta\phi_j$
Petrov–Galerkin
space-time method:
• Taylor–Galerkin
• space-time Galerkin/least-squares
• time-discontinuous Galerkin
• Lagrange–Galerkin
• characteristic Galerkin

Figure 2.5: The hierarchy of discretization methodologies is shown. The Ritz–Galerkin methods can be subdivided into two methodologies: weighted residual and Rayleigh–Ritz. Depending on the definition of test functions, various schemes can be distinguished. Also, various time-integration methods besides a typical ODE solver are listed.

> The original differential equation is rewritten in an integral form through a weak formulation. This approach is less restrictive and can be applied to a physical problem that does not have a variational formulation. Nevertheless, if a variational formulation can be obtained for a certain problem, both variational and weighted-residual methods lead to an identical discretization. The Galerkin weighted-residual method can be further subdivided into Bubnov–Galerkin, and Petrov–Galerkin methods. Bubnov–Galerkin methods, often simply called Galerkin methods, set test functions identical to trial functions, whereas Petrov–Galerkin methods employ different functions. In the case when an approximation of the solution in (2.44) is piecewise *polynomial*, a Bubnov/Petrov–Galerkin method is particularly called a finite-element method. Hence, it is more appropriate to call a DG method, for instance, a discontinuous Bubnov–Galerkin finite-element method.

Here, we employ the method of weighted residuals, more precisely a discontinuous Galerkin method, which is one of the Bubnov–Galerkin methods, to obtain the discretization method. The procedure to obtain the weak formulation is explained in the next section.

### 2.3.2 Weak Formulation

At first, we define a scalar function in the same space as the solution $\mathbf{u}(\boldsymbol{x}, t)$, and denote it as $v(\boldsymbol{x}, t) \in \Omega \times T$. This is called the *test function*. Multiplying the original differential equations (2.43) by a test function, $v(\boldsymbol{x}, t)$, and integrating over the space-time domain $\Omega(t) \times T$ leads to the weak formulation of the original differential equations:

$$\iint\limits_{\Omega(t)\times T} \partial_t \mathbf{u}(\boldsymbol{x},t)\, v(\boldsymbol{x},t)\, d\boldsymbol{x}dt = -\iint\limits_{\Omega(t)\times T} \nabla\cdot\mathbf{f}(\mathbf{u}(\boldsymbol{x},t))\, v(\boldsymbol{x},t)\, d\boldsymbol{x}dt + \iint\limits_{\Omega(t)\times T} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}(\boldsymbol{x},t))\, v(\boldsymbol{x},t)\, d\boldsymbol{x}dt. \tag{2.45}$$

The weak formulation of the problem is:

> Find $\mathbf{u}(\boldsymbol{x},t)$ such that the solution $\mathbf{u}(\boldsymbol{x},t)$ satisfies (2.45) for any test function $v(\boldsymbol{x},t)\in\Omega\times T$.

Note that the problem is still defined in an infinite-dimensional space. In order to remove derivatives of the solution and its flux from the space-time integrals, we integrate by parts, yielding the equalities

$$\int\limits_T \partial_t\mathbf{u}(\boldsymbol{x},t)v(\boldsymbol{x},t)\,dt \equiv [\mathbf{u}(\boldsymbol{x},t)\,v(\boldsymbol{x},t)]_{t_1}^{t_2} - \int\limits_T \mathbf{u}(\boldsymbol{x},t)\,\partial_t v(\boldsymbol{x},t)dt, \tag{2.46a}$$

$$\int\limits_{\Omega(t)} \nabla\cdot\mathbf{f}(\mathbf{u})\,v(\boldsymbol{x})\,d\boldsymbol{x} \equiv \int\limits_{\partial\Omega(t)} v(\boldsymbol{x})\,\mathbf{f}\cdot\mathbf{n}\,d\sigma - \int\limits_{\Omega(t)} \nabla v(\boldsymbol{x})\cdot\mathbf{f}(\mathbf{u})\,d\boldsymbol{x}; \tag{2.46b}$$

these are inserted into the weak formulation (2.45). Here, $\mathbf{n}$ is the outward unit vector normal to the boundary $\partial\Omega(t)$, and $T := [t_1, t_2]$. The integration variable $\sigma$ in the right hand side of (2.46b) is defined along the boundary $\partial\Omega(t)$. See also Figure 2.4(a) illustrating their definitions.

In the course of its derivation, one might wonder why a DG method always relies on integration by parts. The reason is that the integration by parts transfers a differential operator acting on nonlinear functions to test functions. For instance, the divergence operator, $\nabla\cdot$, applied to the nonlinear flux $\mathbf{f}(\mathbf{u})$ on the left hand side of (2.46b) is now acting on a gradient operator, $\nabla$, on a test function $v(\boldsymbol{x})$ on the right hand side. This operation is not unique to DG methods: weighted-residual methods in general take advantage of it. For instance, a finite volume method uses the same technique in its derivation. This can be seen by letting the test function be $v(\boldsymbol{x}) = 1$, then (2.46b) becomes the divergence theorem:

$$\int\limits_{\Omega} \nabla \cdot \mathbf{f}(\mathbf{u})\, d\boldsymbol{x} \equiv \int\limits_{\partial\Omega} \mathbf{f} \cdot \mathbf{n}\, d\sigma, \tag{2.47}$$

which is the core of the derivation of a finite volume method. By seeing the similarity between a finite-volume and Galerkin (finite-element) method, it would be more appropriate and intuitive to state that the multivariable divergence theorem is applied to the flux-divergence term instead of using the term "integration by parts."

Inserting the identities (2.46) into the weak formulation (2.45), and applying Fubini's theorem, which allows to alternate the order of integration in space and time, the weak formulation becomes

$$\begin{aligned} \int\limits_{\Omega(t_2)} \mathbf{u}(\boldsymbol{x}, t_2)\, v(\boldsymbol{x}, t_2)\, d\boldsymbol{x} - \int\limits_{\Omega(t_1)} \mathbf{u}(\boldsymbol{x}, t_1)\, v(\boldsymbol{x}, t_1)\, d\boldsymbol{x} - \iint\limits_{\Omega(t)\times T} \mathbf{u}(\boldsymbol{x}, t)\, \partial_t v(\boldsymbol{x}, t)\, d\boldsymbol{x} dt \\ = - \iint\limits_{\partial\Omega(t)\times T} v(\boldsymbol{x}, t)\, \mathbf{f}(\mathbf{u}(\boldsymbol{x}, t)) \cdot \mathbf{n}\, d\sigma dt + \iint\limits_{\Omega(t)\times T} \nabla v(\boldsymbol{x}, t) \cdot \mathbf{f}(\mathbf{u}(\boldsymbol{x}, t))\, d\boldsymbol{x} dt \\ + \iint\limits_{\Omega(t)\times T} \frac{1}{\epsilon} \mathbf{s}(\mathbf{u}(\boldsymbol{x}, t))\, v(\boldsymbol{x}, t)\, d\boldsymbol{x} dt. \end{aligned} \tag{2.48}$$

When we assume that the spatial domain over the interval of time $T$ is fixed, thus $\Omega(t) = \Omega$, and the test function only varies in space, $v(\boldsymbol{x}, t) = v(\boldsymbol{x})$ thus $\partial_t v(\boldsymbol{x}, t) \equiv 0$, then the above equation further simplifies as follows:

$$\begin{aligned} \int\limits_{\Omega} \big(\mathbf{u}(\boldsymbol{x}, t_2) - \mathbf{u}(\boldsymbol{x}, t_1)\big)\, v(\boldsymbol{x})\, d\boldsymbol{x} = - \iint\limits_{\partial\Omega\times T} v(\boldsymbol{x})\, \mathbf{f}(\mathbf{u}(\boldsymbol{x}, t)) \cdot \mathbf{n}\, d\sigma dt \\ + \iint\limits_{\Omega\times T} \nabla v(\boldsymbol{x}) \cdot \mathbf{f}(\mathbf{u}(\boldsymbol{x}, t))\, d\boldsymbol{x} dt + \iint\limits_{\Omega\times T} \frac{1}{\epsilon} \mathbf{s}(\mathbf{u}(\boldsymbol{x}, t))\, v(\boldsymbol{x})\, d\boldsymbol{x} dt. \end{aligned} \tag{2.49}$$

Note that the above equation is still exact in the weak sense, and the solution is defined in an arbitrary domain $\Omega \times T$ with the boundary $\partial\Omega \times T$.

### 2.3.3 Finite-Dimensional Approximation

In order to derive a numerical method on finite, discrete meshes (elements), at first, we decrease the continuous domain $\Omega$, which requires infinite information to be represented, to a domain that only requires finite information to be defined. A straightforward approach is to assume $\Omega$ to be a convex polygonal domain. Here, we denote this domain as $\Omega_h \subset \Omega$. The subscript $(\cdot)_h$ denotes an approximation of the continuous domain $\Omega$ on a discretized computational grid, where $h$ symbolizes mesh size. See Figure 2.4(b) on page 51 for a schematic of this approximation. Note that the temporal domain is defined as the finite interval $T = [t_1, t_2]$, yet the solution $\mathbf{u}(\boldsymbol{x}, t)$ is still continuous in the interval $T$. Once the entire computational domain $\Omega_h$ is defined, we define non-overlapping elements that compose the entire domain $\Omega_h$. Let $K_j$ be a typical element in the domain; thus $K_j \in \Omega_h$, then its boundary is defined by $\partial K_j$. Each face (3-D) or edge (2-D) of the element $K_j$ is denoted as $e_i$, hence $e_i \in \partial K_j$. We also define the element $K_e \in \Omega_h$, which shares the edge $e_i$ of the element $K_j$. The corresponding schematics are shown in Figures 2.4(c) and (d).

After the continuous domain has been reduced to the finite dimensional domain, we are ready to reduce the continuous solution $\mathbf{u}(\boldsymbol{x}, t)$ to the approximated solution $\mathbf{u}_h(\boldsymbol{x}, t)$, which requires a finite number of data (degrees of freedom) to be represented. In other words, the solution $\mathbf{u}(\boldsymbol{x}, t)$ is projected to the finite-dimensional domain $\Omega_h$. To achieve this goal, we define the finite-dimensional space $W_h$, called a *broken space*, for an approximate solution:

$$W_h(\Omega_h) = \{w_h(\boldsymbol{x}) \in L^\infty(\Omega_h) \colon w_h(\boldsymbol{x})|_{K_j} \in P^k(K_j), \forall K_j \in \Omega_h\}. \tag{2.50}$$

This is a expression we commonly encounter in papers dealing with a discontinuous Galerkin method, but it might not be quite intuitive for those who are new to

Galerkin methods. We therefore go into the details one by one to explain what the above equation means. The first relation on the right hand side is that $w(\boldsymbol{x})$ is a general bounded function defined over the domain $\Omega_h$, since $L^{\infty}(\Omega_h)$ means the space of bounded functions. The second term after the colon is a condition on a function $w_h(\boldsymbol{x})$ at each element $K_j$. It says that $w_h(\boldsymbol{x})|_{K_j}$ is a function restricted to the domain defined by $K_j$, and $P^k(K_j)$ is the space of polynomial functions of at most degree $k$ defined on the element $K_j$. Thus, a function on the element $K_j$ can only be expressed by polynomial functions, and such function representation makes a DG method a finite-element method. Note that a functional space on the element $K_j$, which is a polynomial space $P^k(K_j)$, is defined independently from neighboring elements; continuity along edges is not enforced. Hence, the above functional space $W_h(\Omega_h)$ allows discontinuities along edges. This property subdivides the global problem of finding the approximation of $\mathbf{u}(\boldsymbol{x}, t)$ over the entire domain $\Omega_h$ to the collection of local problems of finding a local solution $\mathbf{u}(\boldsymbol{x}, t)|_{K_j}$ on each element $K_j \in \Omega_h$. Let $\mathbf{u}_h(\boldsymbol{x}, t)$ and $v_h(\boldsymbol{x})$ be an approximate solution and a test function over the domain $\Omega_h$; then the approximate solution and test function at the element $K_j$ are related to these as follows:

solution:

$$\mathbf{u}(\boldsymbol{x}, t) \approx \mathbf{u}_h(\boldsymbol{x}, t) \in W_h(\Omega_h) \quad \longrightarrow \quad \mathbf{u}(\boldsymbol{x}, t)|_{K_j} \approx \mathbf{u}_h(\boldsymbol{x}, t)|_{K_j} \in P^k(K_j), \tag{2.51a}$$

test function:

$$v(\boldsymbol{x}) \approx v_h(\boldsymbol{x}) \in W_h(\Omega_h) \quad \longrightarrow \quad v(\boldsymbol{x})|_{K_j} \approx v_h(\boldsymbol{x})|_{K_j} \in P^k(K_j). \tag{2.51b}$$

When replacing a globally defined continuous solution and test function by a locally defined approximated solution and test function, the weak formulation (2.49) previously defined over the entire computational domain $\Omega_h \times T$ can be recast to the individual element $K_j$ as follows:

$$\int\limits_{K_j} \big(\mathbf{u}_h(\boldsymbol{x},t_2)|_{K_j} - \mathbf{u}_h(\boldsymbol{x},t_1)|_{K_j}\big)\, v_h(\boldsymbol{x})|_{K_j}\, d\boldsymbol{x}$$

$$= - \iint\limits_{\partial K_j \times T} v_h(\boldsymbol{x})|_{K_j}\, \mathbf{f}\big(\mathbf{u}_h(\boldsymbol{x},t)|_{K_j}, \mathbf{u}_h(\boldsymbol{x},t)|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j}\, d\Gamma dt$$

$$+ \iint\limits_{K_j \times T} \nabla v_h(\boldsymbol{x})|_{K_j} \cdot \mathbf{f}(\mathbf{u}_h(\boldsymbol{x},t)|_{K_j})\, d\boldsymbol{x}dt + \iint\limits_{K_j \times T} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(\boldsymbol{x},t)|_{K_j})\, v_h(\boldsymbol{x})|_{K_j}\, d\boldsymbol{x}dt,$$

$$\text{for any } K_j \in \Omega_h, \quad (2.52)$$

where $\mathbf{n}_{e_i,K_j}$ is the outward unit vector normal to an edge $e_i$ of the element $K_j$. The integration variable $\Gamma$ in the surface integral is defined along edges of the element $K_j$. Figure 2.4(d) explains notations schematically. Note that, as mentioned previously, the whole process of approximating the continuous problem (2.49) by the discrete problem (2.52) on a finite dimensional broken subspace $W_h(\Omega_h) \subset W(\Omega)$ is called the discontinuous Galerkin method.

The first term on the right hand side of (2.52) is the surface integral of the flux, $\iint (\cdot)\, d\Gamma dt$, along boundaries of the element $K_j$. As a consequence of the independence of local spaces, the solution along edges can not be uniquely determined because it may be discontinuous. See Figure 2.4(e) on page 51. Thus, the flux at the edge $e_i$ depends on both $\mathbf{u}|_{K_j}$ and $\mathbf{u}|_{K_e}$. Since the boundary $\partial K_j$ is composed of all edges $e_i$, the surface integral can be reformulated as the summation of flux integrations along each edge, thus

$$\iint\limits_{\partial K_j \times T} v_h(\boldsymbol{x})|_{K_j}\, \mathbf{f}(\mathbf{u}_h(\boldsymbol{x},t)|_{K_j}, \mathbf{u}_h(\boldsymbol{x},t)|_{K_e}) \cdot \mathbf{n}_{e_i,K_j}\, d\Gamma dt$$

$$\equiv \sum_{e_i \in \partial K_j} \iint\limits_{e_i \times T} v_h(\boldsymbol{x})|_{K_j}\, \mathbf{f}\big(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j}\, d\Gamma dt. \quad (2.53)$$

This edge-based reformulation is particularly useful for coding, where a numerical flux function can sweep through edges without considering connectivity with other edges. After inserting the above equation into (2.52), the weak formulation with respect to an element $K_j$ is

$$\begin{aligned}
&\int_{K_j} \left(\mathbf{u}_h(\boldsymbol{x},t_2)|_{K_j} - \mathbf{u}_h(\boldsymbol{x},t_1)|_{K_j}\right) v_h(\boldsymbol{x})|_{K_j}\, d\boldsymbol{x} \\
&\qquad\qquad = \sum_{e_i\in\partial K_j} \iint_{e_i\times T} v_h(\boldsymbol{x})|_{K_j}\, \mathbf{f}\left(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\right)\cdot \mathbf{n}_{e_i,K_j}\, d\Gamma dt \\
&+ \iint_{K_j\times T} \nabla v_h(\boldsymbol{x})|_{K_j}\cdot \mathbf{f}(\mathbf{u}_h(\boldsymbol{x},t)|_{K_j})\, d\boldsymbol{x}dt + \iint_{K_j\times T} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(\boldsymbol{x},t)|_{K_j})\, v_h(\boldsymbol{x})|_{K_j}\, d\boldsymbol{x}dt. \qquad (2.54)
\end{aligned}$$

It is important to note that the above equation is still exact for a solution contained in the finite-dimensional subspace $W_h(\Omega_h)$ defined by (2.50). Furthermore, the shape of element $K_j$ is not specified yet; any polygonal element still satisfies the above formula.

### 2.3.4 Polynomial Representation of the Solution

Recall that the individual solution $\mathbf{u}_h(\boldsymbol{x},t)|_{K_j}$ in the element $K_j$ is a polynomial function. Here, we take the Legendre polynomials up to degree $k = 1$ (piecewise linear functions) as the space of polynomial functions, and consider a two-dimensional problem, thus $\boldsymbol{x} = (x, y)$,

$$\mathbf{u}(\boldsymbol{x},t)|_{K_j}, v(\boldsymbol{x})|_{K_j} \in P^1(K_j), \qquad (2.55)$$

where

$$\begin{aligned}
P^1(K_j) &= \text{span}\{\phi_0(\boldsymbol{x}), \phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x})\} \\
&= \text{span}\{1,\, x - x_{c,K_j},\, y - y_{c,K_j}\},
\end{aligned} \qquad (2.56)$$

and $\phi_m(\boldsymbol{x})$, $m = 0, 1, 2$ form the Legendre polynomial basis. The constants $(x_{c,K_j}, y_{c,K_j})$ in the basis function define the centroid of the element $K_j$ defined by

$$x_{c,K_j} := \frac{\displaystyle\iint_{K_j} x\,dxdy}{\displaystyle\iint_{K_j} dxdy}, \qquad y_{c,K_j} := \frac{\displaystyle\iint_{K_j} y\,dxdy}{\displaystyle\iint_{K_j} dxdy}. \tag{2.57}$$

Let $\{\bar{\mathbf{u}}_j(t), \overline{\Delta_x \mathbf{u}}_j(t), \overline{\Delta_y \mathbf{u}}_j(t)\}$ be the degrees of freedom (state variables), which are continuous over the time interval $t \in [t^n, t^{n+1}]$, then the solution and test functions in the element $K_j$ can be expressed by (2.44) as follows:

$$\begin{aligned} \mathbf{u}_h(x, y, t)|_{K_j} &= \bar{\mathbf{u}}_j(t)\phi_0(\boldsymbol{x}) + \overline{\Delta_x \mathbf{u}}_j(t)\phi_1(\boldsymbol{x}) + \overline{\Delta_y \mathbf{u}}_j(t)\phi_2(\boldsymbol{x}) \\ &= \bar{\mathbf{u}}_j(t) + \left(\overline{\Delta_x \mathbf{u}}_j(t), \overline{\Delta_y \mathbf{u}}_j(t)\right) \cdot (x - x_{c,K_j}, y - y_{c,K_j}) \qquad (2.58a) \\ &= \bar{\mathbf{u}}_j(t) + \left(\overline{\Delta_x \mathbf{u}}_j(t), \overline{\Delta_y \mathbf{u}}_j(t)\right) \cdot (\boldsymbol{x} - \boldsymbol{x}_{c,K_j}), \end{aligned}$$

$$v_h(\boldsymbol{x})|_{K_j} \in \operatorname{span}\{1,\ x - x_{c,K_j},\ y - y_{c,K_j}\}. \tag{2.58b}$$

Similar to the 1-D case, the linear distribution of the solution $\mathbf{u}_h(\boldsymbol{x}, t)|_{K_j}$ satisfies the following properties:

$$\iint\limits_{K_j} \mathbf{u}_h(\boldsymbol{x}, t)\,d\boldsymbol{x}dt \equiv \bar{\mathbf{u}}_j(t), \tag{2.59a}$$

$$\mathbf{u}_h(x_{c,K_j}, y_{c,K_j}, t) \equiv \bar{\mathbf{u}}_j(t). \tag{2.59b}$$

The definition of the approximate solution (2.58a) is quite natural in the context of a finite-volume method, i.e., the degrees of freedom possess physical meaning such as cell-average and average slope. However, it is rather uncommon in a Galerkin formulation, typically for a triangular element, to represent a solution based on averaged quantities. The standard practice is that the degrees of freedom are defined at nodes or points along edges of an element. This solution definition makes the method's derivation simple. Conversely, the major drawback of the node-based solution representation, especially for the DG(1)–Hancock method for hyperbolic-relaxation equations, is that the updated cell average can not be obtained until all degrees of freedom have been updated. This means that even though the interface flux calculation is still explicit, the volume integral of the flux needs to be treated implicitly. The cell-average based method does not have this issue. This is the main reason why we prefer to adopt the finite-volume-like approximation. Nevertheless, at least when using linear polynomials, both average-based and nodal-based descriptions lead to identical algebraic equations modulo a similarity transformation.

Once the solution in the element $K_j$ is specified, we can again restate the problem in the weak formulation for a typical element $K_j \in \Omega_h$:

> Find $\mathbf{u}_h(\boldsymbol{x}, t)|_{K_j} \in P^1(K_j) \times T$ such that the solution $\mathbf{u}_h(\boldsymbol{x}, t)|_{K_j}$ satisfies (2.54) for any test function $v_h(\boldsymbol{x})|_{K_j} \in P^1(K_j)$ in the element $K_j$.

The same problem statement is applied to all elements in the discretized domain $\Omega_h$. The last thing needed for an actual discretization method from the weak formulation (2.54) is choosing appropriate test functions. The test function $v(\boldsymbol{x})|_{K_j}$ can be arbitrary as long as it stays in the space of the polynomial functions $P^1(K_j)$. The natural choice is the polynomial basis functions as the test functions, hence

$$v_h(\boldsymbol{x})|_{K_j} = \{\phi_0(\boldsymbol{x}), \phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x})\}. \tag{2.60}$$

With this particular choice, where the test and basis functions are identical, the method is called the Bubnov–Galerkin, or simply Galerkin method. If a test function is not the same as a solution-basis function, then a method is called Petrov–Galerkin method.

### 2.3.5 Evolution Equations of the Degrees of Freedom

Inserting each test function into (2.54) with the solution representation (2.58a) leads to the following update formulas for the degrees of freedom $\{\bar{\mathbf{u}}_j(t)$, $\overline{\Delta_x \mathbf{u}}_j(t)$, $\overline{\Delta_y \mathbf{u}}_j(t)\}$:

- $v_h(\boldsymbol{x})|_{K_j} = 1$:

$$|A_j| \left[\bar{\mathbf{u}}_j(t)\right]_{t^n}^{t^{n+1}} = - \sum_{e_i \in \partial K_j} \iint\limits_{e_i \times T} \mathbf{f}\big(\mathbf{u}_h|_{K_i}, \mathbf{u}_h|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j} \, d\Gamma dt + \iint\limits_{K_j \times T} \frac{1}{\epsilon} \mathbf{s}(\mathbf{u}_h(\boldsymbol{x},t)) \, d\boldsymbol{x} dt, \tag{2.61a}$$

- $v_h(\boldsymbol{x})|_{K_j} = x - x_{c,K_j}$:

$$\begin{aligned}\big[K_{j1}\overline{\Delta_x \mathbf{u}_j}(t) + K_{j2}\overline{\Delta_y \mathbf{u}_j}(t)\big]_{t^n}^{t^{n+1}} &= - \sum_{e_i \in \partial K_j} \iint\limits_{e_i \times T} (x - x_{c,K_j}) \, \mathbf{f}\big(\mathbf{u}_h|_{K_i}, \mathbf{u}_h|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j} \, d\Gamma dt \\ &+ \iint\limits_{K_j \times T} (1,0) \cdot \mathbf{f}(\mathbf{u}_h(\boldsymbol{x},t)|_{K_i}) \, d\boldsymbol{x} dt + \iint\limits_{K_j \times T} (x - x_{c,K_j}) \frac{1}{\epsilon} \mathbf{s}(\mathbf{u}_h(\boldsymbol{x},t)) \, d\boldsymbol{x} dt,\end{aligned} \tag{2.61b}$$

- $v_h(\boldsymbol{x})|_{K_j} = y - y_{c,K_j}$:

$$\begin{aligned}\big[K_{j2}\overline{\Delta_x \mathbf{u}_j}(t) + K_{j3}\overline{\Delta_y \mathbf{u}_j}(t)\big]_{t^n}^{t^{n+1}} &= - \sum_{e_i \in \partial K_j} \iint\limits_{e_i \times T} (y - y_{c,K_j}) \, \mathbf{f}\big(\mathbf{u}_h|_{K_i}, \mathbf{u}_h|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j} \, d\Gamma dt \\ &+ \iint\limits_{K_j \times T} (0,1) \cdot \mathbf{f}(\mathbf{u}_h(\boldsymbol{x},t)|_{K_i}) \, d\boldsymbol{x} dt + \iint\limits_{K_j \times T} (y - y_{c,K_j}) \frac{1}{\epsilon} \mathbf{s}(\mathbf{u}_h(\boldsymbol{x},t)) \, d\boldsymbol{x} dt,\end{aligned} \tag{2.61c}$$

where $\mathbf{f} \in \mathbb{R}^{m \times 2}$ is the 2-D flux tensor, $|A_j|$ is the area of the element $K_j$ defined by

$$|A_j| := \int\limits_{K_j} \phi_0(\boldsymbol{x}) \phi_0(\boldsymbol{x}) \, d\boldsymbol{x} = \iint\limits_{K_j} dx dy, \tag{2.62a}$$

and other tensor products of basis functions are defined by

$$K_{j1} := \int\limits_{K_j} \phi_1(\boldsymbol{x})\phi_1(\boldsymbol{x})\, d\boldsymbol{x} = \iint\limits_{K_j} (x - x_{c,K_j})^2\, dxdy, \tag{2.62b}$$

$$K_{j2} := \int\limits_{K_j} \phi_1(\boldsymbol{x})\phi_2(\boldsymbol{x})\, d\boldsymbol{x} = \iint\limits_{K_j} (x - x_{c,K_j})(y - y_{c,K_j})\, dxdy, \tag{2.62c}$$

$$K_{j3} := \int\limits_{K_j} \phi_2(\boldsymbol{x})\phi_2(\boldsymbol{x})\, d\boldsymbol{x} = \iint\limits_{K_j} (y - y_{c,K_j})^2\, dxdy. \tag{2.62d}$$

These quantities are computed once and stored at the beginning of calculation as long as fixed grids are considered. Let $\mathbf{K}_j$ be the inverse of the partial mass matrix corresponding to the degrees of freedom, $\overline{\Delta_x \mathbf{u}}_j, \overline{\Delta_y \mathbf{u}}_j$, such that

$$\mathbf{K}_j := \frac{1}{K_{j1}K_{j3} - K_{j2}^2} \begin{pmatrix} K_{j3}\,\mathbf{I} & -K_{j2}\,\mathbf{I} \\ -K_{j2}\,\mathbf{I} & K_{j1}\,\mathbf{I} \end{pmatrix}, \tag{2.63}$$

where $\mathbf{I} \in \mathbb{R}^{m\times m}$, then the update formulations in explicit form become

$$\begin{aligned} \bar{\mathbf{u}}_j^{n+1} = \bar{\mathbf{u}}_j^{n} &- \frac{1}{|A_j|} \sum_{e_i \in \partial K_j} \iint\limits_{e_i \times T} \mathbf{f}\big(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j}\, d\Gamma dt \\ &+ \frac{1}{|A_j|} \iint\limits_{K_j \times T} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(\boldsymbol{x},t))\, d\boldsymbol{x}dt, \end{aligned} \tag{2.64a}$$

$$\begin{aligned} \begin{pmatrix} \overline{\Delta_x \mathbf{u}}_j^{n+1} \\ \overline{\Delta_y \mathbf{u}}_j^{n+1} \end{pmatrix} = & \begin{pmatrix} \overline{\Delta_x \mathbf{u}}_j^{n} \\ \overline{\Delta_y \mathbf{u}}_j^{n} \end{pmatrix} \\ &+\mathbf{K}_j \begin{pmatrix} -\sum\limits_{e_i \in \partial K_j} \iint\limits_{e_i \times T} (x - x_{c,K_j})\, \mathbf{f}\big(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j}\, d\Gamma dt \\ -\sum\limits_{e_i \in \partial K_j} \iint\limits_{e_i \times T} (y - y_{c,K_j})\, \mathbf{f}\big(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j}\, d\Gamma dt \end{pmatrix} \\ &+\mathbf{K}_j \begin{pmatrix} \iint\limits_{K_j \times T} (1,0) \cdot \mathbf{f}(\mathbf{u}_h(\boldsymbol{x},t))\, d\boldsymbol{x}dt + \iint\limits_{K_j \times T} (x - x_{c,K_j}) \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(\boldsymbol{x},t))\, d\boldsymbol{x}dt \\ \iint\limits_{K_j \times T} (0,1) \cdot \mathbf{f}(\mathbf{u}_h(\boldsymbol{x},t))\, d\boldsymbol{x}dt + \iint\limits_{K_j \times T} (y - y_{c,K_j}) \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(\boldsymbol{x},t))\, d\boldsymbol{x}dt \end{pmatrix}. \end{aligned} \tag{2.64b}$$

Now we can see the advantage of defining the approximate solution based on the cell-averaged quantities; the cell-average update equation can be solved first, then the update quantity $\bar{\mathbf{u}}_j^{n+1}$ is utilized for slope-update equations. Further numerical approximations are necessary for the interface flux, $\mathbf{f}\big(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\big)$, surface integral of the flux, $\displaystyle\iint_{e_j \times T} (\cdot)\, d\Gamma dt$, and volume integral of the flux and source term, $\displaystyle\iint_{K_j \times T} (\cdot)\, d\boldsymbol{x} dt$.

### 2.3.6 Interface-Flux Approximation and Surface Integral

As we can see in Figure 2.4(e), a solution along the edge $e_j$ can not be uniquely determined because continuity restriction along an edge is not enforced in a discontinuous Galerkin method. Since this discontinuity is the place where wave interactions occur, we borrow a precision tool developed in the finite-volume community, i.e. the interface flux along the edge $e_j$ is obtained numerically by an approximate Riemann solver:

$$\mathbf{f}\big(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j} \approx \hat{\mathbf{f}}_n\big(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\big), \tag{2.65}$$

where $\hat{\mathbf{f}}_n$ is the vector of fluxes normal to the edge $e_i$ projecting from the element $K_j$. Even though multidimensional problems are considered here, a Riemann solver is still based on 1-D physics, as it is in most finite-volume methods. In this respect, a discontinuous Galerkin method is not a truly multidimensional method. There is a whole class of genuinely multidimensional methods called fluctuation-splitting or residual-distribution methods originally proposed by Roe [Roe82, Roe86], yet this rich subject goes beyond the scope of the present thesis, and we restrict ourselves to providing some references: [Roe05a, DRS93, AM07]. The connection of this methodology to the Petrov–Galerkin method was made by Carette et al. [CDPR95].

In an approximate Riemann solver, the flux tensor is projected normal to an element edge $e_i$, then the eigenstructure of the Jacobian of the rotated flux is utilized. The common choices are Roe's [Roe81], HLL's [HLvL83], and Rusanov's [Rus62] approximate Riemann solvers. The Rusanov flux is sometimes called the local Lax–Friedrichs solver, but this only gives rise to a confusion. It is important to note that the original Lax–Friedrichs flux for a DG method is unconditionally unstable. This was first found by Rider and Lowrie [RL02], and supported by further analysis in [SvL07], but has not yet been recognized by the community.

The surface integral of a flux can be evaluated exactly if the flux is linear. When a flux is nonlinear, a standard approach is approximating the integral by a quadrature. An alternative, quadrature-free implementation is developed by Atkins and Shu [AS96, AS98], who approximate a nonlinear flux in terms of the basis functions:

$$\mathbf{f}(\mathbf{u}(\boldsymbol{x})) \approx \sum_i \mathbf{f}(\mathbf{u}(\boldsymbol{x}_i))\, \phi_i(\boldsymbol{x}). \tag{2.66}$$

In this approach, the number of basis functions used to expand a flux has to be at least one higher than the number of basis functions used for the solution approximation. In our formulation of the DG(1)–Hancock method, we adopt the traditional approach: Gaussian quadratures for both surface and volume integrals.

Cockburn and Shu prove that, for a semi-discrete method, if the solution is represented in a polynomial space $P^k$, then a quadrature for the spatial integration along edges must be exact for a polynomial of degree $2k+1$ [CHS90]. Thus, for DG(1)–Hancock, the spatial integration is replaced by a two-point Gaussian quadrature, which is exact for a polynomial of degree 3. The time integration is approximated by the midpoint rule, thus fluxes are evaluated at $t^{n+1/2}$. Figure 2.6 is a schematic of the surface integral over the domain $e_{i,K_j} \times [t^n, t^{n+1}]$. The symbols, ▲, ●, are the quadrature points associated with time $t^{n+1/6}, t^{n+1/2}$ respectively. The

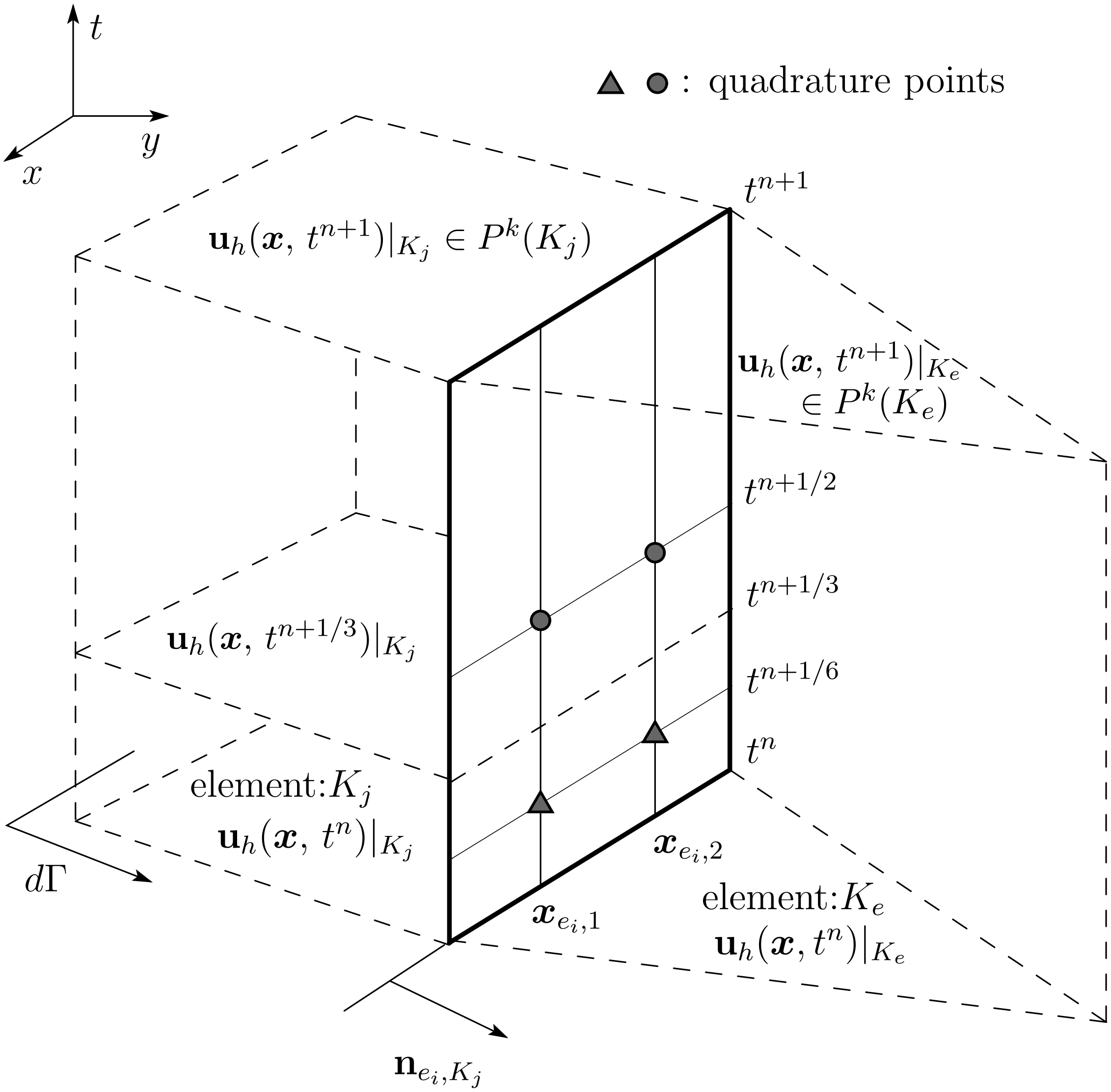

$t$
$x$
$y$
▲ ●: quadrature points
$t^{n+1}$
$\mathbf{u}_h(\boldsymbol{x},\, t^{n+1})|_{K_j} \in P^k(K_j)$
$\mathbf{u}_h(\boldsymbol{x},\, t^{n+1})|_{K_e}$
$\in P^k(K_e)$
$t^{n+1/2}$
$t^{n+1/3}$
$\mathbf{u}_h(\boldsymbol{x},\, t^{n+1/3})|_{K_j}$
$t^{n+1/6}$
$t^n$
element:$K_j$
$\mathbf{u}_h(\boldsymbol{x},\, t^n)|_{K_j}$
$\boldsymbol{x}_{e_i,2}$
$d\Gamma$
$\boldsymbol{x}_{e_i,1}$
element:$K_e$
$\mathbf{u}_h(\boldsymbol{x}, t^n)|_{K_e}$
$\mathbf{n}_{e_i,K_j}$

Figure 2.6: The surface integral of the element-interface flux over the domain $e_{i,K_j} \times [t^n, t^{n+1}]$ is replaced by the two-point Gauss quadrature in space. The quadrature points are denoted by bullets (●, integration over $[t^n, t^{n+1}]$), and triangles (▲, integration over $[t^n, t^{n+1/3}]$). At each point, a Riemann solver is applied to compute a unique interface flux.

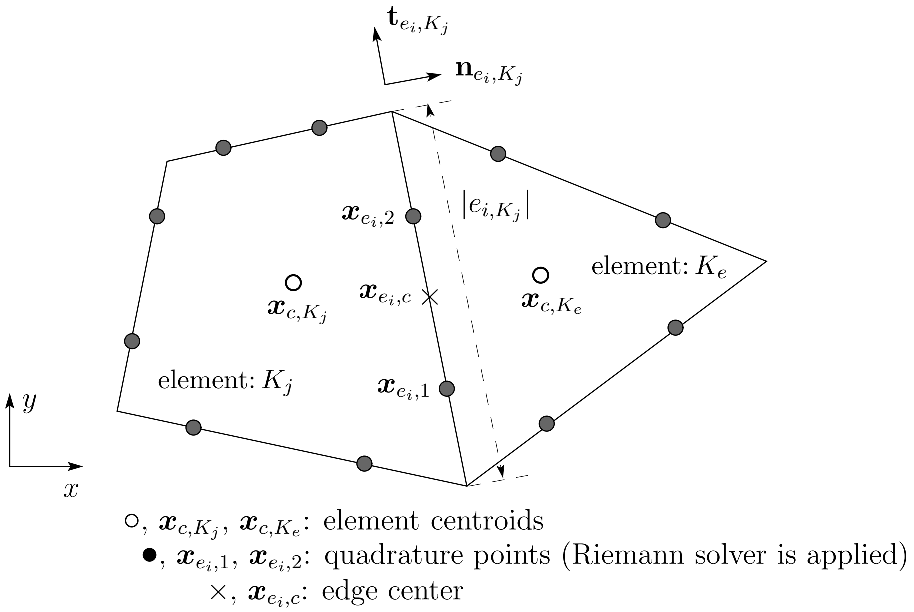

$\mathbf{t}_{e_i,K_j}$
$\mathbf{n}_{e_i,K_j}$
$|e_{i,K_j}|$
$\boldsymbol{x}_{e_i,2}$
element: $K_e$
$\boldsymbol{x}_{c,K_j}$
$\boldsymbol{x}_{e_i,c}$
$\boldsymbol{x}_{c,K_e}$
element: $K_j$
$\boldsymbol{x}_{e_i,1}$
$y$
$x$
○, $\boldsymbol{x}_{c,K_j}$, $\boldsymbol{x}_{c,K_e}$: element centroids
●, $\boldsymbol{x}_{e_i,1}$, $\boldsymbol{x}_{e_i,2}$: quadrature points (Riemann solver is applied)
×, $\boldsymbol{x}_{e_i,c}$: edge center

Figure 2.7: The bullet symbol (●) denotes quadrature points where a Riemann flux is computed. For a $P^1$ method, two Riemann fluxes in the Gauss points along the edge $e_j$ are required to approximate an actual distribution of the interface flux with a polynomial of degree three (cubic function).

time integration of the flux over the interval $[t^n, t^{n+1}]$ is approximated by two fluxes located at the circular symbols (●), while triangular symbols (▲) are used within $[t^n, t^{n+1/3}]$. A similar schematic, but with quadrature points drawn in the $x$,$y$-plane, is seen in Figure 2.7. It shows that a quadrilateral element requires 8 points to evaluate a Riemann flux, and a triangular element 6 points. The resulting quadrature rule is as follows:

$$\begin{aligned}
&\iint\limits_{e_i \times [t^n, t^{n+1}]} \mathbf{f}\big(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\big) \cdot \mathbf{n}_{e_i,K_j}\, d\Gamma dt \\
&\qquad \approx \iint\limits_{e_i \times [t^n, t^{n+1}]} \hat{\mathbf{f}}_n\big(\mathbf{u}_j(\boldsymbol{x}, t), \mathbf{u}_e(\boldsymbol{x}, t)\big)\, d\Gamma dt \\
&\qquad \approx |e_{i,K_j}| \int\limits_{t^n}^{t^{n+1}} \sum_k w_k\, \hat{\mathbf{f}}_n\big(\mathbf{u}_j(\boldsymbol{x}_k, t), \mathbf{u}_e(\boldsymbol{x}_k, t)\big)\, dt \\
&\qquad \approx (t^{n+1} - t^n)\, |e_{i,K_j}| \left[ w_1\, \hat{\mathbf{f}}_n\big(\mathbf{u}_{L_1}^{n+1/2}, \mathbf{u}_{R_1}^{n+1/2}\big) + w_2\, \hat{\mathbf{f}}_n\big(\mathbf{u}_{L_2}^{n+1/2}, \mathbf{u}_{R_2}^{n+1/2}\big) \right],
\end{aligned} \tag{2.67}$$

where the weights at Gauss points are

$$w_1 = w_2 = \frac{1}{2}, \tag{2.68}$$

$|e_{i,K_j}|$ is the length of the edge $e_i$ belonging to the element $K_j$, and the input quantities of the Riemann solver are

$$\mathbf{u}_{L_1}^{n+1/2} = \mathbf{u}_h\big(\boldsymbol{x}_{e_i,1},\, t^{n+1/2}\big)|_{K_j}, \quad \mathbf{u}_{L_2}^{n+1/2} = \mathbf{u}_h\big(\boldsymbol{x}_{e_i,2},\, t^{n+1/2}\big)|_{K_j}, \tag{2.69a}$$

$$\mathbf{u}_{R_1}^{n+1/2} = \mathbf{u}_h\big(\boldsymbol{x}_{e_i,1},\, t^{n+1/2}\big)|_{K_e}, \quad \mathbf{u}_{R_2}^{n+1/2} = \mathbf{u}_h\big(\boldsymbol{x}_{e_i,2},\, t^{n+1/2}\big)|_{K_e}. \tag{2.69b}$$

The $x$-,$y$-coordinates of Gauss points $\boldsymbol{x}_{e_i,1}, \boldsymbol{x}_{e_i,2}$ are computed by

$$\boldsymbol{x}_{e_i,1} = \boldsymbol{x}_{e_i,c} - \frac{|e_{i,K_j}|}{2\sqrt{3}}\,\mathbf{t}_{e_i,K_j}, \tag{2.70a}$$

$$\boldsymbol{x}_{e_i,2} = \boldsymbol{x}_{e_i,c} + \frac{|e_{i,K_j}|}{2\sqrt{3}}\,\mathbf{t}_{e_i,K_j}, \tag{2.70b}$$

where $\mathbf{t}_{e_i,K_j}$ is the unit vector tangential to the edge $e_j$ of the element $K_j$. Other quadrature points denoted by triangular symbols correspond to a time integral over $[t^n, t^{n+1/3}]$. This integral is necessary to obtain the intermediate solution $\mathbf{u}_h(\boldsymbol{x}, t^{n+1/3})$, used for the volume integral of the source term.

### Hancock's Predictor Step

So far, the interface flux has been replaced by a Riemann solver, and the surface integral has been approximated by a Gaussian quadrature. To complete the approximation of the surface integral, the last thing we need to specify is how to compute the input quantities $\mathbf{u}(\boldsymbol{x}_{\text{edge}}, t^{n+1/2})$ for a Riemann solver, associated to the half-time level $t^{n+1/2}$. Based on Hancock's observation, flow quantities evolve over the half time step without any interactions with neighbors. Hence, from the update formula for conserved variables (2.64a) on page 64, element-face interactions can be removed:

$$\bar{\mathbf{u}}_j^{n+1/2} = \bar{\mathbf{u}}_j^n - \frac{1}{2|A_j|} \sum_{e_i \in \partial K_j} \iint_{e_i \times T'} \mathbf{f}\big(\mathbf{u}_h(\boldsymbol{x}_{e_i}, t)|_{K_j}\big) \cdot \mathbf{n}_{e_i,K_j} \, d\Gamma dt \\ + \frac{1}{2|A_j|} \iint_{K_j \times T'} \frac{1}{\epsilon} \mathbf{s}\big(\mathbf{u}_h(\boldsymbol{x}, t)\big) \, d\boldsymbol{x} dt, \quad (2.71)$$

where $T' := [t^n, t^{n+1/2}]$, and the flux tensor $\mathbf{f}$ is simply evaluated from the solution at each quadrature point $\boldsymbol{x}_{e_i}$ of the element $K_j$, obtained by (2.58a) on page 61. The predictor step (2.71) to compute the cell-average quantities at time $t^{n+1/2}$ can be further simplified with the approximations in evaluating the flux integral at time $t^n$. As for the source term, since we are considering a stiff case, that is $\epsilon \ll O(1)$, an $L$-stable method for the temporal integration is preferable even in the predictor step. The simplest one is the backward Euler method, hence the source integral is evaluated at $t^{n+1/2}$; then (2.71) becomes

$$\bar{\mathbf{u}}_j^{n+1/2} = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{2|A_j|} \sum_{e_i \in \partial K_j} \int_{e_i} \mathbf{f}\big(\mathbf{u}_h(\boldsymbol{x}_{e_i}, t^n)|_{K_j}\big) \cdot \mathbf{n}_{e_i,K_j} \, d\Gamma \\ + \frac{\Delta t}{2|A_j|} \int_{K_j} \frac{1}{\epsilon} \mathbf{s}\big(\mathbf{u}_h(\boldsymbol{x}, t^{n+1/2})\big) \, d\boldsymbol{x}, \quad (2.72)$$

where

$$\mathbf{u}_h(\boldsymbol{x}_{e_i}, t^n) = \bar{\mathbf{u}}_j^n + \phi_j^n \left(\overline{\Delta_x \mathbf{u}}_j^n, \overline{\Delta_y \mathbf{u}}_j^n\right) \cdot (\boldsymbol{x}_{e_i} - \boldsymbol{x}_{c,K_j}). \quad (2.73)$$

Here $\boldsymbol{x}_{c,K_j}$ is the centroid of the element $K_j$, and $\phi_j^n$ is a gradient limiter such as the TVB corrected minmod function by Cockburn and Shu [CS98]. The mimmod treatment of computing the slope from a few candidates can be replaced by the Barth–Jespersen limiter [BJ89, Bar90], which is a multidimensional extension of the one-dimensional double-minmod limiter [vL77], or the Venkatakrishnan limiter [Ven95], which is the extension of the Van Alvada limiter [vAvLR82].

The latest development in limiters for DG methods is the hierarchical reconstruction due to Liu [LSTZ07], based on the moment limiter originally developed by Biswas et al. [BDF94]. The approach reduces the limiting process a high-degree polynomial to multiple limiting of piecewise linear functions. Thus, once a 'good' limiter for solution in $P^1$ is developed, the limiter can be applied to any higher-degree polynomial hierarchically. Huynh's new $P^1$ limiter seems to be a good candidate [Huy06b], but it still needs to be extended to an unstructured grid.

Once the predicted cell-average quantity $\bar{\mathbf{u}}_j^{k+1/2}$ at the element $K_j$ is obtained, the distribution of the solution along the edge $e_i$ can be computed by

$$\mathbf{u}_h(\boldsymbol{x}_{e_i}, t^{n+1/2}) = \bar{\mathbf{u}}_j^{n+1/2} + \phi_j^n \left(\overline{\Delta_x \mathbf{u}}_j^n, \overline{\Delta_y \mathbf{u}}_j^n\right) \cdot (\boldsymbol{x}_{e_i} - \boldsymbol{x}_{c,K_j}). \tag{2.74}$$

Note that the slope variables $\overline{\Delta_x \mathbf{u}}_j^n, \overline{\Delta_y \mathbf{u}}_j^n$ are still associated with the time $t^n$. In order to make the time integration of the source term point-implicit, the spatial integral of the source term is evaluated with the solution at just one Gauss point, $(\boldsymbol{x}_{e_i}, t^{n+1/2})$, thus

$$\int\limits_{K_j} \frac{1}{\epsilon}\mathbf{s}\big(\mathbf{u}_h(\boldsymbol{x}, t^{n+1/2})\big)\, d\boldsymbol{x} \approx \frac{|A_j|}{\epsilon}\,\mathbf{s}\big(\mathbf{u}_h(\boldsymbol{x}_{e_i}, t^{n+1/2})\big). \tag{2.75}$$

Inserting the above equation into (2.72) leads to the following implicit predictor step for the input value of a Riemann solver:

$$\begin{aligned}\mathbf{u}_h(\boldsymbol{x}_{e_i}, t^{n+1/2}) = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{2|A_j|} \sum_{e_i \in \partial K_j} \int\limits_{e_i} \mathbf{f}\big(\mathbf{u}_h(\boldsymbol{x}_{e_i}, t^n)|_{K_j}\big) \cdot \mathbf{n}_{e_i,K_j}\, d\Gamma \\ + \phi_j^n \left(\overline{\Delta_x \mathbf{u}}_j^n, \overline{\Delta_y \mathbf{u}}_j^n\right) \cdot (\boldsymbol{x}_{e_i} - \boldsymbol{x}_{c,K_j}) + \frac{\Delta t}{2\epsilon}\mathbf{s}\big(\mathbf{u}_h(\boldsymbol{x}_{e_i}, t^{n+1/2})\big).\end{aligned} \tag{2.76}$$

Similar to the 1-D case, this predictor step is often based on the primitive variables $\mathbf{w}$ instead of the conserved ones. Typically, the primitive form provides a linear source term $\dfrac{1}{\epsilon(\mathbf{w})}\mathbf{s}(\mathbf{w}) = \mathbf{Q}_{\mathrm{w}}\mathbf{w}$, and the above implicit formula can be rewritten as an explicit predictor. This can be achieved by the following two steps. At first, the conserved quantity at a particular Gauss point $\boldsymbol{x}_{e_i}$ is computed without the influence of the source term,

$$\widetilde{\mathbf{u}}_h(\boldsymbol{x}_{e_i}, t^{n+1/2}) = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{2|A_j|} \sum_{e_i \in \partial K_j} \int_{e_i} \mathbf{f}\big(\mathbf{u}_h(\boldsymbol{x}_{e_i}, t^n)|_{K_j}\big) \cdot \mathbf{n}_{e_i,K_j}\, d\Gamma$$

$$+ \phi_j^n \left(\overline{\Delta_x \mathbf{u}}_j^n, \overline{\Delta_y \mathbf{u}}_j^n\right) \cdot (\boldsymbol{x}_{e_i} - \boldsymbol{x}_{c,K_j}). \quad (2.77)$$

Once the predicted conserved quantity $\widetilde{\mathbf{u}}_h(\boldsymbol{x}_{e_i}, t^{n+1/2})$ at the Gauss point is obtained, it is converted to the primitive variable $\widetilde{\mathbf{w}}_h(\boldsymbol{x}_{e_i}, t^{n+1/2})$. The reason we do this is that the $P^1$ distribution of the solution is only defined in terms of the conserved variable $\mathbf{u}_h|_{K_j}$; the distribution of the primitive variable $\mathbf{w}_h|_{K_j}$ in an element is not even a linear function. Thus, the conversion between conserved and primitive variables is only valid at a point, but not in the entire element. The final update formula to obtain the input values for a Riemann solver becomes

$$\mathbf{w}_h(\boldsymbol{x}_{e_i}, t^{n+1/2}) = \widetilde{\mathbf{w}}_h(\boldsymbol{x}_{e_i}, t^{n+1/2}) + \frac{\Delta t}{2\epsilon} \mathbf{s}\big(\mathbf{w}_h(\boldsymbol{x}_{e_i}, t^{n+1/2})\big), \quad (2.78)$$

and in the case of a linear source, $\dfrac{1}{\epsilon(\mathbf{w})}\mathbf{s}(\mathbf{w}) = \mathbf{Q}_{\mathrm{w}}\mathbf{w}$, with a constant relaxation time, $\epsilon(\mathbf{w}) = \text{constant}$, we have

$$\mathbf{w}_h(\boldsymbol{x}_{e_i}, t^{n+1/2}) = \left[\mathbf{I} - \frac{\Delta t}{2}\mathbf{Q}_{\mathrm{w}}\right]^{-1} \widetilde{\mathbf{w}}(\boldsymbol{x}_{e_i} t^{n+1/2}). \quad (2.79)$$

If the relaxation time depends on the solution, as in the 10-moment equations, an iterative method must be applied to solve the implicit formula in the primitive form (2.78), or conserved form (2.76); the Newton–Raphson method is sufficient.

An alternative is treating the whole predictor step in the primitive form:

$$\mathbf{w}_h(\boldsymbol{x}_{e_i}, t^{n+1/2}) = \mathbf{w}(\boldsymbol{x}_{c,K_j}, t^n) - \frac{\Delta t}{2} \left(\mathbf{A}_{\mathrm{w}}\, \Delta_x \mathbf{w}_j^n + \mathbf{B}_{\mathrm{w}}\, \Delta_y \mathbf{w}_j^n\right)$$

$$+ \phi_j^n \left(\Delta_x \mathbf{w}_j^n, \Delta_y \mathbf{w}_j^n\right) \cdot (\boldsymbol{x} - \boldsymbol{x}_{c,K_j}) + \frac{\Delta t}{2\epsilon} \mathbf{s}\big(\mathbf{w}_h(\boldsymbol{x}_{e_i}, t^{n+1/2})\big), \quad (2.80)$$

where $\mathbf{A}_{\mathrm{w}}, \mathbf{B}_{\mathrm{w}}$ are the coefficient matrix of the primitive equations, and the slopes of primitive quantities at the centroid are

$$\left(\Delta_x \mathbf{w}_j^n, \Delta_y \mathbf{w}_j^n\right) = \mathbf{M}^{-1}\left(\overline{\Delta_x \mathbf{u}}_j^n, \overline{\Delta_y \mathbf{u}}_j^n\right); \quad \mathbf{M} := \frac{\partial \mathbf{u}}{\partial \mathbf{w}}. \tag{2.81}$$

### 2.3.7 Volume Integral of Flux and Source Term

In the evolution equations of the degrees of freedom (2.64), three volume integrals need to be evaluated numerically: the source term, the flux, and the moment of the source term. All integrals are evaluated by the $L$-stable two-point Radau IIA quadrature in time in view of the stiffness of the source term. Linearization of the source term allows a simpler implicit treatment, i.e., the cell-average update equations are independent of the updates of the other two degrees of freedom. As for the flux integral, either a Gauss quadrature or a Gauss–Lobatto quadrature in space is adopted. Figure 2.8 shows locations of quadrature points for both quadrilateral and triangular elements. Note that the quadrature points for the flux (●) and the source term (▲) are different. In Chapter III, it may shown that the order of spatial and temporal integration of the flux affects the accuracy of the DG(1)–Hancock method. From numerical results, it was concluded that spatial integration at each time level needs to be carried out first, then a time integration is applied to spatially averaged quantities.

#### Volume Integral of the Source Term

The volume integral of the source term appears in the cell average update equation (2.64a). The quadrature points are shown in Figure 2.8 as the triangular symbol (▲). In order to make the update equations of cell-averaged quantities independent to other two update formulas of the degrees of freedom, the source term

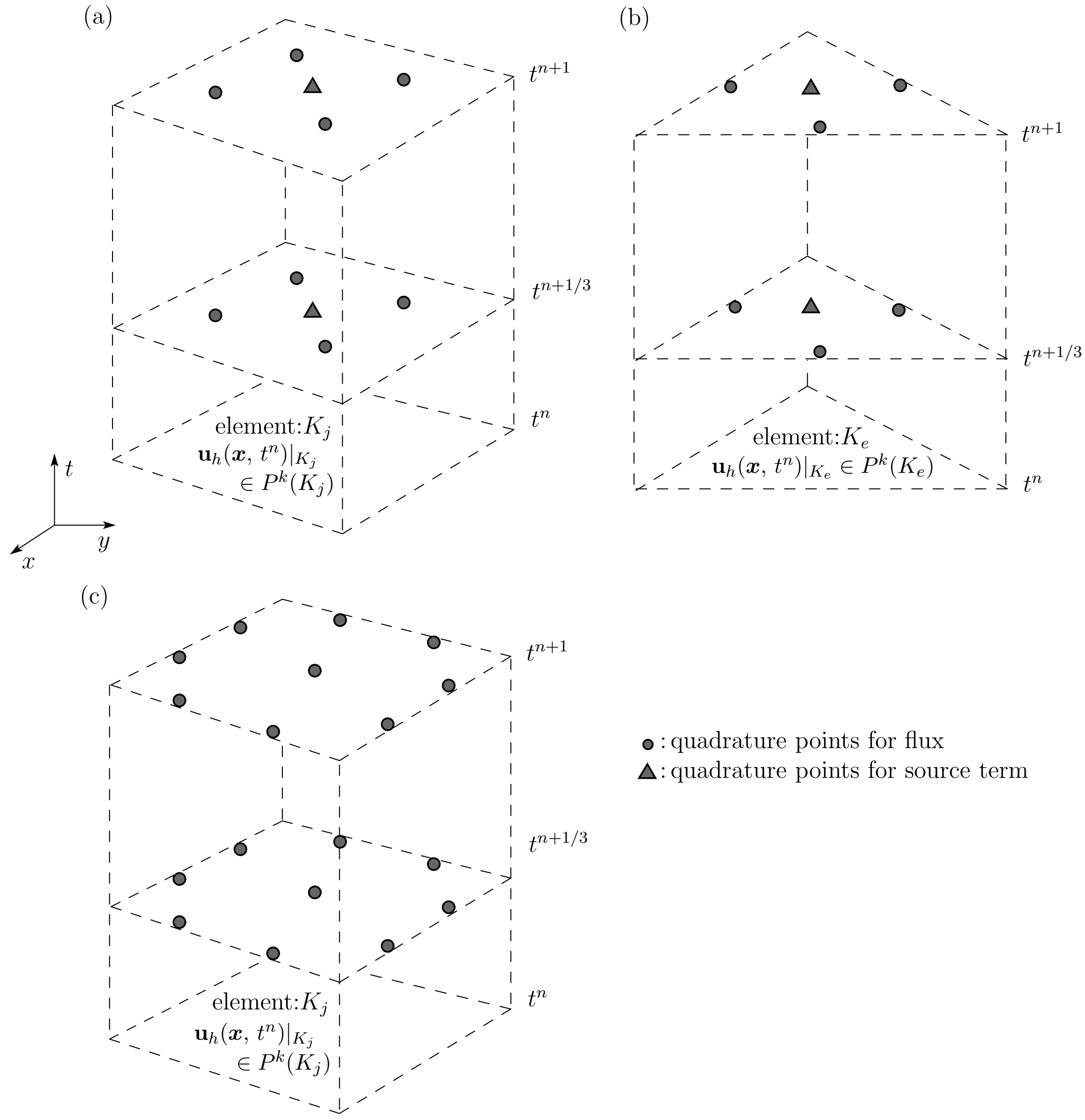

(a)
$t^{n+1}$
$t^{n+1/3}$
$t^n$
element:$K_j$
$\mathbf{u}_h(\boldsymbol{x}, t^n)|_{K_j}$
$\in P^k(K_j)$
$t$
$y$
$x$
(b)
$t^{n+1}$
$t^{n+1/3}$
$t^n$
element:$K_e$
$\mathbf{u}_h(\boldsymbol{x}, t^n)|_{K_e} \in P^k(K_e)$
(c)
$t^{n+1}$
$t^{n+1/3}$
$t^n$
element:$K_j$
$\mathbf{u}_h(\boldsymbol{x}, t^n)|_{K_j}$
$\in P^k(K_j)$
● : quadrature points for flux
▲ : quadrature points for source term

Figure 2.8: The quadrature points required for the volume integrals of flux and source term are denoted as bullet (●) and triangle (▲) respectively. For the flux integration, four points are necessary at each time level for a quadrilateral element (a), while three points along edges are required for a triangular element (b). Alternative quadrature points for a quadrilateral element are shown in (c), with the benefit that conserved quantities at the Gauss points along edges are already computed when a Riemann flux has to be calculated.

is linearized such that

$$\frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_h(\boldsymbol{x},t)) \approx \frac{1}{\epsilon(\bar{\mathbf{u}}_j(t))}\mathbf{s}\big(\bar{\mathbf{u}}_j(t)\big)+\mathbf{Q}\big(\bar{\mathbf{u}}_j(t)\big)\ \Big(\overline{\Delta_x\mathbf{u}}_j(t),\overline{\Delta_y\mathbf{u}}_j(t)\Big)\cdot(\boldsymbol{x}-\boldsymbol{x}_{c,K_j}), \quad (2.82)$$

where $\mathbf{Q}(\mathbf{u}) = \dfrac{\partial(\mathbf{s}/\epsilon)}{\partial\mathbf{u}}$. By approximating the above linearization, integration in space is evaluated analytically, and the volume integral can be simplified together with the two-point Radau IIA method such that

$$\begin{aligned}\iint\limits_{K_j\times T}\frac{1}{\epsilon}\mathbf{s}\big(\mathbf{u}_h(\boldsymbol{x},t)\big)\,d\boldsymbol{x}dt &\approx |A_j|\int\limits_{T}\frac{1}{\epsilon}\mathbf{s}\big(\bar{\mathbf{u}}_j(t)\big)\,dt\\ &\approx |A_j|\,\Delta t\left[\frac{3}{4}\frac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1/3})}{\epsilon(\bar{\mathbf{u}}_j^{n+1/3})}+\frac{1}{4}\frac{\mathbf{s}(\bar{\mathbf{u}}_j^{n+1})}{\epsilon(\bar{\mathbf{u}}_j^{n+1})}\right]. \quad (2.83)\end{aligned}$$

### Volume Integral of the Flux

Volume integrals of the flux appear in the update formula of the slope quantities (2.64b). The space-time volume integral is first approximated by the two-point Radau IIA quadrature in time:

$$\iint\limits_{K_j\times T}\mathbf{f}(\mathbf{u}_h(\boldsymbol{x},t))\,d\boldsymbol{x}dt \approx \Delta t\int\limits_{K_j}\left[\frac{3}{4}\mathbf{f}\big(\mathbf{u}(\boldsymbol{x},t^{n+1/3})\big)+\frac{1}{4}\mathbf{f}\big(\mathbf{u}(\boldsymbol{x},t^{n+1})\big)\right]\,d\boldsymbol{x}. \quad (2.84)$$

The next step is quadrature of the flux in space at the time levels $t^{n+1/3}, t^{n+1}$. Since the shape of an element can be any polygon, a coordinate transformation from the global (physical) coordinate, $\boldsymbol{x} = (x, y)$, to the local (computational) coordinate, $\boldsymbol{\xi} = (\xi, \eta)$, is necessary. In this section, we keep the formulation in general form to make it valid for both quadrilateral and triangular elements. Let $\mathbf{J}$ be the Jacobian matrix of the coordinate transformation from $\boldsymbol{\xi}$ to $\boldsymbol{x}$, thus $\mathbf{J} := \dfrac{\partial\boldsymbol{x}}{\partial\boldsymbol{\xi}} \in \mathbb{R}^{2\times2}$, then the integration over the domain $K_j$ can be transform to the local domain $\widehat{K}_j := [-1, 1] \times [-1, 1]$ such that

$$\int\limits_{K_j}\mathbf{f}(\mathbf{u}(\boldsymbol{x},t))\,d\boldsymbol{x} = \int\limits_{\widehat{K}_j}\mathbf{f}(\mathbf{u}(\boldsymbol{\xi},t))|\mathbf{J}(\boldsymbol{\xi})|\,d\boldsymbol{\xi}, \quad (2.85)$$

where $|\mathbf{J}(\boldsymbol{\xi})|$ is the Jacobian determinant. In the case of a triangular element, the Jacobian determinant is constant and equivalent to the area of a triangle, which further simplifies the above quadrature. Conversely, for a quadrilateral element, both the flux and Jacobian determinant at a quadrature point in the local domain $\widehat{K}_j$ have to be evaluated simultaneously. The quadrature points for quadrilateral and triangular elements are indicated in Figure 2.8(a) and (b) by the bullet symbol (●). Cockburn and Shu propose to recycle extrapolated quantities already computed for a Riemann solver [CS98, p. 206]. This leads to the nine-point quadrature shown in Figure 2.8(c). The only extra computational work is computing the flux at the centroid of the local element $\widehat{K}_j$. Finally, the volume integral of the flux is replaced by quadrature:

$$\iint\limits_{K_j \times T} \mathbf{f}(\mathbf{u}_h(\boldsymbol{x}, t))\, d\boldsymbol{x}dt \approx \Delta t \sum_i w_i |\mathbf{J}(\boldsymbol{\xi}_i)| \left[ \frac{3}{4} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^{n+1/3})) + \frac{1}{4} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^{n+1})) \right], \tag{2.86a}$$

where $w_i$ are the weights at Gauss points. The detailed implementation of the coordinate transformation for quadrilateral and triangular elements are provided in Appendix A on page 343.

### Volume Integral of the Moment of the Source Term

The volume integral of the moment of the source term appears in the evolution equations of slope quantities (2.64b) on page 64. Following the procedure taken for the source-term volume integral, the source term linearization (2.82) is assumed. Hence, the spatial integration is done analytically, while the two-point Radau IIA quadrature is applied in time. The resulting formulas are the following:

$$\begin{aligned}
&\iint\limits_{K_j\times T} (x-x_{c,K_j})\frac{1}{\epsilon}\mathbf{s}\big(\mathbf{u}_h(\boldsymbol{x},t)\big)\,d\boldsymbol{x}dt \\
&\qquad\approx K_{j1}\int\limits_T \mathbf{Q}\big(\bar{\mathbf{u}}(t)\big)\overline{\Delta_x\mathbf{u}}_j(t)\,dt + K_{j2}\int\limits_T \mathbf{Q}\big(\bar{\mathbf{u}}(t)\big)\overline{\Delta_y\mathbf{u}}_j(t)\,dt \\
&\qquad\approx \frac{3}{4}\Delta t\,\mathbf{Q}\big(\bar{\mathbf{u}}_j^{n+1/3}\big)\left(K_{j1}\overline{\Delta_x\mathbf{u}}_j^{n+1/3} + K_{j2}\overline{\Delta_y\mathbf{u}}_j^{n+1/3}\right) \\
&\qquad\quad + \frac{1}{4}\Delta t\,\mathbf{Q}\big(\bar{\mathbf{u}}_j^{n+1}\big)\left(K_{j1}\overline{\Delta_x\mathbf{u}}_j^{n+1} + K_{j2}\overline{\Delta_y\mathbf{u}}_j^{n+1}\right),
\end{aligned} \tag{2.87a}$$

$$\begin{aligned}
&\iint\limits_{K_j\times T} (y-y_{c,K_j})\frac{1}{\epsilon}\mathbf{s}\big(\mathbf{u}_h(\boldsymbol{x},t)\big)\,d\boldsymbol{x}dt \\
&\qquad\approx \frac{3}{4}\Delta t\,\mathbf{Q}\big(\bar{\mathbf{u}}_j^{n+1/3}\big)\left(K_{j2}\overline{\Delta_x\mathbf{u}}_j^{n+1/3} + K_{j3}\overline{\Delta_y\mathbf{u}}_j^{n+1/3}\right) \\
&\qquad\quad + \frac{1}{4}\Delta t\,\mathbf{Q}\big(\bar{\mathbf{u}}_j^{n+1}\big)\left(K_{j2}\overline{\Delta_x\mathbf{u}}_j^{n+1} + K_{j3}\overline{\Delta_y\mathbf{u}}_j^{n+1}\right).
\end{aligned} \tag{2.87b}$$

### 2.3.8 Update Formulas in Discrete Form

In summary, discrete update formulations for the DG(1)–Hancock method are presented. Since the solution **u** is approximated by a piecewise linear function, we have three independent update equations for three vectors of degrees of freedom. Owing to the linearization of the source term, the update equations of the first degree of freedom, the cell averages, can be updated from $t^n$ to $t^{n+1}$ without updating the other two degrees of freedom. The actual discretized form including the intermediate update equation is as follows:

$$
\begin{aligned}
\begin{pmatrix} \bar{\mathbf{u}}_j^{n+1/3} \\ \\ \bar{\mathbf{u}}_j^{n+1} \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{u}}_j^{n} \\ \\ \bar{\mathbf{u}}_j^{n} \end{pmatrix} \\
- \frac{\Delta t}{|A_j|} \underbrace{\begin{pmatrix} \dfrac{1}{3} \displaystyle\sum_{e_i \in \partial K_j} |e_{i,K_j}| \left[ w_1\, \hat{\mathbf{f}}_n\big(\mathbf{u}_{e_{i,1},L}^{n+1/6}, \mathbf{u}_{e_{i,1},R}^{n+1/6}\big) + w_2\, \hat{\mathbf{f}}_n\big(\mathbf{u}_{e_{i,2},L}^{n+1/6}, \mathbf{u}_{e_{i,2},R}^{n+1/6}\big) \right] \\ \displaystyle\sum_{e_i \in \partial K_j} |e_{i,K_j}| \left[ w_1\, \hat{\mathbf{f}}_n\big(\mathbf{u}_{e_{i,1},L}^{n+1/2}, \mathbf{u}_{e_{i,1},R}^{n+1/2}\big) + w_2\, \hat{\mathbf{f}}_n\big(\mathbf{u}_{e_{i,2},L}^{n+1/2}, \mathbf{u}_{e_{i,2},R}^{n+1/2}\big) \right] \end{pmatrix}}_{\text{explicit}} \\
+ \Delta t \begin{pmatrix} \dfrac{5}{12}\mathbf{I} & -\dfrac{1}{12}\mathbf{I} \\ \dfrac{3}{4}\mathbf{I} & \dfrac{1}{4}\mathbf{I} \end{pmatrix} \underbrace{\begin{pmatrix} \dfrac{1}{\epsilon(\bar{\mathbf{u}}_j^{n+1/3})}\mathbf{s}(\bar{\mathbf{u}}_j^{n+1/3}) \\ \dfrac{1}{\epsilon(\bar{\mathbf{u}}_j^{n+1})}\mathbf{s}(\bar{\mathbf{u}}_j^{n+1}) \end{pmatrix}}_{\text{implicit}},
\end{aligned}
\tag{2.88a}
$$

where $\mathbf{I} \in \mathbb{R}^{m\times m}$. The Newton–Raphson method is adopted for the implicit source term.

Once the cell-averaged variables at the three time levels, $t^n, t^{n+1/3}, t^{n+1}$, are known, the volume integrals of the flux can be evaluated explicitly; then the final update formulas for the rest of the degrees of freedom (slope quantities) are only implicit with respect to the source term. Again, the Newton–Raphson method is applied to the following update formulas for the slope quantities:

$$
\begin{pmatrix}
\overline{\Delta_x \mathbf{u}}_j^{n+1/3} \\
\overline{\Delta_y \mathbf{u}}_j^{n+1/3} \\
\overline{\Delta_x \mathbf{u}}_j^{n+1} \\
\overline{\Delta_y \mathbf{u}}_j^{n+1}
\end{pmatrix}
=
\begin{pmatrix}
\overline{\Delta_x \mathbf{u}}_j^{n} \\
\overline{\Delta_y \mathbf{u}}_j^{n} \\
\overline{\Delta_x \mathbf{u}}_j^{n} \\
\overline{\Delta_y \mathbf{u}}_j^{n}
\end{pmatrix}
$$

$$
+ \Delta t
\begin{pmatrix}
\mathbf{K}_j & \mathbf{0} \\
\mathbf{0} & \mathbf{K}_j
\end{pmatrix}
\underbrace{
\begin{pmatrix}
\begin{aligned}
-\frac{1}{3} \sum_{e_i \in \partial K_j} |e_{i,K_j}| \Big[ & w_1 (x_{e_i,1} - x_{c,K_j}) \, \hat{\mathbf{f}}_n \big( \mathbf{u}_{e_{i,1},L}^{n+1/6}, \mathbf{u}_{e_{i,1},R}^{n+1/6} \big) \\
& + w_2 (x_{e_i,2} - x_{c,K_j}) \, \hat{\mathbf{f}}_n \big( \mathbf{u}_{e_{i,2},L}^{n+1/6}, \mathbf{u}_{e_{i,2},R}^{n+1/6} \big) \Big]
\end{aligned} \\
\begin{aligned}
-\frac{1}{3} \sum_{e_i \in \partial K_j} |e_{i,K_j}| \Big[ & w_1 (y_{e_i,1} - y_{c,K_j}) \, \hat{\mathbf{f}}_n \big( \mathbf{u}_{e_{i,1},L}^{n+1/6}, \mathbf{u}_{e_{i,1},R}^{n+1/6} \big) \\
& + w_2 (y_{e_i,2} - y_{c,K_j}) \, \hat{\mathbf{f}}_n \big( \mathbf{u}_{e_{i,2},L}^{n+1/6}, \mathbf{u}_{e_{i,2},R}^{n+1/6} \big) \Big]
\end{aligned} \\
\begin{aligned}
- \sum_{e_i \in \partial K_j} |e_{i,K_j}| \Big[ & w_1 (x_{e_i,1} - x_{c,K_j}) \, \hat{\mathbf{f}}_n \big( \mathbf{u}_{e_{i,1},L}^{n+1/2}, \mathbf{u}_{e_{i,1},R}^{n+1/2} \big) \\
& + w_2 (x_{e_i,2} - x_{c,K_j}) \, \hat{\mathbf{f}}_n \big( \mathbf{u}_{e_{i,2},L}^{n+1/2}, \mathbf{u}_{e_{i,2},R}^{n+1/2} \big) \Big]
\end{aligned} \\
\begin{aligned}
- \sum_{e_i \in \partial K_j} |e_{i,K_j}| \Big[ & w_1 (y_{e_i,1} - y_{c,K_j}) \, \hat{\mathbf{f}}_n \big( \mathbf{u}_{e_{i,1},L}^{n+1/2}, \mathbf{u}_{e_{i,1},R}^{n+1/2} \big) \\
& + w_2 (y_{e_i,2} - y_{c,K_j}) \, \hat{\mathbf{f}}_n \big( \mathbf{u}_{e_{i,2},L}^{n+1/2}, \mathbf{u}_{e_{i,2},R}^{n+1/2} \big) \Big]
\end{aligned}
\end{pmatrix}
}_{\text{explicit}}
$$

$$
+ \Delta t
\begin{pmatrix}
\mathbf{K}_j & \mathbf{0} \\
\mathbf{0} & \mathbf{K}_j
\end{pmatrix}
\underbrace{
\begin{pmatrix}
\sum_i w_i \, |\mathbf{J}(\boldsymbol{\xi}_i)| \left[ \frac{1}{2} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^n)) + \frac{1}{2} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^{n+1/3})) \right] \cdot (1,0) \\
\sum_i w_i \, |\mathbf{J}(\boldsymbol{\xi}_i)| \left[ \frac{1}{2} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^n)) + \frac{1}{2} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^{n+1/3})) \right] \cdot (0,1) \\
\sum_i w_i \, |\mathbf{J}(\boldsymbol{\xi}_i)| \left[ \frac{3}{4} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^{n+1/3})) + \frac{1}{4} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^{n+1})) \right] \cdot (1,0) \\
\sum_i w_i \, |\mathbf{J}(\boldsymbol{\xi}_i)| \left[ \frac{3}{4} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^{n+1/3})) + \frac{1}{4} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i, t^{n+1})) \right] \cdot (0,1)
\end{pmatrix}
}_{\text{explicit}}
$$

$$+\,\Delta t\begin{pmatrix}\frac{5}{12}\mathbf{I} & -\frac{1}{12}\mathbf{I}\\ \frac{3}{4}\mathbf{I} & \frac{1}{4}\mathbf{I}\end{pmatrix}\underbrace{\begin{pmatrix}\mathbf{Q}\big(\mathbf{u}_j^{n+1/3}\big) & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{Q}\big(\mathbf{u}_j^{n+1/3}\big) & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{Q}\big(\mathbf{u}_j^{n+1}\big) & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}\big(\mathbf{u}_j^{n+1}\big)\end{pmatrix}\begin{pmatrix}\overline{\Delta_x\mathbf{u}}_j^{n+1/3}\\ \overline{\Delta_y\mathbf{u}}_j^{n+1/3}\\ \overline{\Delta_x\mathbf{u}}_j^{n+1}\\ \overline{\Delta_y\mathbf{u}}_j^{n+1}\end{pmatrix}}_{\text{implicit}}, \tag{2.88b}$$

where $\mathbf{I}, \mathbf{K}_j \in \mathbb{R}^{m\times m}$. These completes the discretization of the DG(1)–Hancock method for two-dimensional problems.

## 2.4 The Original Hancock Method

The original Hancock method [vAvLR82], which is a second-order finite volume method, is described for the purpose of comparison to the DG(1)–Hancock method. Among finite-volume methods for hyperbolic systems, those of the Godunov-type have been most successful; these require an algorithm for solving the Riemann problem arising at each cell interface, either exactly or approximately. As explained in the previous section, a first-order finite volume method can be seen as the simplest of discontinuous Galerkin methods, with the local solution in $P^0$. Starting with the second method, instead of storing extra degrees of freedom, a finite-volume method reconstructs a higher-order polynomial in a cell by using information from neighboring cells.

The Hancock discretization for the multidimensional hyperbolic-relaxation equations (2.43) can be started from the update equation of cell-averaged variables (2.64a) in the DG formulation (on page 64). In general, evaluation of the volume-averaged source term, $\int\limits_{K_j} \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}(\boldsymbol{x}))\,d\boldsymbol{x}$, requires numerical quadrature; in particular, it is not equivalent to standard finite-volume practice of evaluating the source term based on the cell-average, $\mathbf{s}(\bar{\mathbf{u}}(\boldsymbol{x}_j))$. For instance, in the 1-D case,

$$\begin{aligned}\bar{\mathbf{s}}_j(\mathbf{u}) : &= \frac{1}{\Delta x}\int\limits_{I_j} \mathbf{s}(\mathbf{u}(x))\,dx \\ &= \mathbf{s}(\bar{\mathbf{u}}_j) + O\big(\Delta x^2\big)\,. \end{aligned} \tag{2.89}$$

In a second-order accurate method such as described below, though, the average source term $\bar{\mathbf{s}}_j(\mathbf{u})$ can be replaced by $\mathbf{s}(\bar{\mathbf{u}}_j)$. Similarly, the spatial integral along edges of an element can be approximated by the midpoint rule; the interface flux at the element $e_i$ is evaluated at the edge center $\boldsymbol{x}_{e_i,c}$ such that

$$\iint\limits_{e_i\times T} \mathbf{f}\big(\mathbf{u}_h|_{K_j}, \mathbf{u}_h|_{K_e}\big)\cdot \mathbf{n}_{e_i,K_j}\,d\Gamma dt \approx |e_{i,K_j}| \int\limits_T \hat{\mathbf{f}}_n\big(\mathbf{u}_j(\boldsymbol{x}_{e_i,c},t), \mathbf{u}_e(\boldsymbol{x}_{e_i,c},t)\big)\,dt, \tag{2.90}$$

where $\hat{\mathbf{f}}_n$ is a Riemann flux outward of $K_j$ and normal to the edge $e_j$, and $|e_{i,K_j}|$ is the length of the edge $e_i$ belonging to the element $K_j$. Figure 2.9 shows the quadrature points (midpoint rule) where a Riemann solver is applied. Consequently, finite-volume methods of second-order accuracy in space can be written in fully discrete form as

$$\bar{\mathbf{u}}_j^{n+1} = \bar{\mathbf{u}}_j^n - \frac{1}{|A_j|}\sum_{e_i\in\partial K_j}\left[|e_{i,K_j}|\int\limits_T \hat{\mathbf{f}}_n\big(\mathbf{u}_j(\boldsymbol{x}_{e_i,c},t), \mathbf{u}_e(\boldsymbol{x}_{e_i,c},t)\big)\,dt\right] + \int\limits_T \frac{1}{\epsilon}\mathbf{s}(\bar{\mathbf{u}}_j(t))\,dt. \tag{2.91}$$

Note that time integrations of the flux and source term along the time domain $T := [t^n, t^{n+1}]$ have not been specified yet. Second-order accuracy in space and time is achieved by introducing linear subcell distributions and evaluating fluxes and source terms halfway during the time step:

$$\int\limits_T \hat{\mathbf{f}}_n\big(\mathbf{u}_j(\boldsymbol{x}_{e_i,c},t), \mathbf{u}_e(\boldsymbol{x}_{e_i,c},t)\big)\,dt \approx \Delta t\,\hat{\mathbf{f}}_n\big(\mathbf{u}_j(\boldsymbol{x}_{e_i,c},t^{n+1/2}), \mathbf{u}_e(\boldsymbol{x}_{e_i,c},t^{n+1/2})\big), \tag{2.92a}$$

$$\int\limits_T \frac{1}{\epsilon}\mathbf{s}(\bar{\mathbf{u}}_j(t))\,dt \approx \frac{\Delta t}{\epsilon}\mathbf{s}\big(\bar{\mathbf{u}}_j(t^{n+1/2})\big). \tag{2.92b}$$

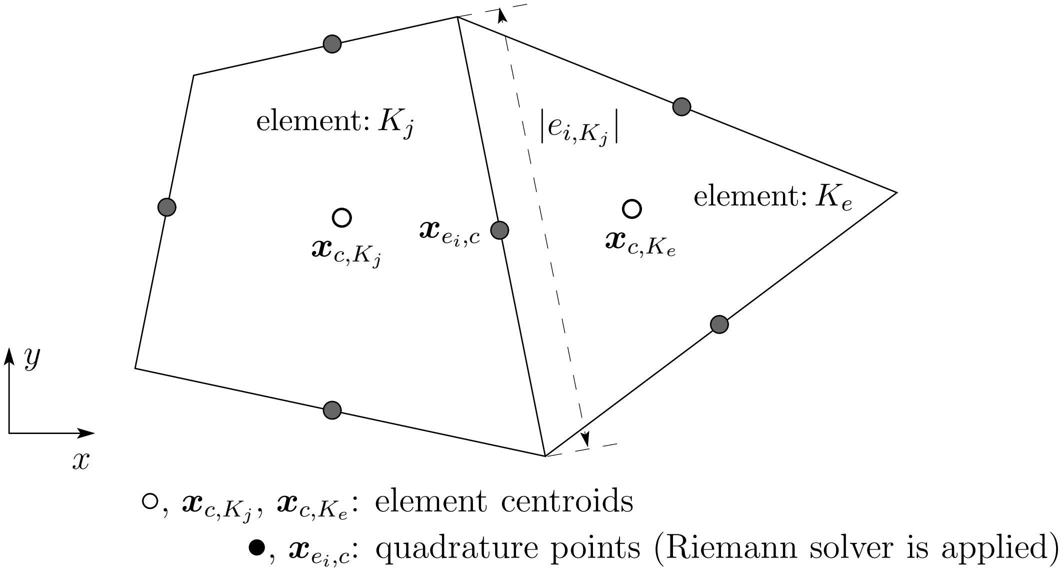

element: $K_j$
$|e_{i,K_j}|$
element: $K_e$
$\boldsymbol{x}_{c,K_j}$
$\boldsymbol{x}_{e_i,c}$
$\boldsymbol{x}_{c,K_e}$
$y$
$x$
○, $\boldsymbol{x}_{c,K_j}$, $\boldsymbol{x}_{c,K_e}$: element centroids
●, $\boldsymbol{x}_{e_i,c}$: quadrature points (Riemann solver is applied)

Figure 2.9: Quadrature points for Hancock's original method. The bullet symbol (●) denotes quadrature points where a Riemann flux is computed. The midpoint rule guarantees the second-order approximation of the surface integral of the flux along the edge $e_j$.

Here, the source term integration is treated explicitly for the sake of simplicity. However, due to the explicit treatment, the time step will suffer by the stiffness of the source term, i.e., $\Delta t \sim \min\left(\frac{h}{|\lambda|_{\max}},\ \epsilon\right)$, where $\lambda_{\max}$ is the maximum eigenvalue of the flux Jacobian.

Inserting the above approximate time integrals into (2.91) results in the fully discrete Hancock method:

$$\bar{\mathbf{u}}_j^{n+1} = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{|A_j|} \sum_{e_i \in \partial K_j} \hat{\mathbf{f}}_n\big(\mathbf{u}_j(\boldsymbol{x}_{e_i,c}, t^{n+1/2}), \mathbf{u}_e(\boldsymbol{x}_{e_i,c}, t^{n+1/2})\big)\, |e_{i,K_j}| + \frac{\Delta t}{\epsilon}\mathbf{s}\big(\bar{\mathbf{u}}_j(t^{n+1/2})\big). \tag{2.93}$$

The half-time (predictor) step, which includes gradient-limiting, to obtain $\mathbf{u}_j(\boldsymbol{x}, t^{n+1/2})$ is done in terms of primitive variables, $\mathbf{w}(\boldsymbol{x}, t)$, instead of conserved variables, $\mathbf{u}(\boldsymbol{x}, t)$, to prevent non-physical values such as negative pressures. Let $\mathbf{M}$ be the Jacobian matrix defined by

$$\mathbf{M} := \frac{\partial \mathbf{u}}{\partial \mathbf{w}}, \tag{2.94}$$

then the two-dimensional hyperbolic-relaxation equations can be reformulated in the primitive form:

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{A}_{\mathrm{w}} \frac{\partial \mathbf{w}}{\partial x} + \mathbf{B}_{\mathrm{w}} \frac{\partial \mathbf{w}}{\partial y} = \mathbf{M}^{-1} \frac{1}{\epsilon} \mathbf{s}(\mathbf{w}), \tag{2.95}$$

where $\mathbf{A}_{\mathrm{w}}, \mathbf{B}_{\mathrm{w}} \in \mathbb{R}^{m \times m}$ are the coefficient matrices of the primitive equations [1] obtained by a similarity transformation of the flux Jacobian:

$$\{\mathbf{A}_{\mathrm{w}}, \mathbf{B}_{\mathrm{w}}\} := \mathbf{M}^{-1} \left( \frac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}} \right) \mathbf{M}; \quad \mathbf{f} \in \mathbb{R}^{m \times 2}. \tag{2.96}$$

When the 10-moment equations are considered, owing to its simple structure, the source term is not affected by the variable transformation: $\mathbf{M}^{-1} \frac{1}{\epsilon} \mathbf{s} \equiv \frac{1}{\epsilon} \mathbf{s}$. Since the divergence theorem can not be applied to the primitive form, we discretize (2.95) by finite-differencing. Applying the forward Euler method in time, the primitive quantities at the half time step, $\mathbf{w}_j^{n+1/2}$, are approximated by

$$\mathbf{w}_j^{n+1/2} = \mathbf{w}_j^n - \frac{\Delta t}{2} \left( \mathbf{A}_{\mathrm{w}} \, \Delta_x \mathbf{w}_j^n + \mathbf{B}_{\mathrm{w}} \, \Delta_y \mathbf{w}_j^n \right) + \frac{\Delta t}{2\epsilon} \mathbf{s}(\mathbf{w}_j^n), \tag{2.97}$$

where the gradients of the primitive variables, $\nabla \mathbf{w}_j^n = (\Delta_x \mathbf{w}_j^n, \Delta_y \mathbf{w}_j^n)$, are obtained by either the Green–Gauss formula [BF90] or solving a least-square problems [Bar93, OGvA02] involving data from all adjacent cells. In general, the least-squares gradient reconstruction is more robust than the Green–Gauss contour integral, yet various ways of weighting residuals are possible in the former method, and the unweighted approach this is often taken is not even the best [Mav07].

Once primitive variables at half-time are obtained, interface fluxes are computed by solving Riemann problems. Finally, the full-time (corrector) step to update

[1] The resulting matrix $A_n$ should not be called as the primitive *Jacobian* since the definition of a Jacobian matrix is the derivative of one vector to another one. The reader is referred to helpful hints in [vL06] for preventing misuse of terminologies.

conservative variables can be written as

$$\bar{\mathbf{u}}_j^{n+1} = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{|A_j|} \sum_{e_i \in \partial K_j} \hat{\mathbf{f}}_n\big(\widetilde{\mathbf{w}}_j(\boldsymbol{x}_{e_i,c}, t^{n+1/2}), \widetilde{\mathbf{w}}_e(\boldsymbol{x}_{e_i,c}, t^{n+1/2})\big)\, |e_{i,K_j}| + \frac{\Delta t}{\epsilon} \mathbf{s}\big(\mathbf{w}_j(t^{n+1/2})\big), \tag{2.98}$$

where the primitive quantities, $\mathbf{w}_j^{n+1/2}$, associated with the centroid of a cell, are linearly extrapolated to the midpoint $\boldsymbol{x}_{e_i,c}$ of the edge $e_i$ by

$$\widetilde{\mathbf{w}}_j(\boldsymbol{x}, t^{n+1/2}) = \mathbf{w}_j^{n+1/2} + \phi_i\ (\nabla \mathbf{w}_j^n) \cdot (\boldsymbol{x} - \boldsymbol{x}_{c,K_j}). \tag{2.99}$$

Here $\phi_i$ is a gradient limiter such as the double-minmod limiter [Bar90], and $\boldsymbol{x}_{c,K_j}$ is the centroid of cell $K_j$. For details of implementation one is referred to [DZ93, pp. 49–57]. Note that the slope quantities, $\nabla \mathbf{w}_j^n$, are used for the reconstruction step; ideally, slopes associated with the time level $t^{n+1/2}$ need to be computed, yet old slopes are sufficient for a second-order method.

## 2.5 Semi-Discrete Methods

The DG(1)–Hancock and Hancock methods introduced in the previous sections are fully discrete methods. This means that the spatial and temporal derivatives are discretized simultaneously. The other approach to discretizing hyperbolic-relaxation equations (2.1) is based on the method-of-lines (MOL), which decouples the spatial and temporal discretizations. The advantages of adopting an MOL are the simplification of the design of a scheme, flexibility to combine a suitable spatial and temporal methods, and ease of implementation. While general-purpose semi-discrete finite-difference, finite-volume, or discontinuous Galerkin methods are employed as spatial discretizations, a number of time discretizations (ODE solvers) have been developed specifically for hyperbolic-relaxation equations.

### 2.5.1 Time Integration with a Stiff Source Term

When a semi-discrete method is considered, the 1-D hyperbolic-relaxation equations can be written in the form:

$$\frac{\partial \mathbf{u}_j(t)}{\partial t} = -\frac{\partial \mathbf{f}(\mathbf{u}_j(t))}{\partial x} + \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_j(t)), \tag{2.100}$$

where the solution $\mathbf{u}_j$ is still a continuous function in time. Due to the stiffness introduced by the relaxation time $\epsilon$ in the source term, the source term needs to be treated implicitly for stability. Conversely, the flux is evaluated explicitly. One of the difficulties in designing such a method is that when the flow is in equilibrium, thus $\epsilon \to 0$, the method has to be consistent with the hyperbolic conservation laws (2.2) at the discretization level. This property guarantees that a underresolved method correctly captures the macroscopic behavior. Jin named such schemes *asymptotic preserving* (AP) [Jin99]. Numerous methods have been proposed to achieve the AP property [Jin95, CJR97, Jin99, LRR00, Rus02]. These methods are summarized as the Butcher array in [PR03].

Among the family of these Runge–Kutta methods, Pareschi and Russo extend the implicit-explicit (IMEX) Runge–Kutta method, originally developed for advection-diffusion problems [ARS97, ARW95], to hyperbolic-relaxation equations [PR05]. The methods utilize an explicit, strongly-stability-preserving (SSP) method for the flux, and an $L$-stable, diagonally implicit Runge–Kutta method for the source term. The methods are shown to possess the AP property at the equilibrium limit. They use the notation IMEX–SSP$k(s, \sigma, p)$ where $k$ is the order of the SSP scheme, $s$ the number of states of the implicit method, $\sigma$ the number of stages of the explicit method, and $p$ the order of the IMEX method. Here, we adopt a second-order IMEX Runge–Kutta method: IMEX–SSP2(3,3,2) as the time integration for the

MOL approach. This time integrator requires three stages for both explicit and implicit terms to achieve second-order accuracy. The update formulas are given by

$$\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}^n + \frac{\Delta t}{4\epsilon}\mathbf{s}(\mathbf{u}^{(1)}), \\
\mathbf{u}^{(2)} &= \mathbf{u}^n - \frac{\Delta t}{2}\partial_x \mathbf{f}(\mathbf{u}^{(1)}) + \frac{\Delta t}{4\epsilon}\mathbf{s}(\mathbf{u}^{(2)}), \\
\mathbf{u}^{(3)} &= \mathbf{u}^n - \frac{\Delta t}{2}\left[\partial_x \mathbf{f}(\mathbf{u}^{(1)}) + \partial_x \mathbf{f}(\mathbf{u}^{(2)})\right] + \frac{\Delta t}{3\epsilon}\left[\mathbf{s}(\mathbf{u}^{(1)}) + \mathbf{s}(\mathbf{u}^{(2)}) + \mathbf{s}(\mathbf{u}^{(3)})\right], \\
\mathbf{u}^{n+1} &= \mathbf{u}^n - \frac{\Delta t}{3}\left[\partial_x \mathbf{f}(\mathbf{u}^{(1)}) + \partial_x \mathbf{f}(\mathbf{u}^{(2)}) + \partial_x \mathbf{f}(\mathbf{u}^{(3)})\right] + \frac{\Delta t}{3\epsilon}\left[\mathbf{s}(\mathbf{u}^{(1)}) + \mathbf{s}(\mathbf{u}^{(2)}) + \mathbf{s}(\mathbf{u}^{(3)})\right].
\end{aligned} \tag{2.101}$$

# CHAPTER III

# ANALYSIS FOR 1-D AND 2-D LINEAR ADVECTION EQUATIONS

## 3.1 Introduction

In this chapter, a Fourier analysis is employed to uncover the linear properties of methods for hyperbolic conservation laws without source terms; hyperbolic-relaxation systems will be analyzed in Chapter IV. The analysis is also called a 'Von Neumann analysis' named after John von Neumann, who originally introduced the analysis for parabolic differential equations [vNR47, vNR63]. The actual applications of the analysis can be found in many textbooks [RM67, TAP97]. The analysis shows the order of accuracy, the dominant numerical dissipation/dispersion errors for the low-frequency mode, and the linear stability of a method. Note that the assumptions required for a Fourier analysis are uniform grids and periodic boundary conditions. The dimensionless 1-D and 2-D linear advection equations,

$$\partial_t u + r\partial_x u = 0, \quad |r| \leq 1, \tag{3.1a}$$

$$\partial_t u + r\partial_x u + s\partial_y u = 0, \quad |r|, |s| \leq 1, \tag{3.1b}$$

are considered as the model equations. Here, the normalization is rather uncommon. The advection speed is normalized by the larger 'frozen' wave speed (= 1) arising further on in hyperbolic-relaxation systems. The motivation of this normalization will be clear once hyperbolic-relaxation systems are considered in Chapter IV.

The Courant number, $\nu$, is defined by the dimensionless frozen wave speed, 1, instead of the advection speed, $r$, thus

$$\nu := 1\frac{\Delta t}{\Delta x}, \quad \Delta x, \Delta t \in \mathbb{R}^+. \tag{3.2}$$

Again, this definition is rather uncommon. Conventionally, the Courant number for a linear advection equation is defined by

$$\tilde{\nu} := r\frac{\Delta t}{\Delta x}, \tag{3.3}$$

where the advection speed, $r$, is normalized by the spatial and temporal scales. To make the analysis consistent with the results later presented for the linear hyperbolic-relaxation equations, we adopt $\nu$ as the Courant number here. The conventional expression can be recovered by substituting

$$r\nu = \tilde{\nu}. \tag{3.4}$$

Recall that, for a linear advection equation, both the upwind moment scheme (with Gauss–Lobatto quadrature for the volume integral) and the proposed method (with Gauss–Radau quadrature) are identical to Van Leer's scheme III [vL77]. The upwind moment method (DG(1)–Hancock) is compared with three other methodologies: a semi-discrete high-resolution Godunov method (HR) with method-of-lines (MOL) or Hancock time integration, and a DG(1)–MOL method. These methods can be regarded as the combination of a spatial and a temporal discretization, and are tabulated in Table 3.1. Here, we adopt the notations HR$s$ and RK$s$, where $s$ is the order of accuracy, and DG($k$) where $k$ is the degree of the polynomial basis. A Fourier analysis for the 1-D advection equation shows that the upwind moment

| | high-resolution Godunov (HR) | discontinuous Galerkin (DG) |
|---|---|---|
| Runge–Kutta (RK) | HR2–RK2, HR2–RK3 | DG(1)–RK2, DG(1)–RK3 |
| Hancock (Ha) | Hancock (HR2–Ha) | upwind moment (DG(1)–Ha) |

Table 3.1: The combinations of space and time discretization methods. First row: semi-discrete methods; second row: the fully discrete methods.

method is linearly stable up to the Courant number 1 with an upwind flux, whereas DG spatial discretizations combined with MOL typically have a more strict stability condition: for DG(1)–RK2 (second-order) the limit is $\frac{1}{3}$, and for the DG(2)–RK3 (third-order) it is $\frac{1}{5}$ [CS01, p. 191].

## 3.2 Methodology

### 3.2.1 Difference Operators in Fourier (Frequency) Space

To investigate and compare the properties of a method, it is useful to write a method in compact form. Let the forward, $\delta^+$, and backward, $\delta^-$, difference operator be

$$\delta^+\mathbf{u}_j = \mathbf{u}_{j+1} - \mathbf{u}_j, \tag{3.5a}$$

$$\delta^-\mathbf{u}_j = \mathbf{u}_j - \mathbf{u}_{j-1}, \tag{3.5b}$$

where $\mathbf{u}_j \in \mathbb{R}^n$. Then, translation over any number of cells can be expressed by applying these difference operators multiple times, e.g.,

$$\mathbf{u}_{j+2} = (I + \delta^+)^2\mathbf{u}_j, \tag{3.6a}$$

$$\mathbf{u}_{j-2} = (I - \delta^-)^2\mathbf{u}_j, \tag{3.6b}$$

where $I$ is the identity operator, that is, $\mathbf{u}_j = I\mathbf{u}_j$. Using these difference operators, the simplest fully discrete methods can be expressed as

$$\mathbf{u}_j^{n+1} = \mathbf{G}(\nu, r, q)\mathbf{u}_j^n, \tag{3.7}$$

where $\mathbf{G} \in \mathbb{R}^{n\times n}$ is an amplification factor or matrix, and $q$ is the dissipation parameter of the $q$-flux (3.27) [vL69]. Since the fully discrete method considered here is a multi-stage one-step method, it can be written as the forward Euler method in time, thus

$$\frac{\mathbf{u}_j^{n+1} - \mathbf{u}_j^n}{\Delta t} = \mathbf{M}(\nu, r, q, \Delta x)\,\mathbf{u}_j^n, \tag{3.8a}$$

or

$$\mathbf{u}_j^{n+1} = [\mathbf{I} + \Delta t\,\mathbf{M}(\nu, r, q, \Delta x)]\,\mathbf{u}_j^n, \tag{3.8b}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ is a 'spatial-temporal' difference operator. Comparing to the (3.7), the amplification matrix, $\mathbf{G}$, can be expressed in terms of $\mathbf{M}$:

$$\text{fully discrete:}\; \mathbf{G}_\mathrm{M} = \mathbf{I} + \Delta t\,\mathbf{M}. \tag{3.9}$$

Conversely, a semi-discrete method is only expressed in an ODE form:

$$\frac{\partial \mathbf{u}_j(t)}{\partial t} = \mathbf{N}(r, q, \Delta x)\,\mathbf{u}_j(t), \tag{3.10}$$

where $\mathbf{N} \in \mathbb{R}^{n\times n}$ is a 'spatial' difference operator. The notation of the two difference operators is made distinct on purpose: to a fully discrete method is assigned $\mathbf{M}$, and to a semi-discrete method $\mathbf{N}$. Note that since a semi-discrete method is only discretized in space, $\mathbf{N}$ is not a function of $\nu$, thus an ODE solver is in charge of the temporal discretization. Here, two RK methods are adopted for the time integration in semi-discrete methods. The two- and three-stage RK methods are second- and third-order accurate in time respectively. For a system of linear equations, a RK method simply generates the series expansion of the matrix exponential $e^{\mathbf{N}\Delta t}$ up to a certain order, thus

$$\text{RK2: } \mathbf{G}_{\text{N}} = \mathbf{I} + \Delta t\, \mathbf{N} + \frac{\Delta t^2}{2}\mathbf{N}^2, \tag{3.11a}$$

$$\text{RK3: } \mathbf{G}_{\text{N}} = \mathbf{I} + \Delta t\, \mathbf{N} + \frac{\Delta t^2}{2}\mathbf{N}^2 + \frac{\Delta t^3}{6}\mathbf{N}^3. \tag{3.11b}$$

In a Fourier analysis, the investigation of the amplification factor $\mathbf{G}$ is the main interest. When a finite-volume method is applied to a scalar linear equation, this factor is a scalar, thus the derivation is straightforward. However, when a DG method is applied, even though the target equation is a scalar equation, the amplification factor becomes a matrix due to the introduction of extra variables in each cell. To extract the behavior of a method, eigenvalues of $\mathbf{G}$ must be obtained by solving a characteristic equation,

$$\det(\mathbf{G} - g\mathbf{I}) = 0, \tag{3.12}$$

where $g$ is an eigenvalue of $\mathbf{G}$. Later, a Fourier analysis is extended to the $2 \times 2$ linear hyperbolic-relaxation equations. In this case, a DG(1) method produces a $4 \times 4$ amplification matrix, and an analysis directly dealing with the amplification matrix, $\mathbf{G}$, becomes cumbersome.

Instead of directly computing eigenvalues from $\mathbf{G}$, the derivation is simplified by assuming that the spatial matrix operator $\mathbf{N}$ is diagonalizable such that $\mathbf{N} = \mathbf{R}\boldsymbol{\Lambda}_{\text{N}}\mathbf{R}^{-1}$. A sufficient condition for this assumption is that the characteristic polynomial of $\mathbf{N}$ has $n$ distinct eigenvalues. Inserting this relation into, for instance, (3.11a) leads to,

$$\text{RK2: } \mathbf{R}^{-1}\mathbf{G}_{\text{N}}\mathbf{R} = \mathbf{I} + \Delta t\boldsymbol{\Lambda}_{\text{N}} + \frac{\Delta t^2}{2}\boldsymbol{\Lambda}_{\text{N}}^2. \tag{3.13}$$

Since the similarity transformation is eigenvalue-invariant, eigenvalues of $\mathbf{N}$ are computed first, and then inserted into the above equation to obtain the eigenvalues of $\mathbf{G}_{\mathrm{N}}$. This significantly reduces the complexity of the derivation for an MOL-based method since we only need eigenvalues of the spatial matrix operator, $\mathbf{N}$, not the amplification matrix, $\mathbf{G}_{\mathrm{N}}$, in the first place. Let $g^{(i)}$ be the amplification factor corresponding to the $i$-th eigenvalue $\lambda_i$ of $\mathbf{N}$, then (3.13) is replaced by

$$\text{RK2: } g^{(i)} = 1 + \Delta t \lambda_i + \frac{\Delta t^2}{2} \lambda_i^2. \tag{3.14}$$

This is an example of the spectral mapping theorem [Var62], [Hir89, p. 297].

For a single Fourier mode of wavelength $N\Delta x$, the solution at cell $j$ can be represented as follows:

$$\begin{aligned} \mathbf{u}_j &= \hat{\mathbf{u}}_0 \exp\left( i \frac{2\pi j}{N} \right) \\ &= \hat{\mathbf{u}}_0 \exp(i\beta j), \quad \beta \in [-\pi, \pi], \end{aligned} \tag{3.15}$$

where $\beta$ is the spatial frequency of the wave. Here, $\beta = 0$ corresponds to the low-frequency limit, and $\beta = \pi$ is the high-frequency limit. Using a Fourier representation, the difference operators are now replaced by exponential functions,

$$\delta^{+} = e^{i\beta} - 1, \tag{3.16a}$$

$$\delta^{-} = 1 - e^{-i\beta}. \tag{3.16b}$$

Inserting these relations into a matrix operator, $\mathbf{M}$ or $\mathbf{N}$, removes the difference operators in the amplification matrix.

### 3.2.2 Exact Solution

An exact solution is critical to examining the order of accuracy of a method. The exact solution of (3.1a) in the harmonic mode is given by

$$\mathbf{u}(x,t) = \hat{\mathbf{u}}_0 e^{ik(x-rt)}, \tag{3.17}$$

where $k$ is the spatial wave number. The exact amplification factor is obtained by expressing $\mathbf{u}(x, t+\Delta t)$ in terms of $\mathbf{u}(x,t)$,

$$\begin{aligned}\mathbf{u}(x, t+\Delta t) &= \hat{\mathbf{u}}_0 e^{ik(x-rt)} e^{-irk\Delta t} \\ &= e^{-irk\Delta t}\mathbf{u}(x,t). \qquad (3.18)\end{aligned}$$

It shows that the exact amplification factor for the time step $\Delta t$, and the exact eigenvalue of the spatial discretization operator in (3.10) are given by

$$g_{\text{exact}} = e^{-irk\Delta t}, \qquad (3.19\text{a})$$

$$\lambda_{\text{exact}} = -irk. \qquad (3.19\text{b})$$

In a Fourier mode, the wave number $k$ is related to the frequency of a wave $\beta$ by

$$k = \frac{\beta}{\Delta x}, \qquad (3.20)$$

thus the amplification factor and spatial eigenvalue become

$$g_{\text{exact}}(\tilde{\nu}, \beta) = e^{-ir\nu\beta} = e^{-i\tilde{\nu}\beta}, \qquad (3.21\text{a})$$

$$\lambda_{\text{exact}} = -\frac{ir}{\Delta x}\beta. \qquad (3.21\text{b})$$

A good understanding of the properties of a method is obtained by rewriting an amplification factor in the polar form. An amplification factor can be expressed by the modulus $|g|$ and the phase angle $\phi \in [-\pi, \pi]$ such that

$$g = |g|e^{i\phi}, \qquad (3.22)$$

where

$$|g| = \sqrt{\Re(g)^2 + \Im(g)^2}, \qquad (3.23\text{a})$$

$$\phi = \arg(g) = \arctan\left[\frac{\Im(g)}{\Re(g)}\right]. \qquad (3.23\text{b})$$

The modulus represents the dissipation, and the phase angle represents the dispersion of a method. For the exact solution, we have

$$|g_{\text{exact}}| = 1, \tag{3.24a}$$

$$\phi_{\text{exact}} = -r\nu\beta = -\tilde{\nu}\beta. \tag{3.24b}$$

Later in an analysis, the low-frequency limit of a Fourier mode is compared with the exact solution. The exact amplification factor in the low-frequency limit is given by expanding in the frequency of a wave, $\beta$, with the fixed Courant number, $r\nu$, thus

$$g_{\text{exact}}(r\nu, \beta) = 1 + (-ir\nu)\beta + \frac{1}{2}(-ir\nu)^2\beta^2 + \frac{1}{6}(-ir\nu)^3\beta^3 + \frac{1}{24}(-ir\nu)^4\beta^4 + O\big(\beta^5\big)\,. \tag{3.25}$$

### 3.2.3 Example of the Analysis (First-Order Method)

Before we start analyzing a higher-order method, the first-order method is analyzed to demonstrate a Fourier analysis for both accuracy and stability. Here, a finite-volume discretization in space, and the forward Euler method in time are employed for the 1-D linear advection equation (3.1a). The resulting scheme is as follows:

$$\frac{\bar{u}_j^{n+1} - \bar{u}_j^n}{\Delta t} = -\frac{1}{\Delta x}\left(\hat{f}_{j+1/2}^n - \hat{f}_{j-1/2}^n\right), \tag{3.26}$$

where the linear flux, $f(u) = ru$, is evaluated at the time level $t^n$ at each cell interface, $j \pm 1/2$. Making the analysis more general, the $q$-flux [vL69], which parameterizes the amount of numerical dissipation added into the flux calculation, is employed as the flux function:

$$f^q_{j+1/2}(u_L, u_R) = \frac{r}{2}(u_L + u_R) - \frac{q}{2}(u_R - u_L), \quad \text{with} \quad q = \begin{cases} |r| & \text{upwind,} \\ 1 & \text{Rusanov, HLL1,} \\ \dfrac{\Delta x}{\Delta t} & \text{Lax–Friedrichs.} \end{cases} \tag{3.27}$$

The unity value in the Rusanov flux comes from the dimensionless frozen wave speed which, appears in the linear hyperbolic-relaxation equations. Here, the dissipation parameter, $q$, satisfies the inequality,

$$|r| \leq q \leq \frac{\Delta x}{\Delta t} = \frac{1}{\nu}. \tag{3.28}$$

Thus, it is easily seen that the upwind flux introduces the least numerical dissipation, whereas the Lax–Friedrichs (LxF) has the most. The first-order approximation of interface fluxes is given by using the cell-averaged values as input, thus for the $q$-flux,

$$\hat{f}^n_{j+1/2} = f^q_{j+1/2}(\bar{u}^n_j, \bar{u}^n_{j+1}), \tag{3.29a}$$

$$\hat{f}^n_{j-1/2} = f^q_{j-1/2}(\bar{u}^n_{j-1}, \bar{u}^n_j). \tag{3.29b}$$

Here, even though the first-order method (3.26) can be seen as a fully discrete method, strictly speaking, it is a semi-discrete method combined with the MOL approach, since the flux formula does not contain time variation. Inserting the difference equations of fluxes into the update equation, the resulting scheme has the three-point formula:

$$\begin{aligned} \frac{\bar{u}^{n+1}_j - \bar{u}^n_j}{\Delta t} &= -\frac{r}{2\Delta x}(\bar{u}^n_{j+1} - \bar{u}^n_{j-1}) + \frac{q}{2\Delta x}(\bar{u}^n_{j+1} - 2\bar{u}^n_j + \bar{u}^n_{j-1}) \\ &= \frac{q-r}{2\Delta x}\bar{u}^n_{j+1} - \frac{q}{\Delta x}\bar{u}^n_j + \frac{q+r}{2\Delta x}\bar{u}^n_{j-1}. \end{aligned} \tag{3.30}$$

Applying the difference operators (3.5) to the above equation leads to the compact form, thus

$$\begin{aligned}\bar{u}_j^{n+1} &= g_{\text{1st-order}}\bar{u}_j^n \\ &= \left(1 + \Delta t \; \mathrm{N}_{\text{1st-order}}\right)\bar{u}_j^n,\end{aligned} \tag{3.31}$$

where

$$\mathrm{N}_{\text{1st-order}} = \frac{q-r}{2\Delta x}\delta^{+} - \frac{q+r}{2\Delta x}\delta^{-}, \tag{3.32a}$$

or for a Fourier mode,

$$\mathrm{N}_{\text{1st-order}} = \frac{q-r}{2\Delta x}e^{i\beta} - \frac{q}{\Delta x} + \frac{q+r}{2\Delta x}e^{-i\beta}. \tag{3.32b}$$

### Shift Condition

When the upwind flux, $q = r$, is used, and the Courant number is set to unity, thus $r\dfrac{\Delta t}{\Delta x} = 1$, then the spatial difference operator multiplied by the time step, $\Delta t$, reduces to

$$\begin{aligned}\Delta t \, \mathbf{N}_{\text{1st-order}} &= -\Delta t\left(\frac{r}{\Delta x}\delta^{-}\right) \\ &= -\delta^{-},\end{aligned} \tag{3.33}$$

which is the exact upwind difference operator. Inserting the above equation into the update formula (3.31) leads to the exact solution:

$$\begin{aligned}\bar{u}_j^{n+1} &= (1 - \delta^{-})\bar{u}_j^n \\ &= \bar{u}_{j-1}^n.\end{aligned} \tag{3.34}$$

The property, which a method reduces to the exact solution with unity Courant number, is called the *shift condition*, or the *unit CFL condition* [LeV02, p. 85]. This property can be found in some fully discrete methods, but almost never in the high-order semi-discrete methods.

### Amplification Factor

Once the compact difference form is obtained, replacing the difference operators $\delta^{\pm}$ by the Fourier mode (3.16) leads to

$$\lambda_{\text{1st-order}} = -\frac{1}{\Delta x}\left[q(1-\cos\beta) + ir\sin\beta\right], \tag{3.35}$$

where we explicitly change the notation from N to $\lambda$, stating that $\lambda$ is the eigenvalue of N. In this example, $\lambda$ and N are identical, however, when a DG method is adopted for a scalar linear equation, or any scheme for a system of linear equations, the difference operator, $\mathbf{N}$, becomes a matrix. In these cases, the characteristic equation,

$$\det(\mathbf{N} - \lambda\mathbf{I}) = 0, \tag{3.36}$$

needs to be solved to obtain eigenvalues of spatial difference operators. Once the eigenvalue of the spatial difference operator, $\mathbf{N}$, is obtained, the amplification factor, $g$, is given by

$$\begin{aligned} g_{\text{1st-order}} &= 1 + \Delta t \lambda_{\text{1st-order}} \\ &= 1 - \nu\left[q(1-\cos\beta) + ir\sin\beta\right] && (3.37\text{a}) \\ &= 1 - \tilde{\nu}\left[\frac{q}{r}(1-\cos\beta) + i\sin\beta\right]. && (3.37\text{b}) \end{aligned}$$

To analyze the properties of the method, it is useful to rewrite the amplification factor in the polar form (3.22), thus

$$g_{\text{1st-order}} = |g_{\text{1st-order}}| e^{i\phi_{\text{1st-order}}}, \tag{3.38}$$

where

$$\begin{aligned} |g_{\text{1st-order}}| &= \sqrt{1 - 4\left[q\nu - (r\nu)^2 - \left((q\nu)^2 - (r\nu)^2\right)\sin^2\frac{\beta}{2}\right]\sin^2\frac{\beta}{2}}, && (3.39\text{a}) \\ \phi_{\text{1st-order}} &= \arctan\left[\frac{-r\nu\sin\beta}{1 - q\nu(1-\cos\beta)}\right]. && (3.39\text{b}) \end{aligned}$$

### Accuracy

Typically, we are interested in the behavior of a method in the low-frequency limit, $\beta \ll 1$. Taking the power-series expansion around $\beta = 0$ of the amplification factor and phase angle leads to

$$|g_{\text{1st-order}}| = 1 - \frac{1}{2}\left[q\nu - (r\nu)^2\right]\beta^2 + O\big(\beta^4\big)\,, \tag{3.40a}$$

$$\phi_{\text{1st-order}} = -r\nu\beta + \frac{1}{6}r\nu\left[1 - 3q\nu + 2(r\nu)^2\right]\beta^3 + O\big(\beta^5\big)\,. \tag{3.40b}$$

The equations (3.40) shows the amount of dissipation (amplitude error) appears in even orders of $\beta$, and the dispersion (phase error) in odd orders. The relative errors of dissipation and dispersion are obtained by comparing with the exact solution (3.24) respectively, thus

$$\frac{|g_{\text{1st-order}}|}{|g_{\text{exact}}|} = 1 + O\big(\beta^2\big)\,, \tag{3.41a}$$

$$\frac{\phi_{\text{1st-order}}}{\phi_{\text{exact}}} = 1 + O\big(\beta^2\big)\,. \tag{3.41b}$$

The overall order of accuracy can be derived by assuming the following form:

$$g_{\text{method}} = e^{\tilde{\lambda}_{\text{method}}\Delta t}, \tag{3.42}$$

where $\tilde{\lambda}_{\text{method}}$ is the eigenvalue containing the information of both spatial and temporal discretizations. This formula assumes that a method has unique exponential form that includes both spatial and temporal discretization errors in the eigenvalue, $\tilde{\lambda}_{\text{method}}$. Note that $\tilde{\lambda}_{\text{1st-order}}$ is somewhat different with $\lambda_{\text{1st-order}}$ in (3.35) since the latter contains only the spatial discretization error. Taking the logarithm of the amplification factor leads to

$$\begin{aligned}\tilde{\lambda}_{\text{1st-order}}\Delta t &= \ln(g_{\text{1st-order}}) \\ &= \ln|g_{\text{1st-order}}| + i\phi_{\text{1st-order}} \\ &= -ir\nu\beta - \frac{1}{2}\left[q\nu - (r\nu)^2\right]\beta^2 + \frac{1}{6}ir\nu\left[1 - 3q\nu + 2(r\nu)^2\right]\beta^3 + O\big(\beta^4\big)\,.\end{aligned} \tag{3.43}$$

The leading error terms are obtained by subtracting the exact eigenvalue (3.21a) from the discretization method,

$$\tilde{\lambda}_{\text{1st-order}}\Delta t - \lambda_{\text{exact}}\Delta t = -\frac{1}{2}\left[q\nu - (r\nu)^2\right]\beta^2 + \frac{1}{6}ir\nu\left[1 - 3q\nu + 2(r\nu)^2\right]\beta^3 + O\big(\beta^4\big)\,. \tag{3.44}$$

To derive the order of accuracy, the frequency, $\beta$, is replaced by the wave number, $k$, given by (3.20). Then, dividing it by the time step, $\Delta t$, leads to the local truncation error (LTE) of the method, thus

$$\begin{aligned}\text{LTE}_{\text{1st-order}} &= \tilde{\lambda}_{\text{1st-order}} - \lambda_{\text{exact}} \\ &= -\frac{1}{2}\left(q\,\boxed{\Delta x} - r^2\,\boxed{\Delta t}\,\right)k^2 + \frac{ir}{6}\left[1 - 3q\nu + 2(r\nu)^2\right]\Delta x^2 k^3 + O\big(k^4\big)\,,\end{aligned} \tag{3.45}$$

here we assume the grid size, $\Delta x$, is fixed to guarantee the correct asymptotic expansion with respect to $k$. The leading error is the $k^2$-term with coefficients $\Delta x$ and $\Delta t$, thus the method is first-order accurate in space and time.

The relation between $\lambda_{\text{1st-order}}$ and $\tilde{\lambda}_{\text{1st-order}}$ becomes clear after taking the series expansion of the eigenvalue (3.35). Following the same procedure, the truncation error of the spatial discretization is given by

$$\lambda_{\text{1st-order}} - \lambda_{\text{exact}} = -\frac{1}{2}q\,\boxed{\Delta x}\,k^2 + \frac{ir\Delta x^2}{6}k^3 + O\big(k^4\big)\,, \tag{3.46}$$

thus the spatial discretization is first-order accurate in space. The identical truncation error is obtained by letting $\Delta t, \nu \to 0$ in (3.45). This example shows that analyzing the eigenvalue of spatial discretization, $\lambda_{\text{method}}$, provides only the order of accuracy in space. Hence, the eigenvalue, $\tilde{\lambda}_{\text{method}}$, defined by (3.42) is necessary to examine the order of accuracy in both space and time.

The order of accuracy can be also obtained by expanding the amplification factor (3.37) directly with respect to $\beta$:

$$g_{\text{1st-order}} = 1 - ir\nu\beta - \frac{1}{2}q\nu\beta^2 + \frac{1}{6}ir\nu\beta^3 + O\big(\beta^4\big)\,. \tag{3.47}$$

Subtracting (3.25) from the above equation leads to the leading error term,

$$g_{\text{1st-order}} - g_{\text{exact}} = -\frac{1}{2}\left[q\nu - (r\nu)^2\right]\beta^2 + \frac{1}{6}ir\nu\left[1 - (r\nu)^2\right]\beta^3 + O\big(\beta^4\big)\,. \tag{3.48}$$

Following the same procedure, replacing $\beta$ by $k$, and dividing by $\Delta t$, leads to the same conclusion. This approach is more straightforward for deriving the order of accuracy in the low-frequency limit; however, the resulting formula does not distinguish whether the higher-order error comes from the dissipation or dispersion any more. This is due to the fact that when (3.42) is expanded,

$$g_{\text{method}} = 1 + \tilde{\lambda}\Delta t + \frac{1}{2}(\tilde{\lambda}\Delta t)^2 + O\Big((\tilde{\lambda}\Delta t)^3\Big)\,, \tag{3.49}$$

and (3.43) is inserted, the $(\tilde{\lambda}\Delta t)^2$ term will produce the $\beta^3$-term which results from the multiplication of $\beta$ (dispersion) and $\beta^2$ (dissipation) terms. Thus, the resulting formula can not distinguish the error coming from two different sources even though it has a similar form of (3.43). Nevertheless, the leading error term, $\beta^2$-term in this case, is always identical in both approaches.

### Stability

The stability of a method can be examined by the modulus of the amplification factor (3.23a). The necessary and sufficient condition for linear stability is

$$|g(\beta,\tilde{\nu})| \leq 1 \quad \text{for any} \quad \beta \in [-\pi,\pi]. \tag{3.50}$$

The stability condition for the particular flux function (3.27) can be easily obtained for the first-order method. The moduli of the amplification factor (3.39a) for the various flux functions are given by

$$|g_{\text{1st, LxF}}| = \sqrt{\cos^2\beta + (r\nu)^2 \sin^2\beta}, \tag{3.51a}$$

$$|g_{\text{1st, upwind}}| = \sqrt{1 - 4r\nu(1 - r\nu)\sin^2\frac{\beta}{2}}, \tag{3.51b}$$

thus the necessary and sufficient condition for linear stability is

$$|r|\nu \leq 1, \tag{3.52}$$

for both flux functions. Here, it is important to distinguish between the stability condition obtained by a Fourier analysis and the Courant–Friedrichs–Lewy (CFL) condition [CFL28, CFL67]. The CFL condition is a necessary condition for the linear or nonlinear stability, but not sufficient. Luckily, a symmetric three-point method supported by $(j-1, j, j+1)$ as described in this example leads to the necessary CFL condition identical to (3.52). Thus, the CFL condition indeed becomes necessary and sufficient for the first-order method. It is also important to notice that the CFL condition itself allows a larger-than-unity Courant number when an explicit method uses a wider stencil. Thus, interpreting the CFL condition as $\tilde{\nu} \leq 1$ for any explicit methods is misleading. In general, for an explicit method utilizing $2m + 1$ cells such that

$$\bar{u}_j^{n+1} = \sum_{k=-m}^{m} c_{j+k}\bar{u}_{j+k}^n, \quad m \geq 1, \tag{3.53}$$

the CFL condition provides the following necessary stability condition:

$$\tilde{\nu} \leq m. \tag{3.54}$$

Consequently, when a higher-order method is considered, a Fourier analysis is necessary to provide the complete linear stability conditions. For instance, the explicit second-order upwind method by Warming and Beam [WB76], [TAP97, p. 119],

$$u_j^{n+1} = u_j^n - \tilde{\nu}(u_j^n - u_{j-1}^n) + \frac{1}{2}\tilde{\nu}(\tilde{\nu} - 1)(u_j^n - 2u_{j-1}^n + u_{j-2}^n), \tag{3.55}$$

using one-sided three-point stencil supported by $(j-2, j-1, j)$, has the CFL condition $0 \leq r\nu \leq 2$. A Fourier analysis also shows that this is the necessary and sufficient for the stability. This is a rare example showing that the weaker CFL condition matches the sufficient condition. Typically, a method using a wider stencil tends to increase the order of accuracy while sacrificing stability. Thus, it has a more restrictive condition on the Courant number provided by a Fourier analysis than the CFL condition. Again, the CFL condition can not provide the complete stability conditions. We also need to keep in mind that when a *compact* explicit high-order method is considered, due to the CFL condition, one can only expect its stability to be given by at most $\tilde{\nu} \leq 1$. More discussion of Courant number and Fourier analysis for an explicit method can be found in [Leo94].

When the stability condition (3.50) is considered for a higher-order method, the modulus of the amplification factor becomes a lengthy expression, and it does not always lead to simple stability conditions. Thus, it is useful to assess the necessary conditions for linear stability by taking the low- and high-frequency limit. The low-frequency limit of the modulus is given by (3.40a). The necessary condition for stability is that the $\beta^2$-term is always negative, thus $q\nu - (r\nu)^2 \geq 0$, or

$$r\nu = \tilde{\nu} \leq \frac{q}{r}. \tag{3.56}$$

In the high-frequency limit, we have

$$|g_{\text{1st-order}}| = |1 - 2q\nu| + O\big((\beta - \pi)^2\big)\,, \tag{3.57}$$

and the necessary condition for stability is $|q\nu| \leq 1$ which is automatically satisfied by the definition of $q$ in (3.28). Thus, overall the necessary condition is given

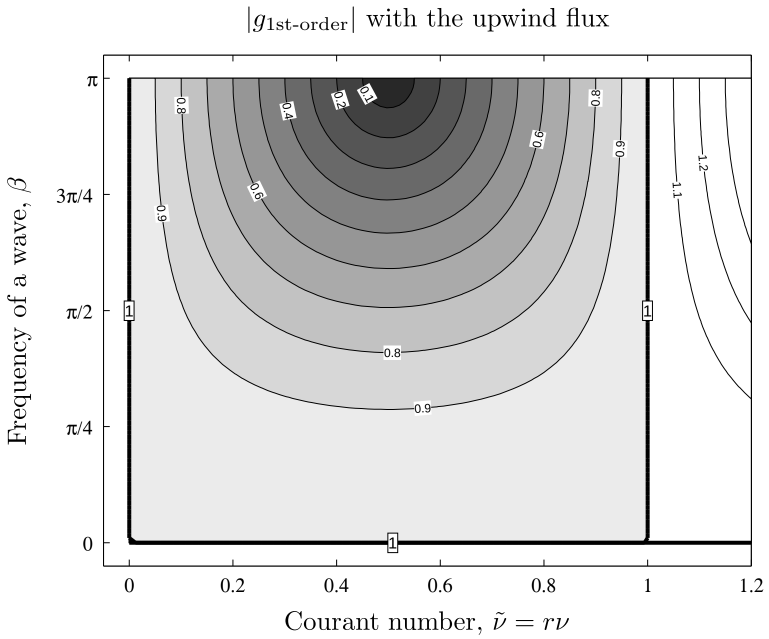

$|g_{\text{1st-order}}|$ with the upwind flux

Figure 3.1: Contour plot of the modulus of the amplification factor, $|g_{\text{1st-order}}(\tilde{\nu}, \beta)|$, computed with the upwind flux. It shows that the first-order method is stable for $\tilde{\nu} \leq 1$.

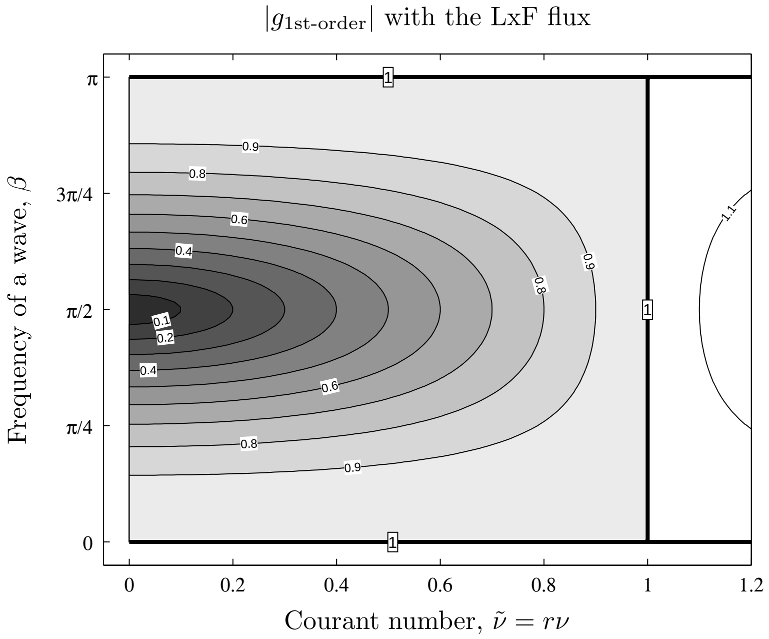

$|g_{\text{1st-order}}|$ with the LxF flux

Figure 3.2: Contour plot of the modulus of the amplification factor, $|g_{\text{1st-order}}(\tilde{\nu}, \beta)|$, computed with the Lax–Friedrichs flux. It shows that the first-order method is stable for $\tilde{\nu} \leq 1$.

by (3.56), and indeed it is sufficient as well for this example. The approach taken here is similar to the heuristic stability analysis based on the modified-equation analysis. The link between the Fourier analysis and the modified-equation analysis is discussed in the next section.

Another approach to obtain a stability condition is purely numerical: plot the modulus of an amplification factor with respect to the Courant number, $\tilde{\nu}$, and the frequency of a wave, $\beta$. Of course, this approach does not provide the rigorous derivation of a stability limit; however, the complex mathematical operations can be eliminated, and the stability domain is identified visually. Even though the complete stability analysis is obtained easily for the first-order method, for future reference, the contour plots of the amplification factor, $|g_{\text{1st-order}}|$, with those for the upwind and Lax–Friedrichs fluxes are shown in Figure 3.1 and 3.2, respectively. The shaded area is where the method is stable, thus $|g_{\text{1st-order}}| \leq 1$, and the stability limit where $|g_{\text{1st-order}}| = 1$ is indicated by a thick line. These two figure clearly show that the 1st-order method with the upwind or Lax–Friedrichs flux inserted is linearly stable for $\tilde{\nu} \leq 1$. Also, it shows that the Lax–Friedrichs flux tends to damp the middle frequencies, whereas the upwind flux rather damps the high frequencies.

### 3.2.4 Fourier Analysis v.s. Modified Equation Analysis

The Fourier analysis and the modified-equation analysis [Hir68, WH74, GSS86] are the standard methods to analyze a linear discretization method. Originally, a Fourier analysis was developed to investigate the properties of a linear method applied to a linear equation; contrarily, the modified equation was for a nonlinear equation where a Fourier analysis can not decompose a wave package down to a single Fourier mode [Hir68]. Even though the assumptions made for a Fourier analysis are more restrictive, these two analyses are interrelated and transferable. When both analyses are applied to a linear equation, a Fourier analysis provides the actual solution of the equation, hence consistency, order of accuracy, and complete conditions for linear stability, while the modified-equation analysis only provides insight into the consistency of a method. It is important to show both stability and consistency of a linear method, since the Lax equivalence theorem (cf. [RM67]) shows that these two conditions are necessary and sufficient for convergence. A Fourier analysis suffices for this purpose. The order of accuracy can be derived by both analyses, even though it is rather common to utilize the modified equation.

Fourier analysis assumes a uniform grid and periodic boundaries, representing a solution in the Fourier domain. It is only applicable to linear equations since a single harmonic can not be isolated in nonlinear equations; a complete Fourier series is required to represent a solution. As shown in the previous section, a Fourier analysis provides the dissipation (amplitude error) and the dispersion (phase error) in the entire frequency domain. An example regarding the first-order method was shown in (3.39).

The modified-equation analysis assumes a uniform grid and ignores the boundary effects. It derives a 'modified' differential (not difference) equation which a numerical method actually solves. The resulting equation has the form of the original PDE with the local truncation error, providing the order of accuracy and consistency of a discretization. The analysis is applicable to nonlinear equations [GM85], however, nonlinear terms in the truncation error make it difficult to predict its effects. Here, the modified equation is derived for the first-order method as an example, starting from (3.30) in the previous section. It will be shown that the truncation error obtained by the modified-equation analysis can be seen as the low-frequency limit of an amplification factor, $g_{\text{1st-order}}$, obtained by a Fourier analysis.

### Example of the Modified Equation Analysis

At first, the solution in neighboring cells is replaced by the following Taylor-series expansions,

$$\bar{u}_j^{n+1} = \bar{u}_j^n + \Delta t \partial_t u + \frac{1}{2}(\Delta t)^2 \partial_{tt} u + \frac{1}{6}(\Delta t)^3 \partial_{ttt} u + O\big((\Delta t)^4\big)\,, \tag{3.58a}$$

$$\bar{u}_{j\pm1}^{n} = \bar{u}_j^n \pm \Delta x \partial_x u + \frac{1}{2}(\Delta x)^2 \partial_{xx} u \pm \frac{1}{6}(\Delta x)^3 \partial_{xxx} u + O\big((\Delta x)^4\big)\,, \tag{3.58b}$$

thus, (3.30) becomes

$$\partial_t u + r\partial_x u = -\frac{\Delta t}{2}\partial_{tt} u + \frac{q\Delta x}{2}\partial_{xx} u - \frac{(\Delta t)^2}{6}\partial_{ttt} u - \frac{r(\Delta x)^2}{6}\partial_{xxx} u + O\big((\Delta x)^3, (\Delta t)^3\big)\,. \tag{3.59}$$

In order to replace all time-derivatives by spatial derivatives, the Cauchy–Kovalevskaya procedure, sometimes called the Lax–Wendroff procedure [Lan98, p. 266] is applied, namely, $\partial_{tt} u$ is replaced by taking $\partial_t(3.59) - r\partial_x(3.59)$, thus

$$\partial_{tt} u = r^2 \partial_{xx} u + \frac{\Delta t}{2}\left(-\partial_{ttt} u + r\partial_{ttx} u\right) + \frac{q\Delta x}{2}\left(\partial_{xxt} u - r\partial_{xxx} u\right) + O\big((\Delta x)^2, (\Delta t)^2\big)\,. \tag{3.60}$$

Inserting the above equation into (3.59) leads to

$$\begin{aligned}\partial_t u + r\partial_x u = \frac{1}{2}\left(q\Delta x - r^2\Delta t\right)\partial_{xx} u + \left(\frac{qr\Delta x\Delta t}{4} - \frac{r\Delta x^2}{6}\right)\partial_{xxx} u - \frac{q\Delta x\Delta t}{4}\partial_{xxt} u \\ - \frac{r\Delta t^2}{4}\partial_{xtt} u + \frac{\Delta t^2}{12}\partial_{ttt} u + O\big((\Delta x)^3, (\Delta x)^2\Delta t, (\Delta t)^3\big)\,,\end{aligned} \tag{3.61}$$

where higher-order time derivatives appear. Repeating the same procedure, these time derivatives are given by

$$\partial_{xxt} u = -r\partial_{xxx} u + O(\Delta x, \Delta t)\,, \tag{3.62a}$$

$$\partial_{xtt} u = \quad r^2\partial_{xxx} u + O(\Delta x, \Delta t)\,, \tag{3.62b}$$

$$\partial_{ttt} u = -r^3\partial_{xxx} u + O(\Delta x, \Delta t)\,, \tag{3.62c}$$

thus the modified equation is now expressed only by spatial derivatives:

$$\begin{aligned}\partial_t u + r\partial_x u &= \frac{1}{2}\left(q\Delta x - r^2\Delta t\right)\partial_{xx}u - \frac{1}{6}\left(r\Delta x^2 - 3qr\Delta x\Delta t + 2r^3\Delta t^2\right)\partial_{xxx}u \\ &\quad + O\big((\Delta x)^3, (\Delta x)^2\Delta t, \Delta x(\Delta t)^2, (\Delta t)^3\big) \\ &= \frac{1}{2}\left(q\,\boxed{\Delta x} - r^2\,\boxed{\Delta t}\,\right)\partial_{xx}u - \frac{r\Delta x^2}{6}\left[1 - 3q\nu + 2(r\nu)^2\right]\partial_{xxx}u + \ldots.\end{aligned} \tag{3.63}$$

It clearly shows that the method is first-order accurate in space and time since the factors $\Delta x$ and $\Delta t$ appear in the leading error (dissipation) term.

The link from the modified equation to a Fourier analysis can be made by inserting a Fourier mode (3.17) on page 92 into the modified equation. This has the effect of replacing the spatial derivatives by multiplication with the wave number, $k$, as follows:

$$\partial_{xx}u = -k^2 u \quad \text{and} \quad \partial_{xxx}u = -ik^3 u. \tag{3.64}$$

Inserting the above relations into (3.63) leads to

$$\partial_t u + r\partial_x u = -\frac{1}{2}\left(q\,\boxed{\Delta x} - r^2\boxed{\Delta t}\,\right)k^2 u + \frac{ir}{6}\left[1 - 3q\nu + 2(r\nu)^2\right]\Delta x^2 k^3 u + \ldots, \tag{3.65}$$

which recovers the result obtained by a Fourier analysis in the low-frequency limit (3.45), multiplied by $u$. Since the modified-equation analysis truncates the higher-order derivatives, which become important for high-frequency mode, it is only valid in the low-frequency limit. In this sense, a Fourier analysis is preferable since it provides the properties of a method in the whole frequency domain.

### Modified Equation → Fourier Analysis

Warming and Hyett have shown that the relative phase error, $\dfrac{\phi_{\text{method}}}{\phi_{\text{exact}}}$, normally obtained by a Fourier analysis can be derived from the modified equation [WH74].

Assume the modified equation has the form:

$$\partial_t u + r\partial_x u = \sum_{n=1}^{\infty}\left(c_{2n}\frac{\partial^{2n}u}{\partial x^{2n}} + c_{2n+1}\frac{\partial^{2n+1}u}{\partial x^{2n+1}}\right), \tag{3.66}$$

then the relative phase error is given by

$$\frac{\phi_{\text{method}}}{\phi_{\text{exact}}} = 1 - \frac{1}{r}\sum_{n=1}^{\infty}(-1)^n\left(\frac{\beta}{\Delta x}\right)^{2n} c_{2n+1}. \tag{3.67}$$

Also, it was shown that a necessary stability condition is given by

$$(-1)^{l-1}c_{2l} > 0, \tag{3.68}$$

where $c_{2l}$ is the coefficient of the lowest even-order derivative term. This is identical to the result of the 'heuristic' stability analysis of Hirt for a nonlinear equation [Hir68].

**Fourier Analysis → Modified Equation**

So far, it was shown that the asymptotic result obtained by a Fourier analysis can be recovered by the modified-equation analysis. Carpenter et al. have shown the opposite way; they derived a systematic procedure to obtain the modified equation from an amplification factor given by a Fourier analysis [CdLBL97]. In summary, they defined the $\tilde{\lambda}$ which corresponds to our previous definition in (3.42) such that

$$\tilde{\lambda} = \frac{1}{\Delta t}\ln\bigl(g(\beta)\bigr), \tag{3.69}$$

then take the formal series expansion with respect to $\beta$,

$$\tilde{\lambda} = \sum_{n=1}^{\infty}\tilde{b}_n\beta^n. \tag{3.70}$$

Once the coefficients, $\tilde{b}_n$, for all $n$ are known, the modified equation is given by the following form:

$$\partial_t u = \sum_{n=1}^{\infty}\tilde{c}_n\frac{\partial^n u}{\partial x^n}, \tag{3.71}$$

where the coefficient, $\tilde{c}_n$, is

$$\tilde{c}_n = \left(\frac{\Delta x}{i}\right)^n \tilde{b}_n. \tag{3.72}$$

The related work by Ramshaw provides a more direct way to obtained the modified equation [Ram94]. The coefficient, $\tilde{c}_n$, is obtained by the $n$-th derivative of the eigenvalue evaluated at $k = 0$:

$$\tilde{c}_n = \frac{1}{i^n n!} \frac{\partial^n \tilde{\lambda}(k)}{\partial k^n}\bigg|_{k=0}. \tag{3.73}$$

This method shows the direct link between a Fourier analysis and the modified-equation analysis.

### Which One Is More Suitable?

The properties of both analyses are summarized in Table 3.2. One of the deficiencies of the modified-equation analysis over a Fourier analysis is that the analysis truncates the higher-order derivatives which represent the high-frequency modes of the solution. Thus, the analysis does not provide the properties of a method in whole frequency domain. Another deficiency is that the method only reveals the truncation error of the principal root [Cha90]. This is critical especially when a DG method or a multi-step method are adopted for the discretization, since the characteristic equation of these methods contains both principal and extraneous roots. Typically, a spurious solution coming from an extraneous root is damped quickly, thus it does not affect the accuracy. However, for stability, all eigenvalues of the amplification matrix have to satisfy the stability condition (3.50). Since the modified-equation analysis does not provide the extraneous roots, we can not confirm if the spurious solution is actually damped, (i.e., the eigenvalue has a negative real part), or not. Consequently, the complete stability analysis can not be performed.

| | Fourier analysis | modified equation analysis |
|---|---|---|
| assumptions | uniform grids<br>periodic boundary | uniform grids<br>no boundary effect |
| strength | valid for all frequencies<br>provides all wave modes | valid for nonlinear equations<br>provides consistency |
| weakness | valid only for linear equations | truncated results<br>provides only the principal root |

Table 3.2: The properties of Fourier and modified-equation analyses are summarized.

Conversely, a Fourier analysis provides both principal and extraneous roots, allowing to assess the complete stability conditions even though the analysis is only valid for linear equations. Thus, we will conduct a Fourier analysis instead of the modified-equation analysis to investigate the dominant truncation errors, the order of accuracy, and the stability conditions of various discretization methods.

### 3.2.5 Methodology of Analysis

The procedure of a Fourier analysis is summarized here in concise formulas.

**Accuracy**

1. Express the difference equation in the form of a linear combination of neighboring cell values. Then, using the difference operator (3.5), rewrite it in the compact operator form. For a fully discrete explicit method:

$$\frac{\mathbf{u}_j^{n+1} - \mathbf{u}_j^n}{\Delta t} = \sum_i a_i \mathbf{u}_i^n = \mathbf{M}(\delta^+, \delta^-)\mathbf{u}_j^n, \tag{3.74a}$$

and for a semi-discrete method:

$$\frac{\partial \mathbf{u}(t)}{\partial t} = \sum_i a_i \mathbf{u}_i(t) = \mathbf{N}(\delta^+, \delta^-)\mathbf{u}_j(t). \tag{3.74b}$$

2. Once a compact form is obtained, replace difference operators for a Fourier mode by (3.16). The eigenvalues of a spatial discretization operator are given by solving the following characteristic equations:

$$\text{fully discrete: } \det\left[\mathbf{M}(\beta) - \lambda(\beta)\mathbf{I}\right] = 0, \tag{3.75a}$$

$$\text{semi-discrete: } \det\left[\mathbf{N}(\beta) - \lambda(\beta)\mathbf{I}\right] = 0. \tag{3.75b}$$

   The order of the spatial accuracy can be found by identifying the lowest-order term of the series expansion of an accurate eigenvalue, given by

$$\lambda(\beta) = \sum_{n=1}^{\infty} b_n \beta^n. \tag{3.76}$$

3. To examine the order of accuracy in both space and time, a fully discrete method is combined with the forward Euler method, and a semi-discrete method is incorporated in an RK$s$ method given by (3.11). The amplification matrices (or factors) are given by

$$\text{fully discrete: } \mathbf{G}_{\mathrm{M}}(\beta) = \mathbf{I} + \Delta t \mathbf{M}(\beta), \tag{3.77a}$$

$$\text{semi-discrete: } \mathbf{G}_{\mathrm{N}}(\beta) = \mathrm{RK}s\left(\mathbf{N}(\beta)\right). \tag{3.77b}$$

   When $\mathbf{G}$ is a matrix, amplification factors, $g(\beta)$, are obtained by solving the following characteristic equation:

$$\det\left[\,\mathbf{G}(\beta) - g(\beta)\mathbf{I}\,\right] = 0. \tag{3.78}$$

   Once an amplification factor is obtained, it can be written in the polar form to decouple the dissipation and dispersion errors:

$$g = |g|e^{i\phi}, \tag{3.79}$$

where

$$|g(\beta)| = \sum_{n=0}^{\infty} b_{2n}\beta^{2n}, \tag{3.80a}$$

$$\phi(\beta) = \sum_{n=0}^{\infty} b_{2n+1}\beta^{2n+1}. \tag{3.80b}$$

Since the formulas of $|g|$ and $\phi$ are lengthly for a higher-order method, they will be presented only in the form of a power series with respect to the frequency of a wave.

4. Once a modulus, $|g|$, and phase angle, $\phi$, are known, the local truncation error is obtained by the following formula:

$$\begin{aligned}
\mathrm{LTE}_{\text{method}} &= \tilde{\lambda}_{\text{method}} - \lambda_{\text{exact}} \\
&= \frac{1}{\Delta t}\ln g - \lambda_{\text{exact}} \\
&= \frac{1}{\Delta t}\left(\ln|g| + i\phi\right) - \lambda_{\text{exact}} \\
&= \frac{1}{\Delta t}\left[ib_1\beta + b_2\beta^2 + ib_3\beta^3 - \frac{1}{2}(b_2^2 - 2b_4)\beta^4 + \dots\right] - \lambda_{\text{exact}}. \qquad (3.81)
\end{aligned}$$

In order to correspond to the result from the modified-equation analysis, wave frequency, $\beta$, is replaced by $\beta = k\Delta x$, thus

$$\begin{aligned}
\mathrm{LTE}_{\text{method}} = \frac{1}{\nu}&\left[ib_1 k + b_2\Delta x k^2 + ib_3\Delta x^2 k^3 + \left(b_4 - \frac{1}{2}b_2^2\right)\Delta x^3 k^4 + \dots\right] \\
&- (-irk). \qquad (3.82)
\end{aligned}$$

Here, we implicitly assume that the grid size, $\Delta x$, is fixed. As long as a method is consistent with the original PDE, it satisfies $b_1 \equiv -r\nu$, thus the above equations is further simplified:

$$\begin{aligned}
\mathrm{LTE}_{\text{method}} &= \frac{1}{\nu}\left[b_2\Delta x k^2 + ib_3\Delta x^2 k^3 + \left(b_4 - \frac{1}{2}b_2\right)\Delta x^3 k^4 + \dots\right] \\
&= \sum_{n=2}^{\infty} c_n k^n, \qquad (3.83)
\end{aligned}$$

where

$$c_2 = \frac{\Delta x}{\nu} b_2, \quad c_3 = i\frac{\Delta x^2}{\nu} b_3, \quad \text{and} \quad c_4 = \frac{\Delta x^3}{\nu}\left(b_4 - \frac{1}{2}b_2\right). \tag{3.84}$$

The overall order of accuracy can be found by the coefficient of the lowest order, $c_n$. For instance, if $c_n = O(\Delta x, \Delta t)$, then a method is first-order in space and time.

An alternative way to obtain the coefficient, $c_n$, is directly computing it from an accurate amplification factor, $g(k)$. Hence, the $c_n$ is given by a Taylor series expansion around $k = 0$:

$$c_n = \frac{1}{n!}\frac{\partial^n \tilde{\lambda}(k)}{\partial k^n}\bigg|_{k=0} = \frac{1}{n!\Delta t g(k=0)}\frac{\partial^n g(k)}{\partial k^n}\bigg|_{k=0}. \tag{3.85}$$

**Stability**

Ideally, the stability criterion is directly obtained from the modulus of an amplification factor, $|g_{\text{method}}|$. However, a high-order method tends to have a lengthly expression for this modulus, and in most of the cases, extracting a stability condition from a modulus is not feasible. The heuristic stability analysis provides a necessary stability condition, (3.68), yet not sufficient. An alternative way to identify a stability condition for a high-order method is based on the numerical contour plot of the modulus, $|g_{\text{method}}(\tilde{\nu}, \beta)|$. Using a contour plot, we find the maximum allowable Courant number, $\tilde{\nu}_{\max}$, such that $|g_{\text{method}}(\tilde{\nu}_{\max}, \beta)| \leq 1$ for any $\beta \in [0, \pi]$.

## 3.3 Difference Operators and Their Properties in 1-D

In this section, a Fourier analysis is employed to uncover the dominant truncation errors in the low-frequency limit, the order of accuracy, and the stability conditions of various discretization methods.

### 3.3.1 HR–MOL Method

A semi-discrete high-resolution Godunov method combined with the method-of-lines (HR–MOL) for the 1-D linear advection equation (3.1a) has the following form:

$$\frac{\partial \bar{u}_j(t)}{\partial t} = -\frac{1}{\Delta x}\left(\hat{f}_{j+1/2}(t) - \hat{f}_{j-1/2}(t)\right), \tag{3.86}$$

where the linear flux, $f(u(t)) = ru(t)$, is evaluated at each cell interface. The interface flux $\hat{f}_{j\pm1/2}(t)$ must be evaluated at various time levels, depending on the time discretization method (ODE solver) chosen. The interface flux is obtained as in the $q$-flux (3.27). For the linear advection equation, it becomes

$$\hat{f}_{j+1/2}(t) = \frac{r}{2}\big(u_{j+1/2,L}(t)+u_{j+1/2,R}(t)\big)-\frac{q}{2}\big(u_{j+1/2,R}(t)-u_{j+1/2,L}(t)\big), \quad q>0, \tag{3.87}$$

where the input values of the flux function are the linearly reconstructed values at the cell interface $j+1/2$:

$$u_{j+1/2,L}(t) = \bar{u}_j + \frac{\Delta x}{2}\left(\frac{\overline{\Delta u}_j}{\Delta x}\right), \tag{3.88a}$$

$$u_{j+1/2,R}(t) = \bar{u}_{j+1} - \frac{\Delta x}{2}\left(\frac{\overline{\Delta u}_{j+1}}{\Delta x}\right). \tag{3.88b}$$

The slopes, $\frac{\overline{\Delta u}_{j,j+1}}{\Delta x}$, approximate the derivative, $\frac{\partial u}{\partial x}$, and are given by the average of the differences with the neighboring cells:

$$\frac{\overline{\Delta u}_j}{\Delta x} = \frac{1}{2}\left(\frac{\bar{u}_{j+1}-\bar{u}_j}{\Delta x} + \frac{\bar{u}_j-\bar{u}_{j-1}}{\Delta x}\right) = \frac{\bar{u}_{j+1}-\bar{u}_{j-1}}{2\Delta x}, \tag{3.89a}$$

$$\frac{\overline{\Delta u}_{j+1}}{\Delta x} = \frac{\bar{u}_{j+2}-\bar{u}_j}{2\Delta x}. \tag{3.89b}$$

The linear reconstruction from the original piecewise constant values leads an HR–MOL method to second-order accuracy in space whereas the original Godunov method, which uses the piecewise constant data, is first-order accurate. Here, we denote the second-order semi-discrete Godunov method as HR2–MOL.

After inserting the cell interface fluxes into the original semi-discrete form (3.86), and some algebra, the spatial difference operator, $\mathrm{N}_{\mathrm{HR2}}$, in the semi-discrete form of (3.10) is given by

$$\mathrm{N}_{\mathrm{HR2}} = -\frac{1}{8\Delta x}\left[(q-r)(\delta^{+})^{2} - 2(q-2r)\delta^{+} + 2(q+2r)\delta^{-} + (q+r)(\delta^{-})^{2}\right], \tag{3.90a}$$

or, for a Fourier mode, using (3.16),

$$\mathrm{N}_{\mathrm{HR2}} = -\frac{1}{8\Delta x}\left[(q-r)e^{2i\beta} + 2(3r-2q)e^{i\beta} + 6q - 2(2q+3r)e^{-i\beta} + (q+r)e^{-2i\beta}\right]. \tag{3.90b}$$

The identical result can be obtained by setting $\nu = 0$ in the spatial-temporal operator for the Hancock method (3.116), derived further below. To check whether the method satisfies the shift condition described on page 96, take the upwind flux, $q = r$, with the unity Courant number, and multiply by the time step, $\Delta t$, then

$$\begin{aligned}\Delta t\ \mathrm{N}_{\mathrm{HR2,\ upwind}} &= -\frac{r\Delta t}{4\Delta x}\left[\delta^{+} + 3\delta^{-} + (\delta^{-})^{2}\right] \\ &= -\frac{1}{4}\left[\delta^{+} + 3\delta^{-} + (\delta^{-})^{2}\right].\end{aligned} \tag{3.91}$$

Since the spatial difference operator contains the forward difference operator, $\delta^{+}$, no RK methods, which yield polynomials of $\Delta t\ \mathrm{N}_{\mathrm{HR2}}$, can produce the exact shift operator, $1-\delta^{-}$. Thus, the HR2–MOL method does not satisfy the shift condition.

### Accuracy

Taking the low-frequency limit of the spatial difference operator, $\mathrm{N}_{\mathrm{HR2}}$, leads to the asymptotic eigenvalue:

$$\lambda_{\mathrm{HR2}} = -\frac{ir}{\Delta x}\beta - \frac{ir}{12\Delta x}\beta^{3} - \frac{q}{8\Delta x}\beta^{4} + O(\beta^{5}). \tag{3.92}$$

The order of accuracy in space is obtained by replacing $\beta$ by the wave number $k$, then

$$\lambda_{\mathrm{HR2}} - \lambda_{\mathrm{exact}} = -\frac{ir}{12}\,\boxed{\Delta x^2}\,k^3 - \frac{q}{8}\Delta x^3\,k^4 + O\big(k^5\big)\,; \tag{3.93}$$

the method appears to be second-order accurate in space.

To examine the overall order of accuracy, the RK2 and RK3 methods (3.11) are employed for the time integration. The amplification factors, $g_{\mathrm{HR2RK2}}$ and $g_{\mathrm{HR2RK3}}$, are expressed in the polar form where the modulus and the phase angle of the HR2–RK2 method are given by

$$|g_{\mathrm{HR2RK2}}| = 1 - \frac{r\nu}{8}\left[\frac{q}{r} - (r\nu)^3\right]\beta^4 + O\big(\beta^6\big)\,, \tag{3.94a}$$

$$\phi_{\mathrm{HR2RK2}} = -r\nu\beta - \frac{r\nu}{12}\left[1 + 2(r\nu)^2\right]\beta^3 + O\big(\beta^5\big)\,, \tag{3.94b}$$

and for the HR2–RK3 method,

$$|g_{\mathrm{HR2RK3}}| = 1 - \frac{r\nu}{8}\left[\frac{q}{r} + \frac{1}{3}(r\nu)^3\right]\beta^4 + O\big(\beta^6\big)\,, \tag{3.95a}$$

$$\phi_{\mathrm{HR2RK3}} = -r\nu\beta - \frac{r\nu}{12}\beta^3 + O\big(\beta^5\big)\,. \tag{3.95b}$$

Following the same procedure, the local truncation errors have the following forms:

$$\begin{aligned}
\mathrm{LTE}_{\mathrm{HR2RK2}} &= \tilde{\lambda}_{\mathrm{HR2RK2}} - \lambda_{\mathrm{exact}} \\
&= \frac{1}{\Delta t}\left(\ln|g_{\mathrm{HR2RK2}}| + i\phi_{\mathrm{HR2RK2}}\right) - \lambda_{\mathrm{exact}} \\
&= -\frac{ir}{12}\left[1 + 2(r\nu)^2\right]\Delta x^2\,k^3 - \frac{r}{8}\left[\frac{q}{r} - (r\nu)^3\right]\Delta x^3 k^4 + O\big(k^5\big) \\
&= -\frac{ir}{12}\left(\boxed{\Delta x^2} + 2r^2\,\boxed{\Delta t^2}\right)k^3 + O\big(k^4\big)\,, && \text{(3.96a)} \\
\mathrm{LTE}_{\mathrm{HR2RK3}} &= \tilde{\lambda}_{\mathrm{HR2RK3}} - \lambda_{\mathrm{exact}} \\
&= -\frac{ir}{12}\,\boxed{\Delta x^2}\,k^3 - \frac{r}{8}\left(\frac{q}{r}\Delta x^3 + \frac{1}{3}r^3\,\boxed{\Delta t^3}\right)k^4 + O\big(k^5\big)\,. && \text{(3.96b)}
\end{aligned}$$

Thus, the HR2–RK2 method is second-order accurate in space and time, and the HR2–RK3 is second-order in space and third-order in time. It clearly shows that the third-order time-integration method (RK3) eliminates the $\Delta t^2$–term of the HR2–RK2 method, making the method third-order accurate in time. However, since a semi-discrete method decouples the space and time discretizations, a higher-order time integration can not eliminate the second-order spatial discretization error. Thus, the HR2–RK3 method is still second-order in space.

### Stability

As mentioned in the example of the first-order method, obtaining the analytical stability condition for a high-order method is not straightforward or sometimes not possible. Thus, we adopt a numerical approach to investigate the linear stability condition. The modulus of the amplification factor, $|g_{\mathrm{HR2RK2}}(\tilde{\nu}, \beta)|$, evaluated with the upwind and Lax–Friedrichs fluxes, is shown in Figures 3.3 and 3.4 respectively. The shaded area indicates the stability region, where $|g_{\mathrm{HR2RK2}}(\tilde{\nu}, \beta)| \leq 1$. These two figures show that the HR2–RK2 method is linearly stable for $\tilde{\nu} \leq 1$ with both the upwind and the Lax–Friedrichs fluxes.

Compared to the first-order method, Figure 3.1 and 3.2 on page 103, the HR2–RK2 method increases the stability region beyond unity around middle to high frequencies. However, the method lost the shift condition: $|g_{\mathrm{HR2RK2}}(1, \beta)| \neq 1$ for some $\beta$.

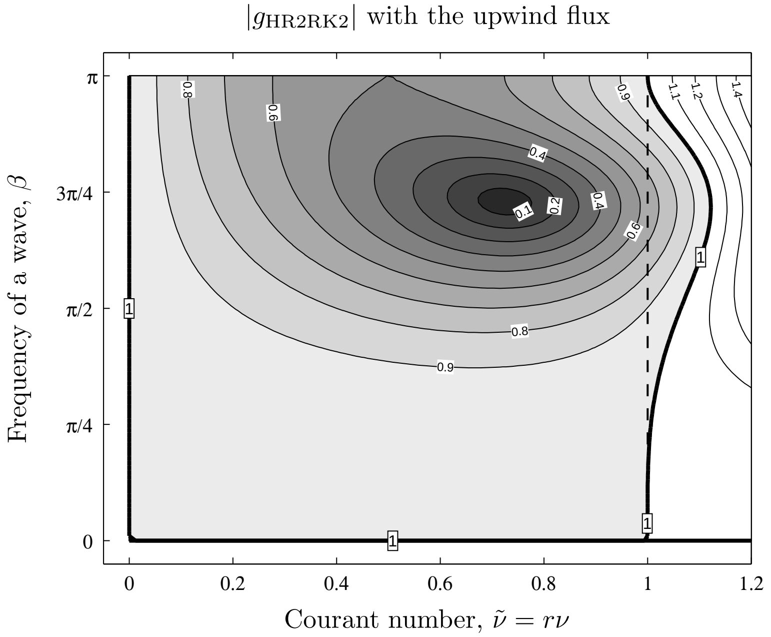

$|g_{\mathrm{HR2RK2}}|$ with the upwind flux

Figure 3.3: Contour plot of the modulus of the amplification factor, $|g_{\mathrm{HR2RK2}}(\tilde{\nu}, \beta)|$, computed with the upwind flux. It shows that the HR2–RK2 method is stable for $\tilde{\nu} \leq 1$.

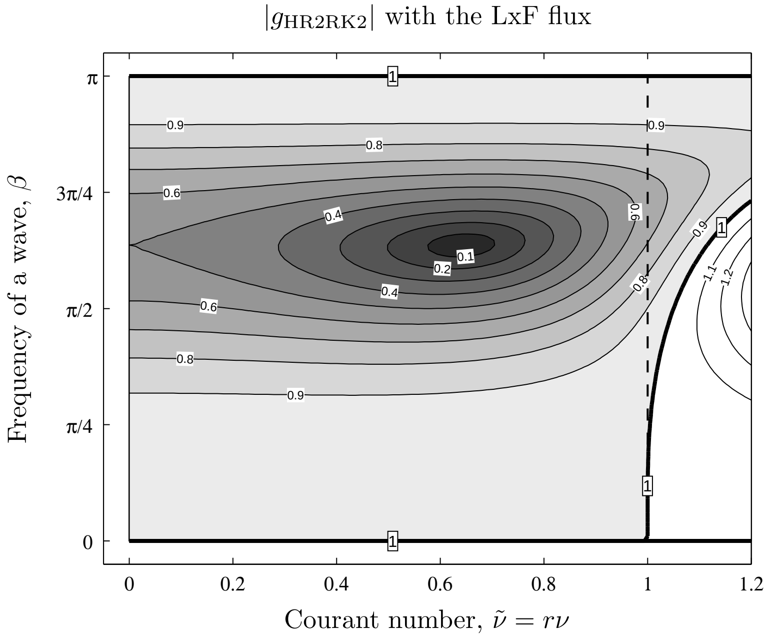

$|g_{\mathrm{HR2RK2}}|$ with the LxF flux

Figure 3.4: Contour plot of the modulus of the amplification factor, $|g_{\mathrm{HR2RK2}}(\tilde{\nu}, \beta)|$, computed with the Lax–Friedrichs flux. It shows that the HR2–RK2 method is stable for $\tilde{\nu} \leq 1$.

### 3.3.2 DG–MOL Method

A semi-discrete DG method combined with the method-of-lines (DG–MOL) for the 1-D linear advection equation (3.1a) has update formulas for both the cell-average and undivided gradient,

$$\frac{\partial \bar{u}_j(t)}{\partial t} = -\frac{1}{\Delta x}\left(\hat{f}_{j+1/2}(t) - \hat{f}_{j-1/2}(t)\right), \tag{3.97a}$$

$$\frac{\partial \overline{\Delta u}_j(t)}{\partial t} = -\frac{1}{\Delta x}6\left(\hat{f}_{j+1/2}(t) + \hat{f}_{j-1/2}(t) - 2r\bar{u}_j(t)\right), \tag{3.97b}$$

where the volume integral of the flux in the second equation simplifies owing to the linearity. The $q$-flux (3.87) is adopted for the cell interface fluxes with linearly interpolated values:

$$u_{j+1/2,L}(t) = \bar{u}_j + \frac{1}{2}\overline{\Delta u}_j, \tag{3.98a}$$

$$u_{j+1/2,R}(t) = \bar{u}_{j+1} - \frac{1}{2}\overline{\Delta u}_{j+1}. \tag{3.98b}$$

Note that a DG method does not need to approximate the slope $\overline{\Delta u}_j(t)$ by using data from the neighboring cells since the slope is also stored as a variable in each cell. After inserting the difference form of fluxes, and some algebra, the spatial difference operator has the following form:

$$\mathbf{N}_{\mathrm{DG}(1)} = \boldsymbol{\mathcal{A}}^{+}\mathbf{D}^{+} + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{A}}^{-}\mathbf{D}^{-}, \tag{3.99}$$

where

$$\boldsymbol{\mathcal{A}}^{+} = \frac{q-r}{2\Delta x}\begin{pmatrix} 1 & -\frac{1}{2} \\ 6 & -3 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}^{-} = \frac{q+r}{2\Delta x}\begin{pmatrix} -1 & -\frac{1}{2} \\ 6 & 3 \end{pmatrix}, \tag{3.100a}$$

$$\boldsymbol{\mathcal{C}} = \frac{r}{\Delta x}\begin{pmatrix} 0 & 0 \\ 0 & -\frac{6q}{r} \end{pmatrix}, \quad \mathbf{D}^{\pm} = \delta^{\pm}\mathbf{I}, \tag{3.100b}$$

or, for a Fourier mode,

$$\mathbf{N}_{\mathrm{DG}(1)} = \boldsymbol{\mathcal{A}}^{+} e^{i\beta\mathbf{I}} + \boldsymbol{\mathcal{C}}' - \boldsymbol{\mathcal{A}}^{-} e^{-i\beta\mathbf{I}}, \tag{3.101}$$

where

$$\boldsymbol{\mathcal{C}}' = -\frac{r}{\Delta x}\begin{pmatrix} \dfrac{q}{r} & \dfrac{1}{2} \\ -6 & \dfrac{3q}{r} \end{pmatrix}. \tag{3.102}$$

Here, the notation "DG(1)" stands for a method representing the solution as piecewise polynomial of degree 1. The identical result can be obtained by setting $\nu = 0$ in the spatial-temporal operator for a DG(1)–Hancock method (3.129)–(3.132). In order to obtain the eigenvalues of the spatial operator, the characteristic equation of $\mathbf{N}_{\mathrm{DG}(1)}$, given by

$$\begin{aligned}(\Delta x\lambda)^2 + \left[(q-r)\delta^+ - (q+r)\delta^- + 6q\right]\Delta x\lambda& \\ + 3q\left[(q-r)\delta^+ - (q+r)\delta^-\right] - 3(q^2-r^2)\delta^+\delta^- &= 0,\end{aligned} \tag{3.103}$$

is solved for $\lambda$. Because of the lengthy expression of the roots in the general form, we present only the result for the upwind flux, $q = r$, as an example:

$$\lambda^{(1),(2)}_{\mathrm{DG}(1),\text{ upwind}} = \frac{r}{\Delta x}\left(-3 + \delta^- \pm \sqrt{9 - 12\delta^- + (\delta^-)^2}\right), \tag{3.104a}$$

or, for a Fourier mode,

$$\lambda^{(1),(2)}_{\mathrm{DG}(1),\text{ upwind}} = \frac{r}{\Delta x}\left(-2 - e^{-i\beta} \pm \sqrt{-2 + 10e^{-i\beta} + e^{-2i\beta}}\right). \tag{3.104b}$$

It shows that the DG(1)–MOL does not satisfy the shift condition.

### Accuracy

The asymptotic eigenvalues in the low-frequency limit are given by

$$\lambda^{(1)}_{\mathrm{DG}(1)} = -\frac{ir}{\Delta x}\beta - \frac{r^2}{72q\Delta x}\beta^4 + O\left(\beta^5\right), \tag{3.105a}$$

$$\lambda^{(2)}_{\mathrm{DG}(1)} = -\frac{6q}{\Delta x} + \frac{3ir}{\Delta x}\beta + O\left(\beta^2\right), \tag{3.105b}$$

where the former is the principal root and the latter is the extraneous one. Since a temporal discretization has not been considered yet, all errors appearing here are attributed solely to the spatial discretization. The order of accuracy in space is obtained by replacing $\beta$ by the wave number $k$, then

$$\lambda^{(1)}_{\mathrm{DG}(1)} - \lambda_{\mathrm{exact}} = -\frac{r^2}{72q}\boxed{\Delta x^3}\, k^4 + O\bigl(k^5\bigr)\,, \tag{3.106a}$$

$$\lambda^{(2)}_{\mathrm{DG}(1)} - \lambda_{\mathrm{exact}} = -\frac{6q}{\Delta x} + O(k)\,, \tag{3.106b}$$

thus the principal root is third-order accurate in space, and the extraneous root is zeroth-order. Fortunately, the extraneous root damps quickly since the leading error, $-\dfrac{6q}{\Delta x}$, is a large negative real value.

To examine the overall accuracy, time integration methods RK2 and RK3 given by (3.11) are employed. The amplification factors of the principal root are expressed in the polar form where the modulus and the phase angle of the DG(1)–RK2 method are given by

$$|g^{(1)}_{\mathrm{DG(1)RK2}}| = 1 - \frac{r\nu}{72}\left[\frac{r}{q} - 9(r\nu)^3\right]\beta^4 + O\bigl(\beta^6\bigr)\,, \tag{3.107a}$$

$$\phi^{(1)}_{\mathrm{DG(1)RK2}} = -r\nu\beta - \frac{1}{6}(r\nu)^3 + O\bigl(\beta^5\bigr)\,, \tag{3.107b}$$

and for the DG(1)–RK3 method,

$$|g^{(2)}_{\mathrm{DG(1)RK3}}| = 1 - \frac{r\nu}{72}\left[\frac{r}{q} + 3(r\nu)^3\right]\,, \tag{3.108a}$$

$$\phi^{(2)}_{\mathrm{DG(1)RK3}} = -r\nu\beta + O\bigl(\beta^5\bigr)\,. \tag{3.108b}$$

Thus, the local truncation errors become

$$\begin{aligned}
\mathrm{LTE}_{\mathrm{DG(1)RK2}} &= \tilde{\lambda}_{\mathrm{DG(1)RK2}} - \lambda_{\mathrm{exact}} \\
&= \frac{1}{\Delta t}\left(\ln|g^{(1)}_{\mathrm{DG(1)RK2}}| + i\phi^{(1)}_{\mathrm{DG(1)RK2}}\right)\lambda_{\mathrm{exact}} \\
&= -\frac{ir}{6}(r\nu)^2\Delta x^2 k^3 - \frac{r}{72}\left(\frac{r}{q} - 9(r\nu)^3\right)\Delta x^3 k^4 \\
&= -\frac{ir^3}{6}\boxed{\Delta t^2}\, k^3 - \frac{r}{72}\left(\frac{r}{q}\boxed{\Delta x^3} - 9r^3\Delta t^3\right) k^4 + O\big(k^5\big)\,. \qquad (3.109a)\\
\mathrm{LTE}_{\mathrm{DG(1)RK3}} &= \tilde{\lambda}_{\mathrm{DG(1)RK3}} - \lambda_{\mathrm{exact}} \\
&= -\frac{r}{72}\left(\frac{r}{q}\boxed{\Delta x^3} + 3r^3\boxed{\Delta t^3}\right) k^4 + O\big(k^5\big)\,. \qquad (3.109b)
\end{aligned}$$

The first equation shows that the temporal discretization RK2 introduces a second-order error in the DG(1)–RK2 method, hence the DG(1)–RK2 method is third-order in space, yet second-order in time. Since the RK3 method is third-order accurate, the DG(1)–RK3 method is third-order in both space and in time.

**Stability**

The stability domain of the DG(1)–RK2 method was first presented by Cockburn and Shu [CS91], and they referred to the stability proof for a simpler case by Chavent and Cockburn [CC87, CC89]. Here, we present the stability limit by plotting the modulus of the two amplification factors independently. The modulus of the accurate and inaccurate amplification factors, $|g^{(1),(2)}_{\mathrm{DG(1)RK2}}|$, computed with the upwind flux are shown in Figures 3.5(a) and 3.5(b) respectively. The figures show that the accurate amplification factor possesses a larger stability domain ($\tilde{\nu}_{\max} = 0.468$) than the inaccurate amplification factor ($\tilde{\nu}_{\max} = 1/3$). Overall, the stability is constrained by the inaccurate amplification factor, so DG(1)–RK2 with the upwind flux is stable for $\tilde{\nu} \leq \dfrac{1}{3}$.

Counterintuitive results are shown in Figures 3.6(a) and 3.6(b), where the Lax–

Friedrichs flux is employed. The contour plots of the modulus show that neither the accurate nor inaccurate amplification factor is stable for any Courant number, even when $\tilde{\nu} = 0$. Thus, the DG(1)–RK2 with the Lax–Friedrichs flux is unconditionally unstable. This was originally found by Rider and Lowrie [RL02]. The same result is obtained for the DG(1)–RK3 method. This is somewhat surprising since the Lax–Friedrichs flux adopts the largest possible dissipation coefficient, $\dfrac{\Delta x}{\Delta t}$, among all $q$-fluxes to stabilize the method.

A reason for the destabilizing result produced by this most dissipative flux function can be found by comparing the dominant numerical dissipation in (3.109) to that in (3.96). For a DG method, the dissipation parameter $q$ appears in the denominator, whereas an HR method contains it in the numerator. Thus, for a DG method, as the numerical dissipation in the flux increases, the method actually becomes less dissipative, at least for low frequencies. This is completely opposite to the behavior of an HR method. Hence, the most dissipative flux leads to the least low-frequency dissipation, resulting in an unconditionally unstable DG method. More specifically, the instability originates in the extraneous root, $\lambda^{(2)}_{\mathrm{DG}(1)}$. The leading error in the extraneous root, multiplied by the time step $\Delta t$ (this product appears when a time integration method is applied), evaluated with the upwind and Lax–Friedrichs fluxes, reads:

$$\text{Lax–Friedreich}: \Delta t\, \lambda^{(2),\mathrm{LxF}}_{\mathrm{DG}(1)} = -\,\Delta t \frac{6q}{\Delta x} = -6, \tag{3.110a}$$

$$\text{upwind}: \Delta t\, \lambda^{(2),\mathrm{upwind}}_{\mathrm{DG}(1)} = -\,\Delta t \frac{6r}{\Delta x} = -6\tilde{\nu}. \tag{3.110b}$$

Assume, for instance, that the RK2 method is used for time integration, then the above eigenvalues should satisfy a necessary condition, $\Re[\Delta t\, \lambda] \in [-2, 0]$, for stability in the low-frequency limit. The second equation, for the upwind flux, satisfies this stability condition as long as $\tilde{\nu} \le \frac{1}{3}$. Conversely, the first equation never satisfies the stability condition, no matter how small the time step is; thus, DG(1) together with the Lax–Friedrichs flux is unconditionally unstable.

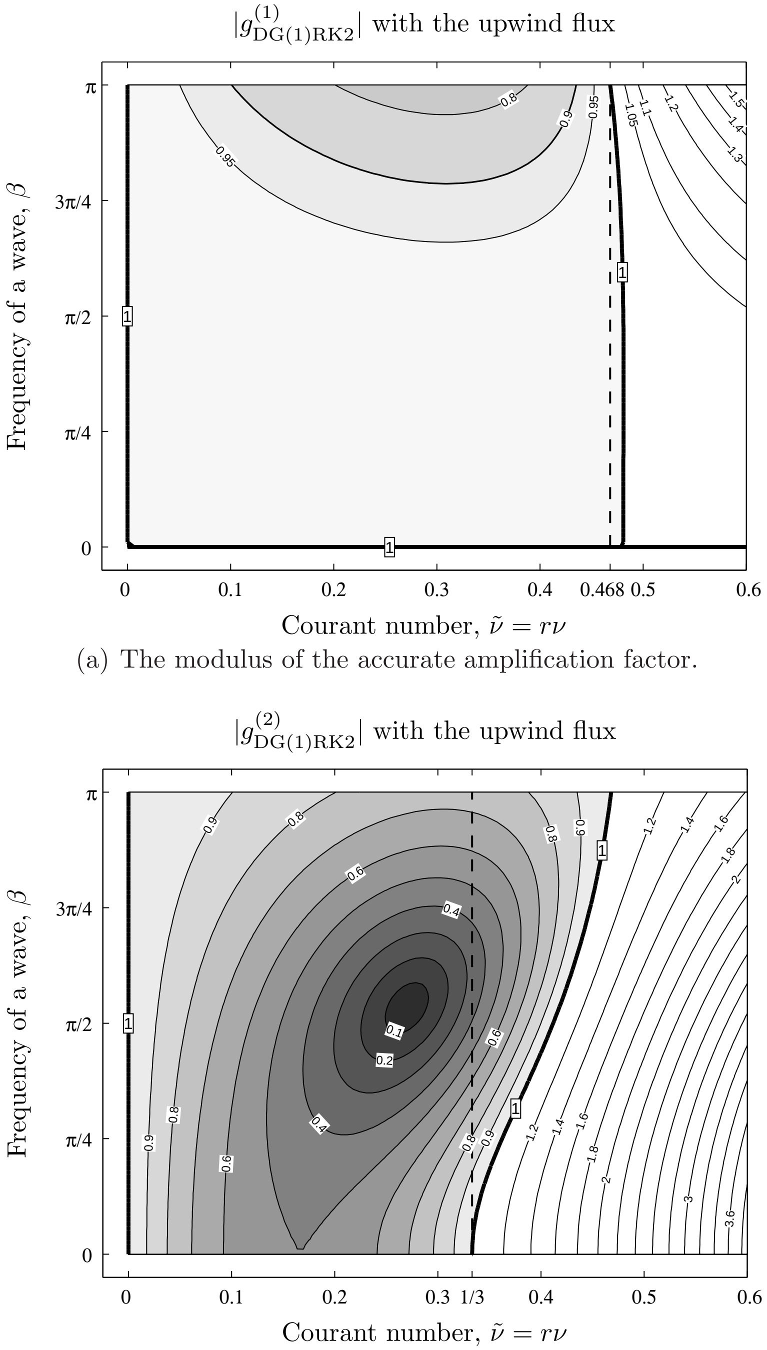
To remedy the instability of the Lax–Friedrichs flux, Rider and Lowrie propose the following modified Lax–Friedrichs flux [RL02]:

$$f_{j+1/2}^{\text{mLxF}}(u_L, u_R) = \frac{r}{2}(u_L + u_R) - \frac{z}{2}\frac{\Delta x}{\Delta x}(u_R - u_L), \tag{3.111}$$

where $z = \frac{1}{3}$ for DG(1), and $z = \frac{1}{5}$ for DG(2). These constants are chosen such that the maximum stable Courant number is the same as for the DG method combined with the upwind flux. The motivation of the choice of constant becomes clear when the leading error is again considered:

$$\begin{aligned} \text{modified Lax–Friedrichs: } \Delta t \lambda_{\text{DG}(1)}^{(2),\text{mLxF}} &= -\Delta t \frac{6\, q_{\text{mLxF}}}{\Delta x} \\ &= -6z, \end{aligned} \tag{3.112}$$

thus as long as $z \le \frac{1}{3}$, the leading error satisfies the stability condition, $\Re[\Delta t\, \lambda] \in [-2, 0]$. Since the condition is merely necessary and not sufficient, the full stability domains based on the modified Lax–Friedrichs flux are obtained numerically and shown in Figures 3.7(a) and 3.7(b). For this flux function, both accurate and inaccurate eigenmodes possess the same stability limit, $\tilde{\nu} \le 0.424$. This is less restrictive than the DG(1)–RK2 with the upwind flux; however, it can be observed that the modified Lax–Friedrichs flux is more dissipative than the upwind flux, especially for high frequency modes.



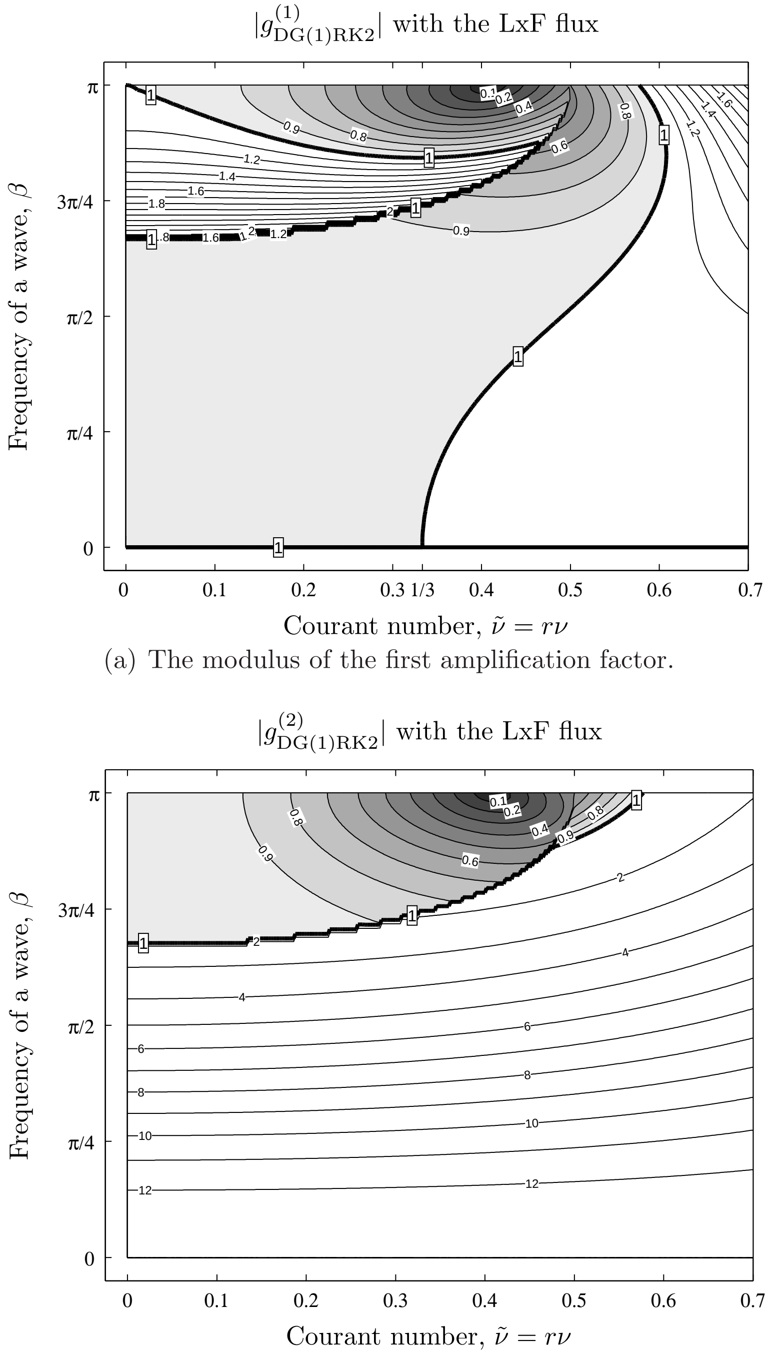
$|g^{(1)}_{\mathrm{DG(1)RK2}}|$ with the upwind flux

(a) The modulus of the accurate amplification factor.

$|g^{(2)}_{\mathrm{DG(1)RK2}}|$ with the upwind flux

(b) The modulus of the inaccurate amplification factor.

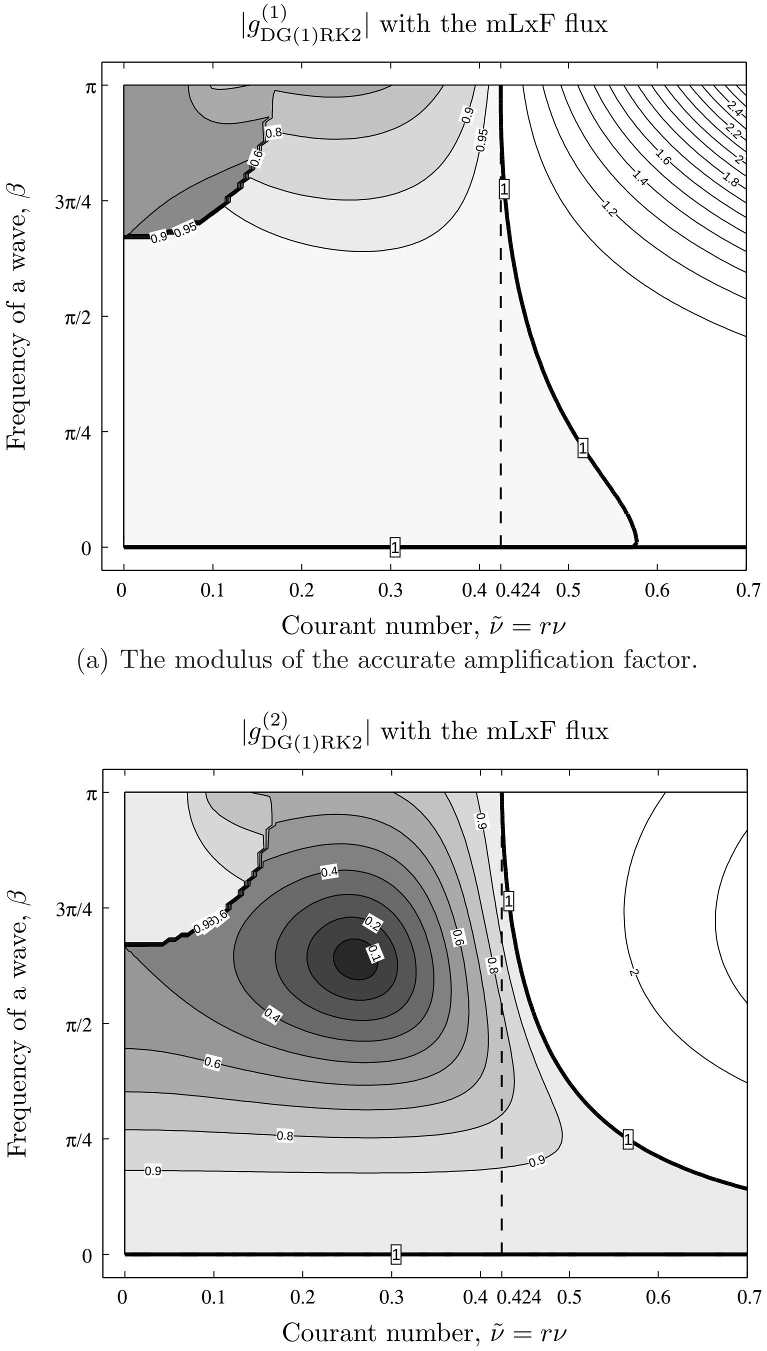
Figure 3.5: Contour plots of the modulus of the amplification factor, $|g_{\mathrm{DG(1)RK2}}(\tilde{\nu}, \beta)|$, computed with the upwind flux. The plots show that the inaccurate amplification factor results in a more strict stability condition, $\tilde{\nu} \leq 1/3$, than the accurate amplification factor, $\nu \leq 0.468$. Thus, the DG(1)–RK2 method with the upwind flux is stabile for $\tilde{\nu} \leq 1/3$.



(a) The modulus of the first amplification factor.

(b) The modulus of the second amplification factor.

Figure 3.6: Contour plots of the modulus of the amplification factor, $|g_{\mathrm{DG(1)RK2}}(\tilde{\nu}, \beta)|$, computed with the Lax–Friedrichs flux. The plots show that there is always a growing mode in a particular frequency at any Courant number. Thus, the DG(1)–RK2 method with the Lax–Friedrichs flux is unconditionally unstable.



(a) The modulus of the accurate amplification factor.

(b) The modulus of the inaccurate amplification factor.

Figure 3.7: Contour plots of the modulus of the amplification factor, $|g_{\mathrm{DG(1)RK2}}(\tilde{\nu}, \beta)|$, computed with the modified Lax–Friedrichs flux. The plots show that the modified flux results in a stable DG(1)–RK2 method for $\tilde{\nu} \leq 0.424$, whereas the original Lax–Friedrichs flux leads to an unconditionally unstable DG method.

### 3.3.3 HR–Hancock Method

The original Hancock method described in Chapter II is a fully discrete one-step method [vAvLR82, vL06]. Here, we denote the method as "HR–Hancock" or "HR–Ha." The update formula is slightly different from that of an HR–MOL method, and given by

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \frac{\Delta t}{\Delta x}\left(\hat{f}_{j+1/2}^{n+1/2} - \hat{f}_{j-1/2}^{n+1/2}\right), \tag{3.113}$$

where the time level of the flux evaluation is already specified: $t = t^{n+1/2}$. Again, the $q$-flux (3.87) is adopted; however, the input values of the flux function are given by a Taylor series expansion in space and time, thus

$$u_{j+1/2,L}^{n+1/2} = \bar{u}_j^n + \frac{1}{2}\left(1 - r\frac{\Delta t}{\Delta x}\right)\overline{\Delta u}_j^{\,n}, \tag{3.114a}$$

$$u_{j+1/2,R}^{n+1/2} = \bar{u}_{j+1}^n - \frac{1}{2}\left(1 + r\frac{\Delta t}{\Delta x}\right)\overline{\Delta u}_{j+1}^{\,n}. \tag{3.114b}$$

As in the HR–MOL method, the slope $\overline{\Delta u_j}$ is obtained by the average of two slopes over cells $(j+1, j, j-1)$, hence

$$\frac{\overline{\Delta u}_j^{\,n}}{\Delta x} = \frac{1}{2}\left(\frac{\bar{u}_{j+1}^n - \bar{u}_j^n}{\Delta x} + \frac{\bar{u}_j^n - \bar{u}_{j-1}^n}{\Delta x}\right) = \frac{\bar{u}_{j+1}^n - \bar{u}_{j-1}^n}{2\Delta x}. \tag{3.115}$$

After inserting the difference form of fluxes, and some algebra, the spatial-temporal difference operator is given by

$$\begin{aligned} \mathrm{M}_{\mathrm{HR2Ha}} = &-\frac{1}{8\Delta x}\left[(q-r)(1+r\nu)(\delta^+)^2 + (q+r)(1-r\nu)(\delta^-)^2\right] \\ &+ \frac{1}{4\Delta x}\left[(q-2r+r^2\nu)\delta^+ - (q+2r+r^2\nu)\delta^-\right], \end{aligned} \tag{3.116a}$$

or, for a Fourier mode, using (3.16),

$$\begin{aligned} \mathrm{M}_{\mathrm{HR2Ha}} = &-\frac{1}{8\Delta x}\left[(q-r)(1+r\nu)e^{2i\beta} + (q+r)(1-r\nu)e^{-2i\beta}\right] \\ &- \frac{1}{4\Delta x}\left[(3r-2q-rq\nu)e^{i\beta} + (3q+r^2\nu) - (3r+2q-rq\nu)e^{-\beta}\right]. \end{aligned} \tag{3.116b}$$

When the upwind flux, $q = r$, is used, and we set the Courant number equal to 1 ($r\nu = \tilde{\nu} = 1$), then the above operator reduces to

$$\Delta t\, \mathrm{M}_{\mathrm{HR2Ha}} = -\delta^{-}, \tag{3.117}$$

which is the exact upwind difference operator. Inserting the above equation into the original update scheme (3.113) leads to

$$\begin{aligned} \bar{u}_j^{n+1} &= \bar{u}_j^n - \delta^{-}\bar{u}_j^n \\ &= \bar{u}_{j-1}^n. \end{aligned} \tag{3.118}$$

Thus, the HR2–Hancock method produces the exact shift (page 96).

### Accuracy

Replacing the difference operators by their Fourier symbols (3.16) and taking the low-frequency limit leads to the asymptotic eigenvalue

$$\lambda_{\mathrm{HR2Ha}} = -\frac{ir}{\Delta x}\beta - \frac{r^2\nu}{2\Delta x}\beta^2 - \left(\frac{ir}{12\Delta x} - \frac{iqr\nu}{4\Delta x}\right)\beta^3 - \left(\frac{q}{8\Delta x} - \frac{r^2\nu}{6\Delta x}\right)\beta^4 + O\big(\beta^5\big)\,. \tag{3.119}$$

The order of accuracy in space is obtained by letting $\nu \to 0$, and replacing $\beta$ by the wave number $k$:

$$\lambda_{\mathrm{HR2Ha}} - \lambda_{\mathrm{exact}} = -\frac{ir}{12}\boxed{\Delta x^2}\, k^3 + O\big(k^4\big)\,, \tag{3.120}$$

thus the method is second-order accurate in space.

To examine the overall order of accuracy, the amplification factor, $g_{\mathrm{HR2Ha}} = 1 + \Delta t\ \mathrm{M}_{\mathrm{HR2Ha}}$, is expressed in the polar form where the modulus and the phase angle are given by

$$|g_{\mathrm{HR2Ha}}| = 1 - \frac{r\nu}{8}\left[\frac{q}{r} - (r\nu)^3 + 2r\nu(q\nu - 1)\right]\beta^4 + O\big(\beta^6\big)\,, \tag{3.121a}$$

$$\phi_{\mathrm{HR2Ha}} = -r\nu\beta - \frac{r\nu}{12}\left[1 - 3q\nu + 2(r\nu)^2\right]\beta^3 + O\big(\beta^5\big)\,. \tag{3.121b}$$

Following the same procedure, the local truncation error is as follows:

$$\begin{aligned}
\mathrm{LTE}_{\mathrm{HR2Ha}} &= \tilde{\lambda}_{\mathrm{HR2Ha}} - \lambda_{\mathrm{exact}} \\
&= \frac{1}{\Delta t}\left(\ln|g_{\mathrm{Hancock}}| + i\phi_{\mathrm{Hancock}}\right) - \lambda_{\mathrm{exact}} \\
&= -\frac{ir}{12}\left[1 - 3q\nu + 2(r\nu)^2\right]\Delta x^2 k^3 \\
&\quad - \frac{r}{8}\left[\frac{q}{r} - (r\nu)^3 + 2r\nu(q\nu - 1)\right]\Delta x^3 k^4 + O\big(k^5\big) \\
&= -\frac{ir}{12}\left(\boxed{\Delta x^2} - 3q\,\boxed{\Delta x \Delta t} + 2r^2\,\boxed{\Delta t^2}\right)k^3 + O\big(k^4\big)\,. \qquad (3.122)
\end{aligned}$$

Here, a new expression, $\Delta x \Delta t$, appears in the leading error term. Since the time step scales as the grid size, $\Delta t \propto \Delta x$, based on the CFL condition, this term is second-order error. Thus, the HR2–Hancock method is second-order in space and time.

### Stability

The modulus of the amplification factors, $|g_{\mathrm{HR2Ha}}(\tilde{\nu}, \beta)|$, with the upwind and Lax–Friedrichs fluxes inserted are shown in Figure 3.8 and 3.9 respectively. The shaded area indicates the stability region, thus $|g_{\mathrm{HR2Ha}}(\tilde{\nu}, \beta)| \leq 1$. These two figures show that the HR2–Hancock method combined with both the upwind and Lax–Friedrichs fluxes is linearly stabile for $\tilde{\nu} \leq 1$. Compared to the HR2–RK2 method shown in Figure 3.3 and 3.4, the HR2–Hancock is less dissipative, and also possesses the shift condition: $|g_{\mathrm{HR2Ha}}(1, \beta)| = 1$ for any $\beta \in [0, \beta]$.

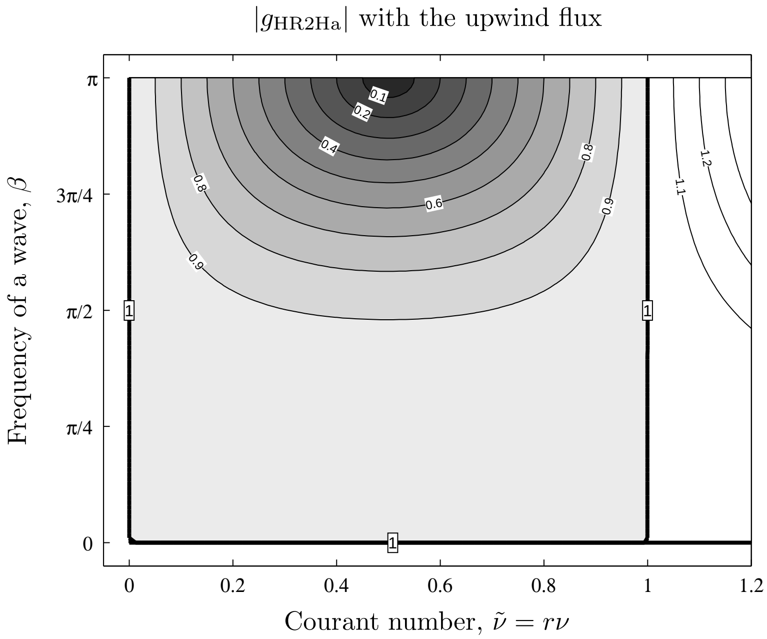

$|g_{\mathrm{HR2Ha}}|$ with the upwind flux

Figure 3.8: The modulus of the amplification factor, $|g_{\mathrm{HR2Ha}}(\tilde{\nu}, \beta)|$, combined with the upwind flux is shown in the contour plot. It shows that the HR2–Hancock method is stabile for $\tilde{\nu} \leq 1$.

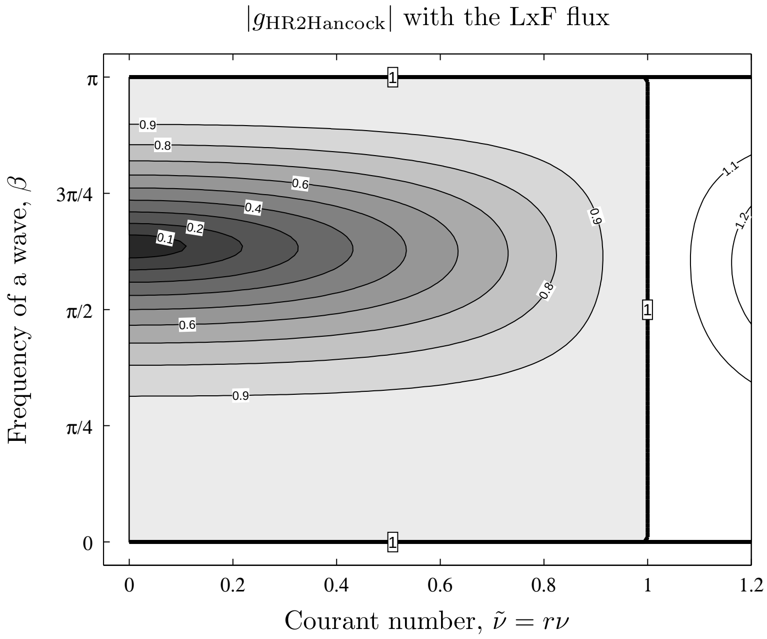

$|g_{\mathrm{HR2Hancock}}|$ with the LxF flux

Figure 3.9: The modulus of the amplification factor, $|g_{\mathrm{HR2Ha}}(\tilde{\nu}, \beta)|$, combined with the Lax–Friedrichs flux is shown in the contour plot. It shows that the HR2–Hancock method is stabile for $\tilde{\nu} \leq 1$.

### 3.3.4 DG–Hancock Method

The DG–Hancock method described in Chapter II is also fully discrete, and introduces two variables for a scalar equation, yielding a 2 × 2 amplification matrix. The update formulas for cell-average and undivided gradient have the following form:

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \frac{\Delta t}{\Delta x}\left(\hat{f}_{j+1/2}^{n+1/2} - \hat{f}_{j-1/2}^{n+1/2}\right), \tag{3.123a}$$

$$\overline{\Delta u}_j^{n+1} = \overline{\Delta u}_j^n - \frac{\Delta t}{\Delta x}6\left(\hat{f}_{j+1/2}^{n+1/2} + \hat{f}_{j-1/2}^{n+1/2} - 2r\check{u}_j\right), \tag{3.123b}$$

where the interface fluxes are evaluated at the time level, $t^{n+1/2}$, by the $q$-flux (3.87). The input values for the flux function at the time level $t^{n+k/2}$ are given by

$$u_{j+1/2,L}^{n+k/2} = \bar{u}_j^n + \frac{1}{2}\left(1 - r\frac{k\Delta t}{\Delta x}\right)\overline{\Delta u}_j^n, \tag{3.124a}$$

$$u_{j+1/2,R}^{n+k/2} = \bar{u}_{j+1}^n - \frac{1}{2}\left(1 + r\frac{k\Delta t}{\Delta x}\right)\overline{\Delta u}_{j+1}^n. \tag{3.124b}$$

The volume integral of the flux over the domain $[x_{j-1/2}, x_{j+1/2}] \times [t^n, t^{n+1}]$ simplifies owing to the linear flux. Hence, the spatial integration is done exactly, and a quadrature is only required in time. Two techniques are employed: 3-point Gauss–Lobatto and 2-point Gauss–Radau quadratures. These quadratures, with their necessary intermediate update equation, are as follows:

*3-point Gauss–Lobatto*

$$\check{u}_{j,\mathrm{GL}} = \frac{1}{6}(\bar{u}_j^n + 4\bar{u}_j^{n+1/2} + \bar{u}_j^{n+1}), \tag{3.125}$$

where

$$\bar{u}_j^{n+1/2} = \bar{u}_j^n - \frac{\Delta t}{2}\frac{1}{\Delta x}\left(\hat{f}_{j+1/2}^{n+1/4} - \hat{f}_{j-1/2}^{n+1/4}\right), \tag{3.126}$$

*2-point Gauss–Radau*

$$\check{u}_{j,\mathrm{GR}} = \frac{1}{4}(3\bar{u}_j^{n+1/3} + \bar{u}_j^{n+1}), \tag{3.127}$$

where

$$\bar{u}_j^{n+1/3} = \bar{u}_j^n - \frac{\Delta t}{3}\frac{1}{\Delta x}\left(\hat{f}_{j+1/2}^{n+1/6} - \hat{f}_{j-1/2}^{n+1/6}\right). \tag{3.128}$$

Here, the cell-interface fluxes, $\hat{f}_{j\pm1/2}^{n+1/6}$ and $\hat{f}_{j\pm1/2}^{n+1/4}$, are again obtained by the $q$-flux, with the input values (3.124) with $k = \frac{1}{6}, \frac{1}{4}$ respectively. Even though the two quadratures use different points, owing to the linear flux both Gauss–Lobatto and Gauss–Radau lead to the identical volume integral of the flux, hence $\hat{u}_{j,\mathrm{GL}} \equiv \hat{u}_{j,\mathrm{GR}}$.

After inserting the flux formula into the update scheme, and some algebra, a space-time difference operator in the form of (3.8a) results; using the notation $\mathbf{u}_j^n = [\bar{u}_j^n, \overline{\Delta u}_j^n]^T$ it can be written in the form

$$\mathbf{M}_{\mathrm{DG(1)Ha}} = \boldsymbol{\mathcal{A}}^+\mathbf{D}^+ + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{A}}^-\mathbf{D}^-, \tag{3.129}$$

where

$$\boldsymbol{\mathcal{A}}^+ = \frac{q-r}{2\Delta x}\begin{pmatrix} 1 & -\frac{1}{2}(1+r\nu) \\ 6(1+r\nu) & -3-6r\nu-2(r\nu)^2 \end{pmatrix}, \tag{3.130a}$$

$$\boldsymbol{\mathcal{A}}^- = \frac{q+r}{2\Delta x}\begin{pmatrix} -1 & -\frac{1}{2}(1-r\nu) \\ 6(1-r\nu) & 3-6r\nu+2(r\nu)^2 \end{pmatrix}, \tag{3.130b}$$

$$\boldsymbol{\mathcal{C}} = \frac{r}{\Delta x}\begin{pmatrix} 0 & 0 \\ 0 & -6\left(\frac{q}{r} - r\nu\right) \end{pmatrix}, \quad \mathbf{D}^\pm = \delta^\pm\mathbf{I}, \tag{3.130c}$$

or, when applied to a Fourier mode,

$$\mathbf{M}_{\mathrm{DG(1)Ha}} = \boldsymbol{\mathcal{A}}^+e^{i\beta\mathbf{I}} + \boldsymbol{\mathcal{C}}' - \boldsymbol{\mathcal{A}}^-e^{-i\beta\mathbf{I}}, \tag{3.131}$$

where

$$\boldsymbol{\mathcal{C}}' = -\frac{r}{\Delta x}\begin{pmatrix} \frac{q}{r} & \frac{1}{2}(1-q\nu) \\ -6(1-q\nu) & \frac{3q}{r} - 2rq\nu^2 \end{pmatrix}. \tag{3.132}$$

When the upwind flux $q = r$ is used, and we set the Courant number equal to 1 ($r\nu = \tilde{\nu} = 1$), the above operator reduces to

$$\Delta t\, \mathbf{M}_{\mathrm{DG(1)Ha}} = -\delta^{-}\mathbf{I}, \tag{3.133}$$

which is again the exact upwind difference operator. Combined with the forward Euler time integrator (3.9), the method reduces to the exact solution $\mathbf{u}_j^{n+1} = \mathbf{u}_{j-1}^{n}$. Thus the DG(1)–Hancock method produces the exact shift.

As mentioned earlier, even though a scalar equation is considered, a DG method produces a difference operator in matrix form. Thus, the characteristic equation of $\mathbf{M}_{\mathrm{DG(1)Ha}}$ is a quadratic form:

$$\begin{aligned}(\Delta x\lambda)^2 + \big[(q-r)\big((r\nu)^2 + 3r\nu + 1\big)\delta^{+} - (q+r)\big((r\nu)^2 - 3r\nu + 1\big)\delta^{-} \\ + 6(q - r^2\nu)\big]\Delta x\lambda - \frac{1}{4}(r\nu)^2\left[(q-r)^2(\delta^{+})^2 + (q+r)^2(\delta^{-})^2\right] \\ + 3(q - r^2\nu)\left[(q-r)\delta^{+} - (q+r)\delta^{-}\right] - \frac{1}{2}(q^2 - r^2)\left[6 - (r\nu)^2\right]\delta^{+}\delta^{-} = 0,\end{aligned} \tag{3.134}$$

which provides two eigenvalues; principal and extraneous. Since the general forms of eigenvalues are lengthy, only the eigenvalues for the upwind flux, $q = r$, are presented here as an example:

$$\begin{aligned}\lambda^{(1,2)}_{\text{DG(1)Ha, upwind}} = \frac{r}{\Delta x}\left[1 - 3r\nu + (r\nu)^2\right]\delta^{-} \\ - \frac{r}{\Delta x}(1 - r\nu)\left[3 \mp \sqrt{9 - 6(2 - r\nu)\delta^{-} + \big(1 - 4r\nu + (r\nu)^2\big)(\delta^{-})^2}\right].\end{aligned} \tag{3.135}$$

The asymptotic analysis that follows, though, is based on the general $q$-flux.

### Accuracy

Replacing the difference operators by their Fourier symbols (3.16), and taking the low-frequency limit, leads to

$$\begin{aligned}\lambda^{(1)}_{\mathrm{DG(1)Ha}} &= -\frac{ir}{\Delta x}\beta - \frac{r^2\nu}{2\Delta x}\beta^2 + \frac{ir^3\nu^2}{6\Delta x}\beta^3 \\ &\quad - \frac{r}{72\Delta x}\left[\frac{r}{q}\frac{\left(1-(r\nu)^2\right)^2}{1-r^2\nu/q} - 3r\nu\big(1-q\nu+(r\nu)^2\big)\right]\beta^4 + O\big(\beta^5\big)\,, \end{aligned} \tag{3.136a}$$

$$\lambda^{(2)}_{\mathrm{DG(1)Ha}} = -\frac{6r}{\Delta x}\left(\frac{q}{r} - r\nu\right) + \frac{ir\left[3-6q\nu+2(r\nu)^2\right]}{\Delta x}\beta + O\big(\beta^2\big)\,. \tag{3.136b}$$

When comparing with the exact eigenvalue (3.21a), it is clear that $\lambda^{(1)}_{\mathrm{DG(1)Ha}}$ is the principal root and $\lambda^{(2)}_{\mathrm{DG(1)Ha}}$ is the extraneous one. Based on the range of the dissipation parameter (3.28), it is easily shown that $\dfrac{q}{r} - r\nu \geq 0$. Thus, the leading term independent of $\beta$ in $\lambda^{(2)}_{\mathrm{DG(1)Ha}}$ is a negative real value, and the corresponding extraneous wave is damped quickly. The order of accuracy in space is obtained by letting $\nu \longrightarrow 0$,

$$\lambda^{(1)}_{\mathrm{DG(1)Ha}} - \lambda_{\mathrm{exact}} = -\frac{r^2}{72q}\boxed{\Delta x^3}\,k^4 + O\big(k^5\big)\,, \tag{3.137a}$$

$$\lambda^{(2)}_{\mathrm{DG(1)Ha}} - \lambda_{\mathrm{exact}} = -\frac{6q}{\Delta x} + O(k)\,, \tag{3.137b}$$

thus the principal root is third-order accurate in space and the extraneous root is zeroth-order.

To examine the dissipation and dispersion of the method, the eigenvalues of the amplification matrix of a fully discrete form, $\mathbf{G}_{\mathrm{DG(1)Ha}} = \mathbf{I} + \Delta t\,\mathbf{M}_{\mathrm{DG(1)Ha}}$, are obtained, and rewritten in the polar form (3.22). The modulus and the phase angle of the principal root are given by

$$|g^{(1)}_{\mathrm{DG(1)Ha}}| = 1 - \frac{r\nu}{72}\left[\frac{r}{q}\frac{\left(1-(r\nu)^2\right)^2}{1-r^2\nu/q} - 3r\nu(1-q\nu)\right]\beta^4 + O\big(\beta^6\big)\,, \tag{3.138a}$$

$$\begin{aligned}\phi^{(1)}_{\mathrm{DG(1)Ha}} = -r\nu\beta + \frac{r\nu}{540}\big(1-(r\nu)^2\big)\Big[3\big(1-4(r\nu)^2\big) \\ - 5r^2\frac{\big(1-(r\nu)^2\big)\big(1-3q\nu+2(r\nu)^2\big)}{(q-r^2\nu)^2}\Big]\beta^5 + O\big(\beta^7\big)\,.\end{aligned} \tag{3.138b}$$

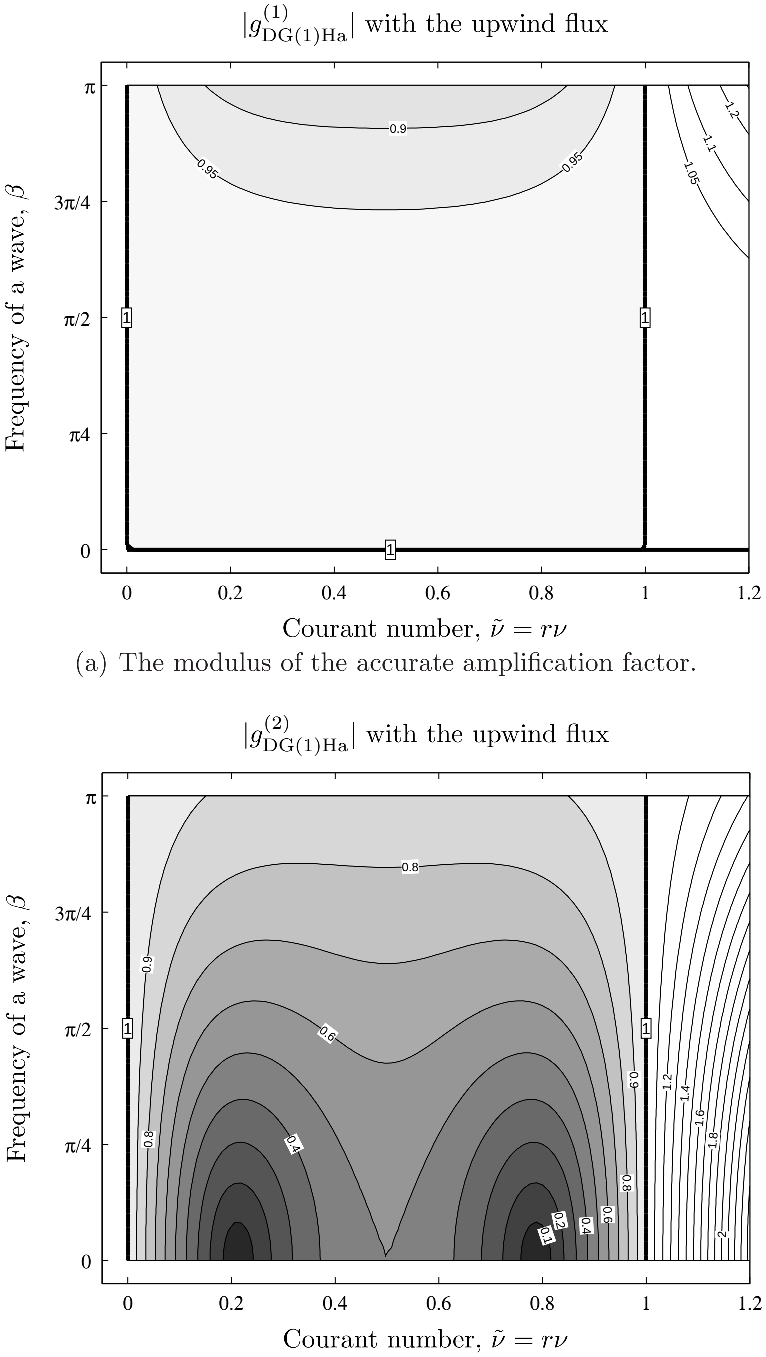
Following the same procedure as before, the local truncation error of an accurate mode is given by

$$\begin{aligned}\mathrm{LTE}^{(1)}_{\mathrm{DG(1)Ha}} &= \tilde{\lambda}^{(1)}_{\mathrm{DG(1)Ha}} - \lambda_{\mathrm{exact}} \\ &= \frac{1}{\Delta t}\left(\ln|g^{(1)}_{\mathrm{DG(1)Ha}}| + i\phi^{(1)}_{\mathrm{DG(1)Ha}}\right) - \lambda_{\mathrm{exact}} \\ &= -\frac{r}{72}\left[\frac{r}{q}\frac{\left(1-(r\nu)^2\right)^2}{1-r^2\nu/q} - 3r\nu(1-q\nu)\right]\boxed{\Delta x^3}\,k^4 + O\big(k^5\big)\,.\end{aligned} \tag{3.139}$$

Therefore, the DG(1)–Hancock method is third-order in space and time.

### Stability

The modulus of the accurate and inaccurate amplification factors, $|g^{(1),(2)}_{\mathrm{DG(1)Ha}}|$, computed with the upwind fluxes are shown in Figure 3.10. Compared to the DG(1)–RK2 method illustrated in Figures 3.5(a) and 3.5(b) on page 125, the DG(1)–Hancock method possesses a wider stability region, $\tilde{\nu} \leq 1$, and also is less dissipative at high frequencies. When the Lax–Friedrichs flux is employed, as for DG(1)–RK2, the DG(1)–Hancock method becomes unconditionally unstable shown in Figure 3.11.



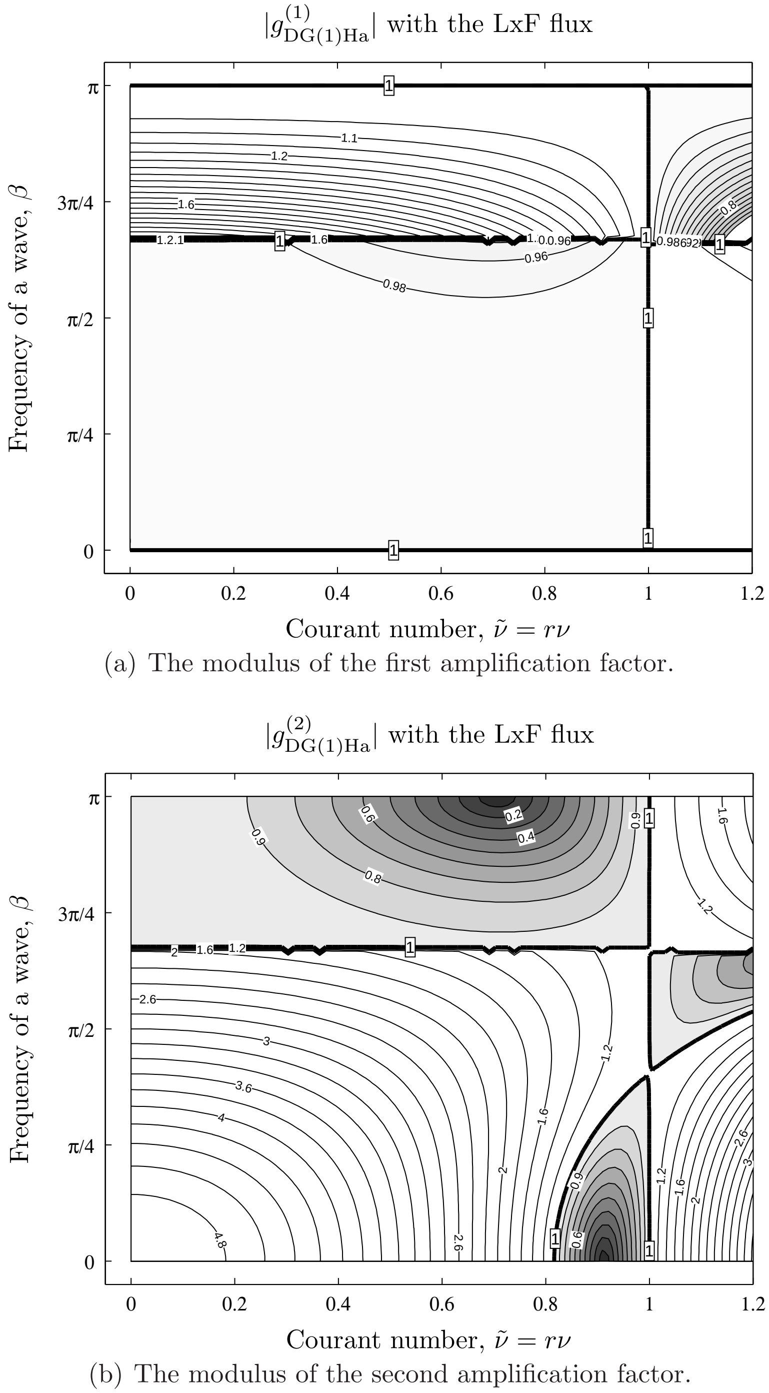
$|g^{(1)}_{\mathrm{DG(1)Ha}}|$ with the upwind flux

(a) The modulus of the accurate amplification factor.

$|g^{(2)}_{\mathrm{DG(1)Ha}}|$ with the upwind flux

(b) The modulus of the inaccurate amplification factor.

Figure 3.10: Contour plots of the modulus of the amplification factor, $|g_{\mathrm{DG(1)Ha}}(\tilde{\nu}, \beta)|$, computed with the upwind flux. These show that the DG(1)–Hancock method with the upwind flux is stable for $\tilde{\nu} \leq 1$.



(a) The modulus of the first amplification factor.

(b) The modulus of the second amplification factor.

Figure 3.11: Contour plots of the modulus of the amplification factor, $|g_{\mathrm{DG(1)Ha}}(\tilde{\nu}, \beta)|$, computed with the Lax–Friedrichs flux. These show that the DG(1)–Hancock method with the Lax–Friedrichs flux is unconditionally unstable.

### 3.3.5 Miscellaneous Methods (SV–MOL, DG–ADER)

Even though our main focus in this section is the analysis and comparison of the DG–Hancock method to the HR/DG–MOL methods, we also present results for two newly developed high-order methods: the spectral finite volume (SV) method [Wan02] and the arbitrarily high-order schemes using derivatives (DG–ADER method) [DM06].

#### SV–MOL Method

The spectral finite volume method subdivides the spectral volume (SV) into a few control volumes (CV). The cell-averaged state variables are defined at each CV inside an SV. For instance, the state variable of the SV over $x \in [x_{j-1/2}, x_{j+1/2}]$ is defined as $\bar{\mathbf{u}}_j = [\bar{u}_{j,1} \ \bar{u}_{j,2}]$ for the second-order method. One of the advantages of an SV method is that the method does not require the volume integral of a flux over the SV, whereas a DG method typically employs a quadrature to compute the volume integral of a flux. Thus, an SV method is computationally less expensive than a DG method. Furthermore, in general, an SV method has a less restrictive stability limit than a DG method. However, a DG method tends to provide more accurate (less dissipative) results than an SV method. A detailed comparison of an SV to a DG method can be found in [SW04, ZS05].

Here, we consider the second-order SV method (SV2) [Wan02, p. 224]. The semi-discrete form of the method is given by

$$\frac{\partial \bar{u}_{j,1}(t)}{\partial t} = -\frac{1}{\Delta x/2}\left(\hat{f}_{j+1/2}(t) - f_j(t)\right), \tag{3.140a}$$

$$\frac{\partial \bar{u}_{j,2}(t)}{\partial t} = -\frac{1}{\Delta x/2}\left(f_j(t) - \hat{f}_{j-1/2}(t)\right), \tag{3.140b}$$

where the interface fluxes across the SV boundaries, $\hat{f}_{j\pm1/2}(t)$, are obtained by the

$q$-flux (3.87) with the following input values:

$$u_{j+1/2,L}(t) = -\frac{1}{2}\bar{u}_{j,1}(t) + \frac{3}{2}\bar{u}_{j,2}(t), \tag{3.141a}$$

$$u_{j+1/2,R}(t) = \frac{3}{2}\bar{u}_{j+1,1}(t) - \frac{1}{2}\bar{u}_{j+1,2}(t). \tag{3.141b}$$

Since the state variable is continuous across the interior CV boundary at $x_j$, the flux, $f_j(t)$, is obtained analytically such that

$$\begin{aligned} f_j(u(t)) &= f_j\left(\frac{1}{2}\big(\bar{u}_{j,1}(t) + \bar{u}_{j,2}(t)\big)\right) \\ &= \frac{r}{2}\left[\bar{u}_{j,1}(t) + \bar{u}_{j,2}(t)\right]. \end{aligned} \tag{3.142}$$

After inserting the flux formula into the update scheme, and some algebra, the spatial difference operator becomes the following compact form:

$$\mathbf{N}_{\text{SV2}} = \boldsymbol{\mathcal{A}}^{+}\mathbf{D}^{+} + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{A}}^{-}\mathbf{D}^{-}, \tag{3.143}$$

where

$$\boldsymbol{\mathcal{A}}^{+} = \frac{q-r}{2\Delta x}\begin{pmatrix} 0 & 0 \\ 3 & -1 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}^{-} = \frac{q+r}{2\Delta x}\begin{pmatrix} 1 & -3 \\ 0 & 0 \end{pmatrix}, \tag{3.144a}$$

$$\boldsymbol{\mathcal{C}} = \frac{2q}{\Delta x}\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \mathbf{D}^{\pm} = \delta^{\pm}\mathbf{I}, \tag{3.144b}$$

or, for a Fourier mode,

$$\mathbf{N}_{\text{SV2}} = \boldsymbol{\mathcal{A}}^{+}e^{i\beta\mathbf{I}} + \boldsymbol{\mathcal{C}}' - \boldsymbol{\mathcal{A}}^{-}e^{-i\beta\mathbf{I}}, \tag{3.145}$$

where

$$\boldsymbol{\mathcal{C}}' = \frac{1}{2\Delta x}\begin{pmatrix} -(3q-r) & q-3r \\ q+3r & -(3q+r) \end{pmatrix}. \tag{3.146}$$

**Accuracy** The eigenvalues of the spatial operator, $\mathbf{N}_{\mathrm{SV2}}$, in the low-frequency limit are as follows:

$$\lambda^{(1)}_{\mathrm{SV2}} = -\frac{ir}{\Delta x}\beta + \frac{ir}{24\Delta x}\beta^3 - \frac{r^2}{32q\Delta x}\beta^4 + O\big(\beta^5\big)\,, \tag{3.147a}$$

$$\lambda^{(2)}_{\mathrm{SV2}} = -\frac{4q}{\Delta x} + \frac{2ir}{\Delta x}\beta + O\big(\beta^2\big)\,. \tag{3.147b}$$

Similar to a DG discretization, the leading term of the extraneous root has a negative real value. Thus, the spurious mode is damped quickly. The order of accuracy in space is obtained by replacing $\beta$ by the wave number $k$:

$$\lambda^{(1)}_{\mathrm{SV2}} - \lambda_{\mathrm{exact}} = \frac{ir}{24}\,\boxed{\Delta x^2}\,k^3 + O\big(k^4\big)\,, \tag{3.148a}$$

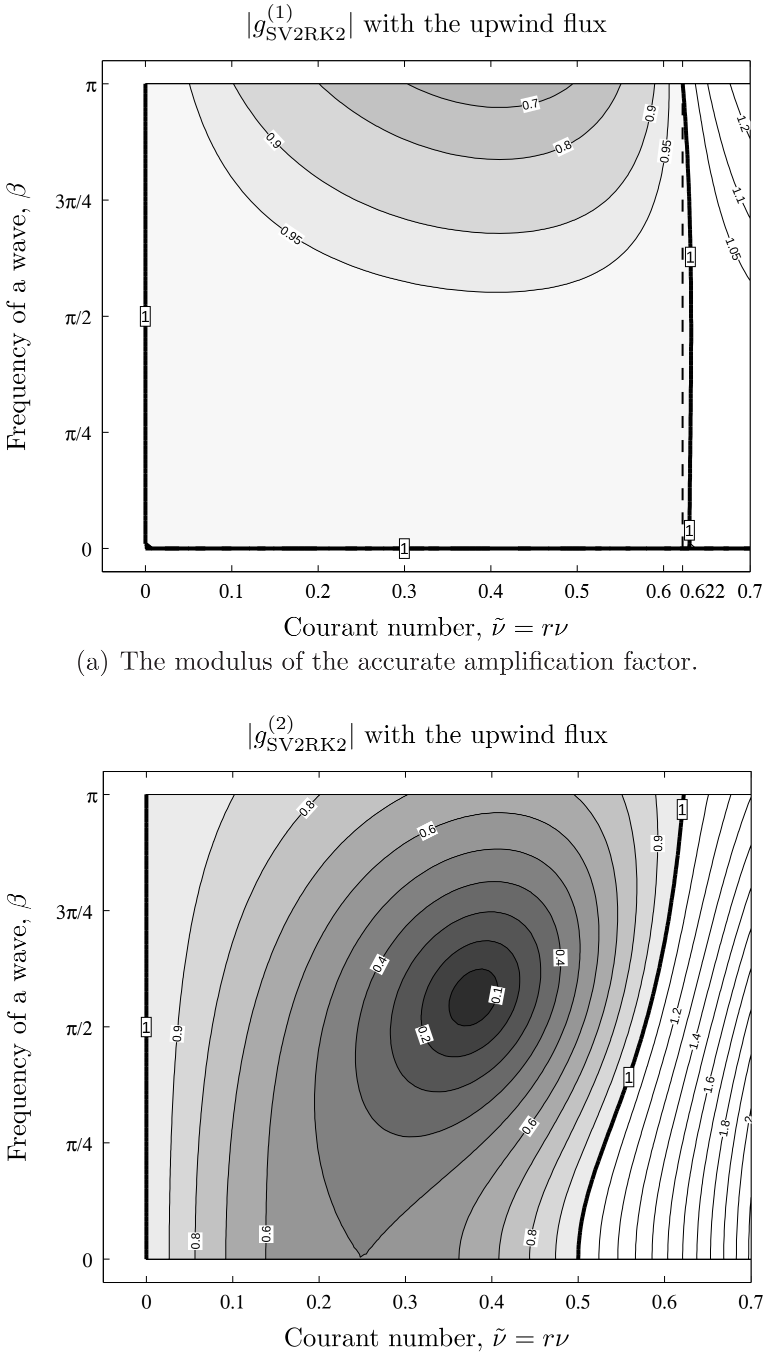
$$\lambda^{(2)}_{\mathrm{SV2}} - \lambda_{\mathrm{exact}} = -\frac{4q}{\Delta x} + O(k)\,. \tag{3.148b}$$

Hence, the SV discretization containing two CVs (SV2) is second-order in space. The overall accuracy can be obtained by combining it with an ODE solver. Here, the second-order Runge–Kutta (RK2) method is adopted for the time integration. Following the same procedure as in the previous analysis, the local truncation error of the SV2–RK2 method is given by

$$\begin{aligned}\mathrm{LTE}_{\mathrm{SV2RK2}} &= \frac{ir}{24}\left[1 - 4(r\nu)^2\right]\Delta x^2 k^3 - \frac{r}{32}\left[\frac{r}{q} - 4(r\nu)^3\right]\Delta x^3 k^4 \\ &= \frac{ir}{24}\left(\boxed{\Delta x^2} - 4r^2\,\boxed{\Delta t^2}\right)k^3 - \frac{r}{32}\left[\frac{r}{q} - 4(r\nu)^3\right]\Delta x^3 k^4 .\end{aligned} \tag{3.149}$$

The above equation shows that the SV2–RK2 method is second-order in space and time. In contrast to the DG(1)–RK2 method, a higher-order time discretization, e.g., SV2–RK3, does not increase the overall accuracy since the spatial discretization error is still second-order. This is clearly shown by comparing the eigenvalues (3.148a) to (3.106a) on page 121.

**Stability** The modulus of the amplification factor, $|g_{\mathrm{SV2RK2}}(\tilde{\nu}, \beta)|$, computed with the upwind flux is shown in Figure 3.12. Interestingly, the dissipation property of the SV2–RK2 method is qualitatively similar to that of the DG(1)–RK2 method shown in Figures 3.5(a) and 3.5(b) on page 125. Both methods are second-order accurate in space and time, however the stability domains are different: the SV2–RK2 method has a wider stability domain $\tilde{\nu} \leq \dfrac{1}{2}$, whereas DG(1)–RK2 has $\tilde{\nu} \leq \dfrac{1}{3}$. This wider stability domain together with the volume-integral free flux makes the SV2–RK2 method less computationally expensive than the DG(1)–RK2 method. However, the SV2–RK2 tends to be more dissipative than the DG(1)–RK2. For instance, at low frequencies, the dominant dissipation of the SV2–RK2 is twice as large as the DG(1)–RK2 method. More detailed comparisons are followed at the end of the section.


$|g^{(1)}_{\text{SV2RK2}}|$ with the upwind flux

(a) The modulus of the accurate amplification factor.

$|g^{(2)}_{\text{SV2RK2}}|$ with the upwind flux

(b) The modulus of the inaccurate amplification factor.

Figure 3.12: Contour plots of the modulus of the amplification factor, $|g_{\text{SV2RK2}}(\tilde{\nu}, \beta)|$, computed with the upwind flux. These show that the inaccurate amplification factor has a more restrictive stability domain. Thus, the SV2–RK2 method with the upwind flux is stabile for $\tilde{\nu} \leq 1/2$.

### DG–ADER Method

A DG–ADER method is a fully discrete method, utilizing the Cauchy–Kovalevskaya procedure to replace the temporal derivatives by the spatial derivatives [DM05]. Even though a DG–ADER method can be extended to an arbitrary high-order of accuracy, we only analyze the second method, DG(1)–ADER, here. The resulting method is similar to the DG(1)–Hancock method except for the evaluation of the volume integral of the flux. For a linear flux, the flux integral in time is approximated by the flux at the intermediate state, $n + 1/2$, thus

$$\frac{1}{\Delta t}\int\limits_{T^n} f(\bar{u}_j)dt \simeq f(\hat{u}_j^{n+1/2}), \tag{3.150}$$

where the predicted value, $\hat{u}_j^{n+1/2}$, is given by

$$\begin{aligned}\hat{u}_j^{n+1/2} &= \bar{u}_j^n - \frac{\Delta t}{2} f_x(u)\\ &= \bar{u}_j^n - \frac{r\Delta t}{2}\frac{\overline{\Delta u}_j^n}{\Delta x}.\end{aligned} \tag{3.151}$$

Note that compared to the DG(1)–Hancock method, the volume integral of the DG(1)–ADER method does not require a Riemann solver; the wave interactions across the cell interfaces are neglected in the volume integral. Here, we omit the derivation of the method; the interested reader is referred to the original papers [DM06, DM05]. The spatial difference operator of the DG(1)–ADER method together with the $q$–flux (3.87) has the following form:

$$\mathbf{M}_{\mathrm{DG(1)ADER}} = \boldsymbol{\mathcal{A}}^{+}\mathbf{D}^{+} + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{A}}^{-}\mathbf{D}^{-}, \tag{3.152}$$

where

$$\boldsymbol{\mathcal{A}}^{+} = \frac{q-r}{2\Delta x}\begin{pmatrix} 1 & -\dfrac{1}{2}(1+r\nu) \\ 6 & -3(1+r\nu) \end{pmatrix}, \tag{3.153a}$$

$$\boldsymbol{\mathcal{A}}^{-} = \frac{q+r}{2\Delta x}\begin{pmatrix} -1 & -\dfrac{1}{2}(1-r\nu) \\ 6 & 3(1-r\nu) \end{pmatrix}, \tag{3.153b}$$

$$\boldsymbol{\mathcal{C}} = \frac{1}{\Delta x}\begin{pmatrix} 0 & 0 \\ 0 & -6q \end{pmatrix}, \quad \mathbf{D}^{\pm} = \delta^{\pm}\mathbf{I}, \tag{3.153c}$$

or, for a Fourier mode,

$$\mathbf{M}_{\mathrm{DG(1)ADER}} = \boldsymbol{\mathcal{A}}^{+}e^{i\beta\mathbf{I}} + \boldsymbol{\mathcal{C}}' - \boldsymbol{\mathcal{A}}^{-}e^{-i\beta\mathbf{I}}, \tag{3.154}$$

where

$$\boldsymbol{\mathcal{C}}' = -\frac{r}{\Delta x}\begin{pmatrix} \dfrac{q}{r} & \dfrac{1}{2}(1-q\nu) \\ -6 & 3\left(\dfrac{q}{r}+r\nu\right) \end{pmatrix}. \tag{3.155}$$

Compared to the DG(1)–Hancock method (3.130) on page 133, the DG(1)–ADER method contains less information due to the crude approximation of the volume integral of the flux given by (3.151). More specifically, the DG(1)–ADER method does not carry the $(r\nu)^2$ term, which is necessary for third-order accuracy in space and time.

When the upwind flux, $q = r$, is used, the difference operator multiplied by $\Delta t$ is reduced to

$$\Delta t\,\mathbf{M}_{\mathrm{DG(1)ADER}} = \begin{pmatrix} -\tilde{\nu}\delta^{-} & -\dfrac{1}{2}\tilde{\nu}(1-\tilde{\nu})\delta^{-} \\ 6\tilde{\nu}\delta^{-} & 3\tilde{\nu}\big((1-\tilde{\nu})\delta^{-}-2\big) \end{pmatrix}, \tag{3.156}$$

where $\tilde{\nu} = r\nu$. The above equation shows that no Courant number $\tilde{\nu} \in [0,1]$ provides $-\delta^{-}\mathbf{I}$. Hence, the DG(1)–ADER method does not satisfy the shift condition.

To obtained the local truncation error, the forward Euler method is adopted for the time integration:

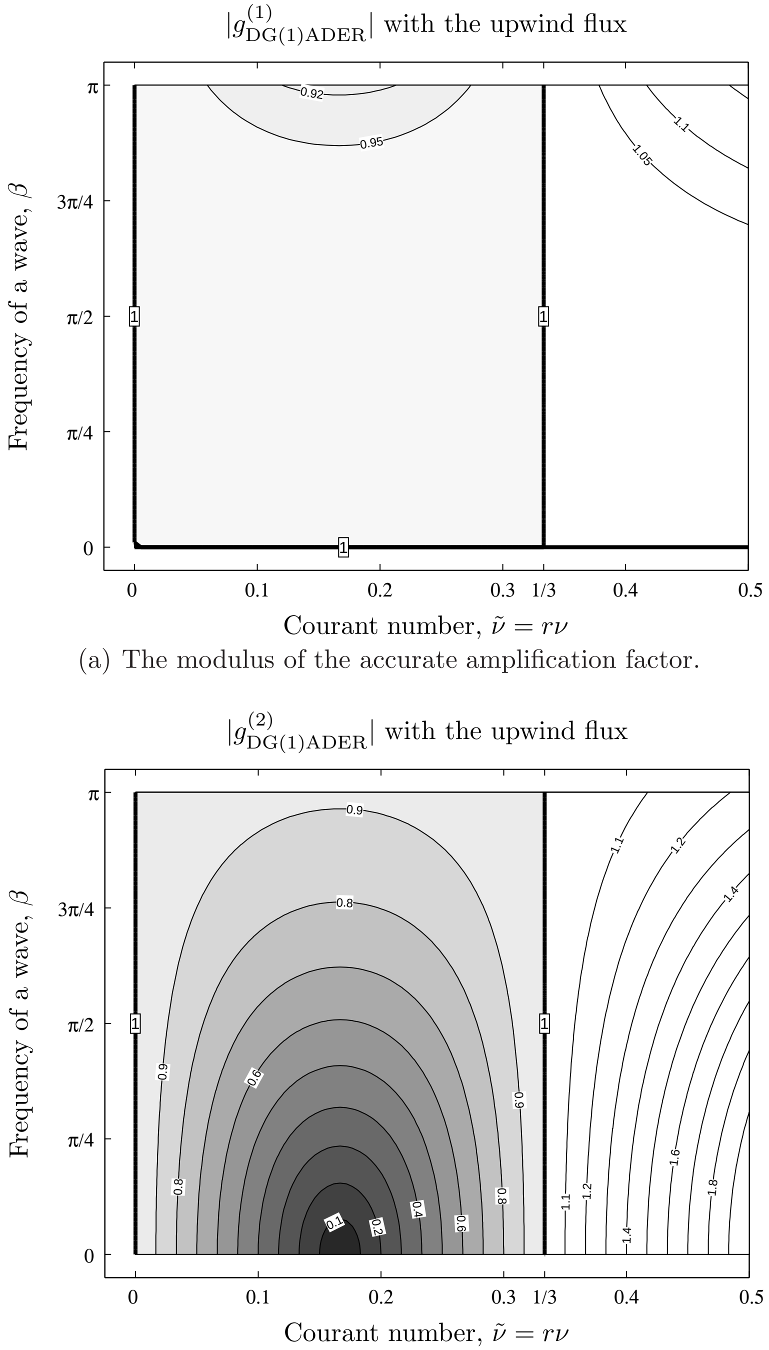
$$\mathbf{G}_{\mathrm{DG(1)ADER}} = \mathbf{I} + \Delta t\, \mathbf{M}_{\mathrm{DG(1)ADER}}. \tag{3.157}$$

**Accuracy** Following the same procedure, the local truncation error of the method becomes

$$\begin{aligned}
\mathrm{LTE}^{(1)}_{\mathrm{DG(1)ADER}} &= \tilde{\lambda}^{(1)}_{\mathrm{DG(1)ADER}} - \lambda_{\mathrm{exact}} \\
&= -\frac{ir}{12} r\nu \left(\frac{r}{q} - r\nu\right) \boxed{\Delta x^2}\, k^3 \\
&\quad - \frac{r}{72}\frac{r}{q}\left[1 - 4\frac{r}{q}(r\nu) + 3(r\nu)^2\right] \Delta x^3 k^4 + O\big(k^5\big)\,.
\end{aligned} \tag{3.158a}$$

Thus, the DG(1)–ADER method is second-order accurate in space and time.

**Stability** The modulus of the amplification factor, $|g_{\mathrm{DG(1)ADER}}(\tilde{\nu}, \beta)|$, computed with the upwind flux, is shown in Figure 3.13. In contrast to the SV2–RK2 method, the dissipation property of the DG(1)–ADER method is qualitatively similar to that of DG(1)–Hancock method shown in Figure 3.10 on page 137. However, the DG(1)–ADER method is second-order accurate and the stability domain is more restrictive, $\tilde{\nu} \leq \frac{1}{3}$, whereas the DG(1)–Hancock method is third-order accurate while stable for $\tilde{\nu} \leq 1$.



(a) The modulus of the accurate amplification factor.

(b) The modulus of the inaccurate amplification factor.

Figure 3.13: Contour plots of the modulus of the amplification factor, $|g_{\mathrm{DG(1)ADER}}(\tilde{\nu}, \beta)|$, computed with the upwind flux. The plots show that both accurate and inaccurate amplification factors are stabile for $\tilde{\nu} \leq 1/3$. Thus, the DG(1)–ADER method with the upwind flux is stabile for $\tilde{\nu} \leq 1/3$.

### 3.3.6 Dominant Dispersion/Dissipation Error and Stability in 1-D

The results of a Fourier analysis for each method are listed below for comparison. The details of the HR3–RK3, DG(2)–RK3, and DG(2)–ADER methods are presented in Appendix B on page 349. The local truncation errors show the dominant dispersion error, $O(k^3)$-term, and the dissipation error, $O(k^4)$-term. Moreover, owing to the equivalence of the Fourier analysis and the modified-equation analysis, the leading error term indicates the order of accuracy. Note that $c_3 \propto \Delta x^2$ and $c_4 \propto \Delta x^3$.

**semi-discrete methods:**

$$\mathrm{LTE}_{\mathrm{HR2RK2}} = c_3 \left[1 + 2(r\nu)^2\right] k^3 \quad + \quad c_4 \left[\frac{q}{r} - (r\nu)^3\right] k^4, \tag{3.159a}$$

$$\mathrm{LTE}_{\mathrm{HR2RK3}} = c_3\ k^3 \quad + \quad c_4 \left[\frac{q}{r} + \frac{1}{3}(r\nu)^3\right] k^4, \tag{3.159b}$$

$$\mathrm{LTE}_{\mathrm{HR3RK3}} = \quad + \quad \frac{2}{3} c_4 \left[\frac{q}{r} + \frac{1}{2}(r\nu)^3\right] k^4, \tag{3.159c}$$

$$\mathrm{LTE}^{(1)}_{\mathrm{DG(1)RK2}} = c_3 \left[2(r\nu)^2\right] k^3 \quad + \quad \frac{1}{9} c_4 \left[\frac{r}{q} - 9(r\nu)^3\right] k^4, \tag{3.159d}$$

$$\mathrm{LTE}^{(1)}_{\mathrm{DG(1)RK3}} = \quad \frac{1}{9} c_4 \left[\frac{r}{q} + 3(r\nu)^3\right] k^4, \tag{3.159e}$$

$$\mathrm{LTE}^{(1),\mathrm{upwind}}_{\mathrm{DG(2)RK3}} = \quad \frac{1}{9} c_4 \left[ \qquad 3(r\nu)^3\right] k^4, \tag{3.159f}$$

$$\mathrm{LTE}^{(1)}_{\mathrm{SV2RK2}} = -\frac{1}{2} c_3 \left[1 - 4(r\nu)^3\right] k^3 \quad + \quad \frac{2}{9} c_4 \left[\frac{r}{q} - 4(r\nu)^3\right] k^4, \tag{3.159g}$$

**fully discrete methods:**

$$\mathrm{LTE}_{\mathrm{HR2Ha}} = c_3 \left[1 - 3q\nu + 2(r\nu)^2\right] k^3 + \quad c_4 \left[\frac{q}{r} - (r\nu)^3 + 2r\nu(q\nu - 1)\right] k^4, \tag{3.159h}$$

$$\mathrm{LTE}^{(1)}_{\mathrm{DG(1)Ha}} = \quad \frac{1}{9} c_4 \left[\frac{r}{q} \frac{\left(1 - (r\nu)^2\right)^2}{1 - r^2\nu/q} - 3r\nu(1 - q\nu)\right] k^4, \tag{3.159i}$$

$$\mathrm{LTE}^{(1)}_{\mathrm{DG(1)ADER}} = c_3 \left[r\nu \left(\frac{r}{q} - r\nu\right)\right] k^3 \quad + \quad \frac{1}{9} c_4 \left[\frac{r}{q}\left(1 - 4\frac{r}{q}(r\nu) + 3(r\nu)^2\right)\right] k^4, \tag{3.159j}$$

$$\mathrm{LTE}^{(1),\mathrm{upwind}}_{\mathrm{DG(2)ADER}} = \quad \frac{1}{15} c_4 \left[r\nu(1 - r\nu)\right] k^4, \tag{3.159k}$$

where

$$c_3 = -\frac{ir}{12}\ \boxed{\Delta x^2}, \quad c_4 = -\frac{r}{8}\ \boxed{\Delta x^3}. \tag{3.160}$$

The above equations show that the leading errors of the HR3–RK3, DG(1)–RK3, DG(2)–RK3, DG(1)–Hancock, and DG(2)–ADER methods are $O(\Delta x^3)$, whereas in the rest of methods they are $O(\Delta x^2)$. Interestingly, the DG(1) spatial discretization can yield a third-order method, at least for a linear equation discretized on a uniform grid, if a proper time integration method is adopted. The Hancock and RK3 method lead to a third-order method; note that, DG(1)–RK3 requires three flux calculations at each cell-interface whereas DG(1)–Hancock requires two to achieve the same order. The DG(2) spatial discretization together with the same (third) order of temporal discretization provides a third-order method: DG(2)–RK3 and DG(2)–ADER. In these methods, the leading error can be attributed to the temporal discretization, as can be seen by letting $\tilde{\nu} = r\nu \rightarrow 0:$ the $O(k^4)$-term disappears. The analysis also shows the lower dissipation of DG discretizations: the leading-error coefficient of a DG(1) method is $\dfrac{1}{9}$ of the value for HR2, $\dfrac{1}{6}$ of the value for HR3, and $\dfrac{1}{2}$ of the value for SV2.

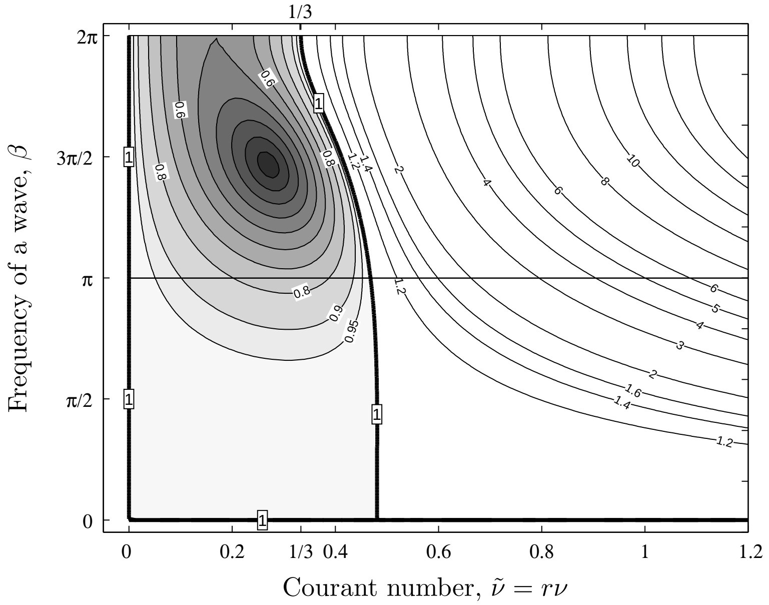
The stability limits of the methods when combined with the upwind and Lax–Friedrichs fluxes are shown in Table 3.3. In an HR (finite-volume) method, the linear stability limit increases as the order of method increases due to the inclusion of wider stencils. Conversely, DG and SV methods reduce their stability domain while increasing the order of accuracy, because increasing the number of unknowns per cell is equivalent to a grid refinement. This suggests that the inaccurate (second) amplification factor indeed contributes to the accuracy, but on the subgrid scale. In order to understand the grid-refinement phenomena of the DG and SV methods, previously presented stability domains are reproduced over the domain $\beta \in [0, 2\pi]$, and shown in Figure 3.14. For each method, the accurate amplification factor is plotted over $\beta \in [0, \pi]$, while the inaccurate one is plotted for $\beta \in [\pi, 2\pi]$. A smooth

| | method | order | maximum Courant number, $\tilde{\nu}_{\max}$ | |
|---|---|---|---|---|
| | | | upwind: $q = r$ | Lax–Friedrichs: $q = \Delta x/\Delta t$ |
| semi-discrete | HR1–RK1 | 1 | 1.0 | 1.0 |
| | HR2–RK2 | 2 | 1.0 | 1.0 |
| | HR2–RK3 | 2 | 1.175 | 1.499 |
| | HR3–RK3 | 3 | 1.625 | 1.681 |
| | DG(1)–RK2 | 2 | 0.333 | unstable |
| | DG(1)–RK3 | 3 | 0.409 | unstable |
| | DG(2)–RK3 | 3 | 0.209 | unstable |
| | SV2–RK2 | 2 | 0.500 | unstable |
| | SV3–RK3 | 3 | 0.333 | (unconfirmed) |
| fully discrete | HR2–Hancock | 2 | 1.0 | 1.0 |
| | DG(1)–Hancock | 3 | 1.0 | unstable |
| | DG(1)–ADER | 2 | 0.333 | unstable |
| | DG(2)–ADER | 3 | 0.170 | unstable |

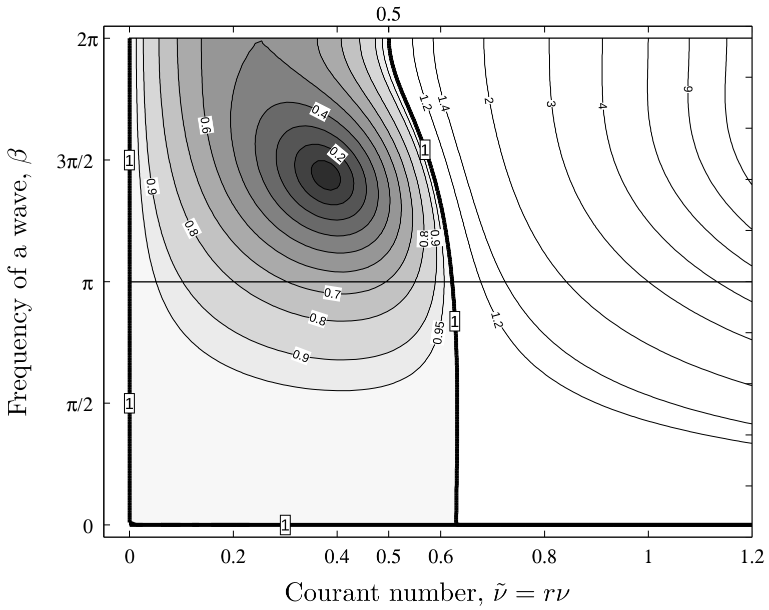
Table 3.3: The maximum stable Courant number, $\tilde{\nu}_{\max} := r\dfrac{\Delta t}{\Delta x}$, of a method applied to the 1-D linear advection equation is tabulated. The DG(1)–Hancock method is seen to possess the largest stability domain among all DG discretizations listed here.

transition of the modulus of the amplification factor across the wave frequency $\pi$ is observed for all four methods, and also the stability of MOL based methods, DG(1)–RK2 and SV2–RK2, is restricted by the wave of highest frequency, $\beta = 2\pi$.

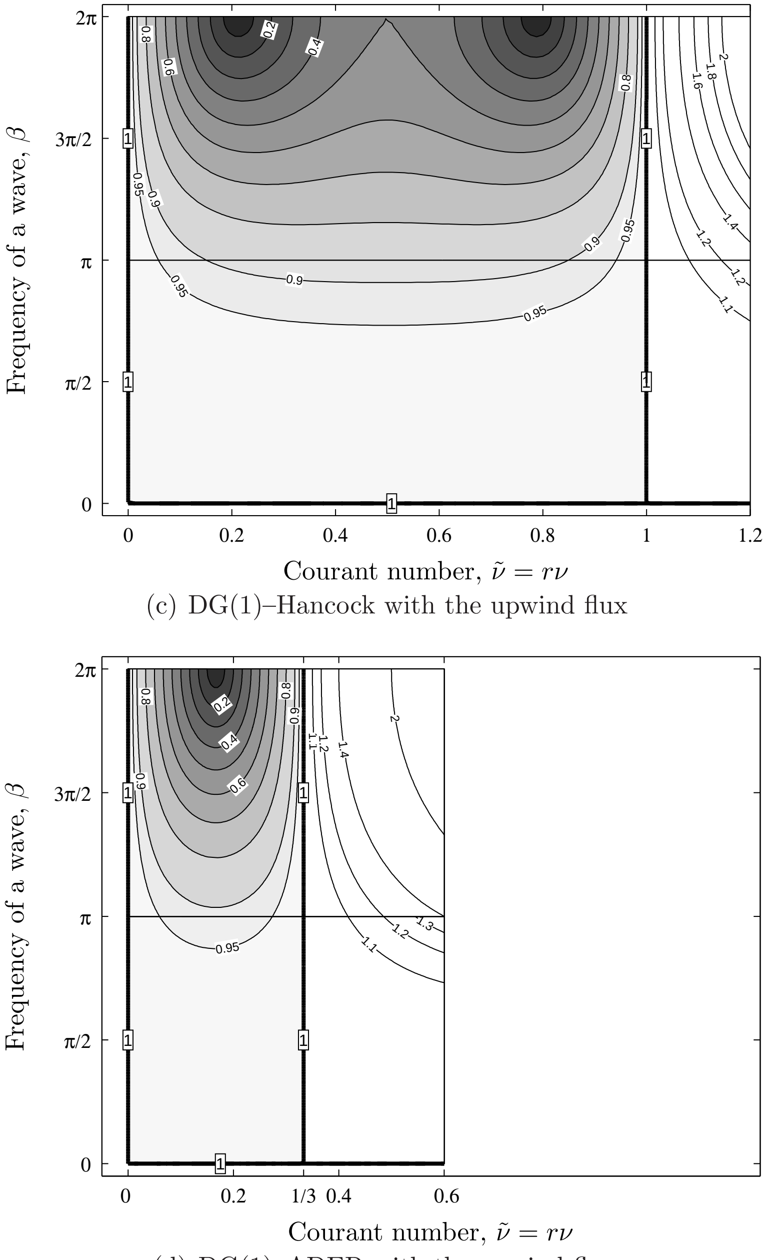
Among DG(1)–Hancock, DG(1)–RK3, and DG(2)–RK3, all third-order accurate, DG(1)–Hancock possesses the largest stability domain, $\tilde{\nu}_{\max} = 1.0$. Nevertheless, DG(1)–Hancock requires only two Riemann solvers per cell-interface per time-step, whereas both DG(1)–RK3 and DG(2)–RK3 need three Riemann solvers.



(a) DG(1)–RK2 with the upwind flux



(b) SV2–RK2 with the upwind flux



(c) DG(1)–Hancock with the upwind flux

(d) DG(1)–ADER with the upwind flux

Figure 3.14: Previously presented stability domains of various upwind methods applied to the 1-D linear advection equation are reproduced over the frequency $\beta \in [0, 2\pi]$. The accurate amplification factor is plotted for $\beta \in [0, \pi]$ while the inaccurate one is plotted for $\beta \in [\pi, 2\pi]$. The shaded area indicates the region where $|g_{\text{method}}(\tilde{\nu})| \leq 1$. Observe that the smooth transition of an amplification factor across the frequency $\pi$, and also that the stability limit for MOL is typically restricted by the highest frequency $2\pi$.

### 3.3.7 Stability of Methods with the Rusanov Flux

When the scalar linear advection equation is considered, the definition of the Rusanov flux, which adopts the largest characteristic speed as the dissipation parameter, becomes vague. One could say the Rusanov flux actually coincides with the upwind flux. Meanwhile, when hyperbolic-relaxation equations are considered later in Chapter IV, the Rusanov and upwind fluxes are defined distinctively owing to the existence of two kinds of waves: frozen and equilibrium waves.

The linear stability conditions of methods with the Rusanov flux are now investigated. Here, let the frozen wave speed be 1, and the equilibrium wave speed is $r \leq 1$, then the Courant number is defined by (3.2) on page 88. Since the equilibrium wave speed $r$ is a free parameter, the stability of a method is conducted for the whole rage of $r \in [0, 1]$. Obviously, when $r = 1$, the analyses recover the previous stability analyses with the upwind flux. Rather surprisingly, as the equilibrium wave speed $r$ approaches zero, some methods increase their stability domains, while the DG(1)–Hancock method reduces its stability. The stability limit of each method is summarized in Table 3.4. As for the DG(1)–Hancock method, the detailed maximum Courant number based on numerical contour plots is tabulated in Table 3.5. Throughout the analysis, it is observed that the most unstable mode occurs at the longest wave ($\beta = 0$) in the rage of $r \in \left[0, \sqrt{3}/2\right]$. Hence, an analytical maximum Courant number can be obtained by inserting $\beta = 0$ into the amplification factor, and solve for $\nu$:

$$\nu_{\max}(r) = \frac{3 - \sqrt{9 - 12r^2}}{6r^2}; \quad r \in [0, \sqrt{3}/2]. \tag{3.161}$$

Once $r$ is greater than $\dfrac{\sqrt{3}}{2}$, the most unstable mode shift from $\beta = 0$ to $\beta = \dfrac{\pi}{2}$. Even though the analytical form of amplification factors for $\beta = \dfrac{\pi}{2}$ are obtained,

| | method | order | maximum Courant number, $\nu_{\max}$<br>Rusanov flux: $q = 1$ |
|---|---|---|---|
| semi-discrete | HR2–RK2 | 1 | 1.0 |
| | HR2–RK3 | 2 | [1.175, 1.256] |
| | HR3–RK3 | 3 | [1.625, 1.884] |
| | DG(1)–RK2 | 2 | 0.333 |
| | DG(1)–RK3 | 3 | [0.409, 0.418] |
| | DG(2)–RK3 | 3 | 0.209 |
| | SV2–RK2 | 2 | 0.500 |
| | SV3–RK3 | 3 | (unconfirmed) |
| fully discrete | HR2–Hancock | 2 | 1.0 |
| | DG(1)–Hancock | 3 | [0.333, 1.0] |
| | DG(1)–ADER | 2 | 0.333 |
| | DG(2)–ADER | 3 | [0.166, 0.170] |

Table 3.4: The maximum stable Courant number, $\nu_{\max} := 1\dfrac{\Delta t}{\Delta x}$, of a method applied to the 1-D linear advection equation is tabulated. The DG(1)–Hancock method reduces its stability as the equilibrium wave speed becomes smaller. If an interval is indicated, the method's stability varies with the value of $r$.

deriving an explicit form of $\nu$ is cumbersome, and further more it does not include the corresponding transition from $\beta = 0$ to $\dfrac{\pi}{2}$. Hence, for the higher $r$-values, we only approximate $\nu$ by a $P^4$ polynomial function based on the data listed in Table 3.5(b). Thus, the maximum Courant number with respect to the equilibrium wave speed can be approximated as follows:

$$\nu_{\max}(r) \approx \begin{cases} \dfrac{1}{3} + \dfrac{1}{9}r^2 + \dfrac{2}{27}r^4 & \text{if } \ 0 \leq r \leq \dfrac{\sqrt{3}}{2}, \\ \displaystyle\sum_{k=0}^{4} c_k r^k & \text{if } \ \dfrac{\sqrt{3}}{2} < r \leq 1, \end{cases} \tag{3.162}$$

where the coefficients are tabulated in Table 3.6. Finally, measured Courant numbers, analytical values, and approximated values (3.162) are plotted in Figure 3.15.

Later, the 10-moment equations are adopted as nonlinear hyperbolic-relaxation

(a) $r \in \left[0, \sqrt{3}/2\right]$

| $r$ | $\nu_{\max}(r)$ |
|---|---|
| 0.0 | 1/3 |
| 0.1 | 0.335 |
| 0.2 | 0.338 |
| 0.3 | 0.344 |
| 0.4 | 0.353 |
| 0.5 | 0.367 |
| 0.6 | 0.387 |
| 0.7 | 0.420 |
| 0.8 | 0.482 |
| 0.82 | 0.504 |
| 0.84 | 0.536 |
| 0.85 | 0.560 |
| 0.86 | 0.596 |
| 0.865 | 0.636 |
| $\sqrt{3}/2 \sim 0.866$ | $2/3 \sim 0.667$ |

(b) $r \in \left[\sqrt{3}/2, 1\right]$

| $r$ | $\nu_{\max}(r)$ |
|---|---|
| 0.866 | 0.758 |
| 0.87 | 0.787 |
| 0.88 | 0.828 |
| 0.90 | 0.877 |
| 0.92 | 0.912 |
| 0.94 | 0.939 |
| 0.96 | 0.962 |
| 0.98 | 0.982 |
| 1.0 | 1.0 |

Table 3.5: The allowable maximum Courant number with respect to the equilibrium wave speed $r \in [0, 1]$ for the DG(1)–Hancock method with the Rusanov flux is tabulated. These values are measured based on contour plots of the modulus of amplification factors. When $r = 1$, the result recovers the stability with the upwind flux, while the stability domain is reduced towards 1/3 as the equilibrium wave gets smaller.

equations. This system has a dimensionless wave speed defined by

$$r(M) := \frac{u + a}{u + \sqrt{3}a} = \frac{M + 1}{M + \sqrt{3}}, \tag{3.163}$$

hence, as the Mach number increases, the maximum stable Courant number also relaxes towards unity. The transition of the Courant number with respect to $r$ is shown in Figure 3.16.

| | |
|---|---|
| $c_0$ | $-1.569446110504144\times10^3$ |
| $c_1$ | $6.612514388329900\times10^3$ |
| $c_2$ | $-1.044389804968620\times10^4$ |
| $c_3$ | $7.331699503711032\times10^3$ |
| $c_4$ | $-1.929870438514635\times10^3$ |

Table 3.6: Coefficients of the polynomial approximation in (3.162).

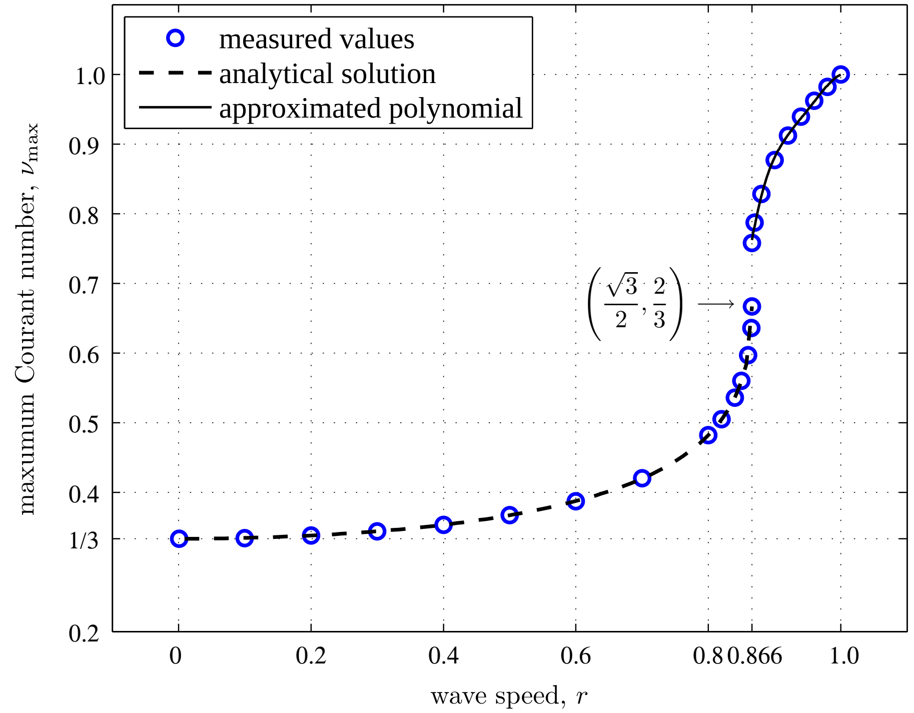


Figure 3.15: The maximum Courant number with respect to the equilibrium wave speed $r$ is shown together with analytical and approximated values.

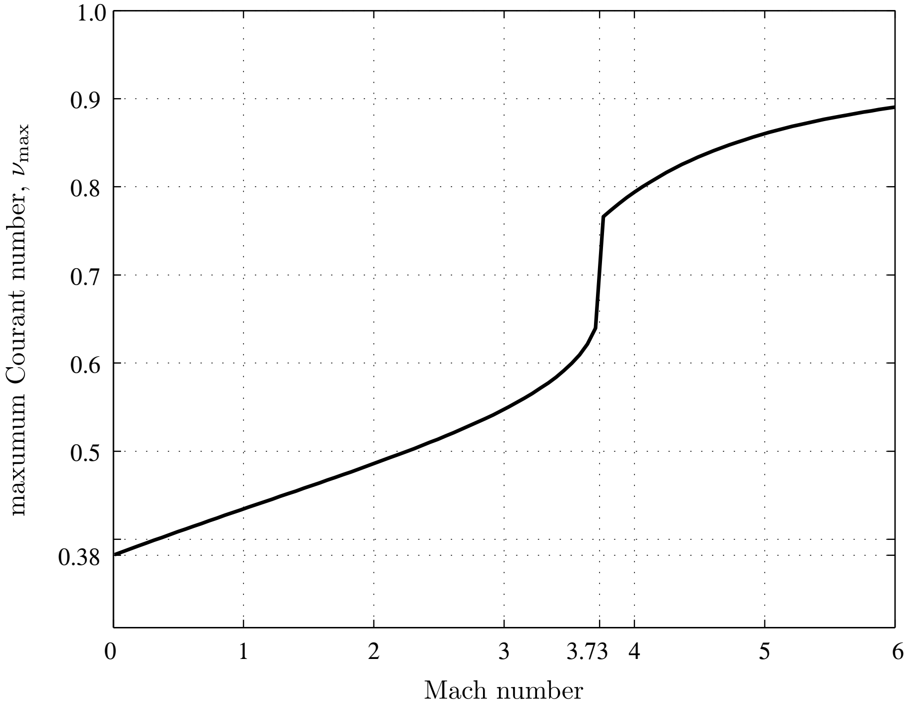


Figure 3.16: The maximum Courant number increases monotonically as the dimensionless equilibrium wave speed for the 10-moment equations increases.

## 3.4 Difference Operators and Their Properties in 2-D

In this section, several discretization methods are applied to the two-dimensional linear advection equation (3.1b), and a Fourier analysis is employed to uncover their dominant dissipation and dispersion errors, order of accuracy, and domain of linear stability. Four methods, HR2–RK2, HR2–Hancock, DG(1)–RK2, and DG(1)–RK3, are compared to the DG(1)–Hancock method. To simplify the analysis, we only consider rectangular grids here; the extension of a Fourier analysis on quadrilateral and triangular grids is shown by Huynh [Huy03] for the upwind (Hancock) and some centered staggered methods.

To write a method in compact form, the difference operators in the $x$-,$y$-directions are defined as follows:

$$\delta_x^+ \mathbf{u}_{i,j} = \mathbf{u}_{i+1,j} - \mathbf{u}_{i,j}, \quad \delta_x^- \mathbf{u}_{i,j} = \mathbf{u}_{i,j} - \mathbf{u}_{i-1,j}, \tag{3.164a}$$

$$\delta_y^+ \mathbf{u}_{i,j} = \mathbf{u}_{i,j+1} - \mathbf{u}_{i,j}, \quad \delta_y^- \mathbf{u}_{i,j} = \mathbf{u}_{i,j} - \mathbf{u}_{i,j-1}. \tag{3.164b}$$

Even though the multi-dimensional problem is considered, Godunov-type and DG methods utilize a one-dimensional Riemann solver. Let $u_E$ and $u_W$ be the point values on either side of a cell face normal to the $x$-direction, and $u_N$ and $u_S$ at the cell faces normal to the $y$-direction. Then, the interface fluxes in the $x$-,$y$-directions are given by

$$f^q_{i+1/2,j}(u_W, u_E) = \frac{r}{2}(u_E + u_W) - \frac{q_x}{2}(u_E - u_W), \tag{3.165a}$$

$$g^q_{i,j+1/2}(u_S, u_N) = \frac{s}{2}(u_N + u_S) - \frac{q_y}{2}(u_N - u_S), \tag{3.165b}$$

where

$$\text{upwind}: \ (q_x, q_y) = (|r|, |s|), \tag{3.166a}$$

$$\text{Rusanov}: \ (q_x, q_y) = (1, 1), \tag{3.166b}$$

$$\text{Lax–Friedrichs}: \ (q_x, q_y) = \left(\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t}\right). \tag{3.166c}$$

Note that the 1-D Lax–Friedrichs flux does not produce the original 2-D Lax–Friedrichs scheme formulated for a staggered grid. In fact, the Lax–Friedrichs flux contains too much dissipation for a 2-D calculation and causes a high-frequency instability, even in a first-order method. We will therefore not apply the Lax–Friedrichs flux beyond one dimension.

The exact solution of (3.1b) for a harmonic mode is

$$\mathbf{u}(x, y, t) = \hat{\mathbf{u}}_0 e^{ik_x(x-rt)} e^{ik_y(y-st)}, \tag{3.167}$$

where $(k_x, k_y)$ are the wave numbers in $(x, y)$. The exact amplification factor is obtained by inserting the time increment, $t + \Delta t$, into the above equation, then insert $\mathbf{u}(x, t)$:

$$\mathbf{u}(x, t + \Delta t) = e^{-i(rk_x + sk_y)\Delta t}\, \mathbf{u}(x, t). \tag{3.168}$$

Hence, the exact amplification factor and exact eigenvalue of the spatial differentiation in the time step, $\Delta t$, are given by

$$g_{\text{exact}} = e^{-i(rk_x + sk_y)\Delta t}, \tag{3.169a}$$

$$\lambda_{\text{exact}} = -i(rk_x + sk_y). \tag{3.169b}$$

Since the wave numbers $(k_x, k_y)$ are related to the spatial frequencies of a wave in the $x$-, and $y$-directions, $(\alpha, \beta)$, such that

$$(k_x, k_y) = \left(\frac{\alpha}{\Delta x}, \frac{\beta}{\Delta y}\right), \tag{3.170}$$

the exact amplification factor and exact eigenvalue of 2-D spatial differentiation can be expressed as

$$g_{\text{exact}}(\tilde{\nu}_x, \tilde{\nu}_y, \alpha, \beta) = e^{-i(r\nu_x\alpha + s\nu_y\beta)}) = e^{-i(\tilde{\nu}_x\alpha + \tilde{\nu}_y\beta)}), \tag{3.171a}$$

$$\lambda_{\text{exact}} = -i\left(\frac{r}{\Delta x}\alpha + \frac{s}{\Delta y}\beta\right). \tag{3.171b}$$

The exact amplification factor in the low-frequency limit is given by expanding in the frequencies of a wave, $(\alpha, \beta)$, at fixed Courant numbers, $(r\nu_x, s\nu_y)$, thus

$$\begin{aligned} g_{\text{exact}}(r\nu_x, s\nu_y, \alpha, \beta) = &\left[1 + (-ir\nu_x)\alpha + \frac{1}{2}(-ir\nu_x)^2\alpha^2 + O\big(\alpha^3\big)\right] \\ &\times\left[1 + (-is\nu_y)\beta + \frac{1}{2}(-is\nu_y)^2\beta^2 + O\big(\beta^3\big)\right]. \end{aligned} \tag{3.172}$$

The asymptotic expansion is further simplified when the wave speeds in the $x$-,$y$-directions are the same, $s = r$, and we also set $\nu_x = \nu_y = \nu$; then

$$g_{\text{exact}}(r\nu, \beta) = 1 + 2(-ir\nu)\beta + 2(-ir\nu)^2\beta^2 + \frac{4}{3}(-ir\nu)^3\beta^3 + \frac{2}{3}(-ir\nu)^4\beta^4 + O\big(\beta^5\big). \tag{3.173}$$

**Accuracy**

The local truncation error of a multi-dimensional method is obtained by straightforward extension of the one-dimensional analysis [CdLBL97, Ram94], described in the previous section. A Taylor series expansion of a space-time eigenvalue, $\tilde{\lambda}_{\text{method}}$, around $k_x = k_y = 0$ leads to a local truncation error:

$$\begin{aligned} \text{LTE}_{\text{method}} &= \tilde{\lambda}_{\text{method}} - \lambda_{\text{exact}} \\ &= \frac{1}{\Delta t}\ln g(r\nu_x, s\nu_y, k_x, k_y) - \lambda_{\text{exact}} \\ &= \sum_{m,n=2}^{\infty} c_{m,n} k_x^m k_y^n, \end{aligned} \tag{3.174}$$

where the coefficient of the modified equation, $c_{m,n}$ is given by

$$\begin{aligned} c_{m,n} &= \frac{1}{m!\ n!} \frac{\partial^{m+n}\tilde{\lambda}}{\partial k_x^m \partial k_y^n}\bigg|_{k_x,k_y=0} \\ &= \frac{1}{m!\ n! \Delta t g(k_x, k_y=0)} \frac{\partial^{m+n} g(k_x,k_y)}{\partial k_x^m \partial k_y^n}\bigg|_{k_x,k_y=0}. \end{aligned} \tag{3.175}$$

In the one-dimensional analysis, the polar form of an amplification factor is obtained first, then a local truncation error is derived. However, in the two-dimensional case, a local truncation error is directly obtained by the above equation due to the complexity of deriving the polar form of the amplification factor.

### Stability

A single Courant number for a two-dimensional problem on rectangular grids can not be uniquely defined; most often used in practice is the definition

$$\begin{aligned} \tilde{\nu}_{\text{2D}} :=& \tilde{\nu}_x + \tilde{\nu}_y \\ =& r\frac{\Delta t}{\Delta x} + s\frac{\Delta t}{\Delta y}. \end{aligned} \tag{3.176}$$

This choice is explained in some detail in Chapter V, as it derives from the more general form (5.13). Following a procedure similar to the one-dimensional analysis, the linear stability conditions of various methods with the upwind flux is obtained numerically.

As a preliminary analysis, the stability domains of the first-order method with the upwind flux is considered. The maximum modulus of the amplification factor $|g_{\text{1st-order}}|_{\max}$ over $\beta \in [0,\pi]$ at the Courant numbers $\tilde{\nu}_x, \tilde{\nu}_y \in [0,1]$ is shown in Figure 3.17. The shaded area represents the stable region: $|g_{\text{1st-order}}| \leq 1$ for any $\beta \in [0,\pi]$. Based on the numerical results, the stability domain for the upwind flux is given by

$$\tilde{\nu}^{\text{upwind}}_{\text{2D, 1st-order}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 1.0. \tag{3.177}$$

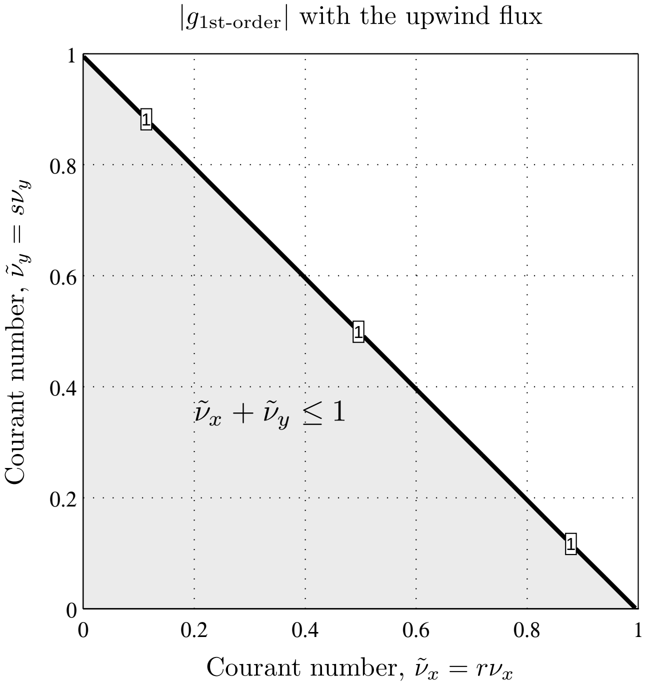


Figure 3.17: Stability domain of the first-order method applied to the 2-D linear advection equation. The shaded area indicates the region where $|g_{\text{first-order}}(\tilde{\nu}_x, \tilde{\nu}_y)| \leq 1$ for $\alpha, \beta \in [0, \pi]$. This shows that the first-order method with the upwind flux is linearly stable for $\tilde{\nu}^{\text{upwind}}_{\text{2D, 1st-order}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 1$.

### 3.4.1 HR–MOL Method

The HR–MOL method on rectangular grids has the following form:

$$\frac{\partial \bar{u}_j(t)}{\partial t} = -\frac{1}{\Delta x}\left(\hat{f}_{i+1/2,j}(t) - \hat{f}_{i-1/2,j}(t)\right) - \frac{1}{\Delta y}\Big(\hat{g}_{i,j+1/2}(t) - \hat{g}_{i,j-1/2}(t)\Big), \quad (3.178)$$

where the interface fluxes are given by the $q$-flux (3.165):

$$\hat{f}(t)_{i+1/2,j} = f^q_{i+1/2,j}\big(u_{i+1/2,j,W}(t), u_{i+1/2,j,E}(t)\big), \quad (3.179a)$$

$$\hat{g}(t)_{i,j+1/2} = g^q_{i,j+1/2}\big(u_{i,j+1/2,S}(t), u_{i,j+1/2,N}(t)\big). \quad (3.179b)$$

The input values for the $q$-flux are obtained by

$$u_{i+1/2,j,W} = \bar{u}_{i,j} + \frac{1}{2}\overline{\Delta_x u_{i,j}}, \quad u_{i+1/2,j,E} = \bar{u}_{i+1,j} - \frac{1}{2}\overline{\Delta_x u_{i+1,j}}, \tag{3.180a}$$

$$u_{i+1/2,j,S} = \bar{u}_{i,j} + \frac{1}{2}\overline{\Delta_y u}_{i,j}, \quad u_{i+1/2,j,N} = \bar{u}_{i,j+1} - \frac{1}{2}\overline{\Delta_y u}_{i,j+1}, \tag{3.180b}$$

where, just as the 1-D analysis, the slopes are obtained by the difference over the domain $(i+1, i, i-1)$ in the $x$-direction, and $(j+1, j, j-1)$ in the $y$-direction respectively:

$$\overline{\Delta_x u}_{i,j} = \frac{1}{2}(\bar{u}_{i+1,j} - \bar{u}_{i-1,j}), \tag{3.181a}$$

$$\overline{\Delta_y u}_{i,j} = \frac{1}{2}(\bar{u}_{i,j+1} - \bar{u}_{i,j-1}). \tag{3.181b}$$

After inserting these difference formulas into the semi-discrete method (3.178), and some algebra, the spatial difference operator of the HR2–MOL becomes

$$\frac{\partial \bar{u}_j(t)}{\partial t} = \mathrm{N}_{\mathrm{HR2}} \bar{u}_j(t), \tag{3.182}$$

where

$$\mathrm{N}_{\mathrm{HR2}} = \sum_{m=1}^{2} \left[ a_{1m}(\delta_x^+)^m + a_{2m}(\delta_x^-)^m + b_{1m}(\delta_y^+)^m + b_{2m}(\delta_y^-)^m \right]. \tag{3.183}$$

Here, the coefficients of the difference operator $\mathbf{N}_{\mathrm{HR2}}$ are given by

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = -\frac{1}{8\Delta x} \begin{pmatrix} 2(2r - q_x) & q_x - r \\ 2(2r + q_x) & q_x + r \end{pmatrix}, \tag{3.184a}$$

$$\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = -\frac{1}{8\Delta y} \begin{pmatrix} 2(2s - q_y) & q_y - s \\ 2(2s + q_y) & q_y + s \end{pmatrix}. \tag{3.184b}$$

**Accuracy** Taking the low-frequency limit of the difference spatial operator $\mathrm{N}_{\mathrm{HR2}}$ leads to the asymptotic eigenvalue:

$$\lambda_{\mathrm{HR2}} = -i\left(\frac{r}{\Delta x}\alpha + \frac{s}{\Delta y}\beta\right) - \frac{i}{12}\left(\frac{r}{\Delta x}\alpha^3 + \frac{s}{\Delta y}\beta^3\right) - \frac{1}{8}\left(\frac{q_x}{\Delta x}\alpha^4 + \frac{q_y}{\Delta y}\beta^4\right) + O\big(\alpha^5, \beta^5\big). \tag{3.185}$$

The order of accuracy in space is obtained by replacing $(\alpha, \beta)$ to the wave numbers, $(k_x, k_y)$, thus

$$\lambda_{\text{HR2}} - \lambda_{\text{exact}} = -\frac{i}{12}\left(r\,\boxed{\Delta x^2}\,k_x^3 + s\,\boxed{\Delta y^2}\,k_y^3\right) + O\big(k_x^4, k_y^4\big)\,. \tag{3.186}$$

Since the HR2–MOL is a semi-discrete method, a suitable time integration method is applied; here, the RK2 method is adopted. Once a time integration method is applied, the overall order of accuracy can be obtained directly by applying the method described in (3.174); then

$$\begin{aligned}
\text{LTE}_{\text{HR2RK2}} &= \tilde{\lambda}_{\text{HR2RK2}} - \lambda_{\text{exact}} \\
&= -\frac{i}{12}\left(r\,\boxed{\Delta x^2}\,k_x^3 + s\,\boxed{\Delta y^2}\,k_y^3\right) \\
&\quad - \frac{i}{6}\left[r^3k_x^3 + s^3k_y^3 + 3rs(rk_x + sk_y)k_xk_y\right]\boxed{\Delta t^2} \\
&\quad + \frac{1}{8}\left[(r^4\Delta t^3 - q_x\Delta x^3)k_x^4 + (s^4\Delta t^3 - q_y\Delta y^3)k_y^4\right] \\
&\quad + \frac{1}{2}rs\Delta t^3(r^2k_x^2 + s^2k_y^2)k_xk_y + \frac{3}{4}r^2s^2\Delta t^3k_x^2k_y^2 + O\big(k_x^4, k_y^4\big)\,.
\end{aligned} \tag{3.187}$$

The above equation shows that the HR2–RK2 method is second-order accurate in space and time. If we further assume a uniform grid, $\Delta x = \Delta y = \Delta h$, and the uniform wave numbers, $k_x = k_y = k$, then the local truncation error becomes

$$\begin{aligned}
\text{LTE}_{\text{HR2RK2}} = -\frac{i}{12}(r+s)&\left[\,\boxed{\Delta h^2} + 2(r+s)^2\,\boxed{\Delta t^2}\,\right]k^3 \\
&- \frac{1}{8}\left[(q_x + q_y)\Delta h^3 - (r+s)^4\Delta t^3\right]k^4 + O\big(k^5\big)\,,
\end{aligned} \tag{3.188}$$

which can be seen as the direct extension of the one-dimensional result shown in (3.96a) on page 116.

**Stability** The stability condition of the two-dimensional HR2–RK2 method with the upwind flux is obtained numerically. The maximum modulus of the amplification factor $|g^{\text{upwind}}_{\text{HR2RK2}}|_{\max}$ over $\beta \in [0, \pi]$ at the Courant numbers $\tilde{\nu}_x, \tilde{\nu}_y \in [0, 1]$ is shown in Figure 3.18. The shaded area represents the stable region: $|g^{\text{upwind}}_{\text{HR2RK2}}| \leq 1$ for any $\beta \in [0, \pi]$. The figure shows that the two-dimensional HR2–RK2 method is stable for

$$\tilde{\nu}^{\text{upwind}}_{\text{2D, HR2RK2}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 1.0. \tag{3.189}$$

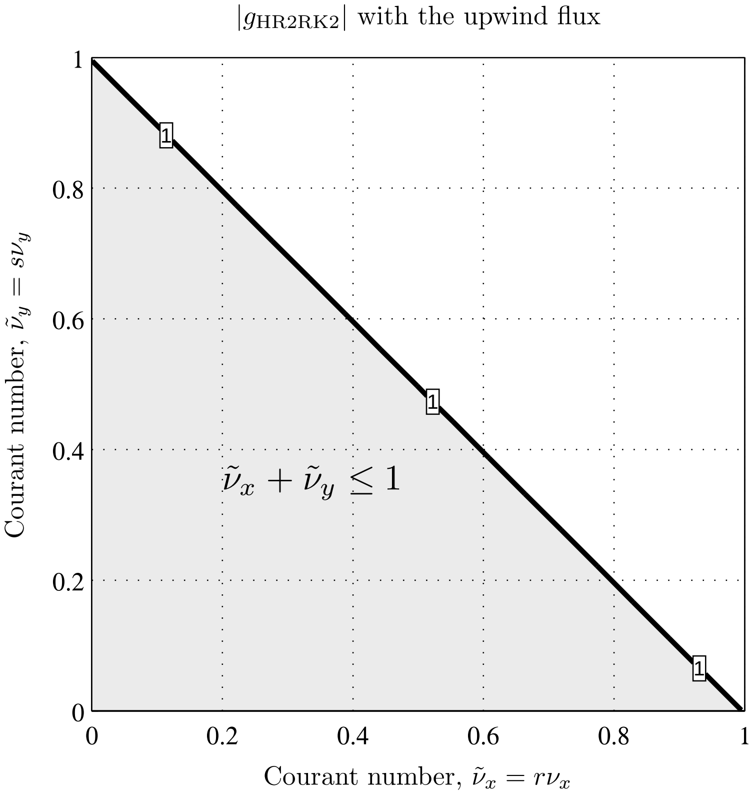


Figure 3.18: Stability domain of the HR2–RK2 method with the upwind flux applied to the 2-D linear advection equation. The shaded area indicates the region where $|g_{\text{HR2RK2}}(\tilde{\nu}_x, \tilde{\nu}_y)| \leq 1$ for $\alpha, \beta \in [0, \pi]$. It shows that the HR2–RK2 method is linearly stable for $\tilde{\nu}^{\text{upwind}}_{\text{2D, HR2RK2}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 1.0$.

### 3.4.2 DG–MOL Method

The DG–MOL method using the $P^1$ Legendre polynomial for both basis and test functions has the following semi-discrete form on the rectangular grids:

$$\frac{\partial \bar{u}_j(t)}{\partial t} = -\frac{1}{\Delta x}\left(\hat{f}_{i+1/2,j}(t) - \hat{f}_{i-1/2,j}(t)\right) - \frac{1}{\Delta y}\left(\hat{g}_{i,j+1/2}(t) - \hat{g}_{i,j-1/2}(t)\right), \tag{3.190a}$$

$$\begin{aligned}\frac{\partial \overline{\Delta_x u}_{i,j}(t)}{\partial t} &= -\frac{6}{\Delta x}\left(\hat{f}_{i+1/2,j}(t) + \hat{f}_{i-1/2,j}(t) - 2r\bar{u}_{i,j}(t)\right)\\ &\quad - \frac{12}{\Delta y}\left[\int_0^1 \left(\xi - \frac{1}{2}\right)\hat{g}_{\xi,j+1/2}(\xi,t)\, d\xi - \int_0^1 \left(\xi - \frac{1}{2}\right)\hat{g}_{\xi,j-1/2}(\xi,t)\, d\xi\right],\end{aligned} \tag{3.190b}$$

$$\begin{aligned}\frac{\partial \overline{\Delta_y u}_{i,j}(t)}{\partial t} &= -\frac{6}{\Delta x}\left(\hat{g}_{i,j+1/2}(t) + \hat{g}_{i,j-1/2}(t) - 2s\bar{u}_{i,j}(t)\right)\\ &\quad - \frac{12}{\Delta x}\left[\int_0^1 \left(\eta - \frac{1}{2}\right)\hat{f}_{i+1/2,\eta}(\eta,t)\, d\eta - \int_0^1 \left(\eta - \frac{1}{2}\right)\hat{f}_{i-1/2,\eta}(\eta,t)\, d\eta\right].\end{aligned} \tag{3.190c}$$

Note that the line integrals of the flux, $\int (\cdot)\, d\xi$ and $\int (\cdot)\, d\eta$, appear in the update formulas of the undivided gradient. Due to the linear variation of the fluxes, $\hat{f}_{i\pm1/2,\eta}$ and $\hat{g}_{\xi,i\pm1/2}$, along cell interfaces, the resulting integrand of the line integral is a quadratic function with respect to $\xi$ or $\eta$. Inserting the $q$-flux (3.165) into the above equations leads to the difference operator of the DG(1)–MOL method such that

$$\frac{\partial \bar{\mathbf{u}}_j(t)}{\partial t} = \mathbf{N}_{\mathrm{DG}(1)}\, \bar{\mathbf{u}}_j(t), \tag{3.191}$$

where

$$\mathbf{N}_{\mathrm{DG}(1)} = \boldsymbol{\mathcal{A}}^+\mathbf{D}_x^+ + \boldsymbol{\mathcal{A}}^-\mathbf{D}_x^- + \boldsymbol{\mathcal{B}}^+\mathbf{D}_y^+ + \boldsymbol{\mathcal{B}}^-\mathbf{D}_y^- + \boldsymbol{\mathcal{C}}. \tag{3.192}$$

Here, the coefficient matrices are given by

$$\boldsymbol{\mathcal{A}}^{+} = \frac{q_x - r}{2\Delta x}\begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ 6 & -3 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}^{-} = \frac{q_x + r}{2\Delta x}\begin{pmatrix} -1 & -\frac{1}{2} & 0 \\ 6 & 3 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \tag{3.193a}$$

$$\boldsymbol{\mathcal{B}}^{+} = \frac{q_y - s}{2\Delta y}\begin{pmatrix} 1 & 0 & -\frac{1}{2} \\ 0 & 1 & 0 \\ 6 & 0 & -3 \end{pmatrix}, \quad \boldsymbol{\mathcal{B}}^{-} = \frac{q_y + s}{2\Delta y}\begin{pmatrix} -1 & 0 & -\frac{1}{2} \\ 0 & -1 & 0 \\ 6 & 0 & 3 \end{pmatrix}, \tag{3.193b}$$

$$\boldsymbol{\mathcal{C}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\frac{6q_x}{\Delta x} & 0 \\ 0 & 0 & -\frac{6q_y}{\Delta y} \end{pmatrix}, \quad \mathbf{D}^{\pm} = \delta^{\pm}\mathbf{I}. \tag{3.193c}$$

**Accuracy** In order to uncover the order of accuracy in space, the asymptotic eigenvalues of the difference operator $\mathbf{N}_{\mathrm{DG}(1)}$ in the low-frequency limit are obtained. Due to the complexity of the formulas, we assume $\alpha = \beta$, then

$$\lambda_{\mathrm{DG}(1)}^{(1)} = -i\left(\frac{r}{\Delta x} + \frac{s}{\Delta y}\right)\beta - \frac{1}{72}\left(\frac{3q_x^2 + r^2}{q_x\Delta x} + \frac{3q_y^2 + s^2}{q_y\Delta y}\right)\beta^4 + O\big(\beta^5\big)\,, \tag{3.194a}$$

$$\lambda_{\mathrm{DG}(1)}^{(2)} = -\frac{6q_x}{\Delta x} + i\left(\frac{3r}{\Delta x} - \frac{s}{\Delta y}\right)\beta + O\big(\beta^2\big)\,, \tag{3.194b}$$

$$\lambda_{\mathrm{DG}(1)}^{(3)} = -\frac{6q_y}{\Delta y} + i\left(\frac{3s}{\Delta y} - \frac{r}{\Delta x}\right)\beta + O\big(\beta^2\big)\,. \tag{3.194c}$$

Furthermore, assuming a uniform grid, $\Delta x = \Delta y = \Delta h$, hence the identical wave numbers $k_x = k_y = k$ in the $x$-,$y$-directions, leads to

$$\lambda_{\mathrm{DG}(1)}^{(1)} - \lambda_{\mathrm{exact}} = -\frac{1}{72}\left[\underbrace{3(q_x + q_y)}_{\text{multi-D error}} + \frac{r^2}{q_x} + \frac{s^2}{q_y}\right]\boxed{\Delta h^3}\; k^4 + O\big(k^5\big)\,, \tag{3.195a}$$

$$\lambda_{\mathrm{DG}(1)}^{(2)} - \lambda_{\mathrm{exact}} = -\frac{6q_x}{\Delta h} + O(k)\,, \tag{3.195b}$$

$$\lambda_{\mathrm{DG}(1)}^{(3)} - \lambda_{\mathrm{exact}} = -\frac{6q_y}{\Delta h} + O(k)\,. \tag{3.195c}$$

The above equations show that the principal root is third-order accurate, and the extraneous roots are damped quickly; the leading errors have negative real values. The high-order accuracy of the DG(1) spatial discretization is preserved for the two-dimensional problem on a rectangular grid. However, compared to the result of the one-dimensional problem (3.106a) on page 121, the extra dissipation error, $-\dfrac{1}{24}\left(q_x + q_y\right)\Delta h^3$, appears in the $O(k^4)$-term. Since this error never disappears even if one-dimensional advection in two-dimensional space (e.g., $r \neq 0, s = 0$) is considered, the error is inherent in the multi-dimensional discretization of a DG method.

To examine the overall accuracy, both RK2 and RK3 method are adopted as the time-integration method. The local truncation error of DG(1)–RK2 and DG(1)–RK3 are obtained directly by (3.174); then

$$\begin{aligned}\mathrm{LTE}_{\mathrm{DG(1)RK2}} &= \tilde{\lambda}^{(1)}_{\mathrm{DG(1)RK2}} - \lambda_{\mathrm{exact}} \\ &= -\frac{i}{6}(r+s)^3 \boxed{\Delta t^2}\, k^3 \\ &\quad - \frac{1}{72}\left[3(q_x+q_y) + \frac{r^2}{q_x} + \frac{s^2}{q_y} - 9(r+s)^4\nu^3\right] \boxed{\Delta h^3}\, k^4 + O\big(k^5\big)\,,\end{aligned} \tag{3.196a}$$

$$\begin{aligned}\mathrm{LTE}_{\mathrm{DG(1)RK3}} &= \tilde{\lambda}^{(1)}_{\mathrm{DG(1)RK3}} - \lambda_{\mathrm{exact}} \\ &= -\frac{1}{72}\left[\left(3(q_x+q_y) + \frac{r^2}{q_x} + \frac{s^2}{q_y}\right)\boxed{\Delta h^3} + 3(r+s)^4\,\boxed{\Delta t^3}\,\right] k^4 + O\big(k^5\big)\,.\end{aligned} \tag{3.196b}$$

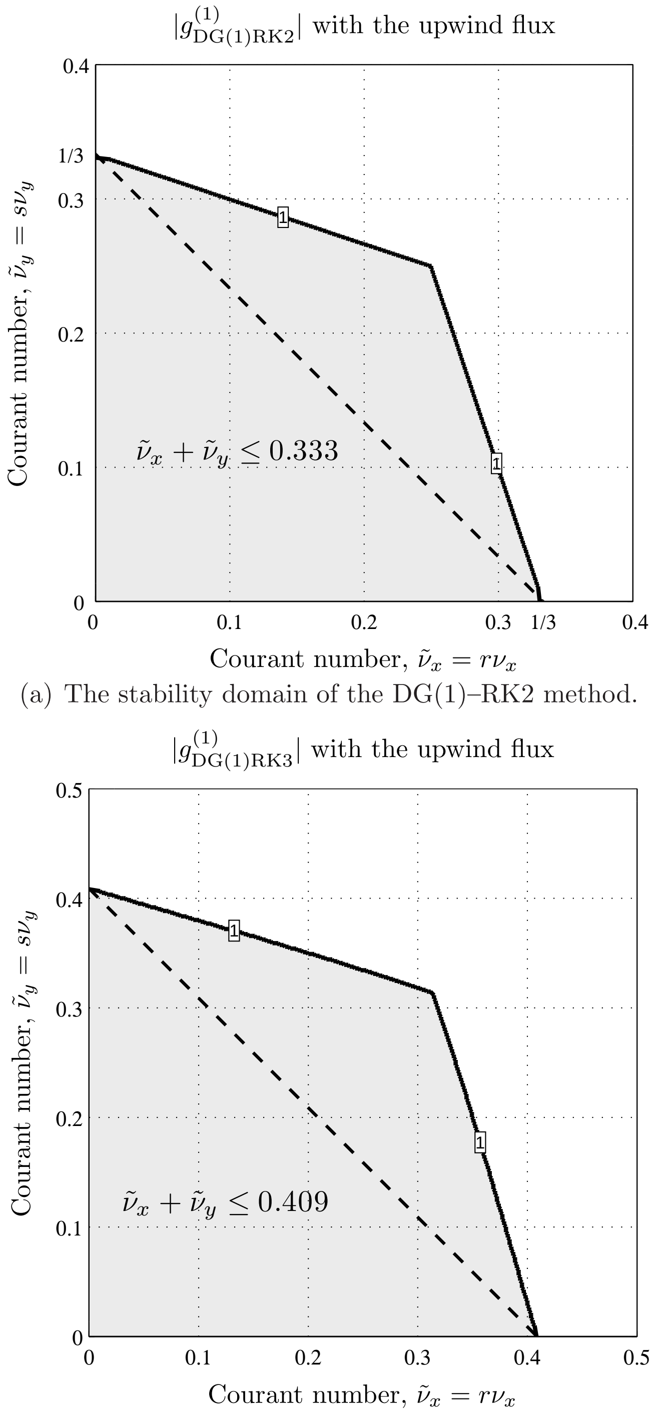
Thus, the DG(1)–RK2 method is third-order accurate in space and second-order in time, just as the method for the one-dimensional problem (3.109a) on page 122. Similarly, the two-dimensional DG(1)–RK3 method is third-order in space and time.

**Stability** The modulus of the amplification factors of both DG(1)–RK2 and DG(1)–RK3 methods with the upwind flux are shown in Figure 3.19. Based on the numerical contour plots of the modulus, the stability condition of each method in terms of $\tilde{\nu}_{\mathrm{2D}}$ is

$$\tilde{\nu}^{\mathrm{upwind}}_{\mathrm{2D,\ DG(1)RK2}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 0.333, \tag{3.197a}$$

$$\tilde{\nu}^{\mathrm{upwind}}_{\mathrm{2D,\ DG(1)RK3}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 0.409, \tag{3.197b}$$

respectively. Thus, adding another stage slightly increases the stability domain, and also the order of accuracy. A drawback is that the RK3 requires an extra Riemann solver to compute at each cell interface.



(a) The stability domain of the DG(1)–RK2 method.

(b) The stability domain of the DG(1)–RK3 method.

Figure 3.19: Stability domain of the DG(1)–RK2/RK3 methods with upwind flux applied to the 2-D linear advection equation. The shaded area indicates the region where $|g_{\mathrm{DG(1)RK2/RK3}}(\tilde{\nu}_x, \tilde{\nu}_y)| \leq 1$ for $\alpha, \beta \in [0, \pi]$. These show that the DG(1)–RK2 method is linearly stable for $\tilde{\nu}^{\mathrm{upwind}}_{\mathrm{2D,\ DG(1)RK2}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 0.333$, and the three-stage RK method slightly increases the stability domain.

### 3.4.3 HR–Hancock Method

The HR–Hancock method on a rectangular grid has the following form:

$$\bar{u}_{i,j}^{n+1} = \bar{u}_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(\hat{f}_{i+1/2,j}^{n+1/2} - \hat{f}_{i-1/2,j}^{n+1/2}\right) - \frac{\Delta t}{\Delta y}\left(\hat{g}_{i+1/2,j}^{n+1/2} - \hat{g}_{i-1/2,j}^{n+1/2}\right), \tag{3.198}$$

where interface fluxes are given by the $q$-flux:

$$\hat{f}_{i+1/2,j}^{n+1/2} = f_{i+1/2,j}^{q}\big(u_{i+1/2,j,W}^{n+1/2}, u_{i+1/2,j,E}^{n+1/2}\big), \tag{3.199a}$$

$$\hat{g}_{i,j+1/2}^{n+1/2} = g_{i,j+1/2}^{q}\big(u_{i,j+1/2,S}^{n+1/2}, u_{i,j+1/2,N}^{n+1/2}\big). \tag{3.199b}$$

The input values of the $q$-flux are obtained by a Taylor series expansion of $u(x, y, t)$ around $(x_i, y_j, t^n)$; then

$$u_{i+1/2,j,E}^{n+k/2} = \bar{u}_{i+1,j}^{n} - \frac{1}{2}\left(1 + r\frac{k\Delta t}{\Delta x}\right)\overline{\Delta_x u}_{i+1,j}^{n} - s\frac{k\Delta t}{\Delta y}\,\overline{\Delta_y u}_{i+1,j}^{n}, \tag{3.200a}$$

$$u_{i+1/2,j,W}^{n+k/2} = \bar{u}_{i,j}^{n} + \frac{1}{2}\left(1 - r\frac{k\Delta t}{\Delta x}\right)\overline{\Delta_x u}_{i,j}^{n} - s\frac{k\Delta t}{\Delta y}\,\overline{\Delta_y u}_{i,j}^{n}, \tag{3.200b}$$

$$u_{i,j+1/2,N}^{n+k/2} = \bar{u}_{i+1,j}^{n} - r\frac{k\Delta t}{\Delta x}\,\overline{\Delta_x u}_{i+1,j}^{n} - \frac{1}{2}\left(1 + s\frac{k\Delta t}{\Delta y}\right)\overline{\Delta_y u}_{i+1,j}^{n}, \tag{3.200c}$$

$$u_{i,j+1/2,S}^{n+k/2} = \bar{u}_{i,j}^{n} - r\frac{k\Delta t}{\Delta x}\,\overline{\Delta_x u}_{i,j}^{n} + \frac{1}{2}\left(1 - s\frac{k\Delta t}{\Delta y}\right)\overline{\Delta_y u}_{i,j}^{n}. \tag{3.200d}$$

Inserting the above equations into the update formula (3.198) leads to the difference operator of the HR2–Hancock method such that

$$u_j^{n+1} = \left(1 + \Delta t\,\mathrm{M}_{\mathrm{Hancock}}\right) u_j^{n}, \tag{3.201}$$

where

$$\mathrm{M}_{\mathrm{Hancock}} = \sum_{m=1}^{2}\left[a_{1m}(\delta_x^+)^m + a_{2m}(\delta_x^-)^m + b_{1m}(\delta_y^+)^m + b_{2m}(\delta_y^-)^m\right] + c_{11}\delta_x^+\delta_y^+ + c_{12}\delta_x^+\delta_y^- + c_{21}\delta_x^-\delta_y^+ + c_{22}\delta_x^-\delta_y^-. \tag{3.202}$$

Here, the coefficients of the difference operator are given by

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = -\frac{1}{8\Delta x}\begin{pmatrix} 2(2r - q_x - r^2\nu_x) & (1 + r\nu_x)(q_x - r) \\ 2(2r + q_x + r^2\nu_x) & (1 - r\nu_x)(q_x + r) \end{pmatrix}, \tag{3.203a}$$

$$\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = -\frac{1}{8\Delta y}\begin{pmatrix} 2(2s - q_y - s^2\nu_y) & (1 + s\nu_y)(q_y - s) \\ 2(2s + q_y + s^2\nu_y) & (1 - s\nu_y)(q_y + s) \end{pmatrix}, \tag{3.203b}$$

$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \frac{\Delta t}{8\Delta x\Delta y}\begin{pmatrix} 2rs - sq_x - rq_y & 2rs - sq_x + rq_y \\ 2rs + sq_x - rq_y & 2rs + sq_x + rq_y \end{pmatrix}. \tag{3.203c}$$

**Accuracy** Taking the low-frequency limit of the difference operator $\mathbf{M}_{\text{HR2Ha}}$ leads to the asymptotic eigenvalue:

$$\begin{aligned}\lambda_{\text{HR2Ha}} = &-i\left(\frac{r}{\Delta x}\alpha + \frac{s}{\Delta y}\beta\right) - \frac{\Delta t}{2}\left[\left(\frac{r}{\Delta x}\right)^2\alpha^2 + \left(\frac{s}{\Delta y}\right)^2\beta^2 + \frac{2rs}{\Delta x\Delta y}\alpha\beta\right] \\ &- \frac{i}{12}\left(\frac{r}{\Delta x}\alpha^3 + \frac{s}{\Delta y}\beta^3\right) + \frac{i\Delta t}{4}\left(\frac{r\alpha}{\Delta x} + \frac{s\beta}{\Delta y}\right)\left(\frac{q_x\alpha^2}{\Delta x} + \frac{q_y\beta^2}{\Delta y}\right) + O\big(\alpha^4, \beta^4\big).\end{aligned} \tag{3.204}$$

The order of accuracy in space is obtained by replacing $(\alpha, \beta)$ to the wave numbers, $(k_x, k_y)$, and let $\Delta t \to 0$, then

$$\lambda_{\text{HR2Ha}} - \lambda_{\text{exact}} = -\frac{i}{12}\left(r\,\boxed{\Delta x^2}\,k_x^3 + s\,\boxed{\Delta y^2}\,k_y^3\right) + O\big(k_x^4, k_y^4\big). \tag{3.205}$$

The above equation shows that the HR2–Hancock method is second-order in space. The overall order of accuracy can be obtained by adopting the forward Euler method as the time integrator; then the local truncation error becomes

$$\begin{aligned}\mathrm{LTE}_{\mathrm{HR2Ha}} &= \tilde{\lambda}_{\mathrm{HR2Ha}} - \lambda_{\mathrm{exact}} \\ &= -\frac{i}{12}\left(r\,\boxed{\Delta x^2}\,k_x^3 + s\,\boxed{\Delta y^2}\,k_y^3\right) - \frac{i}{6}\left(rk_x + sk_y\right)^3\,\boxed{\Delta t^2} \\ &\quad + \frac{i}{4}(rk_x + sk_y)\left(q_x\,\boxed{\Delta x\Delta t}\,k_x^2 + q_y\,\boxed{\Delta y\Delta t}\,k_y^2\right) \\ &\quad + \frac{1}{8}\left[r^4\Delta t^3 - q_x\Delta x^3 + 2r^2\Delta x\Delta t(\Delta x - q_x\Delta t)\right]k_x^4 \\ &\quad + \frac{1}{8}\left[s^4\Delta t^3 - q_y\Delta y^3 + 2s^2\Delta y\Delta t(\Delta y - q_y\Delta t)\right]k_y^4 \\ &\quad + \frac{1}{2}rs\Delta t\left(r^2\Delta t^2 - q_x\Delta x\Delta t + 2\Delta x^2\right)k_x^3k_y \\ &\quad + \frac{1}{2}rs\Delta t\left(s^2\Delta t^2 - q_y\Delta y\Delta t + 2\Delta y^2\right)k_xk_y^3 \\ &\quad + \frac{1}{4}\Delta t^2\left(-s^2q_x\Delta x - r^2q_y\Delta y + 3(rs)^2\Delta t\right)k_x^2k_y^2 + O\left(k_x^4, k_y^4\right). \end{aligned} \tag{3.206}$$

Thus, the HR2–Hancock method is second-order accurate in space and time. If we further assume a uniform grid, $\Delta x = \Delta y = \Delta h$, and uniform wave numbers, $k_x = k_y = k$, then the above equation is further simplified:

$$\mathrm{LTE}_{\mathrm{HR2Ha}} = -\frac{i}{12}(r+s)\left[\,\boxed{\Delta h^2} - 3(q_x + q_y)\,\boxed{\Delta h\Delta t} + 2(r+s)^2\,\boxed{\Delta t^2}\,\right]k^3 + O\left(k^4\right), \tag{3.207}$$

which is identical to the local truncation error of the HR2–Hancock method applied to the one-dimensional problem (3.122) on page 130, if we form 2-D parameters by summation of the wave speeds and the dissipation coefficients.

**Stability** The stability domain of the HR2–Hancock with the upwind flux is shown in Figure 3.20. The shaded area indicates the region where $|g_{\mathrm{HR2Ha}}(\tilde{\nu}_x, \tilde{\nu}_y)| \leq 1$ for any $\alpha, \beta \in [0, \pi]$. As we expected, the two-dimensional HR2–Hancock method is stable for

$$\tilde{\nu}^{\mathrm{upwind}}_{\mathrm{2D,\ HR2Hancock}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 1. \tag{3.208}$$

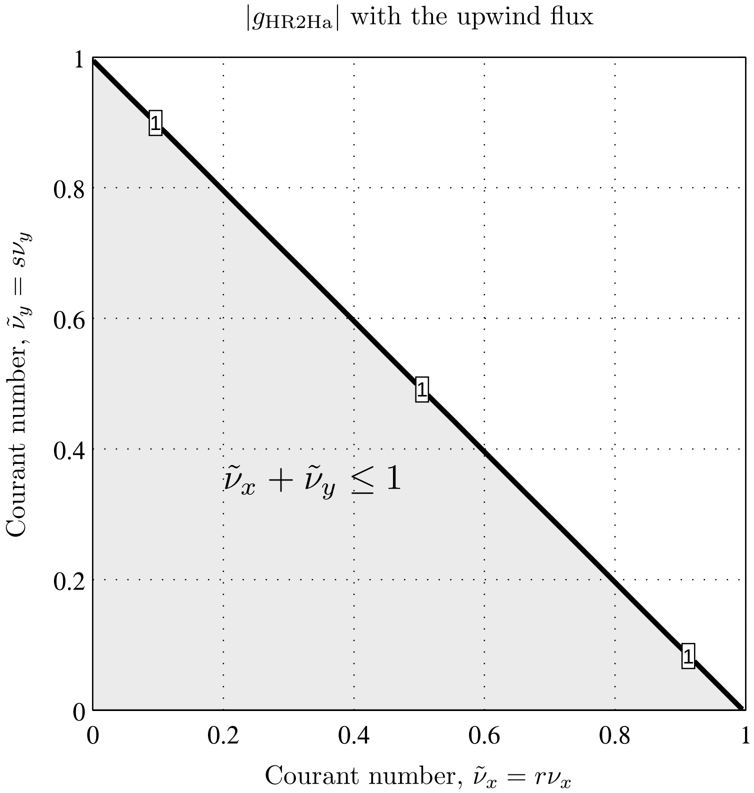


Figure 3.20: Stability domain of the HR2–Hancock method with upwind flux applied to the 2-D linear advection equation. The shaded area indicates the region where $|g_{\mathrm{HR2Ha}}(\tilde{\nu}_x, \tilde{\nu}_y)| \leq 1$ for $\alpha, \beta \in [0, \pi]$. It shows that the HR2–Hancock method is linearly stable for $\tilde{\nu}^{\mathrm{upwind}}_{\mathrm{2D,\ HR2Ha}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 1$.

### 3.4.4 DG–Hancock Method

The DG(1)–Hancock method on a rectangular grid has the following update formulas:

$$\bar{u}_{i,j}^{n+1} = \bar{u}_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(\hat{f}_{i+1/2,j}^{n+1/2} - \hat{f}_{i-1/2,j}^{n+1/2}\right) - \frac{\Delta t}{\Delta y}\left(\hat{g}_{i+1/2,j}^{n+1/2} - \hat{g}_{i-1/2,j}^{n+1/2}\right), \tag{3.209a}$$

$$\begin{aligned}\overline{\Delta_x u}_{i,j}^{n+1} &= \overline{\Delta_x u}_{i,j}^{n} - \frac{\Delta t}{\Delta x}6\left(\hat{f}_{i+1/2,j}^{n+1/2} + \hat{f}_{i-1/2,j}^{n+1/2} - 2r\breve{u}_{i,j}\right)\\ &\quad - \frac{\Delta t}{\Delta y}12\left[\int\limits_0^1 \left(\xi - \frac{1}{2}\right)\hat{g}_{\xi,j+1/2}^{n+1/2}(\xi)\ d\xi - \int\limits_0^1 \left(\xi - \frac{1}{2}\right)\hat{g}_{\xi,j-1/2}^{n+1/2}(\xi)\ d\xi\right],\end{aligned} \tag{3.209b}$$

$$\begin{aligned}\overline{\Delta_y u}_{i,j}^{n+1} &= \overline{\Delta_y u}_{i,j}^{n} - \frac{\Delta t}{\Delta y}6\left(\hat{g}_{i,j+1/2}^{n+1/2} + \hat{g}_{i,j-1/2}^{n+1/2} - 2s\breve{u}_{i,j}\right)\\ &\quad - \frac{\Delta t}{\Delta x}12\left[\int\limits_0^1 \left(\eta - \frac{1}{2}\right)\hat{f}_{i+1/2,\eta}^{n+1/2}(\eta)\ d\eta - \int\limits_0^1 \left(\eta - \frac{1}{2}\right)\hat{f}_{i-1/2,\eta}^{n+1/2}(\eta)\ d\eta\right],\end{aligned} \tag{3.209c}$$

where the fluxes in both coordinate directions are given by the $q$-flux (3.199). The volume integral of the flux simplifies owing to flux linearity, and quadrature is only required in time. Again, both Gauss–Lobatto and Gauss–Radau quadratures in time:

*3-point Gauss–Lobatto*

$$\breve{u}_j = \frac{1}{6}(\bar{u}_j^n + 4\bar{u}_j^{n+1/2} + \bar{u}_j^{n+1}), \tag{3.210}$$

where

$$\bar{u}_j^{n+1/2} = \bar{u}_j^n - \frac{\Delta t}{2}\frac{1}{\Delta x}\left(\hat{f}_{j+1/2}^{n+1/4} - \hat{f}_{j-1/2}^{n+1/4}\right), \tag{3.211}$$

*2-point Gauss–Radau*

$$\breve{u}_j = \frac{1}{4}(3\bar{u}_j^{n+1/3} + \bar{u}_j^{n+1}), \tag{3.212}$$

where

$$\bar{u}_j^{n+1/3} = \bar{u}_j^n - \frac{\Delta t}{3}\frac{1}{\Delta x}\left(\hat{f}_{j+1/2}^{n+1/6} - \hat{f}_{j-1/2}^{n+1/6}\right); \tag{3.213}$$

for $\breve{u}_{i,j}$ these lead to identical final update formulas. After inserting the difference form of the volume integral of the fluxes and the $q$-flux, the difference operator has the form

$$\mathbf{u}_j^{n+1} = \big(\mathbf{I} + \Delta t\,\mathbf{M}_{\mathrm{DG(1)Ha}}\big)\,\mathbf{u}_j^n, \tag{3.214}$$

where

$$\mathbf{M}_{\mathrm{DG(1)Ha}} = \boldsymbol{\mathcal{A}}^{+}\mathbf{D}_x^{+} + \boldsymbol{\mathcal{A}}^{-}\mathbf{D}_x^{-} + \boldsymbol{\mathcal{B}}^{+}\mathbf{D}_y^{+} + \boldsymbol{\mathcal{B}}^{-}\mathbf{D}_y^{-} + \boldsymbol{\mathcal{C}}, \tag{3.215}$$

with

$$\mathbf{D}_x^{\pm} = \delta_x^{\pm}\mathbf{I}, \quad \mathbf{D}_y^{\pm} = \delta_y^{\pm}\mathbf{I}, \tag{3.216}$$

and the coefficient matrices are given by

$$\boldsymbol{\mathcal{A}}^{+} = \frac{q_x - r}{2\Delta x}\begin{pmatrix} 1 & -\frac{1}{2}(1 + r\nu_x) & -\frac{s\nu_y}{2} \\ 6(1 + r\nu_x) & -3 - 6r\nu_x - 2(r\nu_x)^2 & -(3 + 2r\nu_x)s\nu_y \\ 6s\nu_y & -(3 + 2r\nu_x)s\nu_y & 1 - 2(s\nu_y)^2 \end{pmatrix}, \tag{3.217a}$$

$$\boldsymbol{\mathcal{A}}^{-} = \frac{q_x + r}{2\Delta x}\begin{pmatrix} -1 & -\frac{1}{2}(1 - r\nu_x) & \frac{s\nu_y}{2} \\ 6(1 - r\nu_x) & 3 - 6r\nu_x + 2(r\nu_x)^2 & -(3 - 2r\nu_x)s\nu_y \\ -6s\nu_y & -(3 - 2r\nu_x)s\nu_y & -1 + 2(s\nu_y)^2 \end{pmatrix}, \tag{3.217b}$$

$$\boldsymbol{\mathcal{B}}^{+} = \frac{q_y - s}{2\Delta y}\begin{pmatrix} 1 & -\frac{r\nu_x}{2} & -\frac{1}{2}(1 + s\nu_y) \\ 6r\nu_x & 1 - 2(r\nu_x)^2 & -(3 + 2s\nu_y)r\nu_x \\ 6(1 + s\nu_y) & -(3 + 2s\nu_y)r\nu_x & -3 - 6s\nu_y - 2(s\nu_y)^2 \end{pmatrix}, \tag{3.217c}$$

$$\boldsymbol{\mathcal{B}}^{-} = \frac{q_y + s}{2\Delta y}\begin{pmatrix} -1 & \frac{r\nu_x}{2} & -\frac{1}{2}(1 - s\nu_y) \\ -6r\nu_x & -1 + 2(r\nu_x)^2 & -(3 - 2s\nu_y)r\nu_x \\ 6(1 - s\nu_y) & -(3 - 2s\nu_y)r\nu_x & 3 - 6s\nu_y + 2(s\nu_y)^2 \end{pmatrix}, \tag{3.217d}$$

$$\boldsymbol{\mathcal{C}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\dfrac{6r}{\Delta x}\left(\dfrac{q_x}{r} - r\nu_x\right) & \dfrac{6s}{\Delta y} r\nu_x \\ 0 & \dfrac{6r}{\Delta x} s\nu_y & -\dfrac{6s}{\Delta y}\left(\dfrac{q_y}{s} - s\nu_y\right) \end{pmatrix} . \tag{3.217e}$$

**Accuracy** In view of the lengthy formula, let us assume the wave frequencies in the $x$- and $y$-directions are the same, thus $\alpha = \beta$. Furthermore, in the $O(\beta^4)$-term, a square mesh, $\Delta x = \Delta y = \Delta h$, is assumed. Under these assumptions, the asymptotic eigenvalues based on the upwind flux, $(q_x, q_y) = (r, s)$, become

$$\begin{aligned} \lambda^{(1)}_{\mathrm{DG(1)Ha}} = & -i\left(\frac{r}{\Delta x} + \frac{s}{\Delta y}\right)\beta - \frac{1}{2}\left(\frac{r}{\Delta x} + \frac{s}{\Delta y}\right)^2 \Delta t \beta^2 + \frac{i}{6}\left(\frac{r}{\Delta x} + \frac{s}{\Delta y}\right)^3 \Delta t^2 \beta^3 \\ & - \frac{r+s}{72\Delta h}\left[4 - 5(r+s)\frac{\Delta t}{\Delta h} + 2(r+s)^2\left(\frac{\Delta t}{\Delta h}\right)^2 - 4(r+s)^3\left(\frac{\Delta t}{\Delta h}\right)^3\right]\beta^4 \\ & + O\big(\beta^5\big) , \end{aligned} \tag{3.218a}$$

$$\begin{aligned} \lambda^{(2),(3)}_{\mathrm{DG(1)Ha}} = & -\frac{3}{\Delta h}\left[(r+s) - \nu(r^2+s^2) \pm \sqrt{(r-s)^2\big(1 - 2(r+s)\nu\big) + (r^2+s^2)^2\nu^2}\right] \\ & + O(\beta) . \end{aligned} \tag{3.218b}$$

Replacing the wave frequency by the wave number, $k = \dfrac{\beta}{\Delta h}$, and letting $\Delta t \to 0$ brings out the spatial order of accuracy:

$$\lambda^{(1),\mathrm{upwind}}_{\mathrm{DG(1)Ha}} - \lambda_{\mathrm{exact}} = -\frac{r+s}{18}\boxed{\Delta h^3}\, k^4 + O\big(k^5\big) , \tag{3.219a}$$

$$\lambda^{(2),\mathrm{upwind}}_{\mathrm{DG(1)Ha}} - \lambda_{\mathrm{exact}} = -\frac{6r}{\Delta h} + O(k) , \tag{3.219b}$$

$$\lambda^{(3),\mathrm{upwind}}_{\mathrm{DG(1)Ha}} - \lambda_{\mathrm{exact}} = -\frac{6s}{\Delta h} + O(k) , \tag{3.219c}$$

thus, the spatial discretization is third-order accurate. Comparing the dominant dissipation error (3.219a) to the error obtained in the one-dimensional case, (3.137a) on page 135, the multi-dimensionality increases the dissipation by a factor 4. This multi-dimensional error originates with the line integral of the flux along the cell interfaces, the last term in (3.209b) and (3.209c).

When the Lax–Friedrichs flux $\left(q_x = q_y = q = \dfrac{\Delta h}{\Delta t} \text{ in the } O(\beta^4) \text{ term}\right)$ is adopted, the asymptotic eigenvalues become

$$\begin{aligned}\lambda^{(1),\mathrm{LxF}}_{\mathrm{DG(1)Ha}} = {} & -i\left(\frac{r}{\Delta x} + \frac{s}{\Delta y}\right)\beta - \frac{1}{2}\left(\frac{r}{\Delta x} + \frac{s}{\Delta y}\right)^2 \Delta t\beta^2 + \frac{i}{6}\left(\frac{r}{\Delta x} + \frac{s}{\Delta y}\right)^3 \Delta t^2\beta^3 \\ & - \frac{r+s}{72[(r^2+s^2)\Delta t - q\Delta h]}\Bigg[(6q^2 + r^2 + s^2) - 12q(r^2 + rs + s^2)\left(\frac{\Delta t}{\Delta x}\right) \\ & + 2(r+s)^2\big(3q^2 + 2(r^2+s^2)\big)\left(\frac{\Delta t}{\Delta x}\right)^2 - 3q(r+s)^2(3r^2 + 2rs + 3s^2)\left(\frac{\Delta t}{\Delta x}\right)^3 \\ & + 4(r+s)^4(r^2+s^2)\left(\frac{\Delta t}{\Delta x}\right)^4\Bigg]\beta^4 + O\big(\beta^5\big)\,,\end{aligned} \tag{3.220a}$$

$$\lambda^{(2),\mathrm{LxF}}_{\mathrm{DG(1)Ha}} = -\frac{6q}{\Delta h} + O(\beta)\,, \tag{3.220b}$$

$$\lambda^{(3),\mathrm{LxF}}_{\mathrm{DG(1)Ha}} = -\frac{6}{\Delta h}\left[q - (r^2+s^2)\nu\right] + O(\beta)\,. \tag{3.220c}$$

As in the one-dimensional case, the Lax–Friedrichs flux, $q = \dfrac{\Delta h}{\Delta t}$, leads to the constant leading error $-\dfrac{6q}{\Delta h} = -6$ in $\lambda^{(2),\mathrm{LxF}}_{\mathrm{DG(1)Ha}}$, thus the method becomes unconditionally unstable.

The overall order of accuracy for the scheme with the upwind flux becomes

$$\mathrm{LTE}^{(1),\mathrm{upwind}}_{\mathrm{DG(1)Ha}} = -\frac{r+s}{72}\left[4 - 5(r+s)\nu + 2(r+s)^2\nu^2 - (r+s)^3\nu^3\right]\boxed{\Delta h^3}\,k^4 + O\big(k^5\big)\,, \tag{3.221}$$

thus the accurate eigenmode of the two-dimensional DG(1)–Hancock method is third-order in space and time.

**Stability** The unity contour of the amplification-factor modulus $|g_{\mathrm{DG(1)Ha}}|$ with upwind flux is shown in Figure 3.21. The shaded area indicates the region where

$|g_{\mathrm{DG(1)Ha}}(\tilde{\nu}_x, \tilde{\nu}_y)| \leq 1$ for any $\alpha, \beta \in [0, \pi]$. The numerical result shows that a sufficient condition for the two-dimensional DG(1)–Hancock to be stable is

$$\tilde{\nu}^{\mathrm{upwind}}_{\mathrm{2D,\ DG(1)Ha}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 0.664. \tag{3.222}$$

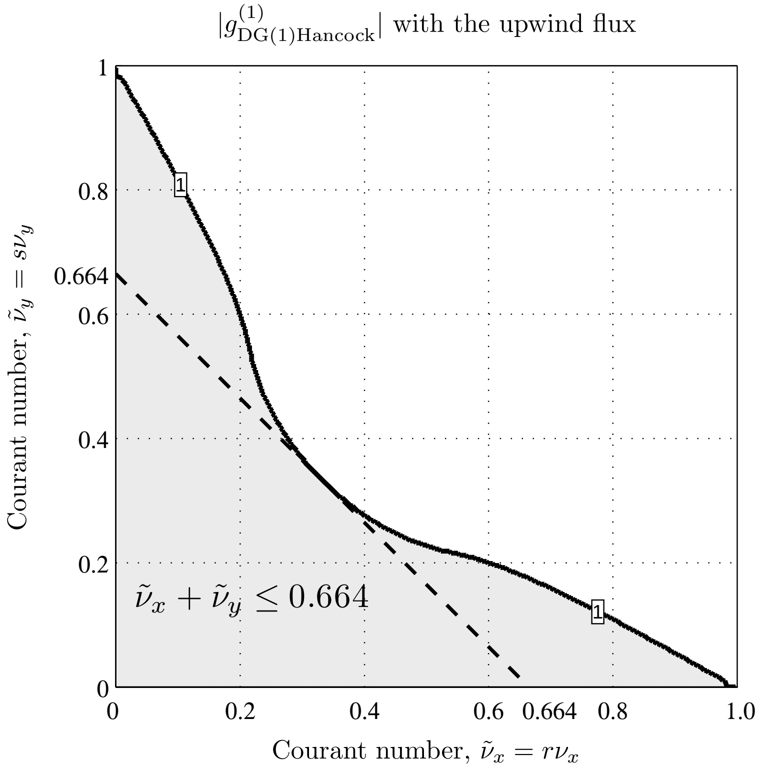


Figure 3.21: Stability domain of the DG(1)–Hancock method with upwind flux applied to the 2-D linear advection equation. The shaded area indicates the region where $|g_{\mathrm{DG(1)Ha}}(\tilde{\nu}_x, \tilde{\nu}_y)| \leq 1$ for $\alpha, \beta \in [0, \pi]$. It shows that the DG(1)–Hancock method is linearly stable for $\tilde{\nu}^{\mathrm{upwind}}_{\mathrm{2D,\ DG(1)Ha}} = \tilde{\nu}_x + \tilde{\nu}_y \leq 0.664$.

### 3.4.5 Dominant Dissipation/Dispersion Error and Stability in 2-D

The results of a Fourier analysis for each method are listed below for comparison:

$$\begin{aligned}
\mathrm{LTE}_{\mathrm{HR2RK2}} &= c_3\left[1+2(r+s)^2\nu^2\right]k^3 \\
&+ \;\; c_4\left[\frac{q_x+q_y}{r+s}-(r+s)^3\nu^3\right]k^4,
\end{aligned} \tag{3.223a}$$

$$\begin{aligned}
\mathrm{LTE}^{\mathrm{upwind}}_{\mathrm{HR2Ha}} &= c_3\left[1-3(r+s)\nu+2(r+s)^2\nu^2\right]k^3 \\
&+ \;\; c_4\left[1-2(r+s)\nu+2(r+s)^2\nu^2-(r+s)^3\nu^3\right]k^4,
\end{aligned} \tag{3.223b}$$

$$\begin{aligned}
\mathrm{LTE}^{(1)}_{\mathrm{DG(1)RK2}} &= c_3\left[2(r+s)^2\nu^2\right]k^3 \\
&+ \frac{1}{9}c_4\left[\frac{3(q_x+q_y)}{r+s}+\frac{1}{r+s}\left(\frac{r^2}{q_x}+\frac{s^2}{q_y}\right)-9(r+s)^3\nu^3\right]k^4,
\end{aligned} \tag{3.223c}$$

$$\mathrm{LTE}^{(1)}_{\mathrm{DG(1)RK3}} = \frac{1}{9}c_4\left[\frac{3(q_x+q_y)}{r+s}+\frac{1}{r+s}\left(\frac{r^2}{q_x}+\frac{s^2}{q_y}\right)+3(r+s)^3\nu^3\right]k^4, \tag{3.223d}$$

$$\mathrm{LTE}^{(1),\mathrm{upwind}}_{\mathrm{DG(1)Ha}} = \frac{1}{9}c_4\left[4-5(r+s)\nu+2(r+s)^2\nu^2-(r+s)^3\nu^3\right]k^4, \tag{3.223e}$$

where

$$c_3 = -\frac{i\,(r+s)}{12}\boxed{\Delta h^2}, \quad c_4 = -\frac{r+s}{8}\boxed{\Delta h^3}. \tag{3.224}$$

The local truncation errors show the dominant dispersion error $\big(O(k^3)\text{-term}\big)$, and the dissipation error, $\big(O(k^4)\text{-term}\big)$. Compared to the one-dimensional results (3.159) on page 149, the leading dissipation error of a two-dimensional DG(1) method increases by a factor 4 due to the multi-dimensionality. In contrast, an HR2 method possesses the same amount of dispersion and dissipation for both one- and two-dimensional discretizations. The DG(1)–Hancock and DG(1)–RK3 methods are superior with a leading error $O(\Delta h^3)$; the rest of the methods have error $O(\Delta h^2)$.

The stability limits for the upwind and Lax–Friedrichs fluxes are also summarized in Table 3.7. The two-dimensional DG(1)–Hancock method is stable for $\tilde{\nu}_{2\mathrm{D}} \leq 0.664$, more restrictive than in one dimension, and also more restrictive than

| | method | order | maximum Courant number:<br>$(\tilde{\nu}_{2\mathrm{D}})_{\max} := (\tilde{\nu}_x + \tilde{\nu}_y)_{\max}$ |
|---|---|---|---|
| semi-discrete | HR1–RK1 | 1 | 1.0 |
| | HR2–RK2 | 2 | 1.0 |
| | DG(1)–RK2 | 2 | 0.333 |
| | DG(1)–RK3 | 3 | 0.409 |
| fully discrete | HR2–Hancock | 2 | 1.0 |
| | DG(1)–Hancock | 3 | $0.664 \simeq 2/3$ |

Table 3.7: Maximum 2-D Courant number, $(\tilde{\nu}_{2\mathrm{D}})_{\max} := (\tilde{\nu}_x + \tilde{\nu}_y)_{\max}$, for various methods combined with the upwind flux $(q_x, q_y) = (r, s)$ are applied to the 2-D linear advection equation. The stability domain of DG(1)–Hancock reduces to $(\tilde{\nu}_{2\mathrm{D}})_{\max} = 0.664$ in two dimensions, yet greater than for DG(1)–RK2/RK3.

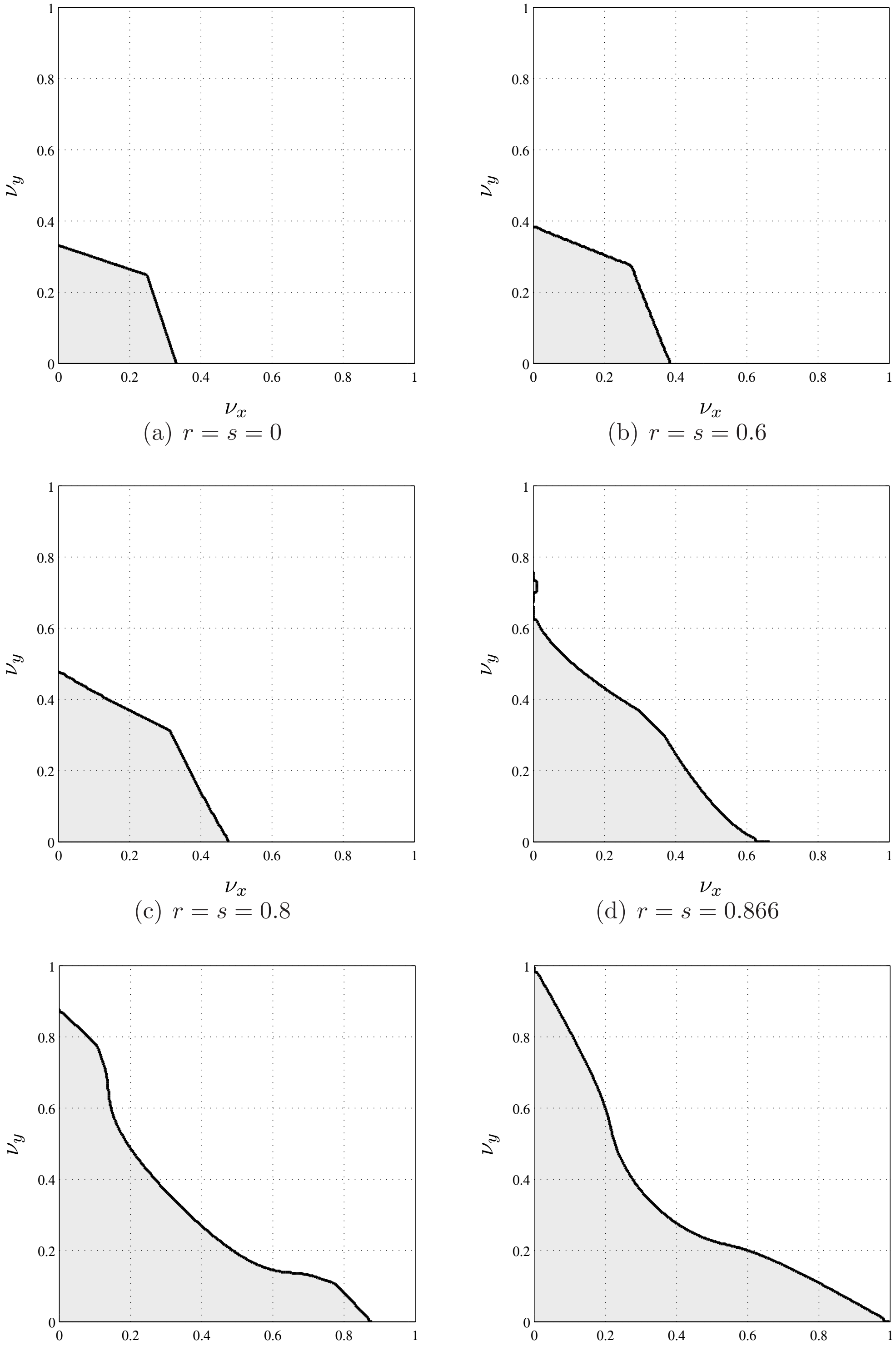
for the two-dimensional HR2–RK2/Hancock method ($\tilde{\nu}_{2\mathrm{D}} \leq 1.0$), but still 50% less restrictive than for the two-dimensional DG(1)–RK2 method ($\tilde{\nu}_{2\mathrm{D}} \leq 0.333$).

Relating to the stability limit for two-dimensional problem, Huynh recently found a factor $\dfrac{1}{\sqrt{2}}$ reduction for a high-order method if tensor-product basis functions are adopted [Huy07, p. 3]. Note that, on a rectangular grid, a tensor-product basis of $P^1$ functions has four degrees of freedom, whereas our minimal $P^1$ basis has three degrees of freedom. Since our analysis is restricted to rectangular grids, further reduction of the stability domain for two-dimensional problems is expected when quadrilateral grids are considered.

### 3.4.6 Stability of Methods with the Rusanov Flux

Similar to the 1-D case, the DG(1)–Hancock method with the Rusanov flux reduces its stability domain as equilibrium wave speeds decrease. The upper bound is obtained by the wave speeds $(r, s) = (1, 1)$, just as for the method with the upwind flux: $(\nu_{2\mathrm{D}})_{\max} = 0.664$. Conversely, the lower bound is identical to that of the DG(1)–RK2 method: $(\nu_{2\mathrm{D}})_{\max} = 0.333$. Owing to the fact that the stability domain is convex while $r, s \leq 0.664$ (see Figure 3.22), the stability condition deduced in the 1-D analysis (Figure 3.15 on page 157) is still applicable. Here, even though we obtain the maximum 2-D Courant number numerically as 0.664, it is more plausible to set it to $\dfrac{2}{3} \simeq 0.666$ owing to the analogy with the 1-D analysis. Finally, let $v := \min(r, s)$, then the maximum Courant number of the 2-D DG(1)–Hancock method on a rectangular grid is given by

$$(\tilde{\nu}_{2\mathrm{D}})_{\max} := (\tilde{\nu}_x + \tilde{\nu}_y)_{\max} \approx \begin{cases} \dfrac{1}{3} + \dfrac{1}{9}v^2 + \dfrac{2}{27}v^4 & \text{if } \; 0 \leq v \leq \dfrac{\sqrt{3}}{2}, \\ \dfrac{2}{3} & \text{if } \; \dfrac{\sqrt{3}}{2} < v \leq 1. \end{cases} \tag{3.225}$$



(a) $r = s = 0$

(b) $r = s = 0.6$

(c) $r = s = 0.8$

(d) $r = s = 0.866$

(e) $r = s = 0.9$

(f) $r = s = 1.0$

Figure 3.22: The stability domains of the 2-D DG(1)–Hancock method with the Rusanov flux are presented. Similar to the 1-D case, the stability domain reduces as wave speeds $(r, s)$ decrease. Note that the Courant numbers in the $x$-,$y$-directions are defined by $\nu_x := \Delta t/\Delta x$ and $\nu_y := \Delta t/\Delta y$ respectively.

## 3.5 Grid Convergence Study in 1-D

To confirm the previous analysis and demonstrate the efficiency of the DG(1)–Hancock method, the 1-D linear advection equation,

$$\partial_t u + r\partial_x u = 0 \quad \text{with} \quad r = \frac{1}{2}, \quad u(x,0) = \cos(2\pi x), \tag{3.226}$$

is solved over the domain $x \in [0,1]$ with periodic boundary conditions. The numerical solution at $t_{\text{end}} = 300$, after the harmonic wave has propagated 150 times across the domain, is compared to the exact solution. The upwind flux is used to compute a cell-interface flux, and we set the Courant number for each method equal to 90% of the method's linear stability limit listed in Table 3.3;

$$\nu_{\text{HR2–RK2}} = \nu_{\text{HR2–Hancock}} = \nu_{\text{DG(1)–Hancock}} = 0.9, \tag{3.227a}$$

$$\nu_{\text{DG(1)–RK2}} = 0.3, \quad \nu_{\text{DG(1)–RK3}} = 0.37, \quad \nu_{\text{DG(2)–RK3}} = 0.19. \tag{3.227b}$$

The $L_2$-, $L_\infty$-errors and CPU time of each method are listed in Table 3.8. At first, to assess the dispersion and dissipation errors of methods qualitatively, the numerical solution by DG(1)–Hancock at $t_{\text{end}}$ is plotted together with solutions by three other methods: HR2–RK2, HR2–Hancock, and DG(1)–RK2. The grid used for the DG methods is twice as coarse ($N = 20$) as for the HR methods ($N = 40$), because the former use two independent date per mesh. The numerical results superimposed on the exact solution are shown in Figure 3.23. The DG(1)–Hancock method produces the least dispersive/dissipative result among the four methods. As shown in the local truncation errors (3.159) on page 149, the leading error of the HR2–RK2, DG(1)–RK2, and HR2–Hancock methods is dispersive, caused by the $O(k^3)$-term. Thus, a traveling wave solution suffers especially with HR2–RK2 and DG(1)–RK2. However, HR2–Hancock has surprisingly little dispersive/dissipative error, which can be understood from the shift condition shown at (3.118) on page 129 which the Hancock method would satisfy at $\tilde{\nu} = 1$.

Secondly, a grid convergence study is conducted, with results shown in Figure 3.24. The $L_2$-norms of solution errors are plotted against the number of degrees of freedom (solution parameters). It is seen that DG(1)–Hancock converges with third-order accuracy, while its error levels are almost comparable to those of DG(2)–RK3.

Even though Figure 3.24(a) provides the accuracy of a method, it does not show its efficiency. Therefor, $L_2$-norms of solution errors are plotted against CPU time in Figure 3.24(b). This figure reveals that the DG(1)–Hancock is actually more efficient than DG(2)–RK3: the former method has only two unknowns $\big($DG(1)$\big)$ and a two-stage update formula (Hancock), whereas the latter method has three unknowns $\big($DG(2)$\big)$ and a three-stage update formula $\big($RK(3)$\big)$.

Finally, CPU time normalized by the CPU time of the DG(1)–RK2 method for a specific error level is shown in Figure 3.25. Remarkably, the DG(1)–Hancock method is almost two orders of magnitude more efficient than the DG(1)–RK2 method. However, this could be a flattering result since the model equation we are solving is merely the 1-D linear advection equation with periodic boundaries. We expect that, when a nonlinear problem is considered, the efficiency of DG(1)–Hancock will be degraded; in fact, the numerical results in the next section still show an order of magnitude difference with DG(1)–MOL for a 1-D nonlinear hyperbolic-relaxation system.

(a) The HR2–RK2 method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---:|---:|---:|---:|---:|---:|---:|
| 10 | 10 | 6.96e–01 | — | 9.36e–01 | — | 0.00e+00 |
| 20 | 20 | 5.99e–01 | 0.22 | 8.43e–01 | 0.15 | 0.00e+00 |
| 40 | 40 | 7.74e–01 | −0.37 | 1.09e+00 | −0.38 | 2.00e–02 |
| 80 | 80 | 8.31e–01 | −0.10 | 1.17e+00 | −0.10 | 8.00e–02 |
| 160 | 160 | 2.23e–01 | 1.90 | 3.16e–01 | 1.90 | 3.00e–01 |
| 320 | 320 | 5.61e–02 | 1.99 | 7.93e–02 | 1.99 | 1.19e+00 |
| 640 | 640 | 1.40e–02 | 2.00 | 1.98e–02 | 2.00 | 4.76e+00 |
| 1280 | 1280 | 3.51e–03 | 2.00 | 4.96e–03 | 2.00 | 2.67e+01 |
| 2560 | 2560 | 8.77e–04 | 2.00 | 1.24e–03 | 2.00 | 1.07e+02 |
| 5120 | 5120 | 2.19e–04 | 2.00 | 3.10e–04 | 2.00 | 4.27e+02 |

(b) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---:|---:|---:|---:|---:|---:|---:|
| 10 | 10 | 7.15e–01 | — | 9.83e–01 | — | 0.00e+00 |
| 20 | 20 | 3.96e–01 | 0.85 | 5.59e–01 | 0.81 | 0.00e+00 |
| 40 | 40 | 1.09e–01 | 1.86 | 1.54e–01 | 1.86 | 2.00e–02 |
| 80 | 80 | 2.75e–02 | 1.99 | 3.88e–02 | 1.99 | 6.00e–02 |
| 160 | 160 | 6.86e–03 | 2.00 | 9.70e–03 | 2.00 | 2.40e–01 |
| 320 | 320 | 1.71e–03 | 2.00 | 2.42e–03 | 2.00 | 9.50e–01 |
| 640 | 640 | 4.28e–04 | 2.00 | 6.06e–04 | 2.00 | 3.78e+00 |
| 1280 | 1280 | 1.07e–04 | 2.00 | 1.51e–04 | 2.00 | 1.79e+01 |
| 2560 | 2560 | 2.68e–05 | 2.00 | 3.78e–05 | 2.00 | 7.14e+01 |
| 5120 | 5120 | 6.68e–06 | 2.00 | 9.44e–06 | 2.00 | 2.85e+02 |

(c) The DG(1)–RK2 method ($\tilde{\nu} = 0.3$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---:|---:|---:|---:|---:|---:|---:|
| 10 | 20 | 8.96e–01 | — | 8.58e–01 | — | 0.00e+00 |
| 20 | 40 | 1.15e+00 | −0.36 | 1.14e+00 | −0.41 | 1.00e–02 |
| 40 | 80 | 3.44e–01 | 1.74 | 3.44e–01 | 1.73 | 5.00e–02 |
| 80 | 160 | 8.72e–02 | 1.98 | 8.72e–02 | 1.98 | 1.80e–01 |
| 160 | 320 | 2.18e–02 | 2.00 | 2.18e–02 | 2.00 | 7.00e–01 |
| 320 | 640 | 5.45e–03 | 2.00 | 5.45e–03 | 2.00 | 2.79e+00 |
| 640 | 1280 | 1.36e–03 | 2.00 | 1.36e–03 | 2.00 | 1.54e+01 |
| 1280 | 2560 | 3.41e–04 | 2.00 | 3.41e–04 | 2.00 | 7.35e+01 |
| 2560 | 5120 | 8.52e–05 | 2.00 | 8.52e–05 | 2.00 | 2.93e+02 |

(d) The DG(1)–RK3 method ($\tilde{\nu} = 0.37$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---:|---:|---|---|---|---|---|
| 10 | 20 | 9.61e–01 | — | 9.19e–01 | — | 0.00e+00 |
| 20 | 40 | 3.70e–01 | 1.38 | 3.69e–01 | 1.32 | 2.00e–02 |
| 40 | 80 | 5.66e–02 | 2.71 | 5.66e–02 | 2.70 | 6.00e–02 |
| 80 | 160 | 7.27e–03 | 2.96 | 7.27e–03 | 2.96 | 2.30e–01 |
| 160 | 320 | 9.13e–04 | 2.99 | 9.13e–04 | 2.99 | 9.20e–01 |
| 320 | 640 | 1.14e–04 | 3.00 | 1.14e–04 | 3.00 | 3.72e+00 |
| 640 | 1280 | 1.43e–05 | 3.00 | 1.43e–05 | 3.00 | 2.02e+01 |
| 1280 | 2560 | 1.78e–06 | 3.00 | 1.78e–06 | 3.00 | 9.33e+01 |
| 2560 | 5120 | 2.23e–07 | 3.00 | 2.23e–07 | 3.00 | 3.73e+02 |

(e) The DG(2)–RK3 method ($\tilde{\nu} = 0.19$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---:|---:|---|---|---|---|---|
| 10 | 30 | 5.31e–02 | — | 7.33e–02 | — | 1.00e–02 |
| 20 | 60 | 6.13e–03 | 3.11 | 8.62e–03 | 3.09 | 4.00e–02 |
| 40 | 120 | 7.46e–04 | 3.04 | 1.05e–03 | 3.03 | 1.50e–01 |
| 80 | 240 | 9.25e–05 | 3.01 | 1.31e–04 | 3.01 | 6.20e–01 |
| 160 | 480 | 1.15e–05 | 3.00 | 1.63e–05 | 3.00 | 2.42e+00 |
| 320 | 960 | 1.44e–06 | 3.00 | 2.04e–06 | 3.00 | 9.69e+00 |
| 640 | 1920 | 1.80e–07 | 3.00 | 2.55e–07 | 3.00 | 4.94e+01 |
| 1280 | 3840 | 2.32e–08 | 2.96 | 3.28e–08 | 2.96 | 2.14e+02 |
| 2560 | 7680 | 7.44e–09 | 1.64 | 1.05e–08 | 1.64 | 8.53e+02 |

(f) The DG(1)–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---:|---:|---|---|---|---|---|
| 10 | 20 | 1.74e–01 | — | 2.42e–01 | — | 0.00e+00 |
| 20 | 40 | 2.54e–02 | 2.78 | 3.57e–02 | 2.76 | 1.00e–02 |
| 40 | 80 | 3.25e–03 | 2.97 | 4.59e–03 | 2.96 | 3.00e–02 |
| 80 | 160 | 4.08e–04 | 2.99 | 5.76e–04 | 2.99 | 1.30e–01 |
| 160 | 320 | 5.10e–05 | 3.00 | 7.21e–05 | 3.00 | 5.00e–01 |
| 320 | 640 | 6.38e–06 | 3.00 | 9.02e–06 | 3.00 | 2.00e+00 |
| 640 | 1280 | 7.97e–07 | 3.00 | 1.13e–06 | 3.00 | 8.05e+00 |
| 1280 | 2560 | 9.96e–08 | 3.00 | 1.41e–07 | 3.00 | 3.57e+01 |
| 2560 | 5120 | 1.28e–08 | 2.96 | 1.81e–08 | 2.96 | 1.43e+02 |

Table 3.8: A grid convergence study by solving the 1-D linear advection equation, $\partial_t u + r\partial_x u = 0$ where $r = \dfrac{1}{2}$, is performed. The numerical solution at $t_{\text{end}} = 300$ is compared to the exact solution, then both $L_p$ errors of $\bar{u}$ and the convergence rates are obtained for each method.

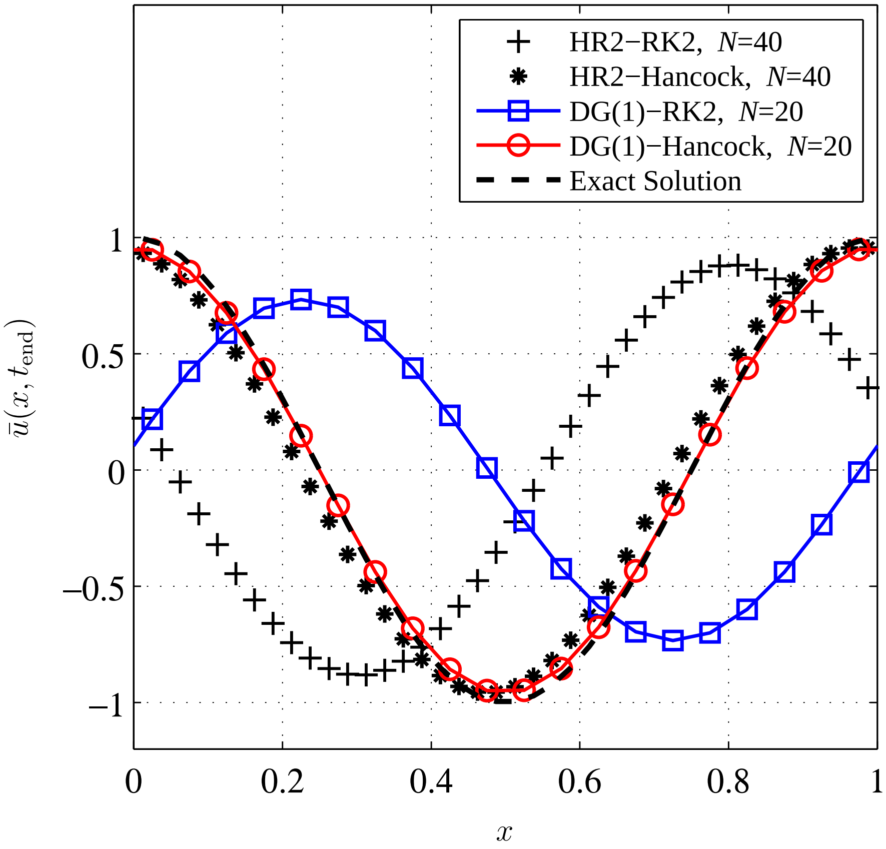
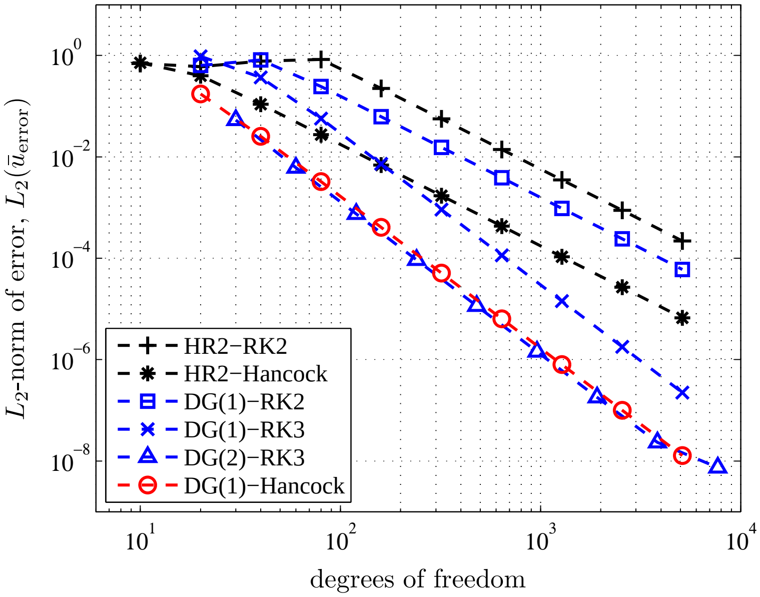
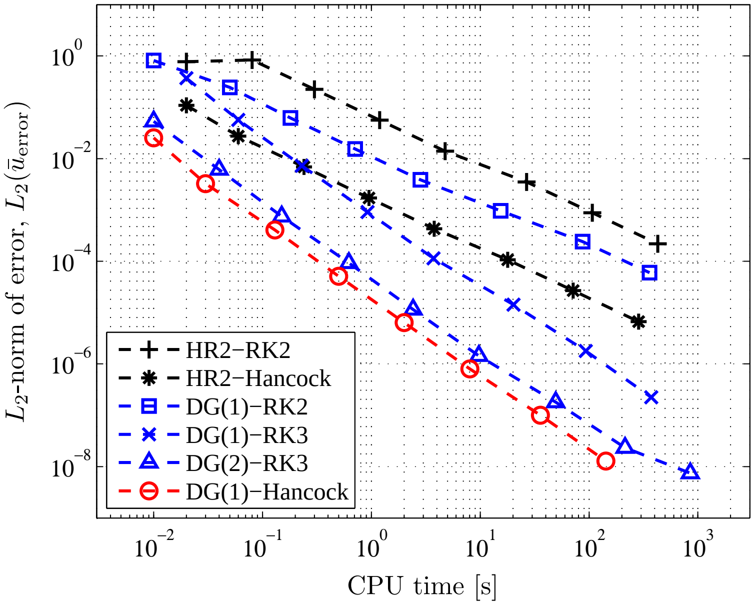


Figure 3.23: Numerical results of four methods at $t_{\text{end}} = 300$ in problem (3.226). The DG(1)–Hancock method appears to be the least dissipative and dispersive.



(a) $L_2$-norms of error plotted against number of degrees of freedom. DG(1)–Hancock is almost comparable to DG(2)–RK3.



(b) $L_2$-norms of error plotted against CPU time. DG(1)–Hancock is the most efficient method.

Figure 3.24: The $L_2$-norms of errors shown in Table 3.8 are plotted in terms of both degrees of freedom and CPU time. The grid convergence study shows the superiority of the DG(1)–Hancock method.

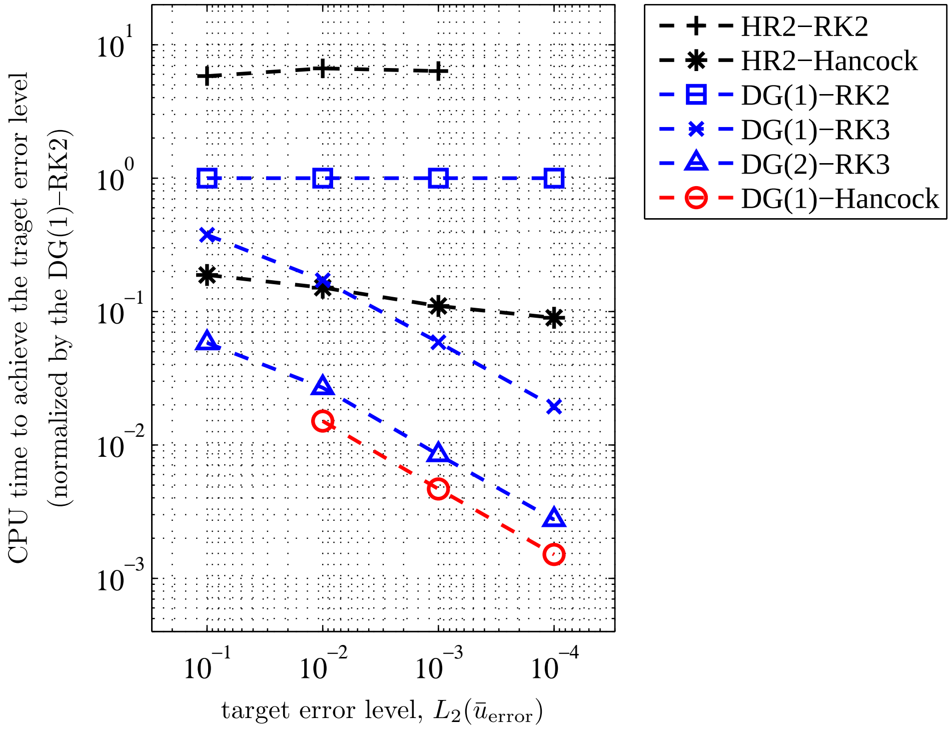


Figure 3.25: CPU time required to achieve the target error level, normalized by the DG(1)–RK2 result. The high efficiency of DG(1)–Hancock is evident.

## 3.6 Grid Convergence Study in 2-D

The previous one-dimensional numerical experiment is extended to a two-dimensional problem. The two-dimensional linear advection equation,

$$\partial_t u + r\partial_x u + s\partial_y u = 0 \quad \text{with} \quad r = \frac{1}{2}, \quad u(x,y,0) = \cos\bigl(2\pi(x+y)\bigr), \tag{3.228}$$

is solved over the domain $(x,y) \in [0,1] \times [0,1]$ with periodic boundary conditions. The numerical solution at $t_{\text{end}} = 150$ is compared to the exact solution. The upwind flux is used at cell interfaces, and we set the Courant number for each method equal to 90% of the method's linear stability limit listed in Table 3.7;

$$\nu_{\text{HR2–RK2}} = \nu_{\text{HR2–Hancock}} = 0.9, \tag{3.229a}$$

$$\nu_{\text{DG(1)–RK2}} = 0.3, \quad \nu_{\text{DG(1)–RK3}} = 0.37, \quad \nu_{\text{DG(1)–Hancock}} = 0.6. \tag{3.229b}$$

Note that the stability limit of the DG(1)–Hancock method in two dimensions reduces to $\tilde{\nu}_{\text{2D}} \leq 0.664$ while the other methods have the same stability limit as in one dimension. The $L_2$-, $L_\infty$-norms of errors and the CPU time of each method are listed in Table 3.9. Figures 3.26(a) and 3.26(b) show the $L_2$-norm of error against the number of degree of freedom and the CPU time. These show that the high accuracy, efficiency, and third-order convergence of the DG(1)–Hancock method are preserved even in two dimensions. A rather surprising result is the superiority of HR2–Hancock to DG(1)–Hancock in the region of high error-level, $L_2(\bar{u}_{\text{error}}) \geq 10^{-2}$. Apparently, for the HR2–Hancock method, the relatively finer grid (compared to one for a DG method with the same number of degrees of freedom) leads to a lower numerical error in this range. However, as higher accuracy is required, the DG(1)–Hancock method takes over in terms of both accuracy and efficiency. The CPU time normalized by the CPU time of the DG(1)–RK2 method for a specific error level is shown in Figure 3.27. Compared to the one-dimensional result shown in Figure 3.25, the efficiency of the DG(1)–Hancock method is reduced, yet it shows that an order of magnitude difference with DG(1)–RK2 method.

(a) The HR2–RK2 method ($\tilde{\nu} = 0.9$)

| $N_x \times N_y$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10× 10 | 100 | 6.84e–01 | — | 9.67e–01 | — | 4.00e–02 |
| 20× 20 | 400 | 5.96e–01 | 0.20 | 8.42e–01 | 0.20 | 3.30e–01 |
| 40× 40 | 1600 | 7.73e–01 | −0.37 | 1.09e+00 | −0.37 | 2.79e+00 |
| 80× 80 | 6400 | 8.31e–01 | −0.10 | 1.17e+00 | −0.11 | 2.42e+01 |
| 160×160 | 25600 | 2.23e–01 | 1.90 | 3.16e–01 | 1.90 | 5.71e+02 |
| 320×320 | 102400 | 5.61e–02 | 1.99 | 7.93e–02 | 1.99 | 4.47e+03 |
| 640×640 | 409600 | 1.40e–02 | 2.00 | 1.98e–02 | 2.00 | 3.49e+04 |

(b) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x \times N_y$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10× 10 | 100 | 7.04e–01 | — | 9.93e–01 | — | 3.00e–02 |
| 20× 20 | 400 | 3.94e–01 | 0.84 | 5.50e–01 | 0.85 | 2.50e–01 |
| 40× 40 | 1600 | 1.09e–01 | 1.85 | 1.54e–01 | 1.83 | 2.06e+00 |
| 80× 80 | 6400 | 2.75e–02 | 1.99 | 3.88e–02 | 1.99 | 1.70e+01 |
| 160×160 | 25600 | 6.86e–03 | 2.00 | 9.70e–03 | 2.00 | 2.63e+02 |
| 320×320 | 102400 | 1.71e–03 | 2.00 | 2.42e–03 | 2.00 | 2.21e+03 |
| 640×640 | 409600 | 4.28e–04 | 2.00 | 6.06e–04 | 2.00 | 1.78e+04 |

(c) The DG(1)–RK2 method ($\tilde{\nu} = 0.3$)

| $N_x \times N_y$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10× 10 | 300 | 6.84e–01 | — | 9.68e–01 | — | 1.20e–01 |
| 20× 20 | 1200 | 6.96e–01 | −0.02 | 9.80e–01 | −0.02 | 9.80e–01 |
| 40× 40 | 4800 | 2.55e–01 | 1.45 | 3.60e–01 | 1.44 | 8.13e+00 |
| 80× 80 | 19200 | 6.32e–02 | 2.01 | 8.94e–02 | 2.01 | 1.23e+02 |
| 160×160 | 76800 | 1.55e–02 | 2.02 | 2.20e–02 | 2.02 | 1.43e+03 |
| 320×320 | 307200 | 3.86e–03 | 2.01 | 5.46e–03 | 2.01 | 1.16e+04 |
| 640×640 | 1228800 | 9.64e–04 | 2.00 | 1.36e–03 | 2.00 | 9.92e+04 |

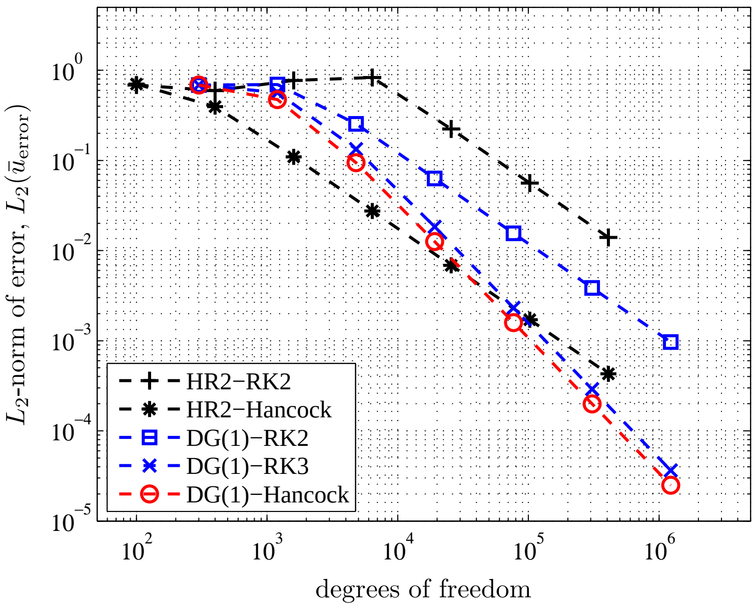
(d) The DG(1)–RK3 method ($\tilde{\nu} = 0.37$)

| $N_x \times N_y$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10× 10 | 300 | 6.84e–01 | — | 9.68e–01 | — | 1.70e–01 |
| 20× 20 | 1200 | 5.66e–01 | 0.27 | 8.00e–01 | 0.27 | 1.44e+00 |
| 40× 40 | 4800 | 1.33e–01 | 2.09 | 1.88e–01 | 2.09 | 1.19e+01 |
| 80× 80 | 19200 | 1.83e–02 | 2.86 | 2.59e–02 | 2.86 | 2.16e+02 |
| 160×160 | 76800 | 2.32e–03 | 2.98 | 3.28e–03 | 2.98 | 1.95e+03 |
| 320×320 | 307200 | 2.91e–04 | 3.00 | 4.11e–04 | 3.00 | 1.74e+04 |
| 640×640 | 1228800 | 3.64e–05 | 3.00 | 5.14e–05 | 3.00 | 1.36e+05 |

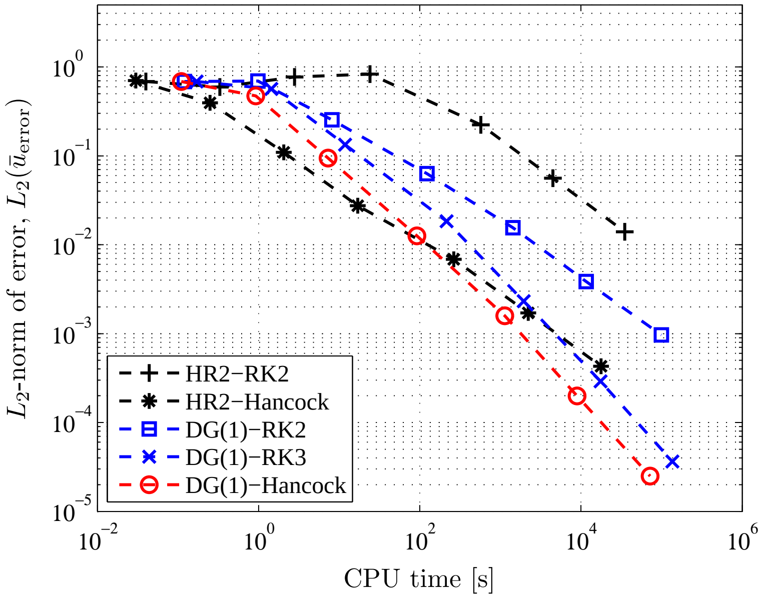
(e) The DG(1)–Hancock method ($\tilde{\nu} = 0.6$)

| $N_x \times N_y$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10× 10 | 300 | 6.84e–01 | — | 9.67e–01 | — | 1.10e–01 |
| 20× 20 | 1200 | 4.73e–01 | 0.53 | 6.69e–01 | 0.53 | 9.20e–01 |
| 40× 40 | 4800 | 9.42e–02 | 2.33 | 1.33e–01 | 2.33 | 7.28e+00 |
| 80× 80 | 19200 | 1.26e–02 | 2.90 | 1.78e–02 | 2.90 | 9.22e+01 |
| 160×160 | 76800 | 1.59e–03 | 2.99 | 2.25e–03 | 2.99 | 1.13e+03 |
| 320×320 | 307200 | 1.99e–04 | 3.00 | 2.82e–04 | 3.00 | 8.94e+03 |
| 640×640 | 1228800 | 2.49e–05 | 3.00 | 3.52e–05 | 3.00 | 7.20e+04 |

Table 3.9: A grid convergence study by solving the 2-D linear advection equation, $\partial_t u + r\partial_x u + s\partial_x u = 0$ where $r = s = \dfrac{1}{2}$, is performed. The numerical solution at $t_{\text{end}} = 150$ is compared to the exact solution, and both $L_p$-errors of $\bar{u}$ and convergence rates are obtained for each method.



(a) $L_2$-norms of error plotted against number of degrees of freedom. The third-order convergence of the DG(1)–Hancock method is observed.



(b) $L_2$-norms of error plotted against CPU time. DG(1)–Hancock is the most efficient method for $L_2(\bar{u}_{\text{error}}) \leq 10^{-2}$.

Figure 3.26: The $L_2$-norms of errors shown in Table 3.9 are plotted in terms of both degrees of freedom and CPU time. The 2-D linear advection equation is solved by various methods. Note that we did not test DG(2)–RK3, as we did in 1D. The grid convergence study shows the superiority of the HR2–Hancock method at low error levels, $L_2(\bar{u}_{\text{error}}) \geq 10^{-2}$, yet the DG(1)–Hancock method takes over in accuracy and efficiency when high accuracy is required.

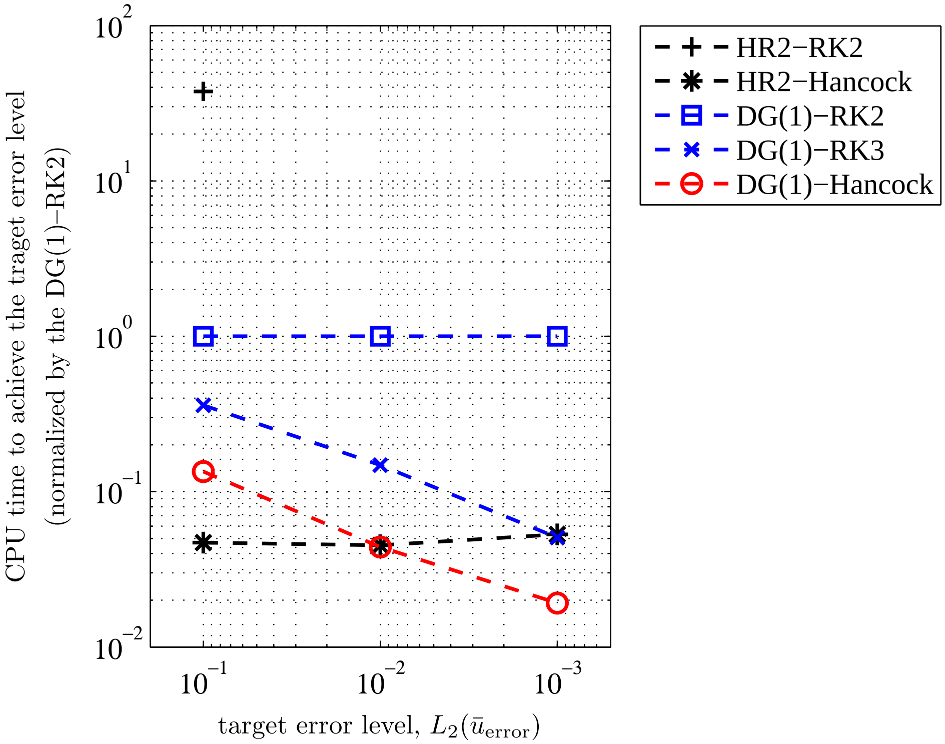


Figure 3.27: CPU time required to achieve the target error level, normalized by the DG(1)–RK2 result. The 2-D linear advection equation is solved by various methods. The high efficiency of DG(1)–Hancock is shown especially when higher accuracy is required.

## 3.7 Grid Convergence Study for Nonlinear Hyperbolic Equations

### 3.7.1 The Inviscid Burgers' Equation

To extend the analysis to nonlinear equations, the inviscid Burgers' equation:

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0; \quad x \in \mathbb{R},\ t > 0, \tag{3.230}$$

which represents the simplest model of the motion of a fluid is considered. The nonlinearity in the flux term leads to the non-constant wave propagation speed $u$ unlike the constant speed in the linear advection equation. Thus the shape of the initial-value distribution is no longer preserved, and it creates either a discontinuity (shock) or a smooth profile (expansion). Also, a sufficiently smooth initial condition could still generate a discontinuity within a finite time due to the nonlinearity. Here, our motivation is to assess the order of convergence of a numerical method, therefore, an initial condition that only creates an expansion wave is chosen. The initial condition is given by

$$u(x,0) = \begin{cases} -1 & x \leq -5, \\ \tanh\left(\dfrac{10x}{25 - x^2}\right) & -5 < x < 5, \\ 1 & x \geq 5. \end{cases} \tag{3.231}$$

The exact solution in the general form can be constructed by the method of characteristics, and is given by an implicit relation:

$$u(x,t) = u(x - ut, 0). \tag{3.232}$$

Note that, in general, the above exact solution is only valid before a shock is formed, but with the initial values (3.231) this is not a concern. The exact solution at time

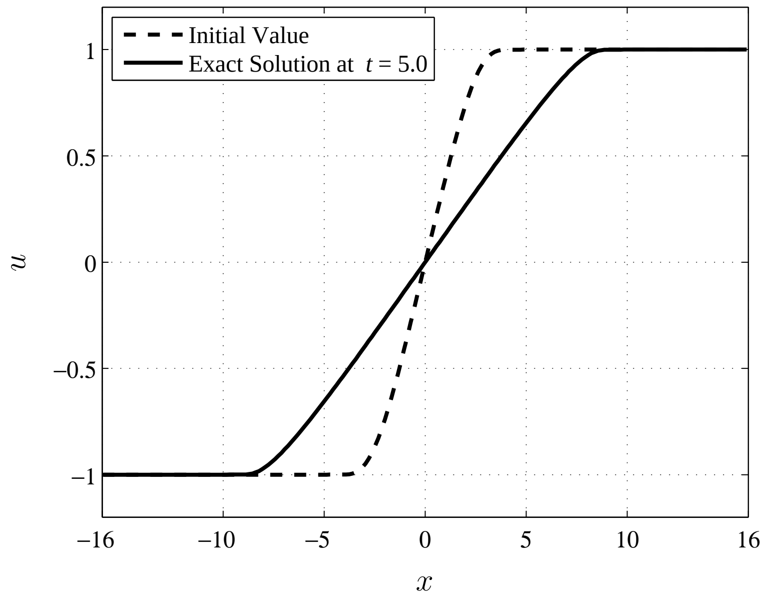


Figure 3.28: The broken line represents the initial condition for the Burgers' equation, and the solid line is the exact solution at time $t_{\text{end}} = 5.0$.

$t_{\text{end}}$ with the initial condition (3.231) is given by

$$u(x,t) = \begin{cases} -1 & x \leq -5 - t_{\text{end}}, \\ u(x - ut_{\text{end}}, 0) & -5 - t_{\text{end}} < x < 5 + t_{\text{end}}, \\ 1 & x \geq 5 + t_{\text{end}}. \end{cases} \tag{3.233}$$

The profiles of the initial condition and exact solution at $t_{\text{end}} = 5.0$ are shown in Figure 3.28. The cell-averaged value in each cell is obtained with sufficient accuracy by three-point Gauss quadrature. The solution at a quadrature point $x_i$ is the solution at the unknown coordinate $x$ at time $t = 0$, which satisfies the implicit relation

$$x = x_i - u(x,0)\, t_{\text{end}}; \tag{3.234}$$

it is computed by Newton's method:

$$x^{n+1} = x^n - \frac{f(x^n)}{f_x(x^n)}, \tag{3.235}$$

where

$$f(x) = x + \tanh\left(\frac{10x}{25 - x^2}\right) t_{\text{end}} - x_i, \tag{3.236a}$$

$$f_x(x) = \frac{10(x^2 + 25)}{(x^2 - 25)^2} \text{sech}^2 \left(\frac{10x}{25 - x^2}\right). \tag{3.236b}$$

The Courant number for each method is identical to the value used in the case of one-dimensional linear advection, equation (3.227) on page 185, that is: 90% of its stability limit. The $L_1$-,$L_\infty$-norms, and corresponding CPU time are listed in Tables 3.10 and 3.11. Table 3.10 shows that DG(1)–RK3 and DG(2)–RK3 are third-order accurate, and HR2–RK2, HR2–Hancock, and DG(1)–RK2 are second-order accurate in both $L_1$- and $L_\infty$-norms. Thus, the order of accuracy obtained by the linear analysis is preserved for a scalar nonlinear problem.

Table 3.11 shows a grid-convergence study of the DG(1)–Hancock method with various volume-integral treatments. Two quadratures in time, Gauss–Lobatto and Gauss–Radau, are compared for the nonlinear flux; recall that these two quadratures give identical results for a linear flux. See Figure 2.3 on page 40 for their quadrature points in space-time domain.

The sequence of quadratures applied in space and time is examined for its influence on the order of accuracy. For the spatial integration in the volume integral of the flux, the Gauss–Lobatto quadrature is always used. Tables 3.11 (a) and (c) show the result when either the Gauss–Lobatto/Radau temporal quadratures are made first, then a spatial quadrature is made later. Conversely, Tables 3.11 (b) and (d) show the result when a spatial quadrature is made first at each time level, then a temporal quadrature is applied. For instance, in the case of Table 3.11 (d), the volume integral of flux is obtained by the Gauss–Radau quadrature in time

$$\iint\limits_{I_j \times T^n} f(u)\, dxdt \approx \frac{\Delta t}{4} \left( 3\bar{f}^{n+1/3} + \bar{f}^{n+1} \right), \tag{3.237}$$

where spatially averaged fluxes at time levels $n + \dfrac{1}{3}$ and $n + 1$ are computed by

$$\bar{f}^{n+1/3} = \frac{\Delta x}{6} \left[ f(u^{n+1/3}_{j+1/2}) + 4f(u^{n+1/3}_{j}) + f(u^{n+1/3}_{j+1/2}) \right], \tag{3.238a}$$

$$\bar{f}^{n+1} = \frac{\Delta x}{6} \left[ f(u^{n+1}_{j+1/2}) + 4f(u^{n+1}_{j}) + f(u^{n+1}_{j+1/2}) \right]. \tag{3.238b}$$

Table 3.11 shows that the order of accuracy is sensitive to the sequence of spatial and temporal quadratures, while both Gauss–Lobatto and Gauss–Radau quadratures in time, when combined in the same sequence, provide similar error levels. Computing a spatial quadrature first is necessary to achieve higher-order accuracy; the convergence rates based on the $L_1$-norms of error shows third-order convergence in this case, yet $L_\infty$-norms converge only at the second-order.

(a) The HR2–RK2 method ($\tilde{\nu} = 0.9$)

| $N$ | DOF | $L_1$ error of $\bar{u}$ | Rate | $L_\infty$ error of $\bar{u}$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10 | 10 | 5.52e–02 | — | 7.91e–02 | — | 1.64e–02 |
| 20 | 20 | 2.17e–02 | 1.35 | 5.17e–02 | 0.61 | 1.69e–02 |
| 40 | 40 | 8.21e–03 | 1.40 | 3.28e–02 | 0.66 | 6.24e–02 |
| 80 | 80 | 2.46e–03 | 1.74 | 1.50e–02 | 1.13 | 2.30e–01 |
| 160 | 160 | 6.34e–04 | 1.95 | 5.44e–03 | 1.46 | 9.11e–01 |
| 320 | 320 | 1.53e–04 | 2.06 | 1.61e–03 | 1.75 | 3.73e+00 |
| 640 | 640 | 3.66e–05 | 2.06 | 4.26e–04 | 1.92 | 1.47e+01 |
| 1280 | 1280 | 9.02e–06 | 2.02 | 1.08e–04 | 1.98 | 5.92e+01 |
| 2560 | 2560 | 2.25e–06 | 2.01 | 2.69e–05 | 2.00 | 2.37e+02 |
| 5120 | 5120 | 5.61e–07 | 2.00 | 6.73e–06 | 2.00 | 9.46e+02 |

(b) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N$ | DOF | $L_1$ error of $\bar{u}$ | Rate | $L_\infty$ error of $\bar{u}$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10 | 10 | 1.19e–02 | — | 2.92e–02 | — | 1.31e–02 |
| 20 | 20 | 6.03e–03 | 0.99 | 1.43e–02 | 1.03 | 8.97e–03 |
| 40 | 40 | 1.84e–03 | 1.71 | 4.79e–03 | 1.57 | 3.00e–02 |
| 80 | 80 | 4.30e–04 | 2.10 | 1.38e–03 | 1.79 | 1.17e–01 |
| 160 | 160 | 9.91e–05 | 2.12 | 3.25e–04 | 2.09 | 4.67e–01 |
| 320 | 320 | 2.36e–05 | 2.07 | 8.42e–05 | 1.95 | 1.87e+00 |
| 640 | 640 | 5.80e–06 | 2.03 | 2.16e–05 | 1.96 | 7.49e+00 |
| 1280 | 1280 | 1.44e–06 | 2.01 | 5.46e–06 | 1.98 | 3.01e+01 |
| 2560 | 2560 | 3.57e–07 | 2.01 | 1.37e–06 | 1.99 | 1.20e+02 |
| 5120 | 5120 | 8.92e–08 | 2.00 | 3.44e–07 | 2.00 | 4.80e+02 |

(c) The DG(1)–RK2 method ($\tilde{\nu} = 0.3$)

| $N$ | DOF | $L_1$ error of $\bar{u}$ | Rate | $L_\infty$ error of $\bar{u}$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10 | 20 | 1.60e–02 | — | 2.98e–02 | — | 2.01e–02 |
| 20 | 40 | 3.78e–03 | 2.08 | 1.28e–02 | 1.22 | 5.04e–02 |
| 40 | 80 | 1.07e–03 | 1.83 | 6.01e–03 | 1.09 | 1.92e–01 |
| 80 | 160 | 2.32e–04 | 2.20 | 1.99e–03 | 1.60 | 7.57e–01 |
| 160 | 320 | 4.97e–05 | 2.22 | 5.19e–04 | 1.94 | 3.02e+00 |
| 320 | 640 | 1.14e–05 | 2.13 | 1.24e–04 | 2.07 | 1.22e+01 |
| 640 | 1280 | 2.75e–06 | 2.05 | 3.02e–05 | 2.04 | 4.91e+01 |
| 1280 | 2560 | 6.79e–07 | 2.02 | 7.44e–06 | 2.02 | 1.97e+02 |
| 2560 | 5120 | 1.69e–07 | 2.01 | 1.85e–06 | 2.01 | 7.91e+02 |
| 5120 | 10240 | 4.20e–08 | 2.00 | 4.61e–07 | 2.00 | 3.18e+03 |

(d) The DG(1)–RK3 method ($\tilde{\nu} = 0.37$)

| $N$ | DOF | $L_1$ error of $\bar{u}$ | Rate | $L_\infty$ error of $\bar{u}$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10 | 20 | 9.36e−03 | — | 2.24e−02 | — | 4.34e−02 |
| 20 | 40 | 2.59e−03 | 1.85 | 8.47e−03 | 1.41 | 6.10e−02 |
| 40 | 80 | 6.82e−04 | 1.93 | 3.06e−03 | 1.47 | 2.25e−01 |
| 80 | 160 | 1.15e−04 | 2.57 | 9.56e−04 | 1.68 | 8.89e−01 |
| 160 | 320 | 1.68e−05 | 2.78 | 2.07e−04 | 2.21 | 3.08e+00 |
| 320 | 640 | 2.20e−06 | 2.93 | 3.16e−05 | 2.71 | 1.24e+01 |
| 640 | 1280 | 2.79e−07 | 2.98 | 4.23e−06 | 2.90 | 4.95e+01 |
| 1280 | 2560 | 3.49e−08 | 2.99 | 5.35e−07 | 2.98 | 1.99e+02 |
| 2560 | 5120 | 4.37e−09 | 3.00 | 6.71e−08 | 3.00 | 7.99e+02 |
| 5120 | 10240 | 5.47e−10 | 3.00 | 8.39e−09 | 3.00 | 3.22e+03 |

(e) The DG(2)–RK3 method ($\tilde{\nu} = 0.19$)

| $N$ | DOF | $L_1$ error of $\bar{u}$ | Rate | $L_\infty$ error of $\bar{u}$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10 | 30 | 1.68e−03 | — | 4.01e−03 | — | 7.60e−02 |
| 20 | 60 | 2.62e−04 | 2.68 | 8.30e−04 | 2.27 | 1.28e−01 |
| 40 | 120 | 6.04e−05 | 2.12 | 4.36e−04 | 0.93 | 4.86e−01 |
| 80 | 240 | 4.61e−06 | 3.71 | 6.58e−05 | 2.73 | 1.87e+00 |
| 160 | 480 | 4.41e−07 | 3.39 | 6.45e−06 | 3.35 | 6.81e+00 |
| 320 | 960 | 4.70e−08 | 3.23 | 6.58e−07 | 3.29 | 2.80e+01 |
| 640 | 1920 | 5.66e−09 | 3.05 | 7.60e−08 | 3.11 | 1.11e+02 |
| 1280 | 3840 | 7.01e−10 | 3.01 | 9.30e−09 | 3.03 | 4.42e+02 |
| 2560 | 7680 | 8.75e−11 | 3.00 | 1.16e−09 | 3.01 | 1.76e+03 |
| 5120 | 15360 | 1.09e−11 | 3.00 | 1.44e−10 | 3.00 | 7.07e+03 |

Table 3.10: A grid convergence study by solving the inviscid Burgers' equation $\partial_t u + u\partial_x u = 0$. The numerical solution at $t_{\text{end}} = 5.0$ is compared to the exact solution, then both $L_p$-norms of error $\bar{u}_{\text{error}}$ and convergence rates are obtained for various methods.

(a) The DG(1)–Hancock method ($\tilde{\nu} = 0.9$): Lobatto quadrature for the volume integral in time (time → space).

| $N$ | DOF | $L_1$ error of $\bar{u}$ | Rate | $L_\infty$ error of $\bar{u}$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10 | 20 | 2.82e−02 | — | 6.57e−02 | — | 3.36e−02 |
| 20 | 40 | 6.07e−03 | 2.22 | 1.68e−02 | 1.97 | 2.80e−02 |
| 40 | 80 | 5.76e−04 | 3.40 | 2.49e−03 | 2.75 | 8.69e−02 |
| 80 | 160 | 8.32e−05 | 2.79 | 5.22e−04 | 2.26 | 3.20e−01 |
| 160 | 320 | 1.83e−05 | 2.19 | 1.21e−04 | 2.11 | 1.21e+00 |
| 320 | 640 | 4.38e−06 | 2.06 | 2.92e−05 | 2.05 | 4.70e+00 |
| 640 | 1280 | 1.06e−06 | 2.04 | 7.16e−06 | 2.03 | 1.86e+01 |
| 1280 | 2560 | 2.62e−07 | 2.02 | 1.78e−06 | 2.01 | 7.17e+01 |
| 2560 | 5120 | 6.50e−08 | 2.01 | 4.42e−07 | 2.01 | 2.82e+02 |
| 5120 | 10240 | 1.62e−08 | 2.01 | 1.10e−07 | 2.00 | 1.12e+03 |

(b) The DG(1)–Hancock method ($\tilde{\nu} = 0.9$): Lobatto quadrature for the volume integral in time (space → time).

| $N$ | DOF | $L_1$ error of $\bar{u}$ | Rate | $L_\infty$ error of $\bar{u}$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10 | 20 | 7.38e−03 | — | 2.26e−02 | — | 6.32e−03 |
| 20 | 40 | 3.42e−03 | 1.11 | 9.81e−03 | 1.21 | 1.95e−02 |
| 40 | 80 | 3.52e−04 | 3.28 | 1.68e−03 | 2.55 | 6.69e−02 |
| 80 | 160 | 3.82e−05 | 3.20 | 3.15e−04 | 2.41 | 2.65e−01 |
| 160 | 320 | 4.44e−06 | 3.11 | 6.69e−05 | 2.23 | 1.07e+00 |
| 320 | 640 | 5.29e−07 | 3.07 | 1.54e−05 | 2.12 | 4.29e+00 |
| 640 | 1280 | 7.19e−08 | 2.88 | 3.69e−06 | 2.06 | 1.72e+01 |
| 1280 | 2560 | 8.61e−09 | 3.06 | 9.03e−07 | 2.03 | 6.89e+01 |
| 2560 | 5120 | 1.03e−09 | 3.06 | 2.23e−07 | 2.02 | 2.74e+02 |
| 5120 | 10240 | 1.43e−10 | 2.85 | 5.56e−08 | 2.01 | 1.10e+03 |

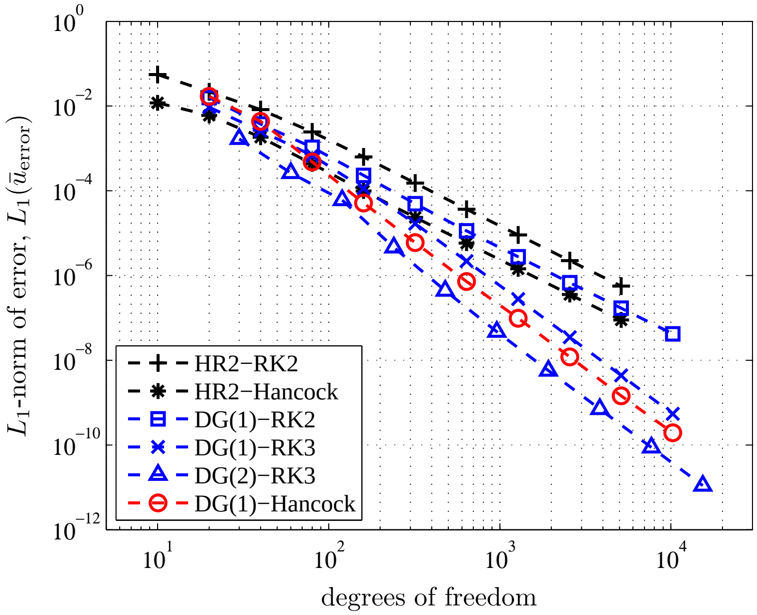
(c) The DG(1)–Hancock method ($\tilde{\nu} = 0.9$): Radau quadrature for the volume integral in time (time → space).

| $N$ | DOF | $L_1$ error of $\bar{u}$ | Rate | $L_\infty$ error of $\bar{u}$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10 | 20 | 2.88e–02 | — | 6.57e–02 | — | 2.05e–02 |
| 20 | 40 | 6.15e–03 | 2.23 | 1.68e–02 | 1.97 | 2.71e–02 |
| 40 | 80 | 5.91e–04 | 3.38 | 2.49e–03 | 2.75 | 8.58e–02 |
| 80 | 160 | 8.34e–05 | 2.82 | 5.22e–04 | 2.26 | 3.17e–01 |
| 160 | 320 | 1.83e–05 | 2.19 | 1.21e–04 | 2.11 | 1.19e+00 |
| 320 | 640 | 4.39e–06 | 2.06 | 2.92e–05 | 2.05 | 4.59e+00 |
| 640 | 1280 | 1.06e–06 | 2.04 | 7.16e–06 | 2.03 | 1.85e+01 |
| 1280 | 2560 | 2.62e–07 | 2.02 | 1.78e–06 | 2.01 | 7.05e+01 |
| 2560 | 5120 | 6.50e–08 | 2.01 | 4.42e–07 | 2.01 | 2.79e+02 |
| 5120 | 10240 | 1.62e–08 | 2.01 | 1.10e–07 | 2.00 | 1.11e+03 |

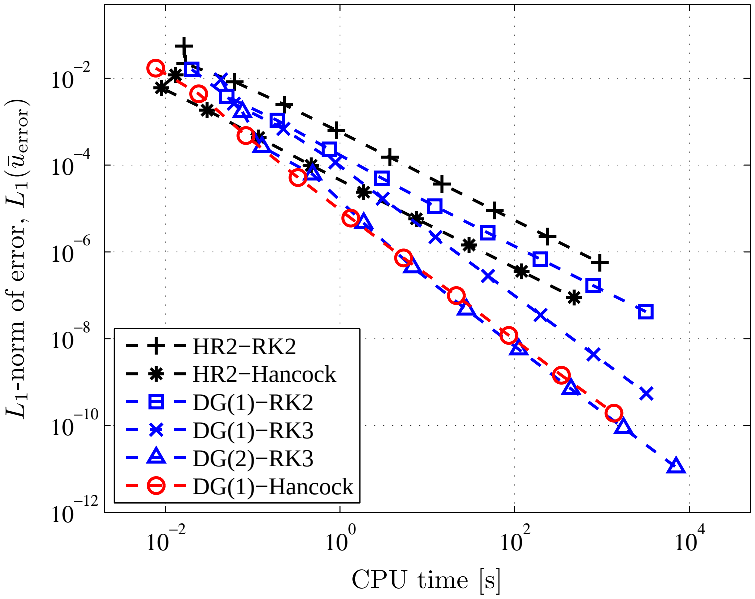
(d) The DG(1)–Hancock method ($\tilde{\nu} = 0.9$): Radau quadrature for the volume integral in time (space → time).

| $N$ | DOF | $L_1$ error of $\bar{u}$ | Rate | $L_\infty$ error of $\bar{u}$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 10 | 20 | 9.47e–03 | — | 2.50e–02 | — | 5.43e–03 |
| 20 | 40 | 3.46e–03 | 1.45 | 1.00e–02 | 1.32 | 1.86e–02 |
| 40 | 80 | 3.26e–04 | 3.41 | 1.63e–03 | 2.61 | 6.31e–02 |
| 80 | 160 | 3.32e–05 | 3.30 | 3.12e–04 | 2.39 | 2.49e–01 |
| 160 | 320 | 3.67e–06 | 3.18 | 6.67e–05 | 2.23 | 1.01e+00 |
| 320 | 640 | 4.30e–07 | 3.09 | 1.54e–05 | 2.12 | 4.08e+00 |
| 640 | 1280 | 5.92e–08 | 2.86 | 3.69e–06 | 2.05 | 1.62e+01 |
| 1280 | 2560 | 7.03e–09 | 3.07 | 9.03e–07 | 2.03 | 6.45e+01 |
| 2560 | 5120 | 8.67e–10 | 3.02 | 2.23e–07 | 2.02 | 2.59e+02 |
| 5120 | 10240 | 1.19e–10 | 2.86 | 5.56e–08 | 2.01 | 1.03e+03 |

Table 3.11: A grid convergence study by solving the inviscid Burgers' equation $\partial_t u + u\partial_x u = 0$. DG(1)–Hancock methods with various volume-integral methods are compared. The results show that doing spatial quadrature first, temporal quadrature later leads to higher accuracy than vice versa.



(a) $L_1$-norms of error plotted against number of degrees of freedom. Unlike the result of 1-D linear advection, DG(2)–RK3 is more accurate than DG(1)–Hancock method.



(b) $L_1$-norms of error plotted against CPU time. DG(1)–Hancock is almost comparable to DG(2)–RK3.

Figure 3.29: The $L_1$-norms of errors shown in Table 3.10 and 3.11 are plotted in terms of both degrees of freedom and CPU time. The grid convergence study shows the DG(2)–RK3 method is more accurate than the DG(1)–Hancock method, yet DG(1)–Hancock is more efficient on coarser grids.

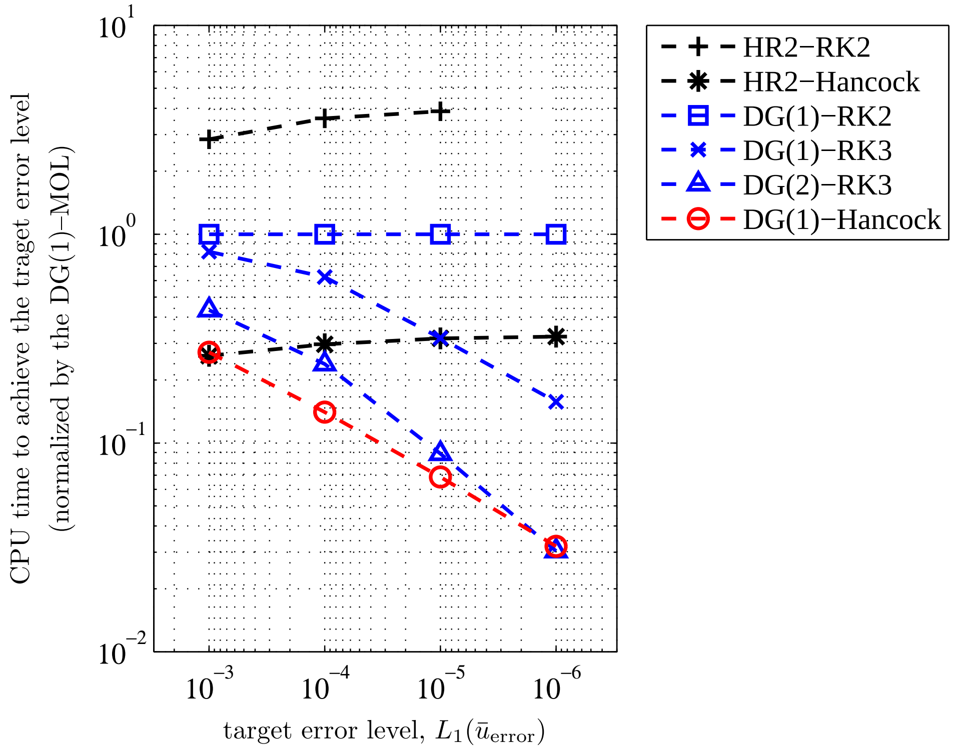


Figure 3.30: CPU time required to achieve the target error level, normalized by the DG(1)–RK2 result. The high efficiency of DG(1)–Hancock is evident; it is matched by DG(2)–RK3 only on the finest grid.

# CHAPTER IV

# ANALYSIS FOR 1-D AND 2-D LINEAR HYPERBOLIC-RELAXATION EQUATIONS

## 4.1 Introduction

In this chapter, numerical methods including the DG(1)–Hancock method for hyperbolic-relaxation equations are investigated analytically and numerically. As a preliminary analysis for hyperbolic-relaxation equations, various methods for hyperbolic conservation laws were analyzed in the previous chapter; Fourier analyses and numerical tests show the superior accuracy and efficiency of the DG(1)–Hancock method over the semi-discrete, method-of-lines approach for both linear and nonlinear equations. We shall now carry out a Fourier analysis of four methods applied to one- and two-dimensional systems of linear hyperbolic-relaxation equations. The local truncation error of the DG(1)–Hancock is compared to HR2–MOL, DG(1)–MOL, and HR2–Hancock; numerical tests confirm the linear analysis. Later, the discretization methods are applied to a system of nonlinear hyperbolic-relaxation equations to examine the validity of the linear analysis.

## 4.2 Model Equations: Generalized Hyperbolic Heat Equations

### 4.2.1 Dimensional Form

The model equation we consider is the generalized hyperbolic heat equations (GHHE) [JL96, Hit00, LM02],

$$\partial_t u + \partial_x v = 0, \tag{4.1a}$$

$$\partial_t v + a_F^2 \partial_x u = -\frac{1}{\tau}(v - a_E u); \quad x \in \mathbb{R},\ t > 0, \tag{4.1b}$$

where $u(x,t) \in \mathbb{R}$ is the conserved variable and $v(x,t) \in \mathbb{R}$ is the flux of $u$. In vector form, $\mathbf{u} = [u, v]^T, \mathbf{f} = [v, a_F^2 u]^T$, and $\mathbf{s} = [0, a_E u - v]^T$ in

$$\partial_t \mathbf{u} + \partial_x \mathbf{f}(\mathbf{u}) = \frac{1}{\tau}\mathbf{s}(\mathbf{u}). \tag{4.2}$$

There are three *constant* parameters: $\tau > 0$ is a relaxation time, $a_F > 0$ is a frozen wave speed, and $a_E > 0$ is an equilibrium wave speed. For stability, $|a_E| \leq a_F$. The constant Jacobian matrix and its eigenvalues are as follows:

$$\mathbf{A} := \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{pmatrix} 0 & 1 \\ a_F^2 & 0 \end{pmatrix} \quad \longrightarrow \quad \lambda_{1,2} := \mathrm{Eig}(\mathbf{A}) = \pm a_F. \tag{4.3}$$

Here, we insist that these three parameters have physical meaning; once the problem is described, these parameters are fixed. The above equations are constructed such that the frozen waves propagate at speed $\pm a_F$ in the beginning; these eventually decay. Simultaneously, equilibrium waves at speed $\pm a_E$ enter the model; one of the equilibrium waves with speed $-a_E$ is quickly damped out, and the other wave with speed $a_E$ dominates the solution. Figure 4.1 describes these waves schematically. The right hand side of (4.1) represents the relaxation process, which always drives the non-equilibrium flux variable $v$ to its equilibrium flux $a_E u$. A detailed

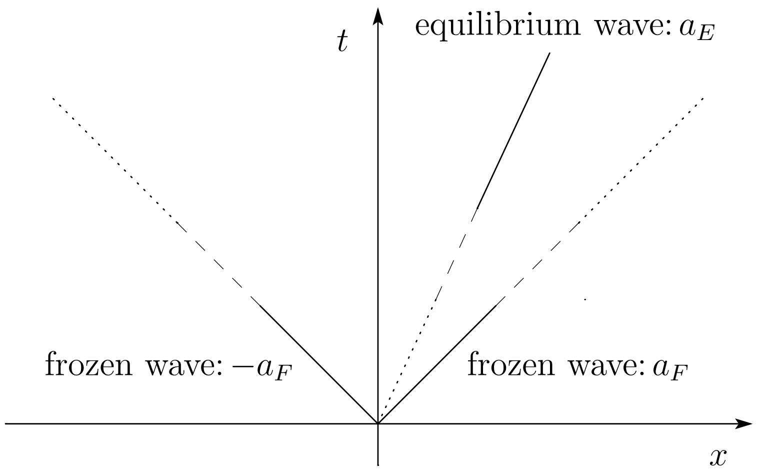


Figure 4.1: Initially, two frozen waves propagate with speed $\pm a_F$; they eventually decay. Meanwhile, the equilibrium wave with speed $a_E$ arises and dominates the flow field in the long-time limit.

dispersion analysis and the exact solution of the Riemann problem are presented by Hittinger [HR04, Hit00].

Let $L$ be a length scale of interest, and $a_F$ serve as a reference wave speed, then a reference time scale can be defined by $T := \dfrac{L}{a_F}$. Note that this is a particular choice of scaling: another reference time may be chosen. Since $a_F$ is a fixed value, changing the length scale of interest affects the reference time. The GHHE can be reduced to a smaller set of equations by a certain choice of $T$ relative to the relaxation time $\tau$, which really means choosing a certain length scale of interest.

When the time of interest is much smaller than the relaxation time ($T \ll \tau$), the relaxation process is not yet important, and the GHHE is reduced to the wave equation,

$$\begin{aligned} \partial_t u + \partial_x v &= 0, \\ \partial_t v + a_F^2 \partial_x u &\simeq 0, \end{aligned} \qquad \longrightarrow \qquad \partial_{tt} u - a_F^2 \partial_{xx} u = 0, \tag{4.4}$$

where the wave speeds are $\pm a_F$. This is the reduced form of the frozen limit.

On the other hand, when the time of interest is much larger than the relaxation time ($T \gg \tau$), the relaxation process is no longer negligible. Asymptotic expansion of $u$ and $v$ for small $\tau$ gives an advection-diffusion equation (the derivation for the particular scaling is given in Appendix C on page 354):

$$\partial_t u + a_E \partial_x u = \tau(a_F^2 - a_E^2)\partial_{xx} u + O\big(\tau^2\big)\,. \tag{4.5}$$

This is the reduced form in the near-equilibrium limit. Note that the leading diffusion coefficient $\tau(a_F^2 - a_E^2)$ always has a positive sign as long as $a_E \leq a_F$; this property is called the sub-characteristic condition for stability [Liu87]. There are two different physical processes included in this equation; the relative strength of the two parameters, advection speed $a_E$ and diffusion coefficient $\epsilon(a_F^2 - a_E^2)$, decides which is the dominant physics. This will be discussed in more detail in a later section.

We further consider the time scale of interest $T$ to be infinite; this is equivalent to letting $\tau \to 0$, so the relaxation process occurs instantaneously, and the above equation becomes a pure advection equation:

$$\partial_t u + a_E \partial_x u = 0, \tag{4.6}$$

where the wave speed is $a_E$. This is the reduced form of the GHHE in the equilibrium limit.

To summarize, let $\bar{t}$ be the dimensionless time normalized by the relaxation time $\tau$ such that

$$\bar{t} := \frac{T}{\tau} = \frac{L}{a_F \tau}. \tag{4.7}$$

The reduced equations of the GHHE corresponding to $\bar{t}$ are shown in Table 4.1. These forms can be seen as consecutive transformations of the GHHE in the time frame.

| dimensionless time | assumption | reduced equation |
|---|---|---|
| $\bar{t} \ll 1$ | frozen limit | $\partial_{tt} u - a_F^2 \partial_{xx} u = 0$ |
| $\bar{t} \gg 1$ | near-equilibrium limit | $\partial_t u + a_E \partial_x u \simeq \tau (a_F^2 - a_E^2) \partial_{xx} u$ |
| $\bar{t} \to \infty$ | equilibrium limit | $\partial_t u + a_E \partial_x u = 0$ |

Table 4.1: The reduced forms of the GHHE are listed in each limit. The characteristic of the GHHE changes with the time scale of interest.

### 4.2.2 Nondimensionalization of the 1-D GHHE

#### Choice of Scaling Parameters

As seen in the previous section, the GHHE changes characteristics in different time scales of interest even though the equations themselves are linear. Thus, when we nondimensionalize the original equations (4.1), the specific choice of reference time $t_0$ affects the behavior of the equations significantly. Here, three different reference times are chosen for nondimensionalization. Let each symbol with subscript 0 serve as a reference parameter to nondimensionalize the variables, and the notation $\hat{(\cdot)}$ represent a dimensionless variable, then

$$\hat{t} := \frac{t}{t_0}, \quad \hat{x} := \frac{x}{x_0}, \quad \hat{u} := \frac{u}{u_0}, \quad \text{and} \quad \hat{v} := \frac{v}{v_0}. \tag{4.8}$$

Inserting these relations into (4.1) leads to

$$\partial_{\hat{t}} \hat{u} + \left( \frac{v_0/u_0}{x_0/t_0} \right) \partial_{\hat{x}} \hat{v} = 0, \tag{4.9a}$$

$$\partial_{\hat{t}} \hat{v} + a_F^2 \left( \frac{u_0/v_0}{x_0/t_0} \right) \partial_{\hat{x}} \hat{u} = -\frac{1}{\tau/t_0} \left( \hat{v} - a_E \frac{u_0}{v_0} \hat{u} \right). \tag{4.9b}$$

Assuming a unity wave speed in (4.9a), hence

$$\frac{v_0/u_0}{x_0/t_0} = 1 \quad \longrightarrow \quad \frac{v_0}{u_0} = \frac{x_0}{t_0}, \tag{4.10}$$

does not change the problem, and the above equations become

$$\partial_{\hat{t}}\hat{u} + \partial_{\hat{x}}\hat{v} = 0, \tag{4.11a}$$

$$\partial_{\hat{t}}\hat{v} + \left(\frac{a_F}{x_0/t_0}\right)^2 \partial_{\hat{x}}\hat{u} = -\frac{1}{\tau/t_0}\left(\hat{v} - \frac{a_E}{x_0/t_0}\hat{u}\right). \tag{4.11b}$$

Now, the proper reference time $t_0$ and reference speed $\dfrac{x_0}{t_0}$ have to be chosen for the nondimensionalization. Available constant parameters are $a_F$ [LT$^{-1}$], $a_E$ [LT$^{-1}$], and $\tau$ [T]. Also, let $L$ [L] be a length scale of interest, which may vary within a problem. As to a reference time, the obvious choice is $t_0 = \tau$; in this scaling, time is measured at a scale of the same order of the relaxation process. A next possible scaling is $t_0 = \dfrac{L}{a_F}$ where time is scaled by the traveling time of frozen waves. The equilibrium speed can be used as scaling when $a_E \neq 0$, thus $t_0 = \dfrac{L}{a_E}$. Note that $\dfrac{L}{a_F} \leq \dfrac{L}{a_E}$. Another nonintuitive choice is $t_0 = \dfrac{L^2}{\tau a_F^2}$.

As a reference speed $\dfrac{x_0}{t_0}$, both frozen speed $a_F$ and equilibrium speed $a_E$ are the obvious choices; the characteristic speed of relaxation $\dfrac{L}{\tau}$ might be a possible choice as well. The specific forms of each scaling are discussed in the next section under the assumption $u_0 = O(1)$.

### Scaling 1: Relaxation Time Scale ($t_0 = \tau$)

Choosing the relaxation time as the reference time ($t_0 = \tau$) eliminates the $\tau$ dependence in the equations when frozen wave speed is selected as reference speed $x_0/t_0 = a_F$, the dimensionless variables (4.8) become

$$\tilde{t} = \frac{t}{\tau}, \quad \tilde{x} = \frac{x}{a_F\tau}, \quad \tilde{u} = \frac{u}{u_0}, \quad \text{and} \quad \tilde{v} = \frac{v}{a_F u_0}, \tag{4.12}$$

and we set $\tilde{t} = \tilde{u} = O(1)$. In this scaling, the dimensionless GHHE (4.11) becomes

$$\partial_{\tilde{t}}\tilde{u} + \partial_{\tilde{x}}\tilde{v} = 0, \tag{4.13a}$$

$$\partial_{\tilde{t}}\tilde{v} + \partial_{\tilde{x}}\tilde{u} = -(\tilde{v} - r\tilde{u}), \tag{4.13b}$$

where

$$r := \frac{a_E}{a_F}, \quad |r| \leq 1, \tag{4.14}$$

is the dimensionless equilibrium speed. Note that in this scaling, there is no reduced asymptotic form since the time scale of interest is set to be that of relaxation process.

### Scaling 2: Frozen-Wave Time Scale ($t_0 = L/a_F$)

In this scaling, the reference time is scaled by $t_0 = \dfrac{L}{a_F}$, and the frozen wave speed $a_F$ is chosen as the reference speed. Then, the dimensionless variables (4.8) become

$$\hat{t} = \frac{t}{L/a_F}, \quad \hat{x} = \frac{x}{L}, \quad \hat{u} = \frac{u}{u_0}, \quad \text{and} \quad \hat{v} = \frac{v}{a_F u_0}, \tag{4.15}$$

and we set $\hat{t} = \hat{u} = O(1)$. Under these scaling, the dimensionless GHHE become

$$\partial_{\hat{t}}\hat{u} + \partial_{\hat{x}}\hat{v} = 0, \tag{4.16a}$$

$$\partial_{\hat{t}}\hat{v} + \partial_{\hat{x}}\hat{u} = -\frac{1}{\epsilon}(\hat{v} - r\hat{u}), \tag{4.16b}$$

where

$$\epsilon := \frac{\tau a_F}{L} > 0, \tag{4.17}$$

is the dimensionless relaxation time, and $r$ is defined by (4.14). Recall that both $\tau$ and $a_F$ are fixed, yet $L$ could be changed. Thus, $\epsilon$ is a function of the length scale of interest $L$. We are particularly interested in the nondimensional form of the GHHE in this frozen-wave time scale since the above equations reduce to the advection-dominated advection-diffusion equation,

$$\partial_{\hat{t}}\hat{u} + r\partial_{\hat{x}}\hat{u} = \epsilon\,(1 - r^2)\partial_{\hat{x}\hat{x}}\hat{u} + O\big(\epsilon^2\big)\,, \tag{4.18}$$

in the near-equilibrium limit ($\epsilon \ll O(1)$); the derivation is shown in Appendix C on page 354. Note that the inverse of the dimensionless relaxation time $\epsilon$ can be seen as the dimensionless time $\bar{t}$ defied by (4.7):

$$\frac{1}{\epsilon} = \frac{L/a_F}{\tau} = \frac{T}{\tau} \quad \rightarrow \quad \bar{t} \equiv \frac{1}{\epsilon}. \tag{4.19}$$

Hence, the smaller the relaxation time $\epsilon$, the longer the time scale for the GHHE, and eventually the GHHE reduces to the advection equation in the limit where $\epsilon \rightarrow 0$.

### Scaling 3: Equilibrium-Wave Time Scale ($t_0 = L/a_E$)

In this scaling, the reference time is scaled by $t_0 = \dfrac{L}{a_E}$, and the equilibrium wave speed $a_E$ is chosen as reference speed. Thus, the dimensionless variables (4.8) become

$$\check{t} = \frac{t}{L/a_E}, \quad \check{x} = \frac{x}{L}, \quad \check{u} = \frac{u}{u_0}, \quad \text{and} \quad \check{v} = \frac{v}{a_E u_0}, \tag{4.20}$$

and we set $\check{t} = \check{u} = O(1)$. Under these scaling, the dimensionless GHHE becomes

$$\partial_{\check{t}}\check{u} + \partial_{\check{x}}\check{v} = 0, \tag{4.21a}$$

$$\partial_{\check{t}}\check{v} + \frac{1}{r^2}\partial_{\check{x}}\check{u} = -\frac{1}{\epsilon r}(\check{v} - \check{u}), \tag{4.21b}$$

where $r$ and $\epsilon$ are defined by (4.14) and (4.17), respectively. The reduced equation in the near-equilibrium limit is

$$\partial_{\check{t}}\check{u} + \partial_{\check{x}}\check{u} = \frac{\epsilon\left(1 - r^2\right)}{r}\partial_{\check{x}\check{x}}\check{u} + O\left(\epsilon^2\right), \tag{4.22}$$

where the dimensionless advection speed is 1, and the viscosity is the inverse of the Peclet number defined by

$$Pe := \frac{r}{\epsilon\left(1 - r^2\right)}. \tag{4.23}$$

### Relation of Dimensionless Parameters

Even though we are interested in the frozen-time scale (4.16), and further analysis focusses on this scaling, the results can be transformed to other scales by using the following relations:

frozen-wave time → relaxation time:

$$\tilde{t} = \frac{\hat{t}}{\epsilon}, \quad \tilde{x} = \frac{\hat{x}}{\epsilon}, \quad \tilde{u} = \hat{u}, \quad \text{and} \quad \tilde{v} = \hat{v}, \tag{4.24a}$$

frozen-wave time → equilibrium-wave time:

$$\check{t} = r\hat{t}, \quad \check{x} = \hat{x}, \quad \check{u} = \hat{u}, \quad \text{and} \quad \check{v} = \frac{\hat{v}}{r}. \tag{4.24b}$$

### 4.2.3 Nondimensional Form

**Symmetric Frozen-Wave-Speeds Model**

Among the various nondimensionalization, we adopt the second scaling, frozen-wave time scale, for the following analysis. For simplicity, the notation $\hat{(\cdot)}$ is henceforth omitted, and our target model equations are written as

$$\partial_t u + \partial_x v = 0, \tag{4.25a}$$

$$\partial_t v + \partial_x u = -\frac{1}{\epsilon}(v - ru). \tag{4.25b}$$

Here, $u$ is the conserved variable, $v$ is the flux of $u$, and $\epsilon > 0$ is a dimensionless relaxation time. In vector form, $\mathbf{u} = [u, v]^T, \mathbf{f} = [v, u]^T$, and $\mathbf{s} = [0, ru - v]^T$ in

$$\partial_t \mathbf{u} + \partial_x \mathbf{f}(\mathbf{u}) = \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}). \tag{4.26}$$

This system has 'frozen' wave speeds ±1 when relaxation is weak ($\epsilon \gg 1$); when relaxation dominates ($\epsilon \ll 1$), it reduces to the advection-diffusion equation,

$$\partial_t u + r\partial_x u = \epsilon(1 - r^2)\partial_{xx} u + O\big(\epsilon^2\big)\,, \tag{4.27}$$

with an 'equilibrium' wave speed $r$. For stability, $|r| \leq 1$. Note that we have written this equation in a form that leads to an advection-dominated advection-diffusion asymptotic limit. This is consistent with a focus on compressible, viscous flow. Other choices of scalings, such as diffusive scalings [LM02, NP00], can lead to more strongly parabolic limits. Indeed, for $r = O(\epsilon)$, the scaling of Lowrie and Morel [LM02] is in effect a long-time, small-advective-flux limit.

### Asymmetric Frozen-Wave-Speeds Model

The system (4.25) can be generalized to break symmetry in the frozen limit [HSvL05]:

$$\partial_t u + \partial_x v = 0, \tag{4.28a}$$

$$\partial_t v + c\partial_x u + (1-c)\partial_x v = -\frac{1}{\epsilon}(v - ru). \tag{4.28b}$$

The frozen wave speeds are thus $-c$ and 1, and the near-equilibrium form is

$$\partial_t u + r\partial_x u = \epsilon(1-r)(c+r)\partial_{xx} u + O\big(\epsilon^2\big)\,. \tag{4.29}$$

Note the modification to the diffusion rate. For stability, $-c \leq r \leq 1$. This model is used only for limited cases due to the complexity of analysis.

### Exact Solution

In the reduced equation of the GHHE (4.27), the exact eigenvalue of the spatial differentiation operator for the harmonic mode $\mathbf{u}(x,t) = \hat{\mathbf{u}}_0 e^{(ikx+\lambda t)}$ is given by

$$\lambda_{\text{exact}}^{\text{GHHE}} = -irk - \epsilon(1-r^2)k^2 - 2i\epsilon^2 r(1-r^2)k^3 + O\big(\epsilon^3\big)\,. \tag{4.30}$$

Note that the above exact solution is indeed an infinite series. Conversely, the exact solution of the spatial differential operator of the genuine advection-diffusion equation runs only up to $O(\epsilon^2)$, thus

$$\lambda_{\text{exact}}^{\text{adv-diff}} = -irk - \epsilon(1-r^2)k^2. \tag{4.31}$$

## 4.3 Difference Operators and Their Properties in 1-D

Various discretization methods are applied to the linear hyperbolic-relaxation equations (4.25), and a Fourier analysis is conducted to uncover those properties. By this we can show, to a given order in $\epsilon$, if a scheme captures the advection-dominated advection-diffusion limit (4.27) with second-order accuracy in $\Delta x$. Similar analyses have been done using modified differential equations [JL96, LM02], though not always using the same scaling and limit. Furthermore, the analysis here also considers temporal discretization to reveal an issue of both spatial and temporal stiffness inherent in the system.

### 4.3.1 Operator-Splitting Method

At first, to demonstrate an extra difficulty arising due to the stiff source term, operator splitting is adopted in the time integrator. This splitting decouples the time evolution of the flux and source terms, allowing us to compute these independently. The great advantage of this method, particularly for hyperbolic-relaxation equations, is that the source term, which yields exponential damping, can be integrated exactly. In order to isolate the error introduced by the operator splitting, we eliminate the spatial discretization error by taking the flux derivative from the exact solution. Thus, the operator-split update operator for (4.25) takes the form

$$\begin{aligned} \mathbf{u}^{(1)} &= e^{\frac{\Delta t}{2}\frac{1}{\epsilon}\mathbf{Q}}\mathbf{u}^{n}, \\ \mathbf{u}^{(2)} &= e^{-ik\Delta t\mathbf{A}}\mathbf{u}^{(1)}, \\ \mathbf{u}^{n+1} &= e^{\frac{\Delta t}{2}\frac{1}{\epsilon}\mathbf{Q}}\mathbf{u}^{(2)}, \end{aligned} \tag{4.32}$$

where

$$\mathbf{A} := \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{Q} = \begin{pmatrix} 0 & 0 \\ r & -1 \end{pmatrix}. \tag{4.33}$$

Following the same procedure as in the previous chapter, the local truncation error in the low-frequency limit is found to be

$$\begin{aligned}\mathrm{LTE}_{\mathrm{splitting}} &= \left[\frac{(1+e^{\Delta t/\epsilon})(1-r^2)\Delta t}{2(1-e^{\Delta t/\epsilon})} + \epsilon(1-r^2)\right] k^2 + O\big(k^3\big) \\ &\simeq -\frac{1-r^2}{12}\frac{\boxed{\Delta t^2}}{\epsilon} k^2 = -\frac{(1-r^2)\nu^2}{12}\frac{\boxed{\Delta x^2}}{\epsilon} k^2, \end{aligned} \tag{4.34}$$

where the Courant number is defined by

$$\nu := 1\frac{\Delta t}{\Delta x}. \tag{4.35}$$

The above equation shows that the splitting is second-order in space and time. However, since the above error is in the $k^2$-term, an extra numerical dissipation is added to the physical dissipation $-\epsilon(1-r^2)k^2$ in (4.30); this leads to an incorrect diffusion coefficient. To ensure the physical dissipation is dominant, the following inequality has to be satisfied:

$$\frac{(1-r^2)\nu^2}{12}\frac{\Delta x^2}{\epsilon} \ll \epsilon(1-r^2) \quad \longrightarrow \quad \nu\Delta x \ll \epsilon. \tag{4.36}$$

Particularly, when the near-equilibrium limit ($\epsilon \ll 1$) is considered, the time step and grid size are severely restricted such that

$$\Delta t = \nu\Delta x \propto \epsilon, \tag{4.37}$$

otherwise the excessive numerical dissipation damps all waves in the domain.

The above example shows that straightforward decoupling of the flux and source term leads to an accurate method in the near-equilibrium limit only when (4.37) is satisfied. In order to overcome this severe restriction, coupling between flux and source term is necessary; for instance, an MOL with several stages, or a fully discrete method in which the flux has strong coupling with the source term.

### 4.3.2 HR–MOL Method

A semi-discrete high-resolution Godunov method (HR–MOL), particularly, the second-order method, is considered. The HR2–MOL method applied to the asymmetric GHHE equation (4.28) has the following generic form:

$$\frac{\partial \bar{\mathbf{u}}_j(t)}{\partial t} = -\frac{1}{\Delta x}\left(\hat{\mathbf{f}}_{j+1/2} - \hat{\mathbf{f}}_{j-1/2}\right) + \frac{1}{\epsilon}\mathbf{s}\left(\bar{\mathbf{u}}_j\right), \tag{4.38}$$

where $\hat{\mathbf{f}}_{j\pm 1/2}$ denotes the approximate flux at interfaces $j \pm 1/2$. We will take this to be the upwind flux

$$\hat{\mathbf{f}}_{j+1/2}\left(\mathbf{u}_L, \mathbf{u}_R\right) = \mathbf{A}^{+}\mathbf{u}_L + \mathbf{A}^{-}\mathbf{u}_R, \tag{4.39}$$

where, if $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{A}$, $\mathbf{A}^{\pm} = \mathbf{R}\mathbf{\Lambda}^{\pm}\mathbf{L}$. In the case of an asymmetric system (4.28),

$$\mathbf{A}^{+} = \frac{1}{1+c}\begin{pmatrix} c & 1 \\ c & 1 \end{pmatrix}, \quad \mathbf{A}^{-} = \frac{c}{1+c}\begin{pmatrix} -1 & 1 \\ c & -c \end{pmatrix}. \tag{4.40}$$

After inserting the difference forms into the original ODE (4.38) and some algebra, the semi-discrete method can be written in the compact form:

$$\frac{\partial \bar{\mathbf{u}}_j(t)}{\partial t} = \left(\mathbf{N}_{\text{HR2}} + \frac{1}{\epsilon}\mathbf{Q}\right)\bar{\mathbf{u}}_j(t), \tag{4.41}$$

where $\bar{\mathbf{u}}_j = \left[\bar{u}_j, \bar{v}_j\right]^T$; the difference operator of the flux derivative $\mathbf{N}_{\text{HR2}}$ is given by

$$\mathbf{N}_{\text{HR2}} = -\frac{1}{4(1+c)\Delta x}\left(\boldsymbol{\mathcal{A}}^{2+}\mathbf{D}^{2+} + \boldsymbol{\mathcal{A}}^{+}\mathbf{D}^{+} + \boldsymbol{\mathcal{A}}^{-}\mathbf{D}^{-} + \boldsymbol{\mathcal{A}}^{2-}\mathbf{D}^{2-}\right), \tag{4.42}$$

where

$$\boldsymbol{\mathcal{A}}^{+} = \begin{pmatrix} -2c & 1+3c \\ c(1+3c) & 1-3c^2 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}^{-} = \begin{pmatrix} 2c & 3+c \\ c(3+c) & 3-c^2 \end{pmatrix} \tag{4.43a}$$

$$\boldsymbol{\mathcal{A}}^{2+} = \begin{pmatrix} c & -c \\ -c^2 & c^2 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}^{2-} = \begin{pmatrix} c & 1 \\ c & 1 \end{pmatrix}, \quad \mathbf{D}^{\pm} = \delta^{\pm}\mathbf{I}, \quad \mathbf{D}^{2\pm} = (\delta^{\pm})^2\mathbf{I}, \tag{4.43b}$$

or, for a Fourier mode,

$$\mathbf{N}_{\mathrm{HR2}} = -\frac{1}{4(1+c)\Delta x}\left(\boldsymbol{\mathcal{A}}^{2+}e^{2i\beta\mathbf{I}} + \boldsymbol{\mathcal{A}}'^{+}e^{i\beta\mathbf{I}} + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{A}}'^{-}e^{-i\beta\mathbf{I}} + \boldsymbol{\mathcal{A}}^{2-}e^{-2i\beta\mathbf{I}}\right), \quad (4.44)$$

where

$$\boldsymbol{\mathcal{A}}'^{+} = \begin{pmatrix} -4c & 1+5c \\ c(1+5c) & 1-5c^2 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}'^{-} = \begin{pmatrix} -4c & -1-5c \\ -c(5+c) & -5+c^2 \end{pmatrix}, \quad (4.45a)$$

$$\boldsymbol{\mathcal{C}} = \begin{pmatrix} 6c & 3(1-c) \\ 3c(1-c) & 3(1+c^2) \end{pmatrix}. \quad (4.45b)$$

In order to obtain the eigenvalues of the HR2 spatial discretization, the quadratic characteristic equation,

$$\det\left(\mathbf{N}_{\mathrm{HR2}} + \frac{1}{\epsilon}\mathbf{Q} - \lambda\mathbf{I}\right) = 0, \quad (4.46)$$

is solved. For simplicity, we only present the case when $c = 1$; the two roots have following form:

$$\lambda^{(1),(2)}_{\mathrm{HR2}} = \frac{(1-\cos\beta)^2}{2\Delta x} + \frac{1}{2\epsilon}\left[1 \mp \sqrt{1 - 2ir\frac{\epsilon}{\Delta x}(3-\cos\beta)\sin\beta + \left(\frac{\epsilon}{\Delta x}(3-\cos\beta)\sin\beta\right)^2}\,\right]. \quad (4.47)$$

**Spatial Accuracy** We expand the trigonometric factors in (4.47) for the long wave-length limit $\beta \ll 1$, and then expand the square root. The results are

$$\begin{aligned} \lambda^{(1)}_{\mathrm{HR2}} = {} & -\frac{ir}{\Delta x}\beta - \frac{\epsilon(1-r^2)}{\Delta x^2}\beta^2 - \left[\frac{ir}{12\Delta x} + \frac{2i\epsilon^2 r(1-r^2)}{\Delta x^3}\right]\beta^3 \\ & - \left[\frac{1}{8\Delta x} + \frac{\epsilon(1-r^2)}{6\Delta x^2} + \frac{\epsilon^2(1-r^2)(1-5r^2)}{\Delta x^4}\right]\beta^4 + O(\beta^5), \end{aligned} \quad (4.48a)$$

$$\lambda^{(2)}_{\mathrm{HR2}} = -\frac{1}{\epsilon} + \frac{ir}{\Delta x}\beta + O(\beta^2). \quad (4.48b)$$

The second root (4.48b) exhibits rapid exponential decay for $\epsilon \ll 1$, while the first root (4.48a) does not, thus the latter $\lambda^{(1)}_{\mathrm{HR2}}$ is the dominant behavior in this asymptotic limit. Following the analysis of the linear advection equation in the previous chapter, the spatial discretization error is obtained by replacing $\beta$ by the wave number $k := \dfrac{\beta}{\Delta x}$, then

$$\lambda^{(1)}_{\text{HR2}} - \lambda^{\text{GHHE}}_{\text{exact}} = \underbrace{-\frac{ir}{12}\boxed{\Delta x^2}}_{\text{dispersion error}}\; k^3 - \underbrace{\left[\frac{1}{8}\boxed{\Delta x^3} + \frac{\epsilon(1-r^2)}{6}\Delta x^2\right]}_{\text{dissipation error}} k^4 + O\big(k^5\big)\,, \tag{4.49a}$$

$$\lambda^{(2)}_{\text{HR2}} - \lambda^{\text{GHHE}}_{\text{exact}} = -\frac{1}{\epsilon} + 2irk + O\big(k^2\big)\,, \tag{4.49b}$$

where the exact spatial differential operator $\lambda^{\text{GHHE}}_{\text{exact}}$ is given in (4.30). In the first equation, both dispersive ($k^3$-term) and dissipative ($k^4$-term) errors are present; the dispersive error is second-order in $\Delta x$, and since the correct limit shows no dispersion, these numerical dispersion errors can not be confused with any physical dispersion. However, the leading dissipative error term, $-\frac{1}{8}\Delta x^3 k^4$, does not scale with $\epsilon$, and so can compete with the physical dissipation $-\epsilon\,(1-r^2)k^2$ in (4.30) if the relaxation scale is unresolved ($\Delta x \gg \epsilon$). For the physical dissipation to dominate,

$$\text{(dominant numerical dissipation)} \ll \text{(physical dissipation)}, \tag{4.50}$$

thus,

$$\frac{1}{8}\Delta x^3 k^4 \ll \epsilon(1-r^2)k^2. \tag{4.51}$$

Solving for $\Delta x$ leads to the threshold grid size $\Delta h^*_{\text{HR2}}$:

$$\Delta x \ll 2\left[\frac{\epsilon(1-r^2)}{k^2}\right]^{1/3} = 2\left(\frac{r}{k^2\,Pe}\right)^{1/3}, \tag{4.52}$$

thus,

$$\Delta h^*_{\text{HR2}} := 2\left(\frac{r}{k^2\,Pe}\right)^{1/3}. \tag{4.53}$$

Hence, the HR2 scheme does not attain the asymptotic limit to second-order in $\Delta x$ with $\Delta x$ independent of $\epsilon$.

For the HR2 scheme, the result (4.52) is well known. It appears for the $r = 0$ case in previous studies, [JL96, LM02] although not necessarily in our scaling. The form (4.49a) for $r \neq 0$ for the HR2 scheme can be obtained from Eq. (3.17) in Jin and Levermore [JL96, p. 461] or Eq. (31) in Lowrie and Morel [LM02, p. 420]. In the scaling of the latter work, the term that leads to (4.52) is actually divergent since it goes like $\dfrac{\Delta x^3}{\epsilon}$ (due to the time dilation of this scaling), but it still leads to the constraint (4.52). Jin and Levermore claimed that the grid size restriction can be removed by averaging the frozen and equilibrium fluxes [Jin95]. However, we find that the grid restriction still exists in their method. The detailed analysis is described in Appendix D.

It is interesting to compare (4.52) with a direct discretization of the asymptotic equation (4.27) on page 215 using the Rusanov flux function and slope reconstruction. The diffusion term is discretized using a three-point, second-order central-difference approximation. From a Fourier analysis, the eigenvalue of the scheme is

$$\lambda_{\mathrm{HR2}}^{\text{adv-diff}} - \lambda_{\mathrm{exact}}^{\text{adv-diff}} = -\frac{ir}{12}\boxed{\Delta x^2}\,k^3 - \left[\frac{1}{8}\boxed{\Delta x^3} - \frac{\epsilon(1-r^2)}{12}\Delta x^2\right]k^4 + O\big(k^5\big)\,, \quad (4.54)$$

for $\Delta x \ll 1$. We see that this has the same fourth-order numerical dissipation term as the HR2 discretization of GHHE (4.49a). This discretization will have the same restriction (4.52) on $\Delta x$ to ensure that the physical dissipation is dominant. When this restriction is satisfied, the above equation shows that the HR2 method is second-order in space owing to $\Delta x^2$ in the $k^3$-term.

**Spatial-Temporal Accuracy** In our previous analysis, we only consider the spatial discretization of the HR2 method under the assumption that the flux and source term are discretized at the same time level [HSvL05]. However, a typically

ODE solver for a stiff problem discretizes the flux and source terms at different time levels due to the implicit treatment of the source term. Furthermore, the system (4.25) possesses both spatial and temporal stiffness, thus, analyzing a fully discrete form of any method is necessary. A great number of stiff ODE solvers have been proposed in the last few decades [Jin95, CJR97, LRR00, NP00, TSL00, HW96, Lam91]. Among these methods, we adopt the implicit-explicit (IMEX) Runge–Kutta methods originally developed by Ascher et al. [ARW95, ARS97] for hyperbolic-parabolic equations, and later extended to hyperbolic-relaxation equations by Pareschi and Russo [PR05]. The methods treat the flux term explicitly by a strong-stability-preserving (SSP) method, and the source term by an $L$-stable diagonally implicit Runge–Kutta method (DIRK). The authors developed a family of second and third-order methods. Here, as an example, we adopt the IMEX–SSP2(3,3,2) method; both explicit and implicit methods require three stages, and overall accuracy is second-order. The actual update formulas are the following:

$$\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}^n + \frac{\Delta t}{4\epsilon}\mathbf{s}(\mathbf{u}^{(1)}), \\
\mathbf{u}^{(2)} &= \mathbf{u}^n - \frac{\Delta t}{2}\partial_x\mathbf{f}(\mathbf{u}^{(1)}) + \frac{\Delta t}{4\epsilon}\mathbf{s}(\mathbf{u}^{(2)}), \\
\mathbf{u}^{(3)} &= \mathbf{u}^n - \frac{\Delta t}{2}\left[\partial_x\mathbf{f}(\mathbf{u}^{(1)}) + \partial_x\mathbf{f}(\mathbf{u}^{(2)})\right] + \frac{\Delta t}{3\epsilon}\left[\mathbf{s}(\mathbf{u}^{(1)}) + \mathbf{s}(\mathbf{u}^{(2)}) + \mathbf{s}(\mathbf{u}^{(3)})\right], \\
\mathbf{u}^{n+1} &= \mathbf{u}^n - \frac{\Delta t}{3}\left[\partial_x\mathbf{f}(\mathbf{u}^{(1)}) + \partial_x\mathbf{f}(\mathbf{u}^{(2)}) + \partial_x\mathbf{f}(\mathbf{u}^{(3)})\right] \\
&\qquad + \frac{\Delta t}{3\epsilon}\left[\mathbf{s}(\mathbf{u}^{(1)}) + \mathbf{s}(\mathbf{u}^{(2)}) + \mathbf{s}(\mathbf{u}^{(3)})\right].
\end{aligned} \tag{4.55}$$

The development of the truncation-error analysis is based on the method described in Chapter III. In order to obtain an explicit form for roots of a characteristic equation, due to the complexity of any amplification matrix $\mathbf{G}$, we assume a power-series form for the amplification factor,

$$g_{\text{method}} = g_0 + g_1 k + g_2 k^2 + O\left(k^3\right), \tag{4.56}$$

for the wave number $k \ll 1$. In brief, once an amplification factor $g$ is obtained from an amplification matrix $\mathbf{G}$, the local truncation error is obtained by

$$\mathrm{LTE}_{\mathrm{method}} = \sum_{n=2}^{\infty} c_n k^n, \tag{4.57}$$

where the coefficient $c_n$ is

$$c_n = \frac{1}{n!\Delta t g(k=0)} \left. \frac{\partial^n g(k)}{\partial k^n} \right|_{k=0}. \tag{4.58}$$

The intermediate formulas for the derivation of the local truncation error are omitted here; only the final result is presented. The local truncation error of the HR2–IMEX method becomes

$$\begin{aligned} \mathrm{LTE}_{\mathrm{HR2IMEX}} = & \left[ \underbrace{-\frac{ir}{12}\left(1+(r\nu)^2\right)\boxed{\Delta x^2}}_{\text{dominant dispersion error}} - \frac{i\epsilon\, r(1-r^2)\nu}{6}\Delta x \right] k^3 \\ & - \left[ \underbrace{\frac{1}{8}\left(1-\frac{r}{3}(r\nu)^3\right)\boxed{\Delta x^3}}_{\text{dominant dissipation error}} + \frac{\epsilon\,(1-r^2)}{6}\Delta x^2 \right] k^4. \end{aligned} \tag{4.59}$$

Under the assumption of near-equilibrium we have $r = O(1)$ and $\epsilon \ll 1$; when the physical dissipation dominates over the dissipation error,

$$\frac{1}{8}\left(1-\frac{r}{3}(r\nu)^3\right)\Delta x^3 k^4 \ll \epsilon\,(1-r^2)k^2, \tag{4.60}$$

then the method is second-order in both space and time owing to the dominant dispersion error in the $k^3$-term. Note that the remaining terms in (4.59) are guaranteed to be always smaller than the physical dispersion and dissipation owing to $\epsilon$ in their coefficients. Finally, the threshold grid size for the fully discrete method is given by

$$\Delta h^*_{\mathrm{HR2IMEX}} := 2\left[\frac{\epsilon(1-r^2)}{\left(1-\frac{r}{3}(r\nu)^3\right)k^2}\right]^{1/3}. \tag{4.61}$$

### 4.3.3 DG–MOL Method

Within cell $j$, the second-order DG method uses a linear basis

$$\mathbf{u}(x,t) = (1-\xi)\,\mathbf{u}_1(t) + \xi\,\mathbf{u}_2(t), \qquad \xi = \frac{x - x_{j-1/2}}{\Delta x} \in [0,1], \tag{4.62}$$

such that $\mathbf{u}(x_{j-1/2}) = \mathbf{u}_1$ and $\mathbf{u}(x_{j+1/2}) = \mathbf{u}_2$ are the solution values in cell $j$ at the extreme left and right edges, respectively. For linear flux ($\mathbf{f}(\mathbf{u}) = \mathbf{A}\mathbf{u}$) and source functions ($\mathbf{s}(\mathbf{u}) = \mathbf{Q}\mathbf{u}$), the semi-discrete DG(1) scheme is then

$$\frac{\partial \mathbf{u}_1(t)}{\partial t} = -\frac{1}{\Delta x}\left[-4\hat{\mathbf{f}}_{j-1/2} - 2\hat{\mathbf{f}}_{j+1/2} + 3\mathbf{f}(\mathbf{u}_1) + 3\mathbf{f}(\mathbf{u}_2)\right] + \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_1), \tag{4.63a}$$
$$\frac{\partial \mathbf{u}_2(t)}{\partial t} = -\frac{1}{\Delta x}\left[4\hat{\mathbf{f}}_{j+1/2} + 2\hat{\mathbf{f}}_{j-1/2} - 3\mathbf{f}(\mathbf{u}_1) - 3\mathbf{f}(\mathbf{u}_2)\right] + \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}_2), \tag{4.63b}$$

where the upwind flux function is to be inserted

$$\hat{\mathbf{f}}_{j+1/2}\left(\mathbf{u}_L, \mathbf{u}_R\right) = \hat{\mathbf{f}}_{j+1/2}\left(\mathbf{u}_{1,j+1}, \mathbf{u}_{2,j}\right). \tag{4.64}$$

To relate (4.63) to the HR2–MOL (4.38) method, we define

$$\bar{\mathbf{u}}_j = \frac{1}{2}\left(\mathbf{u}_1 + \mathbf{u}_2\right) \quad \text{and} \quad \Delta\mathbf{u}_j = \mathbf{u}_2 - \mathbf{u}_1, \tag{4.65}$$

such that $\mathbf{u}(x,t) = \bar{\mathbf{u}}_j(t) + \left(\xi - \frac{1}{2}\right)\overline{\Delta\mathbf{u}}_j(t)$. The DG(1)–MOL scheme (4.63) is then

$$\frac{\partial \bar{\mathbf{u}}_j(t)}{\partial t} = -\frac{1}{\Delta x}\left(\hat{\mathbf{f}}_{j+1/2} - \hat{\mathbf{f}}_{j-1/2}\right) + \frac{1}{\epsilon}\mathbf{s}\left(\bar{\mathbf{u}}_j\right), \tag{4.66a}$$
$$\frac{\partial \overline{\Delta\mathbf{u}}_j(t)}{\partial t} = -\frac{6}{\Delta x}\left(\hat{\mathbf{f}}_{j+1/2} + \hat{\mathbf{f}}_{j-1/2} - 2\mathbf{f}(\bar{\mathbf{u}}_j)\right) + \frac{1}{\epsilon}\mathbf{s}\left(\overline{\Delta\mathbf{u}}_j\right), \tag{4.66b}$$

where the upwind flux function becomes

$$\hat{\mathbf{f}}_{j+1/2}\left(\mathbf{u}_L, \mathbf{u}_R\right) := \hat{\mathbf{f}}_{j+1/2}\left(\bar{\mathbf{u}}_j + \frac{1}{2}\overline{\Delta\mathbf{u}}_j, \bar{\mathbf{u}}_{j+1} - \frac{1}{2}\overline{\Delta\mathbf{u}}_{j+1}\right). \tag{4.67}$$

The first update equation (4.66a) with (4.67) is precisely the HR2 method (modulo limiting) where $\frac{\overline{\Delta\mathbf{u}}_j}{\Delta x}$ is the slope in cell $j$. For the HR2 method, the differences $\overline{\Delta\mathbf{u}}_j$

are approximated at each step by differencing neighboring cell-averaged values $\bar{\mathbf{u}}_{j\pm1}$, whereas in the DG(1) method, the slopes evolve as additional variables.

It is these slopes, whether computed or self-evolving, that are responsible for providing second-order spatial accuracy in the flux evaluation. It is also these slopes that provide the distinction between the two schemes.

For length scales much larger than the relaxation length scale $a\tau$, the flux discretization must approximate the coupling between the two hyperbolic and relaxation operators. For an HR method, the flux function is based solely on the original hyperbolic operator and each slope purely on the initial data. In contrast, the DG method *simultaneously* updates the solution average and slope under the influence of the source.

It is interesting to conduct a Fourier analysis of the one-dimensional DG(1) method (4.63) for the asymmetric system (4.28) on page 216 as $\epsilon \to 0$. Following the previous analysis, take $\mathbf{u}_j = [\bar{\mathbf{u}}_j, \overline{\Delta\mathbf{u}}_j]^T$, then the difference form of the DG(1) method can be written as

$$\frac{\partial \mathbf{u}_j(t)}{\partial t} = \left(\mathbf{N}_{\mathrm{DG}(1)} + \frac{1}{\epsilon}\mathbf{Q}_{\mathrm{DG}(1)}\right)\mathbf{u}_j(t). \tag{4.68}$$

Here, the difference operator of the flux discretization is given by

$$\mathbf{N}_{\mathrm{DG}(1)} = \boldsymbol{\mathcal{A}}^{+}\mathbf{D}^{+} + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{A}}^{-}\mathbf{D}^{-}, \tag{4.69}$$

where

$$\boldsymbol{\mathcal{A}}^{+} = \frac{1}{(1+c)\Delta x}\begin{pmatrix} c & -c & -\dfrac{c}{2} & \dfrac{c}{2} \\ -c^2 & c^2 & \dfrac{c^2}{2} & -\dfrac{c^2}{2} \\ 6c & -6c & -3c & 3c \\ -6c^2 & 6c^2 & 3c^2 & -3c^2 \end{pmatrix}, \tag{4.70a}$$

$$\boldsymbol{\mathcal{A}}^{-} = \frac{1}{(1+c)\Delta x}\begin{pmatrix} -c & -1 & -\dfrac{c}{2} & -\dfrac{1}{2} \\ -c & -1 & -\dfrac{c}{2} & -\dfrac{1}{2} \\ 6c & 6 & 3c & 3 \\ 6c & 6 & 3c & 3 \end{pmatrix}, \tag{4.70b}$$

$$\boldsymbol{\mathcal{C}} = \frac{1}{(1+c)\Delta x}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -12c & -6(1-c) \\ 0 & 0 & -6c(1-c) & -6(1+c^2) \end{pmatrix}, \tag{4.70c}$$

and

$$\mathbf{Q}_{\mathrm{DG}(1)} = \begin{pmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{Q} \end{pmatrix}. \tag{4.71}$$

The difference operator (4.69) can be also written for a Fourier mode,

$$\mathbf{N}_{\mathrm{DG}(1)} = \boldsymbol{\mathcal{A}}^{+}\mathbf{D}^{+} + \boldsymbol{\mathcal{C}}' - \boldsymbol{\mathcal{A}}^{-}\mathbf{D}^{-}, \tag{4.72}$$

where

$$\boldsymbol{\mathcal{C}}' = \frac{1}{(1+c)\Delta x}\begin{pmatrix} -2c & -1+c & 0 & -\dfrac{1}{2}(1+c) \\ -c(1-c) & -(1+c^2) & -\dfrac{1}{2}c(1+c) & -\dfrac{1}{2}(1-c^2) \\ 0 & 6(1+c) & -6c & -3(1-c) \\ 6c(1+c) & 6(1-c^2) & -3c(1-c) & -3(1+c^2) \end{pmatrix}. \tag{4.73}$$

In order to obtain eigenvalues of the DG(1) spatial discretization, we compute the characteristic equation,

$$\det\left(\mathbf{N}_{\mathrm{DG}(1)} + \frac{1}{\epsilon}\mathbf{Q}_{\mathrm{DG}(1)} - \lambda\mathbf{I}\right) = 0, \tag{4.74}$$

and take the equilibrium $\epsilon = 0$ to obtain a quadratic equation in $\lambda$. (This will yield the leading-order term in $\lambda$.) This is

$$\begin{aligned}(\Delta x\lambda)^2 - \left[\left(\frac{2c+r(1-c)}{1+c}\right)(6+\delta^+-\delta^-) - r(\delta^++\delta^-)\right]\Delta x\lambda \\ + 3r\left[r(\delta^--\delta^+) + \left(\frac{2c+r(1-c)}{1+c}\right)(\delta^++\delta^-)\right] = 0,\end{aligned} \tag{4.75}$$

or, for a Fourier mode,

$$\begin{aligned}(\Delta x\lambda)^2 - 2\left[\left(\frac{2c+r(1-c)}{1+c}\right)(2+\cos\beta) - ir\sin\beta\right]\Delta x\lambda \\ + 6r\left[r(1-\cos\beta) + \left(\frac{2c+r(1-c)}{1+c}\right)i\sin\beta\right] = 0.\end{aligned} \tag{4.76}$$

**Spatial Accuracy** The DG(1) scheme in compact form is given in (4.68); we restrict ourselves here to the special case of $c = 1$. Since this is a $4 \times 4$ system, the characteristic polynomial is of degree four. The leading-order behavior is given by the quadratic equation (4.76). Using these to find the terms of $O(\epsilon)$, and, at each order, expanding out the trigonometric terms for $\beta \ll 1$, we find

$$\lambda^{(1)}_{\mathrm{DG}(1)} = -\frac{ir}{\Delta x}\beta - \frac{\epsilon(1-r^2)}{\Delta x^2}\beta^2 + O(\beta^3)\,, \tag{4.77a}$$

$$\lambda^{(2)}_{\mathrm{DG}(1)} = -\frac{1}{\epsilon} + \frac{ir}{\Delta x}\beta + \frac{\epsilon(1-r^2)}{\Delta x^2}\beta^2 + O(\beta^3)\,, \tag{4.77b}$$

$$\lambda^{(3)}_{\mathrm{DG}(1)} = -\frac{6}{\Delta x} + \frac{3ir}{\Delta x}\beta + \frac{\Delta x - 9\epsilon(1-r^2)}{\Delta x^2}\beta^2 + O(\beta^3)\,, \tag{4.77c}$$

$$\lambda^{(4)}_{\mathrm{DG}(1)} = -\frac{1}{\epsilon} - \frac{6}{\Delta x} - \frac{3ir}{\Delta x}\beta + \frac{\Delta x + 9\epsilon(1-r^2)}{\Delta x^2}\beta^2 + O(\beta^3)\,. \tag{4.77d}$$

The last three roots all exhibit rapid exponential decay for $\Delta x \ll 1$ and $\epsilon \ll 1$, while the first root does not. Since $\lambda^{(1)}_{\mathrm{DG}(1)}$ is the dominant root in the asymptotic limit, we continue expanding it; then the spatial discretization error becomes

$$\lambda^{(1)}_{\mathrm{DG}(1)} - \lambda^{\mathrm{GHHE}}_{\mathrm{exact}} = -\frac{1}{72}\left[\Delta x^3 - \frac{1-r^2}{\Delta x + 6\epsilon}\Delta x^4\right]k^4 + O(k^5)\,, \tag{4.78}$$

where we have made the substitution $k = \dfrac{\beta}{\Delta x}$. Since we are considering the near-equilibrium limit $\epsilon \ll 1$, expanding the above error with respect to $\epsilon$ provides

$$\lambda^{(1)}_{\mathrm{DG}(1)} - \lambda^{\mathrm{GHHE}}_{\mathrm{exact}} = -\frac{1}{72}\left[r^2 \boxed{\Delta x^3} + 6\epsilon(1-r^2)\Delta x^2 - 36\epsilon^2(1-r^2)\Delta x\right] k^4 + O\big(\epsilon^3 k^4, k^5\big)\,, \tag{4.79}$$

Again, we find a third-order numerical dissipation term independent of $\epsilon$ in the $k^4$-term that can compete with the physical second-order dissipation. The criterion for the physical dissipation to dominate is

$$\frac{1}{72} r^2 \Delta x^3 k^4 \ll \epsilon(1-r^2)k^2. \tag{4.80}$$

Solving for $\Delta x$ leads to the threshold grid size $\Delta h^*_{\mathrm{DG}(1)}$:

$$\Delta x \ll 2\left[\frac{9\epsilon(1-r^2)}{r^2k^2}\right]^{1/3} = 2\left(\frac{9}{rk^2\,Pe}\right)^{1/3}, \tag{4.81}$$

thus,

$$\Delta h^*_{\mathrm{DG}(1)} := 2\left(\frac{9}{rk^2\,Pe}\right)^{1/3}, \tag{4.82}$$

which is a factor of $\left(\dfrac{9}{r^2}\right)^{1/3}$ larger than for the HR2 scheme. When rescaled, this is the same result as Eq. (32) in Lowrie and Morel [LM02, p. 420] which was obtained from a modified differential-equation analysis.

We directly discretize the advection-diffusion limit (4.27) using the DG(1) scheme with the Rusanov flux function; this is equivalent to the HLL1 flux with $c = 1$. The diffusion term is discretized using the recently developed 'recovery method' [vLN05, vLLvR07]. The eigenvalues of spatial discretization from the Fourier analysis are

$$\lambda^{(1),\text{ adv-diff}}_{\mathrm{DG}(1)} - \lambda^{\text{adv-diff}}_{\mathrm{exact}} = -\frac{r^2\Delta x^4}{36\left[2\Delta x + 5\epsilon(1-r^2)\right]}k^4 + O\big(k^5\big)\,, \tag{4.83a}$$

$$\lambda^{(2),\text{ adv-diff}}_{\mathrm{DG}(1)} - \lambda^{\text{adv-diff}}_{\mathrm{exact}} = -\frac{6}{\Delta x} - \frac{15\epsilon(1-r^2)}{\Delta x^2} + O(k)\,. \tag{4.83b}$$

The dominant eigenvalue, the first equation, can be further expanded in terms of $\epsilon$ since we are assuming the near-equilibrium limit $\epsilon \ll 1$; then

$$\lambda^{\text{adv-diff}}_{\text{DG}(1)} - \lambda^{\text{adv-diff}}_{\text{exact}} = -\frac{1}{72}\left[r^2\,\boxed{\Delta x^3} - \frac{5\epsilon r^2(1-r^2)}{2}\Delta x^2\right]k^4 + O\big(\epsilon^2 k^4, k^5\big)\,. \quad (4.84)$$

Again, the dominant numerical dissipation is of precisely the same form as for the GHHE discretization (4.79).

Finally, we note that, for $r = O(\epsilon)$ with $\epsilon \ll 1$, the DG(1) scheme exhibits an interesting property. In this case, the fourth-order numerical dissipation term in (4.79) becomes higher-order in $\epsilon$, and the constraint (4.81) on $\Delta x$ is removed. Thus, the DG(1) scheme should converge with second-order accuracy with $\Delta x$ independent of $\epsilon$, since the higher-even-order terms are too small to compete with the physical dissipation. This case is included in the diffusive limit considered by Lowrie and Morel [LM02], and our result agrees with theirs when one accounts for the time dilation of their scaling.

**Spatial-Temporal Accuracy** Following the procedure used earlier, the local truncation error of the DG(1)–MOL combined with the IMEX–SSP2(3,3,2) is found to have following form:

$$\text{LTE}^{(1)}_{\text{DG(1)IMEX}} = -\left[\underbrace{\frac{ir(r\nu)^2}{12}\boxed{\Delta x^2}}_{\text{dominant dispersion error}} + \frac{i\epsilon\, r(1-r^2)\nu}{6}\Delta x\right]k^3 - \left[\underbrace{\frac{r\big(r-3(r\nu)^3\big)}{72}\boxed{\Delta x^3}}_{\text{dominant dissipation error}} + \frac{\epsilon(1-r^2)}{12}\Delta x^2\right]k^4. \quad (4.85)$$

Just as for the HR2 method, if the physical dissipation is dominant over the dissipation error,

$$\frac{r\big(r-3(r\nu)^3\big)}{72}\Delta x^3 k^4 \ll \epsilon(1-r^2)k^2, \quad (4.86)$$

the method is second-order in space and time. Solving the above equation for $\Delta x$ leads to the threshold grid size:

$$\Delta h^*_{\mathrm{DG(1)IMEX}} := 2\left[\frac{9\epsilon\left(1-r^2\right)}{r\big(r-3(r\nu)^3\big)k^2}\right]^{1/3}. \tag{4.87}$$

### 4.3.4 HR–Hancock Method

It is also interesting to compare the HR2–Hancock method to the DG(1)–Hancock method. The Hancock methodology is applied to the flux discretization, while a method for the source term is still to be chosen. Here, similar to the choice for the DG(1)–Hancock method, we apply the Radau IIA method for the source term. As described in Chapter II, the update formulas for the cell average are the following:

$$\bar{\mathbf{u}}_j^{n+1/3} = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{3}\frac{1}{\Delta x}\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/6} - \hat{\mathbf{f}}_{j-1/2}^{n+1/6}\right] + \frac{\Delta t}{3}\frac{1}{\epsilon}\left[\frac{5}{4}\mathbf{Q}\bar{\mathbf{u}}_j^{n+1/3} - \frac{1}{4}\mathbf{Q}\bar{\mathbf{u}}_j^{n+1}\right], \tag{4.88a}$$

$$\bar{\mathbf{u}}_j^{n+1} = \bar{\mathbf{u}}_j^n - \frac{\Delta t}{\Delta x}\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/2} - \hat{\mathbf{f}}_{j-1/2}^{n+1/2}\right] + \frac{\Delta t}{\epsilon}\left[\frac{3}{4}\mathbf{Q}\bar{\mathbf{u}}_j^{n+1/3} + \frac{1}{4}\mathbf{Q}\bar{\mathbf{u}}_j^{n+1}\right]. \tag{4.88b}$$

where $\mathbf{u} = [u, v]^T$ and the inputs for a Riemann solver are computed by (2.18) on page 39.

In view of the lengthy expressions, intermediate formulas are omitted, and the final result for the local truncation error is presented here:

$$\begin{aligned}\mathrm{LTE}_{\mathrm{HR2Ha}} = &-\left[\underbrace{\frac{ir\left(1-3\nu+2(r\nu)^2\right)}{12}\boxed{\Delta x^2}}_{\text{dominant dispersion error}} + \frac{i\epsilon\, r(1-r^2)\nu}{2}\Delta x\right]k^3 \\ &-\left[\underbrace{\frac{\left(1-r(r\nu)^3+2r^2\nu(\nu-1)\right)}{8}\boxed{\Delta x^3}}_{\text{dominant dissipation error}} - \frac{\epsilon(1-r^2)(4+3\nu)}{12}\Delta x^2\right]k^4.\end{aligned} \tag{4.89}$$

When the grid size is less than the threshold grid size:

$$\Delta h^*_{\mathrm{HR2Ha}} := 2\left[\frac{\epsilon(1-r^2)}{\left(1-r(r\nu)^3+2r^2\nu(\nu-1)\right)k^2}\right]^{1/3}, \tag{4.90}$$

the physical dissipation dominates over the numerical dissipation under the assumption of near-equilibrium ($r = O(1)\,, \epsilon \ll 1$), and the method becomes second-order in space and time.

### 4.3.5 DG–Hancock Method

The DG(1)–Hancock method for the 1-D GHHE has the form (4.88) together with the update equations for $\overline{\Delta\mathbf{u}}_j$:

$$\begin{aligned}\overline{\Delta\mathbf{u}}_j^{n+1/3} &= \overline{\Delta\mathbf{u}}_j^{n} - \frac{\Delta t}{3}\frac{6}{\Delta x}\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/6} + \hat{\mathbf{f}}_{j-1/2}^{n+1/6} - \frac{2}{\Delta x\Delta t}\overline{\mathbf{f}(\mathbf{u}_j^{n+1/6})}\right] \\ &\quad + \frac{\Delta t}{3}\frac{1}{\epsilon}\left[\frac{5}{4}\mathbf{Q}\overline{\Delta\mathbf{u}}_j^{n+1/3} - \frac{1}{4}\mathbf{Q}\overline{\Delta\mathbf{u}}_j^{n+1}\right], \end{aligned} \tag{4.91a}$$

$$\begin{aligned}\overline{\Delta\mathbf{u}}_j^{n+1} &= \overline{\Delta\mathbf{u}}_j^{n} - \frac{\Delta t}{\Delta x}6\left[\hat{\mathbf{f}}_{j+1/2}^{n+1/2} + \hat{\mathbf{f}}_{j-1/2}^{n+1/2} - \frac{2}{\Delta x\Delta t}\overline{\mathbf{f}(\mathbf{u}_j^{n+1/2})}\right] \\ &\quad + \frac{\Delta t}{\epsilon}\left[\frac{3}{4}\mathbf{Q}\overline{\Delta\mathbf{u}}_j^{n+1/3} + \frac{1}{4}\mathbf{Q}\overline{\Delta\mathbf{u}}_j^{n+1}\right]. \end{aligned} \tag{4.91b}$$

After some algebra, the local truncation error of the dominant eigenvalue is found to be given by

$$\begin{aligned}\mathrm{LTE}_{\mathrm{DG(1)Ha}}^{(1)} = &-\Bigg[\underbrace{\frac{r}{72}\left(\frac{r\big(1-(r\nu)^2\big)^2}{1-r^2\nu} - 3r\nu(1-\nu)\right)\boxed{\Delta x^3}}_{\text{dominant dissipation error}} + \frac{\epsilon\,(1-r^2)}{12(1-r^2\nu)^2} \\ &\times\left(1+\frac{1}{3}r^2+\frac{\nu^2}{6}\big(2r^2(r^2-9)+3r^4\nu+r^4(9-7r^2)\nu^2+3r^6\nu^3\big)\right)\Delta x^2\Bigg]\,k^4. \end{aligned} \tag{4.92}$$

Note that the leading error is a $k^4$-term, hence a dissipation, whereas in other methods possess a leading dispersion error. The threshold grid size to guarantee the method be third-order accurate is

$$\Delta h_{\mathrm{DG(1)Ha}}^{*} := 2\left[\frac{9\epsilon\,(1-r^2)(1-r^2\nu)}{r^2\big(1-3\nu+(3+r^2)\nu^2-3r^2\nu^3+(r\nu)^4\big)k^2}\right]^{1/3}. \tag{4.93}$$

### 4.3.6 Limiting Flux Function

Here, we show by a Fourier analysis of the semi-discrete HR2–MOL and DG(1)–MOL methods with the upwind flux based on the frozen wave speeds ±1, that this flux reduces to the Rusanov flux ($q = 1$) when $\epsilon \to 0$ [HSvL05]. Hence, at the discretization level, solving

$$\partial_t u + \partial_x v = 0, \tag{4.94a}$$

$$\partial_t v + \partial_x u = -\frac{1}{\epsilon}(v - ru); \quad \mathrm{Eig}(\mathbf{A}) = \pm 1, \tag{4.94b}$$

by using the upwind flux (also with $q = 1$ for this system) is identical to directly solving

$$\partial_t u + r\partial_x u = 0; \quad \mathrm{Eig}(\mathbf{A}) = r, \tag{4.95}$$

with the Rusanov flux ($q = 1$). Note that the genuine upwind flux for the above advection equation is obtained by $q = r$. This means that the semi-discrete HR2–MOL and DG(1)–MOL methods for (4.94) are not strictly upwind in the equilibrium limit ($\epsilon \to 0$).

The dominant dispersion/dissipation errors of semi-discrete methods for advection-diffusion in the low-frequency limit are given by (3.93) on page 116 for HR2–MOL and (3.106) on page 121 for DG(1)–MOL with $q = 1$, while the exact solution for the GHHE in this case is (4.30).

Consider the DG(1) discretization of the equilibrium advection equation,

$$\partial_t u + r\partial_x u = 0, \tag{4.96}$$

obtained by taking $\epsilon = 0$ in (4.27):

$$\frac{\partial u_{1,j}}{\partial t} = -\frac{1}{\Delta x}\left[-4f^q_{j-1/2} - 2f^q_{j+1/2} + 3r(u_{1,j} + u_{2,j})\right], \tag{4.97a}$$

$$\frac{\partial u_{2,j}}{\partial t} = -\frac{1}{\Delta x}\left[4f^q_{j+1/2} + 2f^q_{j-1/2} - 3r(u_{1,j} + u_{2,j})\right]. \tag{4.97b}$$

We substitute for $f^q$ a $q$-flux [vL69], that is,

$$\begin{aligned} f^q_{j+1/2}(u_{2,j}, u_{1,j+1}) &= \frac{r}{2}(u_{2,j} + u_{1,j+1}) - \frac{q}{2}(u_{1,j+1} - u_{2,j}) \\ &= \frac{1}{2}\left[2r + (r-q)\delta^+\right]\bar{u}_j + \frac{1}{4}\left[2q + (q-r)\delta^+\right]\overline{\Delta u}_j. \end{aligned} \tag{4.98}$$

In the manner of (4.68), we solve for the characteristic equation of the discrete system and insert a harmonic modes to find

$$(\Delta x \lambda)^2 - 2\left[q(2+\cos\beta) - ir\sin\beta\right]\Delta x\lambda + 6r\left[r(1-\cos\beta) + iq\sin\beta\right] = 0, \quad (4.99)$$

which is precisely the form of (4.76) on page 228 with

$$q = \frac{2c + r(1-c)}{1+c}. \quad (4.100)$$

This $q$ corresponds to the first Harten–Lax–Van Leer (HLL1) flux function [HLvL83] based on the frozen wave speeds $a_L = |-c|$ and $a_R = 1$, where

$$\begin{aligned} \hat{f}_{\mathrm{HLL1}}(u_L, u_R) &= \frac{a_L}{a_R + a_L} f(u_R) + \frac{a_R}{a_R + a_L} f(u_L) - \frac{a_R a_L}{a_R + a_L}(u_R - u_L) \\ &= \frac{c}{1+c} r u_R + \frac{1}{1+c} r u_L - \frac{c}{1+c}(u_R - u_L) \\ &= \frac{r}{2}(u_R + u_L) - \frac{1}{2}\frac{2c + r(1-c)}{1+c}(u_R - u_L), \end{aligned} \quad (4.101)$$

with $f_{L|R} = r u_{L|R}$. Thus the equilibrium (and near-equilibrium) asymptotic flux function for the DG(1) scheme applied to the simple one-dimensional problem is just the HLL1 flux function based on the frozen wave speeds 1 and $-c$. This is not completely surprising, as a stable relaxation system has a subcharacteristic condition, that is, each equilibrium wave speed must be bounded by two frozen wave speeds [Liu87, CLL94]. Since the upwind flux function in the DG discretization only knows about the two frozen wave speeds, it continues to use them to approximate the included equilibrium wave speed. Of course, this means that the equilibrium method is not strictly upwind.

Note that when the symmetric system (4.25) is considered, the above flux function reduces to the Rusanov flux:

$$\hat{f}_{\mathrm{Rusanov}}(u_L, u_R) = \frac{r}{2}(u_R + u_L) - \frac{1}{2}(u_R - u_L). \quad (4.102)$$

Thus, the result of the analysis for the linear advection equation obtained in Chapter III can be directly related here by letting $q = 1$. This is the reason why we defined in chapter III the Courant number (3.2) on page 88 based on the frozen wave speed 1, instead of the equilibrium wave speed $r$.

### 4.3.7 Dominant Dispersion/Dissipation Error in 1-D

In summary, the local truncation error of each method is listed for comparison.

**semi-discrete methods:**

$$\mathrm{LTE}_{\mathrm{HR2MOL}} = \left[c_3\left(1+(r\nu)^2\right)+\frac{1}{6}\tilde{c}_3\nu\right]k^3+\left[\ c_4\left(\frac{1}{r}-\frac{1}{3}(r\nu)^3\right)+\frac{1}{6}\tilde{c}_4\right]k^4, \tag{4.103a}$$

$$\mathrm{LTE}^{(1)}_{\mathrm{DG(1)MOL}} = \left[c_3(r\nu)^2 \qquad\qquad +\frac{1}{6}\tilde{c}_3\nu\right]k^3+\left[\frac{1}{9}c_4\left(r-3(r\nu)^3\right)\quad+\frac{1}{12}\tilde{c}_4\right]k^4, \tag{4.103b}$$

**fully discrete methods:**

$$\begin{aligned}\mathrm{LTE}_{\mathrm{HR2Ha}} &= \left[c_3\left(1-3\nu+2(r\nu)^2\right)+\frac{1}{2}\tilde{c}_3\nu\right]k^3\\ &\quad+\left[c_4\left(\frac{1}{r}-(r\nu)^3+2r\nu(\nu-1)\right)-\frac{1}{12}\tilde{c}_4\left(4+3\nu\right)\right]k^4,\end{aligned} \tag{4.103c}$$

$$\begin{aligned}\mathrm{LTE}^{(1)}_{\mathrm{DG(1)Ha}} &= \left[\frac{1}{9}c_4\left(\frac{r\left(1-(r\nu)^2\right)^2}{1-r^2\nu}-3r\nu(1-\nu)\right)+\frac{\tilde{c}_4}{12(1-r^2\nu)^2}\right.\\ &\quad\left.\times\left(1+\frac{1}{3}r^2+\frac{\nu^2}{6}\left(2r^2(r^2-9)+3r^4\nu+r^4(9-7r^2)\nu^2+3r^6\nu^3\right)\right)\right]k^4,\end{aligned} \tag{4.103d}$$

where the coefficients of errors attributed to the explicit flux discretization are

$$c_3 = -\frac{ir}{12}\,\boxed{\Delta x^2}, \quad c_4 = -\frac{r}{8}\,\boxed{\Delta x^3}, \tag{4.104a}$$

and the implicit source-term errors have coefficients

$$\tilde{c}_3 = -i\epsilon r(1-r^2)\,\boxed{\Delta x}, \quad \tilde{c}_4 = -\epsilon(1-r^2)\,\boxed{\Delta x^2}. \tag{4.104b}$$

The above equations show that HR2–MOL, DG(1)–MOL, and HR2–Hancock have a *first-order* error $\sim k^3$, as $\tilde{c}_3 = O(\epsilon\Delta x)$. However, numerically, this error term is not pronounced in the near-equilibrium limit since $\epsilon \ll 1$, thus the dominant error of the methods stem from $c_3 = O(\Delta x^2)$. Similarly, the dominant error of DG(1)–Hancock is $c_4 = O(\Delta x^3)$.

As it was described previously, when $r = O(\epsilon)$ with $\epsilon \ll 1$, the DG(1) method reveals uniform spatial convergence. To demonstrate this, we simply set $r = 0$, then the local truncation error of each method becomes

$$\mathrm{LTE}_{\mathrm{HR2MOL}} = \left[-\frac{1}{8}\boxed{\Delta x^3} - \frac{1}{6}\epsilon\Delta x^2\right] k^4, \tag{4.105a}$$

$$\mathrm{LTE}_{\mathrm{HR2Ha}} = \left[-\frac{1}{8}\boxed{\Delta x^3} + \frac{1}{12}\epsilon\left(4+3\nu\right)\Delta x^2\right] k^4, \tag{4.105b}$$

$$\mathrm{LTE}^{(1)}_{\mathrm{DG(1)MOL}} = -\frac{1}{12}\epsilon\Delta x^2\, k^4, \tag{4.105c}$$

$$\mathrm{LTE}^{(1)}_{\mathrm{DG(1)Ha}} = -\frac{1}{12}\epsilon\Delta x^2\, k^4. \tag{4.105d}$$

All dispersions, $O(k^3)$-terms, have disappeared, and the dominate error is the dissipation. The dominant dissipation errors of both DG(1) methods are proportional to $\epsilon$, while HR2 methods have the term, $\frac{1}{8}\Delta x^3$, which is independent of $\epsilon$. Hence, DG(1) methods lose their grid size restrictions, but HR2 methods still need to satisfy the following inequality:

$$\frac{1}{8}\Delta x^3 k^4 \ll \epsilon k^2 \quad \longrightarrow \quad \Delta x \ll 2\left(\frac{\epsilon}{k^2}\right)^{1/3}, \tag{4.106}$$

to guarantee physical dissipation is dominant. Since the leading errors of HR2 methods are proportional to $\Delta x^3$, we expect third-order convergence on coarse grids.

### 4.3.8 Stability of Methods

All four methods described previously include implicit treatment of the source term. Particularly, an $L$-stable time integrator is employed. Hence, the stability condition is restricted solely by an explicit discretization of the flux even in the stiff regime. Recall that, in the near-equilibrium (stiff) regime, discretizing hyperbolic-relaxation equations with the upwind flux is equivalent to discretizing the advection equation directly with the Rusanov flux. Thus, it is beneficial to look into the stability of each method applied to the advection equation. In Chapter III on page 154, stability analyses of methods with the Rusanov flux was conducted. A rather surprising property was found for the DG(1)–Hancock method; the stability domain reduces as the equilibrium wave speed gets smaller, while other methods possess constant stability domains. This property suggests that, in the frozen limit, the maximum Courant number of the DG(1)–Hancock method is still unity, yet in the near-equilibrium limit, it will depend on the magnitude of the dimensionless equilibrium wave speed $|r|$; the maximum Courant number can be found from (3.162) on page 155, or Figure 3.15 on page 157.

## 4.4 Model Equations for Two-Dimensional Problem

In two dimensions we consider the simple system

$$\partial_t u + \partial_x v + \partial_y w = 0, \tag{4.107a}$$

$$\partial_t v + \partial_x u + r\partial_y w = -\frac{1}{\epsilon}(v - ru), \tag{4.107b}$$

$$\partial_t w + s\partial_x v + \partial_y u = -\frac{1}{\epsilon}(w - su), \tag{4.107c}$$

where $v$ and $w$ are the fluxes in the $x$- and $y$-directions, respectively. The above equations can be written in vector form:

$$\partial_t \mathbf{u}(\boldsymbol{x}, t) + \partial_x \mathbf{f}(\mathbf{u}) + \partial_y \mathbf{g}(\mathbf{u}) = \frac{1}{\epsilon}\mathbf{s}(\mathbf{u}); \quad \boldsymbol{x} \in \mathbb{R}^2,\ t > 0, \tag{4.108}$$

with linear fluxes and source, $\mathbf{f}(\mathbf{u}) = \mathbf{A}\mathbf{u}$, $\mathbf{g}(\mathbf{u}) = \mathbf{B}\mathbf{u}$, and $\mathbf{s}(\mathbf{u}) = \mathbf{Q}\mathbf{u}$, where

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & s & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & r \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{Q} = \frac{1}{\epsilon} \begin{pmatrix} 0 & 0 & 0 \\ r & -1 & 0 \\ s & 0 & -1 \end{pmatrix}. \tag{4.109}$$

The near-equilibrium limit is formally

$$\partial_t u + r\partial_x u + s\partial_y u = \epsilon \left[(1-r^2)\partial_{xx}u + (1-s^2)\partial_{yy}u\right] + O\left(\epsilon^2\right), \tag{4.110}$$

with the equilibrium wave speeds $r$ and $s$ in the $x$- and $y$-directions, respectively. The derivation is shown in Appendix C on page 354. For a harmonic mode with wave vector $\mathbf{k} = (k_x, k_y)$, a stability criterion in the near-equilibrium limit is found by insisting that the second-order derivative terms are dissipative; mathematically, this is $|rk_x + sk_y| \leq |\mathbf{k}|$. Due to the complexity of the analysis, we restrict the discussion to a uniform grid with unit aspect ratio ($\Delta h := \Delta x = \Delta y$), and the wave frequencies in the $x$- and $y$-directions are the same, thus $\alpha = \beta$ and $k_x = k_y = k$. Based on these assumptions, the exact solution of the reduced equation (4.110) in the near-equilibrium limit is

$$\lambda_{\text{exact}}^{\text{GHHE}} = -i(r+s)k - \epsilon(2-r^2-s^2)k^2 - i\epsilon^2(r+s)(3-2r^2+rs-2s^2)k^3 + O\left(\epsilon^3\right). \tag{4.111}$$

It has been shown that the near-equilibrium limit is formed by coupling between the flux and relaxation operators [Hit00, HR04], and this is clear by inspection of the wave speeds and diffusion coefficients in (4.27), (4.29) on page 216, and (4.110).

## 4.5 Difference Operators and Their Properties in 2-D

In this section, due to the complexity of multidimensional Fourier analysis, we restrict ourselves to semi-discrete methods, namely, HR2–MOL and DG(1)–MOL methods.

### 4.5.1 HR–MOL Method

The HR2–MOL method applied to the two-dimensional GHHE equation (4.107) has the form:

$$\frac{\partial \bar{\mathbf{u}}_{jk}(t)}{\partial t} = -\frac{1}{\Delta x}\left(\hat{\mathbf{f}}_{j+1/2,k} - \hat{\mathbf{f}}_{j-1/2,k}\right) - \frac{1}{\Delta y}\left(\hat{\mathbf{g}}_{j,k+1/2} - \hat{\mathbf{g}}_{j,k-1/2}\right) + \frac{1}{\epsilon}\mathbf{Q}\,\bar{\mathbf{u}}_{jk}, \quad (4.112)$$

and its compact form is

$$\frac{\partial \bar{\mathbf{u}}_{j}(t)}{\partial t} = \left(\mathbf{N}_{\mathrm{HR2}} + \frac{1}{\epsilon}\mathbf{Q}\right)\bar{\mathbf{u}}_{j}(t), \quad (4.113)$$

where $\bar{\mathbf{u}}_j = [\bar{u}_j, \bar{v}_j, \bar{w}_j]^T$ The flux-difference operator $\mathbf{N}_{\mathrm{HR2}}$ is given by

$$\mathbf{N}_{\mathrm{HR2}} = \boldsymbol{\mathcal{A}}^{2+}\mathbf{D}_x^{2+} + \boldsymbol{\mathcal{A}}^{+}\mathbf{D}_x^{+} + \boldsymbol{\mathcal{A}}^{-}\mathbf{D}_x^{-} + \boldsymbol{\mathcal{A}}^{2-}\mathbf{D}_x^{2-}$$
$$+ \boldsymbol{\mathcal{B}}^{2+}\mathbf{D}_y^{2+} + \boldsymbol{\mathcal{B}}^{+}\mathbf{D}_y^{+} + \boldsymbol{\mathcal{B}}^{-}\mathbf{D}_y^{-} + \boldsymbol{\mathcal{B}}^{2-}\mathbf{D}_y^{2-}. \quad (4.114)$$

Here, the coefficients matrices are

$$\boldsymbol{\mathcal{A}}^{+} = \frac{1}{4\Delta x}\begin{pmatrix} 1 & -2 & 0 \\ -2 & 1 & 0 \\ s & -2s & 0 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}^{-} = -\frac{1}{4\Delta x}\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ s & 2s & 0 \end{pmatrix} \quad (4.115a)$$

$$\boldsymbol{\mathcal{A}}^{2+} = \frac{1}{8\Delta x}\begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ -s & s & 0 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}^{2-} = -\frac{1}{8\Delta x}\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ s & s & 0 \end{pmatrix}, \quad (4.115b)$$

$$\boldsymbol{\mathcal{B}}^{+} = \frac{1}{4\Delta y}\begin{pmatrix} 1 & 0 & -2 \\ r & 0 & -2r \\ -2 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\mathcal{B}}^{-} = -\frac{1}{4\Delta y}\begin{pmatrix} 1 & 0 & 2 \\ r & 0 & 2r \\ 2 & 0 & 1 \end{pmatrix}, \quad (4.115c)$$

$$\boldsymbol{\mathcal{B}}^{2+} = \frac{1}{8\Delta y}\begin{pmatrix} -1 & 0 & 1 \\ -r & 0 & r \\ 1 & 0 & -1 \end{pmatrix}, \quad \boldsymbol{\mathcal{B}}^{2-} = -\frac{1}{8\Delta y}\begin{pmatrix} 1 & 0 & 1 \\ r & 0 & r \\ 1 & 0 & 1 \end{pmatrix}, \quad (4.115d)$$

$$\mathbf{D}_x^{\pm} = \delta_x^{\pm}\mathbf{I}, \quad \mathbf{D}_x^{2\pm} = (\delta_x^{\pm})^2\mathbf{I}, \quad \mathbf{D}_y^{\pm} = \delta_y^{\pm}\mathbf{I}, \quad \mathbf{D}_y^{2\pm} = (\delta_y^{\pm})^2\mathbf{I}. \tag{4.115e}$$

**Accuracy** Following the previous analysis, the eigenvalues of the spatial discretization are obtained by solving the characteristic equations:

$$\det\left(\mathbf{N}_{\mathrm{HR2}} + \frac{1}{\epsilon}\mathbf{Q} - \lambda\mathbf{I}\right) = 0. \tag{4.116}$$

Taking the low-frequency limit, under the assumptions of $\Delta x = \Delta y = \Delta h$ and $\alpha = \beta$, leads to asymptotic eigenvalues,

$$\begin{aligned}\lambda_{\mathrm{HR2}}^{(1)} &= -\frac{i(r+s)}{\Delta h}\beta - \frac{\epsilon(2-r^2-s^2)}{\Delta h^2}\beta^2 \\ &\quad - \left[\frac{i(r+s)}{12\Delta h} + \frac{i\epsilon^2(r+s)(3+2r^2-rs+2s^2)}{\Delta h^3}\right]\beta^3 + O\left(\beta^4\right)\end{aligned} \tag{4.117a}$$

$$\lambda_{\mathrm{HR2}}^{(2)} = -\frac{1}{\epsilon} + \frac{ir}{\Delta h}\beta + O\left(\beta^2\right), \tag{4.117b}$$

$$\lambda_{\mathrm{HR2}}^{(3)} = -\frac{1}{\epsilon} + \frac{is}{\Delta h}\beta + O\left(\beta^2\right), \tag{4.117c}$$

where the first eigenvalue represents the dominant wave in the asymptotic limit, and the other waves damp quickly since the leading errors are large negative real. The spatial discretization error corresponding to the dominant wave is derived by replacing the wave frequency by the wave number, then

$$\lambda_{\mathrm{HR2}}^{(1)} - \lambda_{\mathrm{exact}}^{\mathrm{2D\text{-}GHHE}} = -\frac{i(r+s)}{12}\boxed{\Delta h^2}k^3 - \left[\frac{1}{4}\boxed{\Delta h^3} + \frac{\epsilon(2-r^2-s^2)}{6}\Delta h^2\right]k^4 + O\left(k^5\right). \tag{4.118}$$

Thus, the spatial discretization error of the dominant wave is second-order in space. In order to ensure the physical dissipation is dominant in the near-equilibrium limit $\epsilon \ll 1$, the following relation has to be satisfied:

$$\frac{1}{4}\Delta h^3 k^4 \ll \epsilon(2-r^2-s^2)k^2. \tag{4.119}$$

Solving for $\Delta h$ results in the threshold grid size:

$$\Delta h_{\mathrm{HR2}} \ll \left[\frac{4\epsilon\left(2-r^2-s^2\right)}{k^2}\right]^{1/3}. \tag{4.120}$$

### 4.5.2 DG–MOL Method

The two-dimensional DG(1)–MOL method utilizes a linear solution representation:

$$\mathbf{u}(x,y,t) = \bar{\mathbf{u}}(t) + \left(\xi - \frac{1}{2}\right)\overline{\Delta_x \mathbf{u}}(t) + \left(\eta - \frac{1}{2}\right)\overline{\Delta_y \mathbf{u}}(t), \tag{4.121}$$

where $\xi \in [0,1]$ is as before and $\eta = \dfrac{y - y_{k-1/2}}{\Delta y} \in [0,1]$. The semi-discrete DG(1)–MOL method is

$$\frac{\partial \bar{\mathbf{u}}_{jk}(t)}{\partial t} = -\frac{1}{\Delta x}\left(\hat{\mathbf{f}}_{j+1/2,k} - \hat{\mathbf{f}}_{j-1/2,k}\right) - \frac{1}{\Delta y}\left(\hat{\mathbf{g}}_{j,k+1/2} - \hat{\mathbf{g}}_{j,k-1/2}\right) + \frac{1}{\epsilon}\mathbf{Q}\,\bar{\mathbf{u}}_{jk}, \tag{4.122a}$$

$$\begin{aligned}\frac{\partial \overline{\Delta_x \mathbf{u}}_{jk}(t)}{\partial t} &= -\frac{6}{\Delta x}\left[\hat{\mathbf{f}}_{j+1/2,k} + \hat{\mathbf{f}}_{j-1/2,k} - 2\mathbf{f}(\bar{\mathbf{u}}_{jk})\right] \\ &- \frac{12}{\Delta y}\left[\int_0^1 \left(\xi - \frac{1}{2}\right)\hat{\mathbf{g}}_{\xi,k+1/2}(\xi)\,d\xi - \int_0^1 \left(\xi - \frac{1}{2}\right)\hat{\mathbf{g}}_{\xi,k-1/2}(\xi)\,d\xi\right] + \frac{1}{\epsilon}\mathbf{Q}\,\overline{\Delta_x \mathbf{u}}_{jk},\end{aligned} \tag{4.122b}$$

$$\begin{aligned}\frac{\partial \overline{\Delta_y \mathbf{u}_{jk}}(t)}{\partial t} &= -\frac{6}{\Delta y}\left[\hat{\mathbf{g}}_{j,k+1/2} + \hat{\mathbf{g}}_{j,k-1/2} - 2\mathbf{g}(\bar{\mathbf{u}}_{jk})\right] \\ &- \frac{12}{\Delta x}\left[\int_0^1 \left(\eta - \frac{1}{2}\right)\hat{\mathbf{f}}_{j+1/2,\eta}(\eta)\,d\eta - \int_0^1 \left(\eta - \frac{1}{2}\right)\hat{\mathbf{f}}_{j+1/2,\eta}(\eta)\,d\eta\right] + \frac{1}{\epsilon}\mathbf{Q}\,\overline{\Delta_y \mathbf{u}}_{jk},\end{aligned} \tag{4.122c}$$

where

$$\hat{\mathbf{f}}_{j+1/2,k} = \hat{\mathbf{f}}\left(\bar{\mathbf{u}}_{j,k} + \frac{1}{2}\Delta_x \mathbf{u}_{j,k}, \bar{\mathbf{u}}_{j+1,k} - \frac{1}{2}\overline{\Delta_x \mathbf{u}}_{j+1,k}\right), \tag{4.123a}$$

$$\hat{\mathbf{g}}_{j,k+1/2} = \hat{\mathbf{g}}\left(\bar{\mathbf{u}}_{j,k} + \frac{1}{2}\Delta_y \mathbf{u}_{j,k}, \bar{\mathbf{u}}_{j,k+1} - \frac{1}{2}\overline{\Delta_y \mathbf{u}}_{j,k+1}\right), \tag{4.123b}$$

are one-dimensional upwind flux functions (4.39). The two-dimensional DG(1)–MOL discretization can be written in the compact form

$$\frac{\partial \mathbf{u}_{jk}(t)}{\partial t} = \left(\mathbf{N}_{\mathrm{DG}(1)} + \mathbf{Q}_{\mathrm{DG}(1)}\right) \mathbf{u}_{jk}(t), \tag{4.124}$$

where

$$\begin{aligned}\mathbf{u}_{jk} &= [\overline{\mathbf{u}}_{jk}, \overline{\Delta_x \mathbf{u}}_{jk}, \overline{\Delta_y \mathbf{u}}_{jk}]^T \\ &= [\bar{u}_{jk}, \bar{v}_{jk}, \bar{w}_{jk}, \overline{\Delta_x u}_{jk}, \overline{\Delta_x v}_{jk}, \overline{\Delta_x w}_{jk}, \overline{\Delta_y u}_{jk}, \overline{\Delta_y v}_{jk}, \overline{\Delta_y w}_{jk}]^T,\end{aligned} \tag{4.125}$$

and the matrix operator of spatial differencing is given by

$$\mathbf{N}_{\mathrm{DG}(1)} = \boldsymbol{\mathcal{A}}^+ \mathbf{D}_x^+ + \boldsymbol{\mathcal{A}}^- \mathbf{D}_x^- + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{B}}^+ \mathbf{D}_y^+ + \boldsymbol{\mathcal{B}}^- \mathbf{D}_y^-, \tag{4.126}$$

where

$$\boldsymbol{\mathcal{A}}^+ = \frac{1}{4\Delta x}\begin{pmatrix} 2\mathbf{A}_1 & -\mathbf{A}_1 & \mathbf{0} \\ 12\mathbf{A}_1 & -6\mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 2\mathbf{A}_1 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}^- = \frac{1}{4\Delta x}\begin{pmatrix} -2\mathbf{A}_2 & -\mathbf{A}_2 & \mathbf{0} \\ 12\mathbf{A}_2 & 6\mathbf{A}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -2\mathbf{A}_2 \end{pmatrix}, \tag{4.127a}$$

$$\boldsymbol{\mathcal{B}}^+ = \frac{1}{4\Delta y}\begin{pmatrix} 2\mathbf{B}_1 & \mathbf{0} & -\mathbf{B}_1 \\ \mathbf{0} & 2\mathbf{B}_1 & \mathbf{0} \\ 12\mathbf{B}_1 & \mathbf{0} & -6\mathbf{B}_1 \end{pmatrix}, \quad \boldsymbol{\mathcal{B}}^- = \frac{1}{4\Delta y}\begin{pmatrix} -2\mathbf{B}_2 & \mathbf{0} & -\mathbf{B}_2 \\ \mathbf{0} & -2\mathbf{B}_2 & \mathbf{0} \\ 12\mathbf{B}_2 & \mathbf{0} & 6\mathbf{B}_2 \end{pmatrix}, \tag{4.127b}$$

$$\boldsymbol{\mathcal{C}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_2 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ s & -s & 0 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ s & s & 0 \end{pmatrix}, \tag{4.127c}$$

$$\mathbf{B}_1 = \begin{pmatrix} 1 & 0 & -1 \\ r & 0 & -r \\ -1 & 0 & 1 \end{pmatrix}, \quad \mathbf{B}_2 = \begin{pmatrix} 1 & 0 & 1 \\ r & 0 & r \\ 1 & 0 & 1 \end{pmatrix}, \tag{4.127d}$$

$$\mathbf{C}_1 = -\frac{6}{\Delta x}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ s & 0 & 0 \end{pmatrix}, \quad \mathbf{C}_2 = -\frac{6}{\Delta y}\begin{pmatrix} 1 & 0 & 0 \\ r & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.127e}$$

The source-term matrix is defined to be

$$\mathbf{Q}_{\mathrm{DG}(1)} = \begin{pmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q} \end{pmatrix}. \tag{4.128}$$

We look for roots of the characteristic polynomial

$$\det\left(\mathbf{N}_{\mathrm{DG}(1)} + \frac{1}{\epsilon}\mathbf{Q}_{\mathrm{DG}(1)} - \lambda\mathbf{I}\right) = 0, \tag{4.129}$$

under the assumption $\alpha = \beta$. As in the 1-D case, we assume a power-series form for the eigenvalue, and solve for the eigenvalue order-by-order in $\beta$. We find the following eigenvalues:

$$\lambda^{(1)}_{\mathrm{DG}(1)} = -i\left(\frac{r}{\Delta x} + \frac{s}{\Delta y}\right)\beta - \epsilon\left(\frac{1-r^2}{\Delta x^2} + \frac{1-s^2}{\Delta y^2}\right)\beta^2 + O\big(\beta^3\big), \tag{4.130a}$$

$$\lambda^{(2)}_{\mathrm{DG}(1)} = -\frac{6}{\Delta x} + O(\beta), \qquad \lambda^{(3)}_{\mathrm{DG}(1)} = -\frac{6}{\Delta y} + O(\beta), \tag{4.130b}$$

$$\lambda^{(4)}_{\mathrm{DG}(1)} = -\frac{6}{\Delta x} - \frac{1}{\epsilon} + O(\beta), \quad \lambda^{(5)}_{\mathrm{DG}(1)} = -\frac{6}{\Delta y} - \frac{1}{\epsilon} + O(\beta), \tag{4.130c}$$

$$\lambda^{(6,7,8,9)}_{\mathrm{DG}(1)} = -\frac{1}{\epsilon} + O(\beta). \tag{4.130d}$$

To simplify the analysis, we further assume a uniform grid, $\Delta x = \Delta y = \Delta h$, then the spatial-discretization error is obtained by comparing the dominant eigenvalue to the exact solution:

$$\begin{aligned}\lambda^{(1)}_{\mathrm{DG}(1)} - \lambda^{\text{2D-GHHE}}_{\text{exact}} &= -\left[\frac{1}{9}\Delta h^3 - \frac{2-r^2-s^2}{72(\Delta h+6\epsilon)}\Delta h^4\right]k^4 + O\big(k^5\big)\\ &= -\frac{1}{72}\left[(\underbrace{6}_{\text{multi-D error}} + r^2+s^2)\boxed{\Delta h^3} + 6\epsilon(2-r^2-s^2)\Delta h^2\right]k^4\\ &\qquad + O\big(\epsilon^2 k^4, k^5\big)\,.\end{aligned} \tag{4.131}$$

The above equation shows that in the near-equilibrium limit $\epsilon \ll 1$, the dominant error is $O(\Delta h^3)$, thus the DG(1) spatial discretization is third-order in space. To ensure the physical dissipation is dominant, the mesh size has the following constrain:

$$\frac{6+r^2+s^2}{72}\Delta h^3 k^4 \ll \epsilon(2-r^2-s^2)k^2. \tag{4.132}$$

Solving for $\Delta h$ leads to

$$\Delta h_{\mathrm{DG}(1)} \ll 2\left[\frac{9\epsilon\,(2-r^2-s^2)}{(6+r^2+s^2)k^2}\right]^{1/3}. \tag{4.133}$$

As pointed out in a Fourier analysis of the DG(1)–MOL method for the 2-D advection equation on page 169, the two-dimensional DG(1) discretization contains a multidimensional error, $-\frac{1}{12}\Delta h^3$, in the $O(k^4)$-term. Since the upwind flux for the GHHE in the near-equilibrium limit is equivalent to the direct discretization of the advection equation with the Rusanov flux ($q_x = q_y = 1$), the above error really comes from the term $-\frac{1}{24}(q_x+q_y)\Delta h^3$. This extra multidimensional error eliminates the uniform-convergence property which the DG(1) method possess in one dimensional problem with a certain scaling.

In the 1-D case, for the specific scaling where the equilibrium speed is $r = O(\epsilon)$, the DG(1) method does not have any grid size restriction to achieve second-order accuracy. However, due to the multidimensional error independent of the equilibrium wave speeds $r, s$, there is always a grid size restriction even in the case where $r = O(\epsilon)$ with $\epsilon \ll 1$.

### 4.5.3 Dominant Dispersion/Dissipation Error

In summary, the local truncation errors of spatial discretization methods, HR2 and DG(1), are listed for comparison:

$$\mathrm{LTE}_{\mathrm{HR2}} = c_3\, k^3 + \left[ \quad c_4 \frac{2}{r+s} \qquad\qquad\qquad + \frac{1}{6}\tilde{c}_4 \right] k^4, \tag{4.134a}$$

$$\mathrm{LTE}_{\mathrm{DG(1)}} = \qquad \left[ \frac{1}{9} c_4 \left( \frac{6}{r+s} + \frac{r^2+s^2}{r+s} \right) + \frac{1}{12}\tilde{c}_4 \right] k^4, \tag{4.134b}$$

where the coefficients of errors attributed to the flux discretization are

$$c_3 = -\frac{i(r+s)}{12} \boxed{\Delta h^2}, \quad c_4 = -\frac{r+s}{8} \boxed{\Delta h^3}, \tag{4.135a}$$

and the source-error coefficients are

$$\tilde{c}_4 = -\epsilon(2 - r^2 - s^2) \boxed{\Delta h^2}. \tag{4.135b}$$

Note that the above errors are merely the spatial discretization errors; unlike in the 1-D analysis given by (4.103) on page 235, the temporal errors are not considered. Meanwhile, the above results can be related to Fourier analyses for the 2-D advection equation given by (3.223) on page 181 combined with the Rusanov flux ($q_x = q_y = 1$), and zero temporal error ($\nu = 0$). The HR2 method has a leading second-order dispersion error, whereas the DG(1) method is third-order accurate as long as grids are coarse, hence $\Delta h \gg O(\epsilon)$.

When we consider the case $r = s = 0$, the above truncation errors become

$$\mathrm{LTE}_{\mathrm{HR2}} = -\frac{1}{4} \boxed{\Delta h^3} - \frac{1}{3}\epsilon \Delta h^2, \tag{4.136a}$$

$$\mathrm{LTE}_{\mathrm{DG(1)}} = -\frac{1}{12} \boxed{\Delta h^3} - \frac{1}{6}\epsilon \Delta h^2. \tag{4.136b}$$

Unlike in one dimension, the DG(1) method possesses a dominant error independent to $\epsilon$. Hence, both spatial discretizations yield grid-size restrictions to ensure the physical dissipation is dominant.

### 4.5.4 Stability of Methods

Similar to the 1-D case, the stability of the 2-D DG(1)–Hancock method in the near-equilibrium limit varies as the dimensionless wave speed changes. The stability limit is given by (3.225) on page 183. The rest of methods remain same stability properties as the upwind flux; stability limits are listed in Table 3.7 on page 182.

## 4.6 Grid-Convergence Study in 1-D

### 4.6.1 Problem Definition

We consider the symmetric model problem (4.25) on page 215. To confirm the analysis, consider an initial-value problem on a periodic domain with a harmonic initial condition,

$$\mathbf{u}(x,0) = \Re\{\mathbf{u}_0 \exp(ikx)\} = \mathbf{u}_0 \cos(kx), \tag{4.137}$$

where $k = 2\pi$ and $\mathbf{u}_0 = (1,1)^T$. A dispersion analysis provides the exact solution for the computation of the $L_2$-norm [Hit00]. The DG(1)–Hancock method is compared with the HR2–MOL and DG(1)–MOL methods, both incorporating the IMEX–RK method [PR05]. For each method, we consider both a frozen limit ($\epsilon \gg 1$, non-stiff) and a near-equilibrium limit ($\epsilon \ll 1$, stiff) with two equilibrium wave speeds in the stiff regime:

$$\text{frozen (non-stiff) limit}: r = \frac{1}{2}, \quad \epsilon = 10^3, \quad t_{\text{end}} = 100, \tag{4.138a}$$

$$\text{near-equilibrium (stiff) limit}: r = 0, \frac{1}{2}, \quad \epsilon = 10^{-5}, \quad t_{\text{end}} = 300, \tag{4.138b}$$

and compute the error norms relative to an exact analytic solution. The non-stiff case represents *frozen* waves with speed ±1 propagate 100 times over the computation domain. Similarly, in the stiff case with $r = \dfrac{1}{2}$, the *equilibrium* wave travels 150 wave lengths until $t_{\text{end}} = 300$. Note that $r = 0$ corresponds to no advection, hence the system reduces to a pure diffusion equation:

$$\partial_t u = \epsilon\, \partial_{xx} u. \tag{4.139}$$

In both cases, the final analytical damping factor is obtained by

$$u(t_{\text{end}}) = e^{-\epsilon(1-r^2)k^2\, t_{\text{end}}} \approx 0.915, \tag{4.140}$$

thus, a sinusoidal wave is physically dissipated by 8.5%. The Courant numbers based on the unit wave speed:

$$\nu_{\text{method}} = 1\frac{\Delta t}{\Delta x}, \tag{4.141}$$

are set to be

$$\nu_{\text{HR2–MOL}} = \nu_{\text{HR2–Hancock}} = 0.9, \tag{4.142a}$$

$$\nu_{\text{DG(1)–MOL}} = 0.3, \tag{4.142b}$$

$$\nu_{\text{DG(1)–Hancock}} = \begin{cases} 0.9 & \text{frozen limit} \\ 0.3 & \text{near-equilibrium limit.} \end{cases} \tag{4.142c}$$

The DG(1) method requires cell averages $\bar{\mathbf{u}}_j$ and slopes $\overline{\Delta\mathbf{u}}_j$ in each cell for initial conditions. Given an initial function $\mathbf{u}(x, 0)$ in (4.137), the initial cell values are obtained from

$$\bar{\mathbf{u}}_j(t=0) := \frac{1}{\Delta x}\int_{I_j} \mathbf{u}(x,0)dx, \tag{4.143a}$$

$$\overline{\Delta_x \mathbf{u}}_j(t=0) := \frac{\displaystyle\int_{I_j} \mathbf{u}(x,0)\left(\frac{x-x_j}{\Delta x}\right)dx}{\displaystyle\int_{I_j}\left(\frac{x-x_j}{\Delta x}\right)^2 dx} = \frac{12}{\Delta x^2}\int_{I_j} \mathbf{u}(x,0)(x-x_j)dx, \tag{4.143b}$$

where $I_j \in [x_{j-1/2}, x_{j+1/2}]$.

### 4.6.2 Convergence in the Frozen Limit

When the relaxation time is large relative to the residence time, the effect of the source term is negligible, and the model equations behave like pure advection equations showed in (4.4) on page 209. The time step based on frozen wave speeds ±1 provides a stable method. Table 4.2 and Figure 4.2 demonstrate second-order convergence in this limit for HR2–IMEX, DG(1)–IMEX, and HR2–Hancock. As we expected for the DG(1)–Hancock method, third-order convergence is observed. Figure 4.3 shows the normalized CPU time needed to achieve the target error level. As in the case of advection, the high accuracy and efficiency of the DG(1)–Hancock method are evident.

(a) The HR2–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 6.78e−01 | — | 9.12e−01 | — | 6.78e−01 | — | 9.12e−01 | — | 0.00e+00 |
| 20 | 20 | 7.92e−01 | −0.22 | 1.11e+00 | −0.29 | 7.92e−01 | −0.22 | 1.11e+00 | −0.29 | 1.00e−02 |
| 40 | 40 | 1.14e+00 | −0.52 | 1.60e+00 | −0.53 | 1.14e+00 | −0.52 | 1.60e+00 | −0.53 | 4.00e−02 |
| 80 | 80 | 3.91e−01 | 1.54 | 5.54e−01 | 1.54 | 3.91e−01 | 1.54 | 5.54e−01 | 1.54 | 1.50e−01 |
| 160 | 160 | 1.00e−01 | 1.96 | 1.42e−01 | 1.96 | 1.01e−01 | 1.96 | 1.42e−01 | 1.96 | 6.00e−01 |
| 320 | 320 | 2.52e−02 | 2.00 | 3.56e−02 | 2.00 | 2.52e−02 | 2.00 | 3.56e−02 | 2.00 | 2.43e+00 |
| 640 | 640 | 6.30e−03 | 2.00 | 8.91e−03 | 2.00 | 6.30e−03 | 2.00 | 8.91e−03 | 2.00 | 1.22e+01 |
| 1280 | 1280 | 1.57e−03 | 2.00 | 2.23e−03 | 2.00 | 1.57e−03 | 2.00 | 2.23e−03 | 2.00 | 5.13e+01 |
| 2560 | 2560 | 3.94e−04 | 2.00 | 5.57e−04 | 2.00 | 3.94e−04 | 2.00 | 5.57e−04 | 2.00 | 2.05e+02 |
| 5120 | 5120 | 9.84e−05 | 2.00 | 1.39e−04 | 2.00 | 9.84e−05 | 2.00 | 1.39e−04 | 2.00 | 1.24e+03 |

(b) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 6.51e−01 | — | 9.14e−01 | — | 6.51e−01 | — | 9.14e−01 | — | 0.00e+00 |
| 20 | 20 | 2.73e−01 | 1.25 | 3.85e−01 | 1.25 | 2.73e−01 | 1.25 | 3.85e−01 | 1.25 | 1.00e−02 |
| 40 | 40 | 7.16e−02 | 1.93 | 1.01e−01 | 1.93 | 7.16e−02 | 1.93 | 1.01e−01 | 1.93 | 5.00e−02 |
| 80 | 80 | 1.79e−02 | 2.00 | 2.53e−02 | 2.00 | 1.79e−02 | 2.00 | 2.53e−02 | 2.00 | 2.10e−01 |
| 160 | 160 | 4.46e−03 | 2.00 | 6.31e−03 | 2.00 | 4.46e−03 | 2.00 | 6.31e−03 | 2.00 | 8.30e−01 |
| 320 | 320 | 1.11e−03 | 2.00 | 1.58e−03 | 2.00 | 1.11e−03 | 2.00 | 1.58e−03 | 2.00 | 3.34e+00 |
| 640 | 640 | 2.78e−04 | 2.00 | 3.94e−04 | 2.00 | 2.78e−04 | 2.00 | 3.94e−04 | 2.00 | 1.42e+01 |
| 1280 | 1280 | 6.96e−05 | 2.00 | 9.84e−05 | 2.00 | 6.96e−05 | 2.00 | 9.84e−05 | 2.00 | 5.81e+01 |
| 2560 | 2560 | 1.74e−05 | 2.00 | 2.46e−05 | 2.00 | 1.74e−05 | 2.00 | 2.46e−05 | 2.00 | 2.33e+02 |
| 5120 | 5120 | 4.34e−06 | 2.00 | 6.14e−06 | 2.00 | 4.34e−06 | 2.00 | 6.14e−06 | 2.00 | 9.32e+02 |

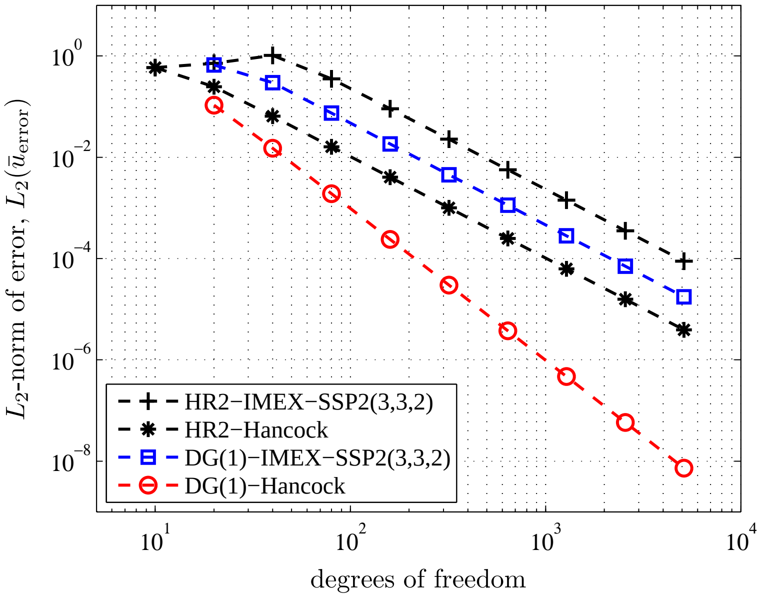
(c) The DG(1)–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.3$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---:|---:|---|---|---|---|---|---|---|---|---|
| 10 | 20 | 7.41e–01 | — | 1.03e+00 | — | 7.41e–01 | — | 1.03e+00 | — | 1.00e–02 |
| 20 | 40 | 3.29e–01 | 1.17 | 4.64e–01 | 1.14 | 3.29e–01 | 1.17 | 4.64e–01 | 1.14 | 3.00e–02 |
| 40 | 80 | 8.25e–02 | 2.00 | 1.16e–01 | 1.99 | 8.25e–02 | 2.00 | 1.16e–01 | 1.99 | 1.40e–01 |
| 80 | 160 | 2.02e–02 | 2.03 | 2.86e–02 | 2.02 | 2.02e–02 | 2.03 | 2.86e–02 | 2.02 | 5.80e–01 |
| 160 | 320 | 5.03e–03 | 2.01 | 7.11e–03 | 2.01 | 5.03e–03 | 2.01 | 7.11e–03 | 2.01 | 2.33e+00 |
| 320 | 640 | 1.25e–03 | 2.00 | 1.77e–03 | 2.00 | 1.25e–03 | 2.00 | 1.77e–03 | 2.00 | 1.15e+01 |
| 640 | 1280 | 3.13e–04 | 2.00 | 4.43e–04 | 2.00 | 3.13e–04 | 2.00 | 4.43e–04 | 2.00 | 4.97e+01 |
| 1280 | 2560 | 7.83e–05 | 2.00 | 1.11e–04 | 2.00 | 7.83e–05 | 2.00 | 1.11e–04 | 2.00 | 2.01e+02 |
| 2560 | 5120 | 1.96e–05 | 2.00 | 2.77e–05 | 2.00 | 1.96e–05 | 2.00 | 2.77e–05 | 2.00 | 8.63e+02 |

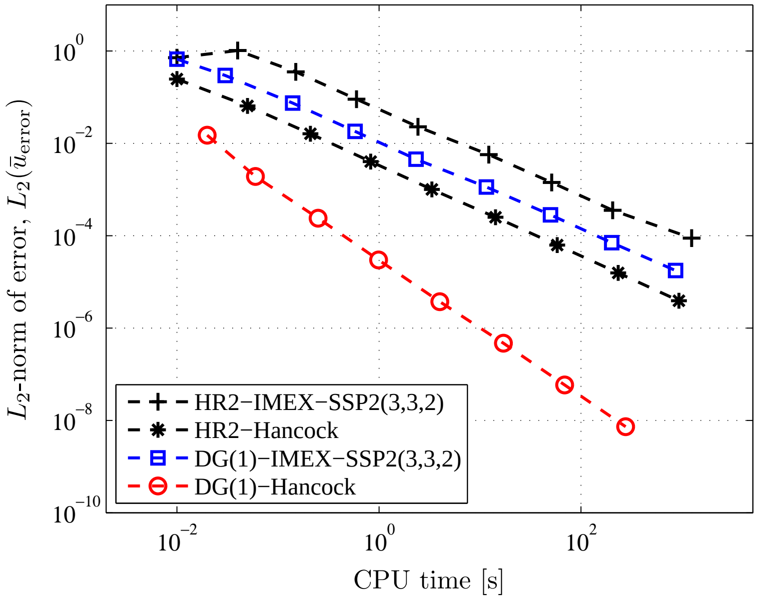
(d) The DG(1)–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---:|---:|---|---|---|---|---|---|---|---|---|
| 10 | 20 | 1.19e–01 | — | 1.65e–01 | — | 1.19e–01 | — | 1.65e–01 | — | 0.00e+00 |
| 20 | 40 | 1.66e–02 | 2.84 | 2.34e–02 | 2.82 | 1.66e–02 | 2.84 | 2.34e–02 | 2.82 | 2.00e–02 |
| 40 | 80 | 2.11e–03 | 2.97 | 2.99e–03 | 2.97 | 2.11e–03 | 2.97 | 2.99e–03 | 2.97 | 6.00e–02 |
| 80 | 160 | 2.65e–04 | 3.00 | 3.75e–04 | 2.99 | 2.65e–04 | 3.00 | 3.75e–04 | 2.99 | 2.50e–01 |
| 160 | 320 | 3.32e–05 | 3.00 | 4.69e–05 | 3.00 | 3.32e–05 | 3.00 | 4.69e–05 | 3.00 | 9.90e–01 |
| 320 | 640 | 4.15e–06 | 3.00 | 5.86e–06 | 3.00 | 4.15e–06 | 3.00 | 5.86e–06 | 3.00 | 4.00e+00 |
| 640 | 1280 | 5.18e–07 | 3.00 | 7.33e–07 | 3.00 | 5.18e–07 | 3.00 | 7.33e–07 | 3.00 | 1.71e+01 |
| 1280 | 2560 | 6.48e–08 | 3.00 | 9.16e–08 | 3.00 | 6.48e–08 | 3.00 | 9.16e–08 | 3.00 | 6.89e+01 |
| 2560 | 5120 | 8.12e–09 | 3.00 | 1.15e–08 | 3.00 | 8.12e–09 | 3.00 | 1.15e–08 | 3.00 | 2.75e+02 |

Table 4.2: A grid convergence study by solving the 1-D GHHE in the frozen limit ($r = 1/2, \epsilon = 10^3$) is performed. $L_2$, $L_\infty$-norms, rates of convergence, and CPU times of each method are tabulated.



(a) $L_2$-norms of error plotted against number of degrees of freedom. DG(1)–Hancock is the most accurate for a given number of degrees of freedom.



(b) $L_2$-norms of error plotted against CPU time. DG(1)–Hancock is the most efficient method.

Figure 4.2: 1-D GHHE grid convergence study in the frozen limit for $r = \dfrac{1}{2}$ and $\epsilon = 10^3$. The $L_2$-norms of errors shown in Table 4.2 are plotted against both degrees of freedom and CPU time. The grid convergence study shows the superiority of the DG(1)–Hancock method.

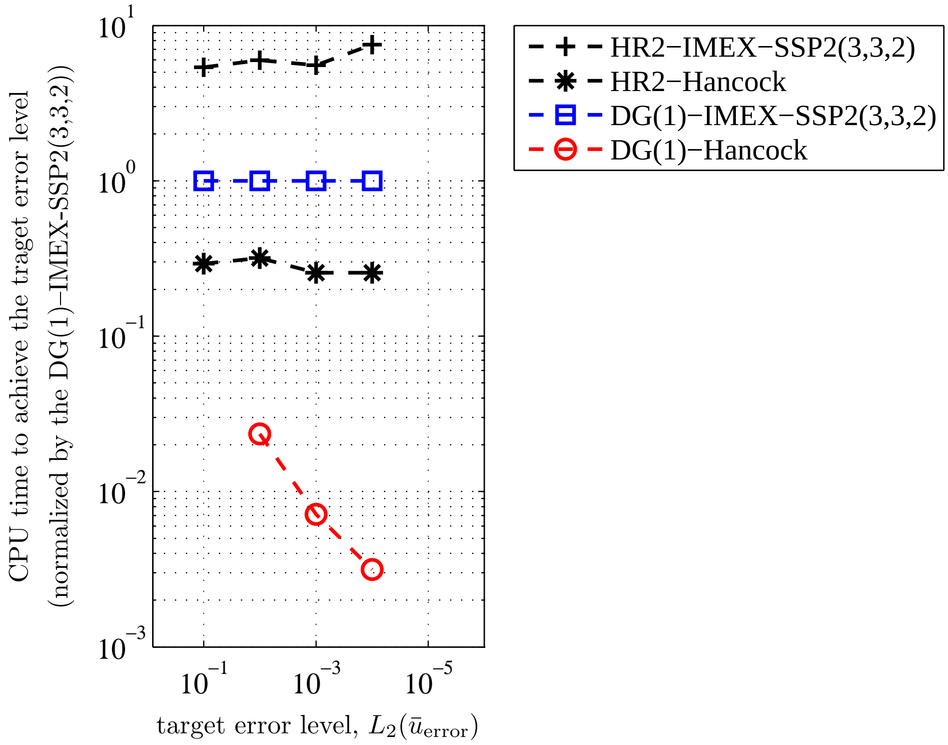


Figure 4.3: CPU time required to achieve the target error level, normalized by the DG(1)–IMEX-SSP2(3,3,2) result. The high efficiency of DG(1)–Hancock is evident.

### 4.6.3 Convergence in the Near-Equilibrium Limit I

When the relaxation time $\epsilon$ is small relative to the residence time, the source term is dominant, and the asymptotic equation is the advection-dominated advection-diffusion equation (4.5) on page 210. We have found that, for both HR2 and DG(1) methods, when $r \neq 0$, each spatial discretization method has a mesh size threshold restriction,

$$\Delta x \ \sim \ \left(\frac{\epsilon}{k^2}\right)^{\frac{1}{3}}, \tag{4.144}$$

above which numerical dissipation dominates and below which physical dissipation dominates. Note that when $r = 0$, the DG(1) method loses this restriction. We demonstrate this numerically in the next section. As before, the time step is based solely on the frozen wave speed CFL condition, yet the Courant number of the DG(1)–Hancock method is reduced to 0.3. Since the diffusion is weak, the simulations are run for many time steps until a sufficient amplitude reduction can be observed. In our particular choice of parameters, a sinusoidal wave damps 8.5% compared to its initial profile.

Firstly, to demonstrate the accuracy of the DG(1)–Hancock method qualitatively, computational results at $t_{\text{end}} = 300$ are presented in Figure 4.4. It shows that the DG(1)–Hancock method is the least dissipative and dispersive of all, whereas the HR2–IMEX method produces a completely inaccurate solution.

Secondly, in order to assess performance quantitatively, a grid-convergence study of the solution at the final time $t_{\text{end}} = 300$ is summarized in Table 4.5 and Figure 4.5. Figure 4.6 shows the normalized CPU time to achieve the target error level. Third-order convergence is observed for the DG(1)–Hancock method, whereas the HR2–IMEX, HR2–Hancock, and DG(1)–IMEX methods show second-order conver-

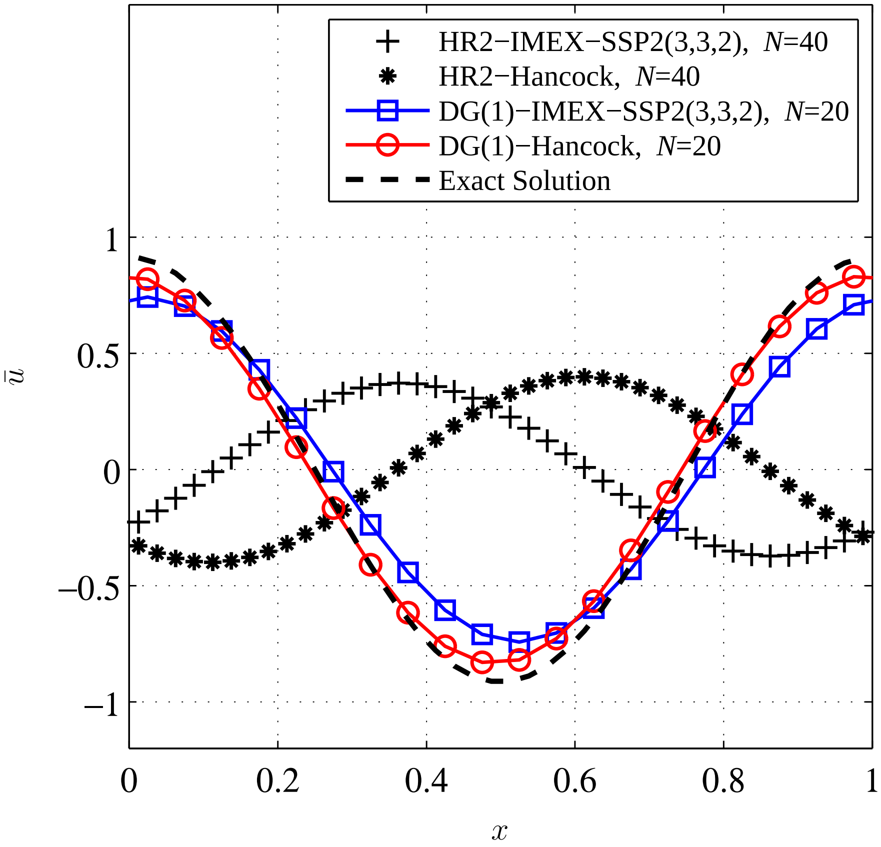


Figure 4.4: Numerical solutions at the final time $t_{\text{end}} = 300$ in the near-equilibrium limit. The DG(1)–Hancock method is the most accurate of all. Note that the exact solution itself is slightly damped by the physical dissipation.

| $N_x$ | $L_2(\bar{u}_{\text{exact}})$ | $L_\infty(\bar{u}_{\text{exact}})$ | $L_2(\bar{v}_{\text{exact}})$ | $L_\infty(\bar{v}_{\text{exact}})$ |
|---|---|---|---|---|
| 10 | 6.36e-01 | 8.56e-01 | 3.18e-01 | 4.28e-01 |
| 20 | 6.44e-01 | 9.00e-01 | 3.22e-01 | 4.50e-01 |
| 40 | 6.46e-01 | 9.11e-01 | 3.23e-01 | 4.56e-01 |

Table 4.3: $L_2$- and $L_\infty$-norms of the exact solution at $t_{\text{end}} = 300$.

| equilibrium<br>wave speed $r$ | HR2–IMEX<br>($\nu = 0.9$) | HR2–Hancock<br>($\nu = 0.9$) | DG(1)–IMEX<br>($\nu = 0.3$) | DG(1)–Hancock<br>($\nu = 0.3$) |
|---|---|---|---|---|
| 0.0 | 79 | 79 | no restriction | no restriction |
| 0.25 | 81 | 80 | 15 | 11 |
| 0.50 | 87 | 84 | 26 | 19 |
| 0.75 | 101 | 91 | 41 | 32 |
| 1.0 | physical dissipation vanishes, no restriction | | | |

Table 4.4: The threshold number of meshes $N_x^* := 1/\Delta h^*$ of each method in the near-equilibrium limit $\epsilon = 10^{-5}$ with the wave number $k = 2\pi$. The DG(1)–Hancock method requires the fewest meshes to ensure that the physical dissipation is dominant.

gence in the $L_2$-norm. The HR2–IMEX and HR2–Hancock methods only begin to converge when $N_x > 80$, while the DG(1)–IMEX and DG(1)–Hancock methods converge for $N_x > 20$. The utter lack of convergence for HR2 methods can be understood by considering the values of the $L_2$-error norms, which are roughly the $L_2$-norms of the exact solution. In other words, the numerical dissipation has so swamped the physical dissipation, that there is effectively no signal left. The actual $L_2$- and $L_\infty$-norms of exact solutions for $N_x = 10, 20, 40$ are listed in Table 4.3. Considering Table 4.5, all four methods begin to exhibit convergence precisely when the mesh size $\Delta x$ becomes smaller than the theoretical limit $\Delta h^*_{\text{method}}$ obtained by a Fourier analysis. The threshold number of meshes $N_x^*$ for each method is tabulated in Table 4.4.

As for DG(1) methods, we observe that the order of accuracy seem to decrease on finer grids. This can be understood by the Fourier analysis given in (4.103). For instance, the DG(1)–MOL has two leading dispersion errors: $c_3 \propto \Delta x^2$ and $\tilde{c}_3 \propto \epsilon\Delta x$. On coarse grids, hence $c_3 \gg \tilde{c}_3$, second-order convergence is pronounced. However, as grids get finder and start resolving the relaxation scale, these two coefficients become comparable, and eventually the first-order error, $\tilde{c}_3$, dominates. A similar observation can be made for the DG(1)–Hancock method regarding its convergence reduction from third- to second-order.

Lastly, it can be observed that the HR2–Hancock method in this limit loses the high accurate observed in the frozen limit. The reason is that the shift condition of the HR2–Hancock method owing to the upwind flux is no longer preserved in the near-equilibrium limit. Note that in this limit the flux becomes the Rusanov/HLL1 type (on page 234); excessive numerical dissipation reduces the accuracy to almost the same level as for the HR2–IMEX method.

(a) The HR2–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $\Delta x/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 8.65 | 6.36e−01 | — | 8.56e−01 | — | 3.18e−01 | — | 4.28e−01 | — | 0.00e+00 |
| 20 | 20 | 4.33 | 6.45e−01 | −0.02 | 9.01e−01 | −0.07 | 3.22e−01 | −0.02 | 4.50e−01 | −0.07 | 3.00e−02 |
| 40 | 40 | 2.16 | 8.46e−01 | −0.39 | 1.20e+00 | −0.41 | 4.23e−01 | −0.39 | 5.98e−01 | −0.41 | 1.20e−01 |
| 80 | 80 | 1.08 | 3.57e−01 | 1.24 | 5.05e−01 | 1.24 | 1.79e−01 | 1.24 | 2.52e−01 | 1.24 | 4.50e−01 |
| 160 | 160 | 0.54 | 9.40e−02 | 1.93 | 1.33e−01 | 1.92 | 4.70e−02 | 1.93 | 6.65e−02 | 1.92 | 1.79e+00 |
| 320 | 320 | 0.27 | 2.36e−02 | 1.99 | 3.34e−02 | 1.99 | 1.18e−02 | 1.99 | 1.67e−02 | 1.99 | 7.37e+00 |
| 640 | 640 | 0.14 | 5.93e−03 | 1.99 | 8.39e−03 | 1.99 | 2.97e−03 | 1.99 | 4.19e−03 | 1.99 | 3.64e+01 |
| 1280 | 1280 | 0.07 | 1.49e−03 | 1.99 | 2.11e−03 | 1.99 | 7.46e−04 | 1.99 | 1.06e−03 | 1.99 | 1.53e+02 |
| 2560 | 2560 | 0.03 | 3.78e−04 | 1.98 | 5.34e−04 | 1.98 | 1.89e−04 | 1.98 | 2.67e−04 | 1.98 | 6.13e+02 |
| 5120 | 5120 | 0.02 | 9.64e−05 | 1.97 | 1.36e−04 | 1.97 | 4.82e−05 | 1.97 | 6.81e−05 | 1.97 | 3.89e+03 |

(b) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $\Delta x/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 8.43 | 6.36e−01 | — | 8.56e−01 | — | 3.18e−01 | — | 4.28e−01 | — | 1.00e−02 |
| 20 | 20 | 4.21 | 6.45e−01 | −0.02 | 9.01e−01 | −0.07 | 3.23e−01 | −0.02 | 4.51e−01 | −0.07 | 4.00e−02 |
| 40 | 40 | 2.11 | 8.83e−01 | −0.45 | 1.25e+00 | −0.47 | 4.42e−01 | −0.45 | 6.24e−01 | −0.47 | 1.60e−01 |
| 80 | 80 | 1.05 | 3.81e−01 | 1.21 | 5.39e−01 | 1.21 | 1.91e−01 | 1.21 | 2.70e−01 | 1.21 | 6.30e−01 |
| 160 | 160 | 0.53 | 1.00e−01 | 1.93 | 1.42e−01 | 1.92 | 5.02e−02 | 1.93 | 7.10e−02 | 1.92 | 2.52e+00 |
| 320 | 320 | 0.26 | 2.51e−02 | 2.00 | 3.55e−02 | 2.00 | 1.25e−02 | 2.00 | 1.77e−02 | 2.00 | 1.01e+01 |
| 640 | 640 | 0.13 | 6.20e−03 | 2.02 | 8.77e−03 | 2.02 | 3.10e−03 | 2.02 | 4.38e−03 | 2.02 | 4.25e+01 |
| 1280 | 1280 | 0.07 | 1.51e−03 | 2.04 | 2.13e−03 | 2.04 | 7.53e−04 | 2.04 | 1.06e−03 | 2.04 | 1.74e+02 |
| 2560 | 2560 | 0.03 | 3.50e−04 | 2.11 | 4.95e−04 | 2.11 | 1.75e−04 | 2.11 | 2.47e−04 | 2.11 | 6.98e+02 |
| 5120 | 5120 | 0.02 | 7.12e−05 | 2.30 | 1.01e−04 | 2.30 | 3.56e−05 | 2.30 | 5.04e−05 | 2.30 | 2.85e+03 |

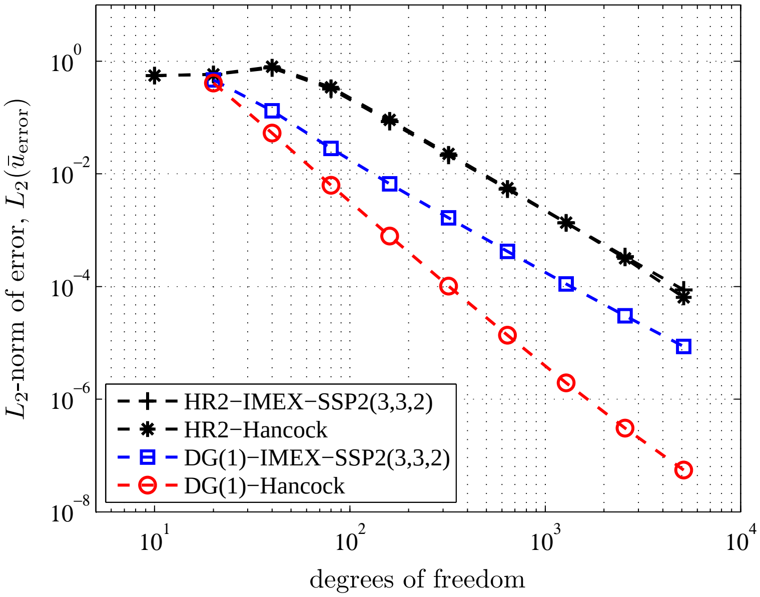
(c) The DG(1)–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.3$)

| $N_x$ | DOF | $\Delta x/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 20 | 2.62 | 5.29e–01 | — | 7.21e–01 | — | 2.65e–01 | — | 3.61e–01 | — | 3.00e–02 |
| 20 | 40 | 1.31 | 1.46e–01 | 1.86 | 2.06e–01 | 1.81 | 7.34e–02 | 1.85 | 1.03e–01 | 1.80 | 1.10e–01 |
| 40 | 80 | 0.65 | 3.13e–02 | 2.22 | 4.43e–02 | 2.22 | 1.57e–02 | 2.22 | 2.22e–02 | 2.22 | 4.60e–01 |
| 80 | 160 | 0.33 | 7.37e–03 | 2.09 | 1.04e–02 | 2.09 | 3.69e–03 | 2.09 | 5.22e–03 | 2.09 | 1.76e+00 |
| 160 | 320 | 0.16 | 1.83e–03 | 2.01 | 2.59e–03 | 2.01 | 9.17e–04 | 2.01 | 1.30e–03 | 2.01 | 7.12e+00 |
| 320 | 640 | 0.08 | 4.69e–04 | 1.97 | 6.63e–04 | 1.97 | 2.35e–04 | 1.97 | 3.32e–04 | 1.97 | 3.46e+01 |
| 640 | 1280 | 0.04 | 1.23e–04 | 1.93 | 1.74e–04 | 1.93 | 6.16e–05 | 1.93 | 8.72e–05 | 1.93 | 1.49e+02 |
| 1280 | 2560 | 0.02 | 3.36e–05 | 1.88 | 4.75e–05 | 1.88 | 1.68e–05 | 1.88 | 2.38e–05 | 1.88 | 6.00e+02 |
| 2560 | 5120 | 0.01 | 9.53e–06 | 1.82 | 1.35e–05 | 1.82 | 4.77e–06 | 1.82 | 6.74e–06 | 1.82 | 2.84e+03 |

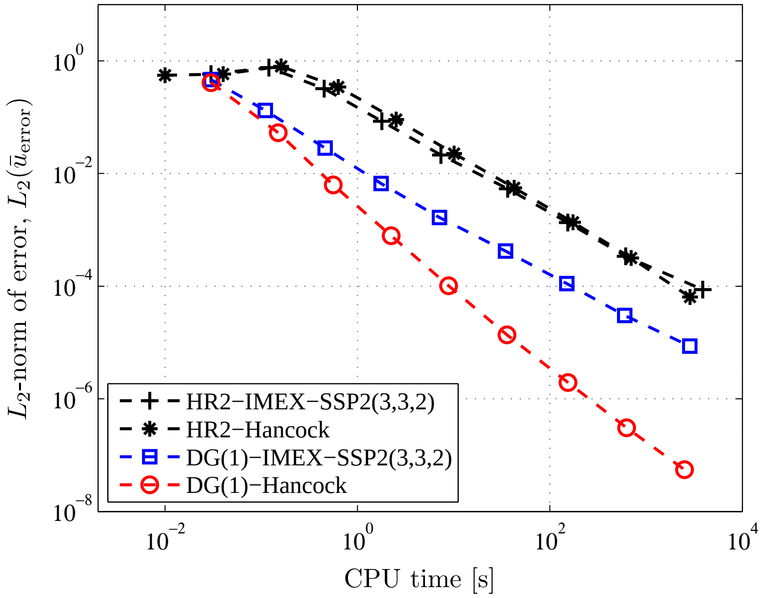
(d) The DG(1)–Hancock method ($\tilde{\nu} = 0.3$)

| $N_x$ | DOF | $\Delta x/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 20 | 1.95 | 4.48e–01 | — | 6.30e–01 | — | 2.24e–01 | — | 3.15e–01 | — | 3.00e–02 |
| 20 | 40 | 0.97 | 5.91e–02 | 2.92 | 8.35e–02 | 2.91 | 2.96e–02 | 2.92 | 4.18e–02 | 2.91 | 1.50e–01 |
| 40 | 80 | 0.49 | 6.99e–03 | 3.08 | 9.88e–03 | 3.08 | 3.49e–03 | 3.08 | 4.94e–03 | 3.08 | 5.60e–01 |
| 80 | 160 | 0.24 | 8.70e–04 | 3.01 | 1.23e–03 | 3.01 | 4.35e–04 | 3.01 | 6.15e–04 | 3.01 | 2.23e+00 |
| 160 | 320 | 0.12 | 1.12e–04 | 2.95 | 1.59e–04 | 2.95 | 5.62e–05 | 2.95 | 7.95e–05 | 2.95 | 8.86e+00 |
| 320 | 640 | 0.06 | 1.51e–05 | 2.89 | 2.14e–05 | 2.89 | 7.57e–06 | 2.89 | 1.07e–05 | 2.89 | 3.57e+01 |
| 640 | 1280 | 0.03 | 2.17e–06 | 2.80 | 3.07e–06 | 2.80 | 1.09e–06 | 2.80 | 1.54e–06 | 2.80 | 1.54e+02 |
| 1280 | 2560 | 0.02 | 3.41e–07 | 2.67 | 4.82e–07 | 2.67 | 1.70e–07 | 2.67 | 2.41e–07 | 2.67 | 6.26e+02 |
| 2560 | 5120 | 0.01 | 6.13e–08 | 2.48 | 8.66e–08 | 2.48 | 3.06e–08 | 2.48 | 4.33e–08 | 2.48 | 2.50e+03 |

Table 4.5: A grid convergence study by solving the 1-D GHHE in the near-equilibrium limit ($r = 1/2, \epsilon = 10^{-5}$) is performed. $L_2$, $L_\infty$-norms, rates of convergence, and CPU times of each method are tabulated.



(a) $L_2$-norms of error plotted against number of degrees of freedom. DG(1)–Hancock is the most accurate in a given number of degrees of freedom.



(b) $L_2$-norms of error plotted against CPU time. DG(1)–Hancock is the most efficient method.

Figure 4.5: 1-D GHHE grid convergence study in the near-equilibrium limit for $r = 1/2$ and $\epsilon = 10^{-5}$. The $L_2$-norms of errors shown in Table 4.5 are plotted against both degrees of freedom and CPU time. The grid convergence study shows the superiority of the DG(1)–Hancock method.

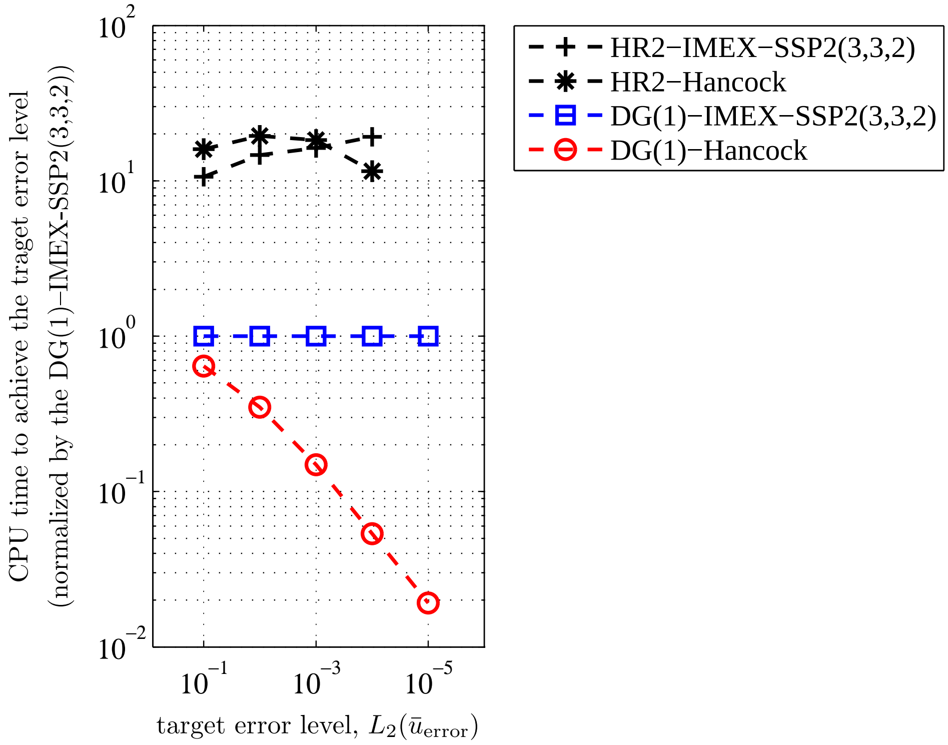


Figure 4.6: CPU time required to achieve the target error level, normalized by the DG(1)–IMEX-SSP2(3,3,2) result. The high efficiency of DG(1)–Hancock is evident.

### 4.6.4 Convergence in the Near-Equilibrium Limit II

When the equilibrium wave speed vanishes ($r = 0$), both DG(1)–IMEX and DG(1)–Hancock methods should exhibit second-order convergence without the threshold. Results are provided in Table 4.6 and Figure 4.7. Figure 4.8 shows the normalized CPU time to achieve the target error level. Indeed, DG(1) methods converge at a second-order rate and exhibit no threshold. This suggests that the threshold for $r = \frac{1}{2}$ was in fact a demonstration of the behavior of the spatial discretization.

As before, both HR2–IMEX and HR2–Hancock methods exhibit a threshold, and do not appear to begin to converge until $\Delta x < \Delta h^*_{\mathrm{HR2}}$. See Table 4.4 for a threshold number of mesh size with respect to $r$. For $N_x > 80$, both HR2 methods appear to converge at a rate greater than two. This can be explained by the fact that, for our choice of parameters, the $\frac{1}{8}\boxed{\Delta x^3}\,k^4$ error term in (4.105) on page 236 still dominates the numerical error, even if it is smaller than the physical dissipation. Nevertheless, due to the large coefficient of that term, DG(1) methods still yield more accurate results. The almost identical error levels of DG(1)–IMEX and DG(1)–Hancock are explained by Fourier analyses presented in (4.105) on page 236; both method possess identical dominant dissipation errors.

Rather surprisingly, the convergence of the HR2–Hancock method stalls on finer grids. The cause is unclear, yet it might be related to the anti-diffusive (positive) error appearing in (4.105b) on page 236. Since the method is space-time coupled, we are not able to trace back the source of error to either spatial or temporal discretization.

(a) The HR2–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $\Delta x/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 7.90 | 6.18e−01 | — | 8.31e−01 | — | 3.88e−05 | — | 5.49e−05 | — | 0.00e+00 |
| 20 | 20 | 3.95 | 6.25e−01 | −0.02 | 8.73e−01 | −0.07 | 3.93e−05 | −0.02 | 5.49e−05 | 0.00 | 3.00e−02 |
| 40 | 40 | 1.98 | 3.75e−01 | 0.74 | 5.29e−01 | 0.72 | 2.36e−05 | 0.74 | 3.33e−05 | 0.72 | 1.10e−01 |
| 80 | 80 | 0.99 | 6.77e−02 | 2.47 | 9.57e−02 | 2.46 | 4.28e−06 | 2.46 | 6.05e−06 | 2.46 | 4.50e−01 |
| 160 | 160 | 0.49 | 8.92e−03 | 2.93 | 1.26e−02 | 2.92 | 5.57e−07 | 2.94 | 7.87e−07 | 2.94 | 1.78e+00 |
| 320 | 320 | 0.25 | 1.12e−03 | 2.99 | 1.59e−03 | 2.99 | 7.11e−08 | 2.97 | 1.00e−07 | 2.97 | 7.29e+00 |
| 640 | 640 | 0.12 | 1.41e−04 | 2.99 | 2.00e−04 | 2.99 | 8.58e−09 | 3.05 | 1.21e−08 | 3.05 | 3.65e+01 |
| 1280 | 1280 | 0.06 | 1.78e−05 | 2.99 | 2.52e−05 | 2.99 | 1.12e−09 | 2.93 | 1.59e−09 | 2.93 | 1.55e+02 |
| 2560 | 2560 | 0.03 | 2.26e−06 | 2.98 | 3.20e−06 | 2.98 | 1.17e−10 | 3.26 | 1.66e−10 | 3.26 | 6.18e+02 |
| 5120 | 5120 | 0.02 | 2.93e−07 | 2.95 | 4.14e−07 | 2.95 | 1.65e−11 | 2.83 | 2.34e−11 | 2.83 | 4.03e+03 |

(b) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $\Delta x/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 7.90 | 6.18e−01 | — | 8.31e−01 | — | 3.88e−05 | — | 5.49e−05 | — | 1.00e−02 |
| 20 | 20 | 3.95 | 6.25e−01 | −0.02 | 8.73e−01 | −0.07 | 3.93e−05 | −0.02 | 5.49e−05 | 0.00 | 4.00e−02 |
| 40 | 40 | 1.98 | 3.74e−01 | 0.74 | 5.28e−01 | 0.73 | 2.35e−05 | 0.74 | 3.31e−05 | 0.73 | 1.60e−01 |
| 80 | 80 | 0.99 | 6.75e−02 | 2.47 | 9.53e−02 | 2.47 | 4.22e−06 | 2.48 | 5.96e−06 | 2.47 | 6.40e−01 |
| 160 | 160 | 0.49 | 8.83e−03 | 2.93 | 1.25e−02 | 2.93 | 5.50e−07 | 2.94 | 7.78e−07 | 2.94 | 2.54e+00 |
| 320 | 320 | 0.25 | 1.10e−03 | 3.00 | 1.56e−03 | 3.00 | 6.81e−08 | 3.01 | 9.63e−08 | 3.01 | 1.01e+01 |
| 640 | 640 | 0.12 | 1.41e−04 | 2.97 | 2.00e−04 | 2.97 | 8.57e−09 | 2.99 | 1.21e−08 | 2.99 | 4.26e+01 |
| 1280 | 1280 | 0.06 | 3.90e−05 | 1.85 | 5.52e−05 | 1.85 | 2.42e−09 | 1.82 | 3.42e−09 | 1.82 | 1.74e+02 |
| 2560 | 2560 | 0.03 | 3.20e−05 | 0.29 | 4.52e−05 | 0.29 | 2.01e−09 | 0.27 | 2.84e−09 | 0.27 | 6.96e+02 |
| 5120 | 5120 | 0.02 | 2.64e−05 | 0.27 | 3.74e−05 | 0.27 | 1.66e−09 | 0.27 | 2.35e−09 | 0.27 | 2.84e+03 |

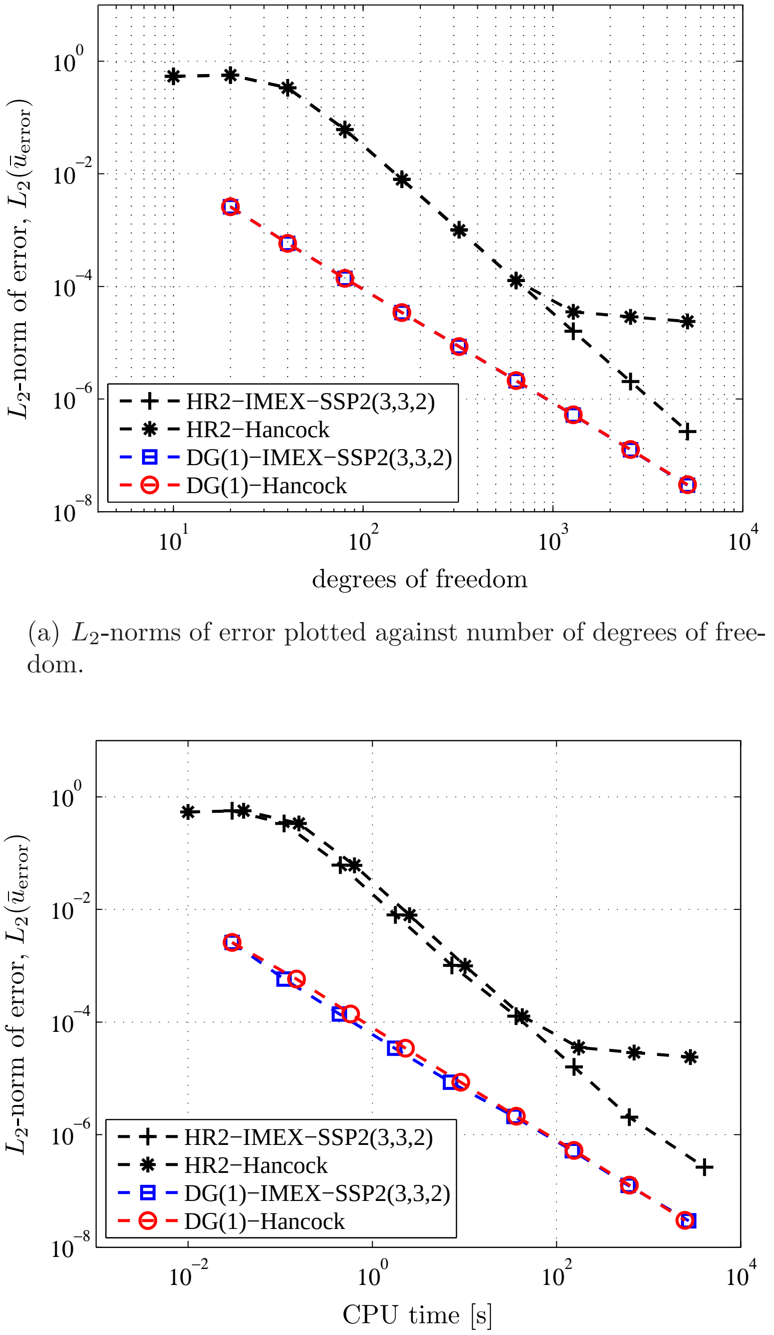
(c) The DG(1)–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.3$)

| $N_x$ | DOF | $\Delta x/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 20 | — | 2.99e–03 | — | 4.02e–03 | — | 6.11e–07 | — | 8.64e–07 | — | 3.00e–02 |
| 20 | 40 | — | 6.45e–04 | 2.21 | 9.00e–04 | 2.16 | 1.65e–07 | 1.89 | 2.31e–07 | 1.91 | 1.10e–01 |
| 40 | 80 | — | 1.55e–04 | 2.06 | 2.18e–04 | 2.05 | 4.22e–08 | 1.97 | 5.95e–08 | 1.96 | 4.40e–01 |
| 80 | 160 | — | 3.82e–05 | 2.02 | 5.40e–05 | 2.01 | 1.06e–08 | 1.99 | 1.50e–08 | 1.99 | 1.75e+00 |
| 160 | 320 | — | 9.49e–06 | 2.01 | 1.34e–05 | 2.01 | 2.67e–09 | 1.99 | 3.78e–09 | 1.99 | 7.06e+00 |
| 320 | 640 | — | 2.35e–06 | 2.01 | 3.32e–06 | 2.01 | 6.74e–10 | 1.99 | 9.54e–10 | 1.99 | 3.44e+01 |
| 640 | 1280 | — | 5.78e–07 | 2.02 | 8.17e–07 | 2.02 | 1.70e–10 | 1.99 | 2.41e–10 | 1.99 | 1.49e+02 |
| 1280 | 2560 | — | 1.40e–07 | 2.05 | 1.98e–07 | 2.05 | 4.27e–11 | 1.99 | 6.04e–11 | 1.99 | 5.98e+02 |
| 2560 | 5120 | — | 3.28e–08 | 2.09 | 4.64e–08 | 2.09 | 1.04e–11 | 2.04 | 1.47e–11 | 2.04 | 2.72e+03 |

(d) The DG(1)–Hancock method ($\tilde{\nu} = 0.3$)

| $N_x$ | DOF | $\Delta x/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 20 | — | 2.99e–03 | — | 4.02e–03 | — | 1.14e–06 | — | 1.61e–06 | — | 3.00e–02 |
| 20 | 40 | — | 6.45e–04 | 2.21 | 9.01e–04 | 2.16 | 2.85e–07 | 1.99 | 3.99e–07 | 2.01 | 1.50e–01 |
| 40 | 80 | — | 1.55e–04 | 2.06 | 2.18e–04 | 2.05 | 7.14e–08 | 2.00 | 1.01e–07 | 1.99 | 5.80e–01 |
| 80 | 160 | — | 3.82e–05 | 2.02 | 5.40e–05 | 2.01 | 1.79e–08 | 2.00 | 2.52e–08 | 2.00 | 2.29e+00 |
| 160 | 320 | — | 9.50e–06 | 2.01 | 1.34e–05 | 2.01 | 4.46e–09 | 2.00 | 6.30e–09 | 2.00 | 9.14e+00 |
| 320 | 640 | — | 2.36e–06 | 2.01 | 3.34e–06 | 2.01 | 1.11e–09 | 2.01 | 1.57e–09 | 2.01 | 3.66e+01 |
| 640 | 1280 | — | 5.81e–07 | 2.02 | 8.22e–07 | 2.02 | 2.75e–10 | 2.01 | 3.89e–10 | 2.01 | 1.54e+02 |
| 1280 | 2560 | — | 1.41e–07 | 2.04 | 2.00e–07 | 2.04 | 6.74e–11 | 2.03 | 9.54e–11 | 2.03 | 6.18e+02 |
| 2560 | 5120 | — | 3.33e–08 | 2.09 | 4.70e–08 | 2.09 | 1.61e–11 | 2.07 | 2.27e–11 | 2.07 | 2.47e+03 |

Table 4.6: A grid convergence study by solving the 1-D GHHE in the near-equilibrium limit ($r = 0, \epsilon = 10^{-5}$) is performed. $L_2$, $L_\infty$-norms, rates of convergence, and CPU times of each method are tabulated.



(a) $L_2$-norms of error plotted against number of degrees of freedom.

(b) $L_2$-norms of error plotted against CPU time.

Figure 4.7: 1-D GHHE grid convergence study in the near-equilibrium limit for $r = 0$ and $\epsilon = 10^{-5}$. The $L_2$-norms of errors shown in Table 4.6 are plotted against both degrees of freedom and CPU time. HR2 methods converge at the third order, while DG(1) methods are second order.

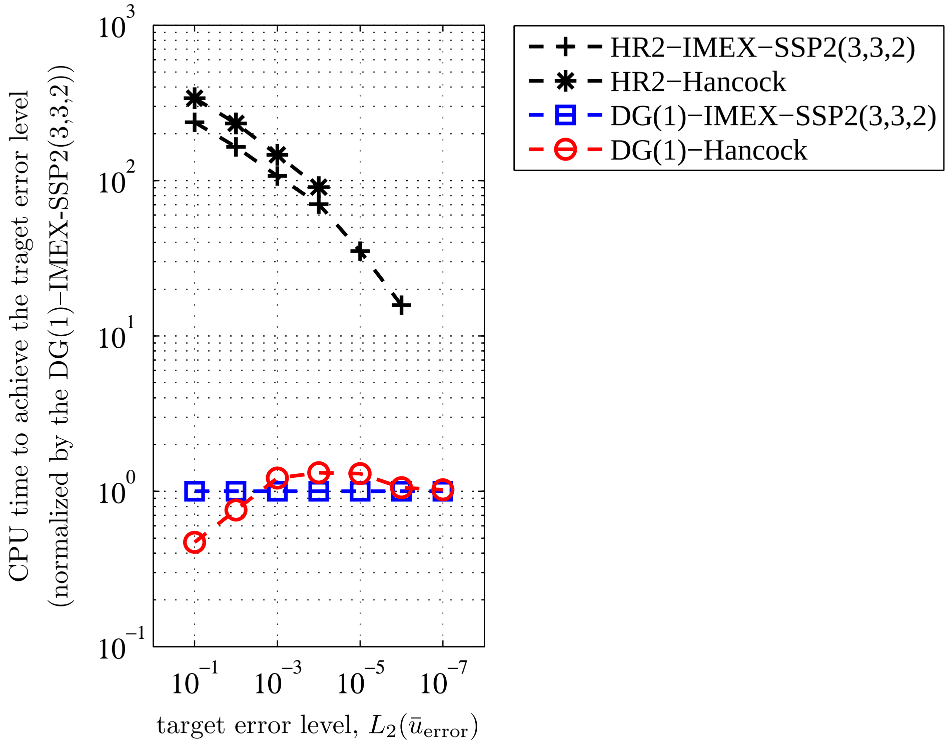


Figure 4.8: CPU time required to achieve the target error level, normalized by the DG(1)–RK2 result. The superiority of the DG(1)–Hancock method over the DG(1)–IMEX method is lost when zero equilibrium speed is considered.

## 4.7 Grid Convergence Study in 2-D

### 4.7.1 Problem Definition

We consider the two-dimensional model problem (4.107) on page 237. To confirm the analysis, consider an initial-value problem on a periodic domain $(x, y) \in [0, 1] \times [0, 1]$ in each direction, and use the harmonic initial condition,

$$\begin{aligned}\mathbf{u}(x, y, 0) &= \Re\{\mathbf{u}_0 \exp(ik_x x) \exp(ik_y y)\} \\ &= \mathbf{u}_0[\cos(k_x x)\cos(k_y y) - \sin(k_x x)\sin(k_y y)], \end{aligned} \tag{4.145}$$

where $\mathbf{u}_0 = [1, 1, 1]^T$ and $k_x = k_y = 2\pi$. This initial condition has variation in the diagonal direction (at 45° angle with the $x$-axis) and is uniform in the other diagonal direction. A dispersion analysis, included in Appendix C on page 354, provides the exact solution for the computation of the $L_2$-norm.

Following the 1-D analysis, the DG(1)–Hancock method is compared to the HR2–IMEX, HR2–Hancock, and DG(1)–IMEX methods. For each method, we consider both a frozen limit ($\epsilon \gg 1$, non-stiff) and a near-equilibrium limit ($\epsilon \ll 1$, stiff), with two equilibrium wave speeds in the stiff regime:

$$\text{frozen (non-stiff) limit: } r = s = \frac{1}{2}, \quad \epsilon = 10^3, \quad t_{\text{end}} = 100, \tag{4.146a}$$

$$\text{near-equilibrium (stiff) limit: } r = s = 0, \frac{1}{2}, \quad \epsilon = 10^{-5}, \quad t_{\text{end}} = 150, \tag{4.146b}$$

and we compute the error norms relative to an exact analytic solution. Note that $r = s = 0$ corresponds to no advection, hence the system reduces to a pure diffusion equation:

$$\partial_t u = \epsilon \left(\partial_{xx} u + \partial_{yy} u\right). \tag{4.147}$$

In both cases, the analytical damping factor is obtained by

$$u(t_{\text{end}}) = e^{-\epsilon(2-r^2-s^2)k^2\, t_{\text{end}}} \approx 0.915, \tag{4.148}$$

thus, a sinusoidal wave is physically dissipated by 8.5%. The Courant numbers based on the unit wave speed:

$$(\nu_{\text{2D}})_{\text{method}} := 1\frac{\Delta t}{\Delta x} + 1\frac{\Delta t}{\Delta y}, \tag{4.149}$$

are set as

$$\nu_{\text{HR2–MOL}} = \nu_{\text{HR2–Hancock}} = 0.9, \tag{4.150a}$$

$$\nu_{\text{DG(1)–MOL}} = 0.3, \tag{4.150b}$$

$$\nu_{\text{DG(1)–Hancock}} = \begin{cases} 0.6 & \text{frozen limit} \\ 0.3 & \text{near-equilibrium limit.} \end{cases} \tag{4.150c}$$

Note that the maximum stable Courant number of the DG(1)–Hancock method is reduced to 0.66 in two dimensions. A DG(1) method requires cell averages $\bar{\mathbf{u}}_{jk}$ and slopes $\overline{\Delta_x \mathbf{u}}_{jk}, \overline{\Delta_y \mathbf{u}}_{jk}$ in each cell for initial conditions. Given an initial function $\mathbf{u}(x, y, 0)$, the initial cell values are obtained from

$$\bar{\mathbf{u}}_{jk} = \frac{1}{\Delta x \Delta y} \iint\limits_{C_{jk}} \mathbf{u}(x, y, 0)\, dxdy, \tag{4.151a}$$

$$\overline{\Delta_x \mathbf{u}}_{jk} = \frac{12}{\Delta x^2 \Delta y} \iint\limits_{C_{jk}} \mathbf{u}(x, y, 0)(x - x_j)\, dxdy, \tag{4.151b}$$

$$\overline{\Delta_y \mathbf{u}}_{jk} = \frac{12}{\Delta x \Delta y^2} \iint\limits_{C_{jk}} \mathbf{u}(x, y, 0)(y - y_k)\, dxdy, \tag{4.151c}$$

where the integration is over the cell $C_{jk} := [x_{j-1/2}, x_{j+1/2}] \times [y_{k-1/2}, y_{k+1/2}]$.

### 4.7.2 Convergence in the Frozen Limit

Table 4.7 and Figure 4.9 demonstrate the expected third-order convergence for the 2-D DG(1)–Hancock method, and second-order convergence for the DG(1)–IMEX, HR2–IMEX, and HR2–Hancock methods in the frozen limit, although, the onset of the asymptotic convergence rate requires slightly more points per dimension than in 1D. We note that in this frozen problem the initial-value distribution propagates along the diagonal of the domain, which produces an effective mesh size in the propagation direction larger than $h$ by a factor $\sqrt{2}$. This factor is enough to explain the difference between the 1-D and 2-D results. Figure 4.10 shows the normalized CPU time to achieve the target error level. The high efficiency of the DG(1)–Hancock method is evident.

(a) The HR2–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.9$)

| $N_x \times N_y$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$, $L_2(\bar{w}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$, $L_\infty(\bar{w}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 100 | 7.35e–01 | — | 1.02e+00 | — | 6.32e–01 | — | 8.93e–01 | — | 2.00e–01 |
| 20× 20 | 400 | 7.33e–01 | 0.00 | 1.03e+00 | −0.01 | 6.38e–01 | −0.01 | 9.02e–01 | −0.01 | 1.76e+00 |
| 40× 40 | 1600 | 1.19e+00 | −0.70 | 1.68e+00 | −0.71 | 1.07e+00 | −0.75 | 1.52e+00 | −0.75 | 1.49e+01 |
| 80× 80 | 6400 | 5.98e–01 | 0.99 | 8.46e–01 | 0.99 | 5.35e–01 | 1.00 | 7.56e–01 | 1.00 | 2.38e+02 |
| 160×160 | 25600 | 1.63e–01 | 1.88 | 2.30e–01 | 1.88 | 1.38e–01 | 1.95 | 1.96e–01 | 1.95 | 2.10e+03 |
| 320×320 | 102400 | 4.12e–02 | 1.98 | 5.83e–02 | 1.98 | 3.46e–02 | 2.00 | 4.89e–02 | 2.00 | 1.75e+04 |
| 640×640 | 409600 | 1.03e–02 | 2.00 | 1.46e–02 | 2.00 | 8.63e–03 | 2.00 | 1.22e–02 | 2.00 | 1.34e+05 |

(b) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x \times N_y$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$, $L_2(\bar{w}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$, $L_\infty(\bar{w}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 100 | 7.35e–01 | — | 1.02e+00 | — | 6.32e–01 | — | 8.93e–01 | — | 2.70e–01 |
| 20× 20 | 400 | 8.69e–01 | −0.24 | 1.22e+00 | −0.26 | 7.06e–01 | −0.16 | 9.98e–01 | −0.16 | 2.24e+00 |
| 40× 40 | 1600 | 3.70e–01 | 1.23 | 5.24e–01 | 1.22 | 2.91e–01 | 1.28 | 4.12e–01 | 1.28 | 1.84e+01 |
| 80× 80 | 6400 | 9.63e–02 | 1.94 | 1.36e–01 | 1.94 | 7.79e–02 | 1.90 | 1.10e–01 | 1.90 | 2.21e+02 |
| 160×160 | 25600 | 2.39e–02 | 2.01 | 3.38e–02 | 2.01 | 1.97e–02 | 1.98 | 2.78e–02 | 1.98 | 2.41e+03 |
| 320×320 | 102400 | 5.96e–03 | 2.01 | 8.42e–03 | 2.01 | 4.94e–03 | 1.99 | 6.98e–03 | 1.99 | 1.63e+04 |
| 640×640 | 409600 | 1.49e–03 | 2.00 | 2.10e–03 | 2.00 | 1.24e–03 | 2.00 | 1.75e–03 | 2.00 | 1.17e+05 |

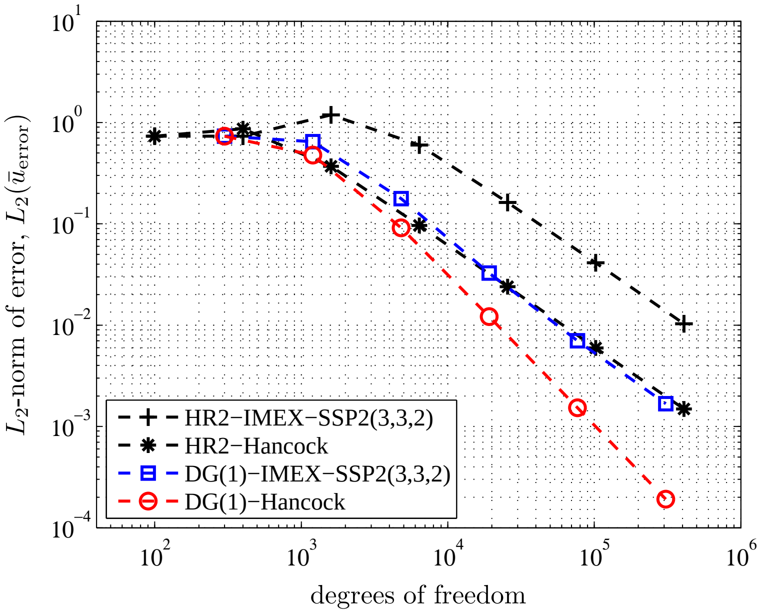
(c) The DG(1)–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.3$)

| $N_x \times N_y$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$, $L_2(\bar{w}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$, $L_\infty(\bar{w}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 300 | 7.35e–01 | — | 1.02e+00 | — | 6.32e–01 | — | 8.93e–01 | — | 1.08e+00 |
| 20× 20 | 1200 | 6.49e–01 | 0.18 | 9.15e–01 | 0.16 | 5.65e–01 | 0.16 | 7.97e–01 | 0.16 | 9.33e+00 |
| 40× 40 | 4800 | 1.78e–01 | 1.87 | 2.51e–01 | 1.87 | 1.61e–01 | 1.81 | 2.28e–01 | 1.81 | 1.05e+02 |
| 80× 80 | 19200 | 3.27e–02 | 2.44 | 4.63e–02 | 2.44 | 3.00e–02 | 2.43 | 4.24e–02 | 2.42 | 1.62e+03 |
| 160×160 | 76800 | 6.99e–03 | 2.23 | 9.89e–03 | 2.23 | 6.24e–03 | 2.27 | 8.82e–03 | 2.27 | 1.26e+04 |
| 320×320 | 307200 | 1.68e–03 | 2.06 | 2.37e–03 | 2.06 | 1.45e–03 | 2.10 | 2.06e–03 | 2.10 | 9.83e+04 |

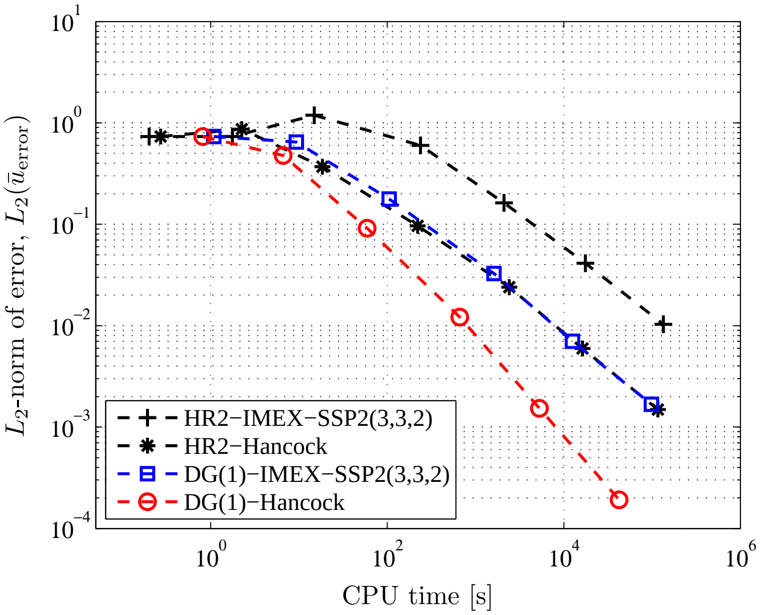
(d) The DG(1)–Hancock method ($\tilde{\nu} = 0.6$)

| $N_x \times N_y$ | DOF | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$, $L_2(\bar{w}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$, $L_\infty(\bar{w}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 300 | 7.34e–01 | — | 1.02e+00 | — | 6.31e–01 | — | 8.92e–01 | — | 8.10e–01 |
| 20× 20 | 1200 | 4.78e–01 | 0.62 | 6.67e–01 | 0.61 | 4.08e–01 | 0.63 | 5.77e–01 | 0.63 | 6.60e+00 |
| 40× 40 | 4800 | 9.13e–02 | 2.39 | 1.29e–01 | 2.37 | 7.80e–02 | 2.39 | 1.10e–01 | 2.39 | 5.90e+01 |
| 80× 80 | 19200 | 1.21e–02 | 2.91 | 1.72e–02 | 2.91 | 1.04e–02 | 2.91 | 1.47e–02 | 2.91 | 6.61e+02 |
| 160×160 | 76800 | 1.53e–03 | 2.99 | 2.16e–03 | 2.99 | 1.31e–03 | 2.99 | 1.85e–03 | 2.99 | 5.29e+03 |
| 320×320 | 307200 | 1.91e–04 | 3.00 | 2.71e–04 | 3.00 | 1.64e–04 | 3.00 | 2.32e–04 | 3.00 | 4.23e+04 |

Table 4.7: $L_2$-, $L_\infty$-norms and rates of convergence for $r = s = \dfrac{1}{2}$ in the frozen limit ($\epsilon = 10^3$) for the 2-D GHHE system.



(a) $L_2$-norms of error plotted against number of degrees of freedom. DG(1)–Hancock is the most accurate for a given number of degrees of freedom.



(b) $L_2$-norms of error plotted against CPU time. DG(1)–Hancock is the most efficient method.

Figure 4.9: 2-D GHHE grid convergence study in the frozen limit for $r = s = \dfrac{1}{2}$ and $\epsilon = 10^3$. The $L_2$-norms of errors shown in Table 4.7 are plotted against both degrees of freedom and CPU time. The grid-convergence study shows the superiority of the DG(1)–Hancock method.

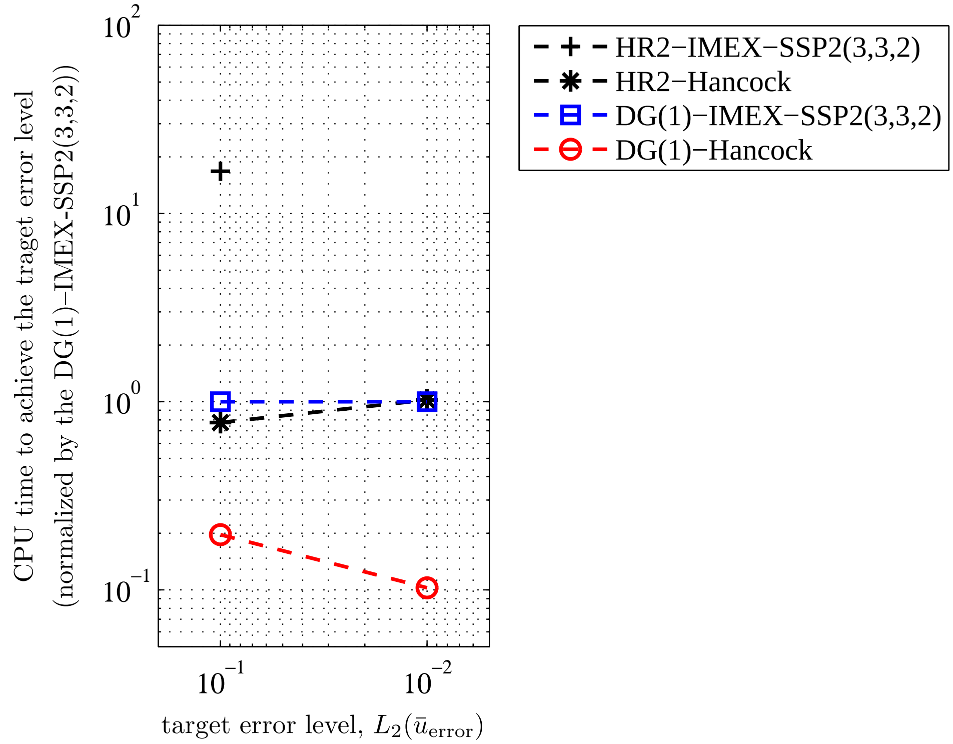


Figure 4.10: CPU time required to achieve the target error level, normalized by the DG(1)–IMEX-SSP2(3,3,2) result. The high efficiency of DG(1)–Hancock is preserved for a two-dimensional problem.

| equilibrium<br>wave speed $r$ | HR2 ($\nu = 0.9$) | DG(1) ($\nu = 0.3$) |
|---|---|---|
| 0.0 | 79 | 55 |
| 0.25 | 81 | 56 |
| 0.50 | 87 | 62 |
| 0.75 | 104 | 76 |
| 1.0 | physical dissipation vanishes, no restriction | |

Table 4.8: The threshold number of meshes $N_x^* := 1/\Delta h^*$ of each method in the near-equilibrium limit $\epsilon = 10^{-5}$ with wave number $k_x = k_y = 2\pi$.

### 4.7.3 Convergence in the Near-Equilibrium Limit I

Results are shown in Table 4.9 and Figure 4.11. Figure 4.12 shows the normalized CPU time to achieve the target error level. Both HR2–IMEX and HR2–Hancock methods follow the Fourier analysis: second-order convergence is observed. The DG(1)–Hancock method preserves third-order accuracy in two-dimensional problems. Compared to the 1-D result, an unexpected result is obtained for the DG(1)–IMEX method. First, it converges as third-order, then deteriorates towards second-order. Since Fourier analysis shows that DG(1) method is third-order in space, this reduction must be related to the temporal error. A similar trend is also observed in the results for the frozen limit in Table 4.7(c). Unfortunately, a Fourier analysis of the fully discrete method could not be performed due to the complexity of the formulas, so the exact source of error is still unknown.

These grid-convergence results reveal that if one allows a moderately large error, then both DG(1)–Hancock and DG(1)–IMEX are comparable in both accuracy and efficiency. However, if a low error level is required, then the DG(1)–Hancock method is superior to other methods because the method is truly third-order accurate.

(a) The HR2–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.9$)

| $N_x \times N_y$ | DOF | $\Delta h/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 100 | 8.70 | 6.26e−01 | — | 8.85e−01 | — | 3.13e−01 | — | 4.43e−01 | — | 3.10e−01 |
| 20× 20 | 400 | 4.35 | 6.42e−01 | −0.04 | 9.08e−01 | −0.04 | 3.21e−01 | −0.04 | 4.54e−01 | −0.04 | 2.68e+00 |
| 40× 40 | 1600 | 2.17 | 8.45e−01 | −0.40 | 1.19e+00 | −0.39 | 4.22e−01 | −0.40 | 5.95e−01 | −0.39 | 2.22e+01 |
| 80× 80 | 6400 | 1.09 | 3.57e−01 | 1.24 | 5.05e−01 | 1.24 | 1.79e−01 | 1.24 | 2.52e−01 | 1.24 | 3.76e+02 |
| 160×160 | 25600 | 0.54 | 9.40e−02 | 1.93 | 1.33e−01 | 1.93 | 4.70e−02 | 1.93 | 6.65e−02 | 1.93 | 3.14e+03 |
| 320×320 | 102400 | 0.27 | 2.36e−02 | 1.99 | 3.34e−02 | 1.99 | 1.18e−02 | 1.99 | 1.67e−02 | 1.99 | 2.50e+04 |
| 640×640 | 409600 | 0.14 | 5.93e−03 | 2.00 | 8.39e−03 | 2.00 | 2.97e−03 | 2.00 | 4.19e−03 | 2.00 | 2.04e+05 |

(b) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x \times N_y$ | DOF | $\Delta h/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 100 | 8.70 | 6.26e−01 | — | 8.85e−01 | — | 3.13e−01 | — | 4.43e−01 | — | 4.00e−01 |
| 20× 20 | 400 | 4.35 | 6.43e−01 | −0.04 | 9.09e−01 | −0.04 | 3.21e−01 | −0.04 | 4.54e−01 | −0.04 | 3.35e+00 |
| 40× 40 | 1600 | 2.17 | 8.82e−01 | −0.46 | 1.25e+00 | −0.46 | 4.41e−01 | −0.46 | 6.23e−01 | −0.46 | 2.70e+01 |
| 80× 80 | 6400 | 1.09 | 3.81e−01 | 1.21 | 5.39e−01 | 1.21 | 1.91e−01 | 1.21 | 2.70e−01 | 1.21 | 3.78e+02 |
| 160×160 | 25600 | 0.54 | 1.00e−01 | 1.93 | 1.42e−01 | 1.93 | 5.02e−02 | 1.93 | 7.10e−02 | 1.93 | 3.05e+03 |
| 320×320 | 102400 | 0.27 | 2.51e−02 | 2.00 | 3.55e−02 | 2.00 | 1.25e−02 | 2.00 | 1.77e−02 | 2.00 | 2.41e+04 |
| 640×640 | 409600 | 0.14 | 6.18e−03 | 2.02 | 8.74e−03 | 2.02 | 3.09e−03 | 2.02 | 4.37e−03 | 2.02 | 1.85e+05 |

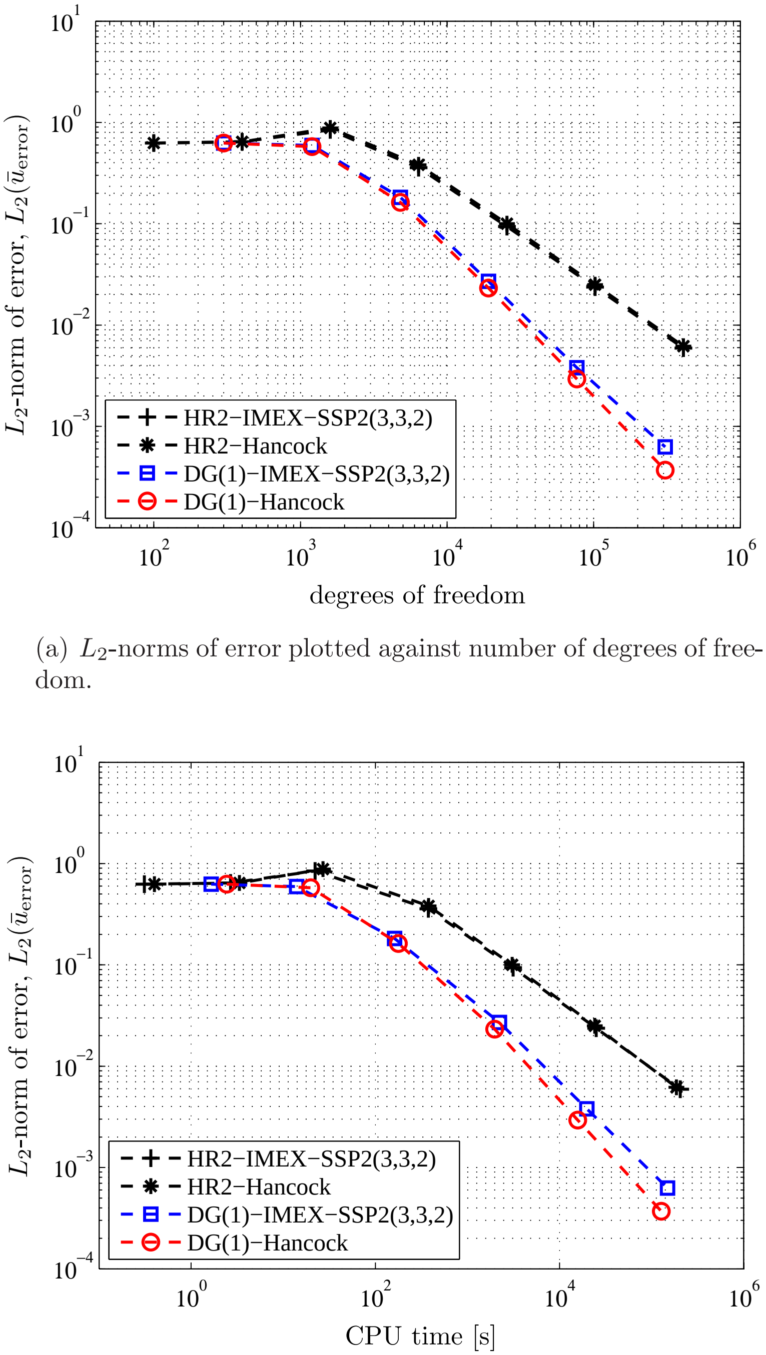
(c) The DG(1)–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.3$)

| $N_x \times N_y$ | DOF | $\Delta h/\Delta h^{\star}$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 300 | 6.19 | 6.26e−01 | — | 8.85e−01 | — | 3.13e−01 | — | 4.43e−01 | — | 1.65e+00 |
| 20× 20 | 1200 | 3.10 | 5.94e−01 | 0.08 | 8.40e−01 | 0.08 | 2.97e−01 | 0.08 | 4.20e−01 | 0.08 | 1.40e+01 |
| 40× 40 | 4800 | 1.55 | 1.82e−01 | 1.71 | 2.57e−01 | 1.71 | 9.11e−02 | 1.71 | 1.29e−01 | 1.71 | 1.62e+02 |
| 80× 80 | 19200 | 0.77 | 2.70e−02 | 2.75 | 3.82e−02 | 2.75 | 1.35e−02 | 2.75 | 1.91e−02 | 2.75 | 2.22e+03 |
| 160×160 | 76800 | 0.39 | 3.79e−03 | 2.83 | 5.35e−03 | 2.83 | 1.90e−03 | 2.83 | 2.68e−03 | 2.83 | 1.97e+04 |
| 320×320 | 307200 | 0.19 | 6.26e−04 | 2.60 | 8.86e−04 | 2.60 | 3.14e−04 | 2.60 | 4.44e−04 | 2.60 | 1.49e+05 |

(d) The DG(1)–Hancock method ($\tilde{\nu} = 0.3$)

| $N_x \times N_y$ | DOF | $\Delta h/\Delta h^{\star}$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 300 | 6.19 | 6.26e−01 | — | 8.85e−01 | — | 3.13e−01 | — | 4.43e−01 | — | 2.43e+00 |
| 20× 20 | 1200 | 3.10 | 5.77e−01 | 0.12 | 8.16e−01 | 0.12 | 2.89e−01 | 0.12 | 4.08e−01 | 0.12 | 2.00e+01 |
| 40× 40 | 4800 | 1.55 | 1.63e−01 | 1.83 | 2.30e−01 | 1.83 | 8.14e−02 | 1.83 | 1.15e−01 | 1.83 | 1.78e+02 |
| 80× 80 | 19200 | 0.77 | 2.31e−02 | 2.81 | 3.27e−02 | 2.81 | 1.16e−02 | 2.81 | 1.64e−02 | 2.81 | 1.98e+03 |
| 160×160 | 76800 | 0.39 | 2.94e−03 | 2.97 | 4.16e−03 | 2.97 | 1.47e−03 | 2.97 | 2.08e−03 | 2.97 | 1.57e+04 |
| 320×320 | 307200 | 0.19 | 3.69e−04 | 3.00 | 5.22e−04 | 3.00 | 1.85e−04 | 3.00 | 2.61e−04 | 3.00 | 1.27e+05 |

Table 4.9: $L_2$-, $L_\infty$-norms and rates of convergence for $r = \dfrac{1}{2}$ in the near-equilibrium limit ($\epsilon = 10^{-5}$) for the 2-D GHHE system.



(a) $L_2$-norms of error plotted against number of degrees of freedom.

(b) $L_2$-norms of error plotted against CPU time.

Figure 4.11: 2-D GHHE grid convergence study in the near-equilibrium limit for $r = s = \dfrac{1}{2}$ and $\epsilon = 10^{-5}$. The $L_2$-norms of errors shown in Table 4.9 are plotted against both degrees of freedom and CPU time. The grid-convergence study still shows a slight superiority of the DG(1)–Hancock method.

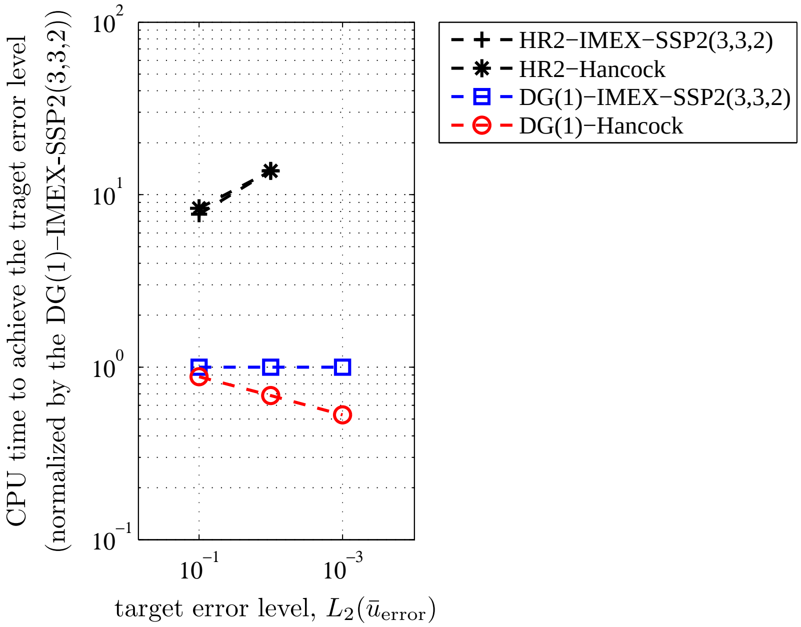


Figure 4.12: CPU time required to achieve the target error level, normalized by the DG(1)–IMEX-SSP2(3,3,2) result.

### 4.7.4 Convergence in the Near-Equilibrium Limit II

In a single dimension, both DG(1)–IMEX and DG(1)–Hancock methods for $r = 0$ do not appear to have a threshold above which numerical dissipation dominates over physical dissipation. However, problems of practical significance are multidimensional, and multiple dimensions frequently introduce additional couplings. Analysis suggests that part of the multidimensional error is independent of the wave speeds, and causes the mesh-size restriction even when zero wave speeds, $r = s = 0$, are considered. To gauge whether 2-D DG(1) methods actually exhibit this behavior, we conduct a two-dimensional numerical experiment.

The results are shown in Table 4.10 and Figure 4.13. Figure 4.14 shows the normalized CPU time to achieve the target error level. It is observed that all four methods converge with third-order accuracy. This can be understood again by the Fourier analysis given in (4.136) on page 245. Even though the analyses do not include the temporal discretization error, they predict third-order convergence on coarse grids for both HR2 and DG(1) spatial discretizations. When the grids become finer, we expect a reduction to second-order convergence because the spatial error is proportional to $\epsilon \Delta h^2$. The dominant dissipation error of the DG(1) method is $\frac{1}{3}$ of that of HR2, but Figure 4.13 appears to indicate the opposite; the reason is that DG(1) uses three times as more degrees of freedom as HR2. In the end, the errors of both methods are quite comparable. These numerical results show that the DG(1) spatial discretization loses its superiority to HR2 in the near-equilibrium limit with $r = s = 0$.

(a) The HR2–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.9$)

| $N_x \times N_y$ | DOF | $\Delta h/\Delta h^{\star}$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 100 | 7.90 | 6.08e−01 | — | 8.59e−01 | — | 3.82e−05 | — | 5.14e−05 | — | 3.10e−01 |
| 20× 20 | 400 | 3.95 | 6.23e−01 | −0.03 | 8.80e−01 | −0.03 | 3.91e−05 | −0.03 | 5.53e−05 | −0.11 | 2.66e+00 |
| 40× 40 | 1600 | 1.98 | 3.75e−01 | 0.73 | 5.30e−01 | 0.73 | 2.35e−05 | 0.73 | 3.33e−05 | 0.73 | 2.23e+01 |
| 80× 80 | 6400 | 0.99 | 6.77e−02 | 2.47 | 9.58e−02 | 2.47 | 4.26e−06 | 2.47 | 6.02e−06 | 2.47 | 3.73e+02 |
| 160×160 | 25600 | 0.49 | 8.91e−03 | 2.93 | 1.26e−02 | 2.93 | 5.56e−07 | 2.94 | 7.86e−07 | 2.94 | 3.16e+03 |
| 320×320 | 102400 | 0.25 | 1.12e−03 | 2.99 | 1.59e−03 | 2.99 | 7.01e−08 | 2.99 | 9.92e−08 | 2.99 | 2.46e+04 |
| 640×640 | 409600 | 0.12 | 1.41e−04 | 2.99 | 2.00e−04 | 2.99 | 8.54e−09 | 3.04 | 1.21e−08 | 3.04 | 1.95e+05 |

(b) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x \times N_y$ | DOF | $\Delta h/\Delta h^{\star}$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 100 | 7.90 | 6.08e−01 | — | 8.59e−01 | — | 3.82e−05 | — | 5.14e−05 | — | 4.00e−01 |
| 20× 20 | 400 | 3.95 | 6.23e−01 | −0.03 | 8.80e−01 | −0.03 | 3.91e−05 | −0.03 | 5.53e−05 | −0.11 | 3.30e+00 |
| 40× 40 | 1600 | 1.98 | 3.74e−01 | 0.74 | 5.29e−01 | 0.74 | 2.35e−05 | 0.74 | 3.32e−05 | 0.74 | 2.63e+01 |
| 80× 80 | 6400 | 0.99 | 6.74e−02 | 2.47 | 9.54e−02 | 2.47 | 4.22e−06 | 2.48 | 5.97e−06 | 2.48 | 3.28e+02 |
| 160×160 | 25600 | 0.49 | 8.83e−03 | 2.93 | 1.25e−02 | 2.93 | 5.50e−07 | 2.94 | 7.78e−07 | 2.94 | 2.65e+03 |
| 320×320 | 102400 | 0.25 | 1.11e−03 | 3.00 | 1.56e−03 | 3.00 | 6.82e−08 | 3.01 | 9.65e−08 | 3.01 | 2.08e+04 |
| 640×640 | 409600 | 0.12 | 1.53e−04 | 2.85 | 2.17e−04 | 2.85 | 9.37e−09 | 2.87 | 1.32e−08 | 2.87 | 1.67e+05 |

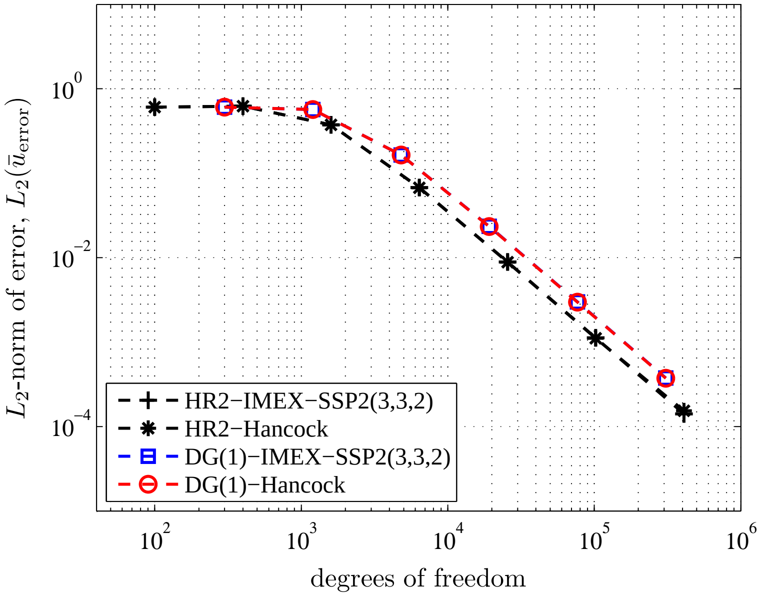
(c) The DG(1)–IMEX-SSP2(3,3,2) method ($\tilde{\nu} = 0.3$)

| $N_x \times N_y$ | DOF | $\Delta h/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 300 | 5.48 | 6.08e–01 | — | 8.59e–01 | — | 3.82e–05 | — | 5.14e–05 | — | 1.56e+00 |
| 20× 20 | 1200 | 2.74 | 5.66e–01 | 0.10 | 8.01e–01 | 0.10 | 3.56e–05 | 0.10 | 5.04e–05 | 0.03 | 1.34e+01 |
| 40× 40 | 4800 | 1.37 | 1.64e–01 | 1.79 | 2.32e–01 | 1.79 | 1.03e–05 | 1.79 | 1.45e–05 | 1.79 | 1.55e+02 |
| 80× 80 | 19200 | 0.68 | 2.35e–02 | 2.81 | 3.32e–02 | 2.81 | 1.46e–06 | 2.82 | 2.06e–06 | 2.82 | 2.20e+03 |
| 160×160 | 76800 | 0.34 | 2.99e–03 | 2.97 | 4.23e–03 | 2.97 | 1.84e–07 | 2.99 | 2.60e–07 | 2.99 | 1.80e+04 |
| 320×320 | 307200 | 0.17 | 3.76e–04 | 2.99 | 5.31e–04 | 2.99 | 2.26e–08 | 3.02 | 3.19e–08 | 3.02 | 1.44e+05 |

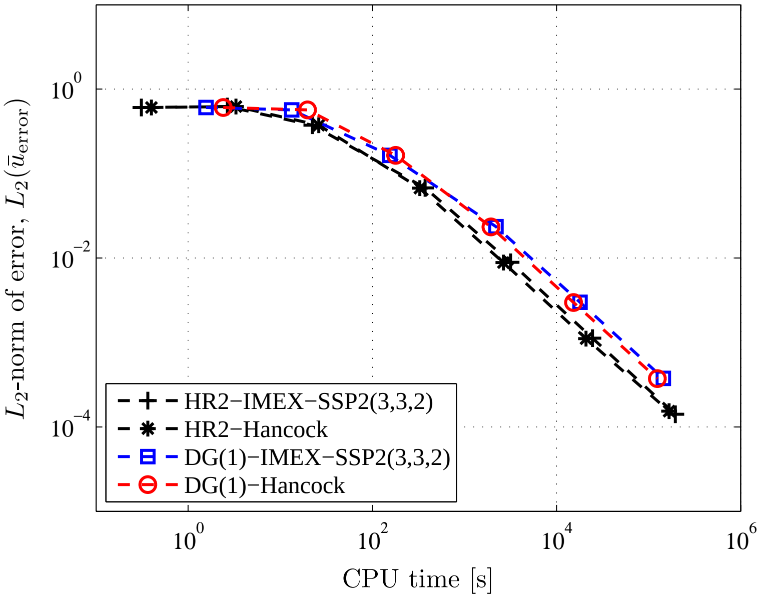
(d) The DG(1)–Hancock method ($\tilde{\nu} = 0.3$)

| $N_x \times N_y$ | DOF | $\Delta h/\Delta h^\star$ | $L_2(\bar{u}_{\text{error}})$ | Rate | $L_\infty(\bar{u}_{\text{error}})$ | Rate | $L_2(\bar{v}_{\text{error}})$ | Rate | $L_\infty(\bar{v}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10× 10 | 300 | 5.48 | 6.08e–01 | — | 8.59e–01 | — | 3.82e–05 | — | 5.14e–05 | — | 2.41e+00 |
| 20× 20 | 1200 | 2.74 | 5.66e–01 | 0.10 | 8.01e–01 | 0.10 | 3.55e–05 | 0.10 | 5.03e–05 | 0.03 | 1.98e+01 |
| 40× 40 | 4800 | 1.37 | 1.64e–01 | 1.79 | 2.32e–01 | 1.79 | 1.02e–05 | 1.80 | 1.45e–05 | 1.80 | 1.79e+02 |
| 80× 80 | 19200 | 0.68 | 2.34e–02 | 2.81 | 3.31e–02 | 2.81 | 1.45e–06 | 2.82 | 2.05e–06 | 2.82 | 1.94e+03 |
| 160×160 | 76800 | 0.34 | 2.98e–03 | 2.97 | 4.21e–03 | 2.97 | 1.82e–07 | 2.99 | 2.58e–07 | 2.99 | 1.53e+04 |
| 320×320 | 307200 | 0.17 | 3.73e–04 | 3.00 | 5.28e–04 | 3.00 | 2.22e–08 | 3.04 | 3.14e–08 | 3.04 | 1.24e+05 |

Table 4.10: $L_2$-, $L_\infty$-norms and rates of convergence for $r = s = 0$ in the near-equilibrium limit ($\epsilon = 10^{-5}$) for the 2-D GHHE system.



(a) $L_2$-norms of error plotted against number of degrees of freedom.



(b) $L_2$-norms of error plotted against CPU time.

Figure 4.13: 2-D GHHE grid convergence study in the near-equilibrium limit for $r = s = 0$ and $\epsilon = 10^{-5}$. The $L_2$-norms of errors shown in Table 4.10 are plotted against both degrees of freedom and CPU time. The grid-convergence study shows all methods converge in third-order, and error levels are comparable.

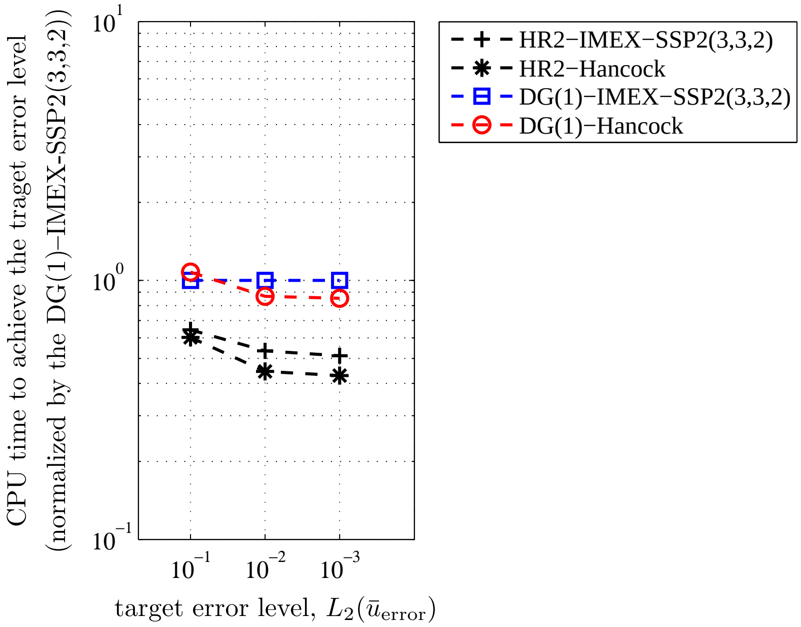


Figure 4.14: CPU time required to achieve the target error level, normalized by the DG(1)–IMEX-SSP2(3,3,2) result. DG(1) methods are now less efficient than HR2 methods due to the extra degrees of freedom. They require for achieving high accuracy.

## 4.8 Grid-Convergence Study for Nonlinear Hyperbolic–Relaxation Equations

### 4.8.1 The Euler Equations with Heat Transfer

To demonstrate the accuracy of the DG(1)–Hancock method when applied to a nonlinear hyperbolic-relaxation system, the Euler equations with heat transfer, which reduce to the isothermal Euler equations in the equilibrium limit, are adopted as a model equation [Pem93b]:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho u H\end{pmatrix}=-\frac{1}{\epsilon}\begin{pmatrix}0\\ 0\\ \rho(T-T_0)\end{pmatrix}, \tag{4.152}$$

where the pressure is given by the ideal gas law, $p:=(\gamma-1)\rho e=\rho RT$. The frozen characteristic speeds are $u\pm a, u$, where the speed of sound is given by $a:=\sqrt{\gamma p/\rho}$. In the equilibrium limit ($\epsilon\to 0$), the nonequilibrium temperature $T$ converges to the constant equilibrium temperature $T_0$ instantaneously. As a result, the above equations tend asymptotically to the following isothermal Euler equations:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p^*\end{pmatrix}=\begin{pmatrix}0\\ 0\end{pmatrix}, \tag{4.153}$$

where the gas becomes polytropic with equilibrium $\gamma=1$ and the pressure is given by $p^*(\rho):=\rho RT_0$. The equilibrium characteristic speeds are $u\pm a^*$, where the constant speed of sound is $a^*:=\sqrt{p^*/\rho}=\sqrt{RT_0}$.

Consider an initial-value problem with the following $\mathbb{C}^{\infty}$ initial distributions:

$$\rho_0(x) = \exp\left(\frac{u_0(x)}{a^*}\right), \tag{4.154a}$$

$$u_0(x) = \begin{cases} -a^*, & x < -5, \\ a^* \tanh\left[-\dfrac{10x}{(x+5)(x-5)}\right], & x \in [-5, 5], \\ a^*, & x > 5, \end{cases} \tag{4.154b}$$

$$p_0(x) = (a^*)^2 \rho_0(x), \tag{4.154c}$$

plotted in Figure 4.15. The initial conditions are chosen such that the analytical solution of the isothermal Euler equations becomes a simple wave solution; one of the Riemann invariants remains constant: $J^{-}_{\text{iso}}(x,t) = \ln \rho_0 - \dfrac{u_0}{a^*} \equiv 0$. Also, the flow properties are non-equilibrium ($T \neq T_0$) only within the domain $x \in [-5, 5]$. The equilibrium speed of sound, equilibrium temperature, and the ratio of specific heats are taken to be:

$$a^* = \sqrt{0.4}, \quad T_0 = 1, \quad \gamma = 1.4. \tag{4.155}$$

The computational domain is confined to $x \in [-16, 16]$ with uniform meshes, and the solution at the time $t_{\text{end}} = 5.0$ is used for checking grid convergence and comparison among methods.

### Richardson Extrapolation for Grid Convergence

In order to determine the order of accuracy of the various schemes, we need to know the exact solution of (4.152) at the time $t_{\text{end}}$. When the frozen limit ($\epsilon \to \infty$) is considered, the exact solution can be obtained with the regular Euler equations. Conversely, at the equilibrium limit ($\epsilon \to 0$), the exact solution is derived from the isothermal Euler equations, (4.153). Simple-wave solutions are available for these two conservation laws; however, the resulting exact solutions are not strictly the exact solution of (4.152). For instance, when $O(\epsilon) \ll 1$, an asymptotic expansion shows that a series of $O\left(\epsilon^k\right)$-term appears on the right-hand side of the conservation laws,

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = O(\epsilon)\, \partial_{xx} \mathbf{U} + \dots . \tag{4.156}$$

Thus, the actual exact solution should be derived from (4.156), which contains an infinite series in terms of $\epsilon$. This could be possible if a simple system is considered; however, the derivation can be cumbersome.

To overcome this difficulty, Richardson extrapolation, which does not require knowledge of the exact solution, is adopted for the grid-convergence study. In brief, successive grid solutions provide an estimated exact solution, $\bar{u}_{i,\text{exact}}$, and the coefficients of the local truncation error, $c_j$, in the following form:

$$\bar{\mathbf{u}}_i = \bar{\mathbf{u}}_{i,\text{exact}} + \mathbf{c}_1 \Delta x + \mathbf{c}_2 \Delta x^2 + \mathbf{c}_3 \Delta x^3 + \dots . \tag{4.157}$$

Thus, once the right-hand side of the above equation is computed, the error at the cell $i$ is given by

$$\begin{aligned} \text{error}_i(\mathbf{u}) &:= \bar{\mathbf{u}}_i - \bar{\mathbf{u}}_{i,\text{exact}} \\ &= (\mathbf{c}_1)_i \Delta x + (\mathbf{c}_2)_i \Delta x^2 + (\mathbf{c}_3)_i \Delta x^3 + \dots , \end{aligned} \tag{4.158}$$

after which, the $L_p$-norm on the uniform grid is obtained by

$$L_p(\mathbf{u}) := \left[ \frac{1}{N} \sum_{i=1}^{N} \left| \text{error}_i(\mathbf{u}) \right|^p \right]^{1/p} . \tag{4.159}$$

### Numerical Results

The DG(1)–Hancock method is compared to two semi-discrete methods: HR2–MOL and DG(1)–MOL. As to the time integrator, we adopt the IMEX–SSP2(3,3,2) (2.101).

In order to verify the accuracy of a method in the stiff regime ($\epsilon \ll O(1)$), the relaxation time is taken as

$$\epsilon = 10^{-8}. \tag{4.160}$$

Due to the implicit treatment of the source term, the time step is solely constrained by the maximum acoustic wave speed, thus

$$\Delta t = \nu_{\text{method}} \frac{\Delta x}{|u| + a}, \tag{4.161}$$

where $\nu_{\text{method}}$ is the Courant number of the method used. Here, we set a Courant number as 90% of a method's linear stability limit:

$$\begin{aligned} \nu_{\text{HR2–MOL}} &= 0.9, \\ \nu_{\text{DG(1)–MOL}} &= 0.3, \\ \nu_{\text{DG(1)–Hancock}} &= 0.8. \end{aligned} \tag{4.162}$$

As to the stability of the DG(1)–Hancock method, the maximum Courant number depends on the dimensionless equilibrium wave speed $|r|$. For the Euler equations with heat transfer, $r$ is defined by

$$r := \frac{|u| + a^*}{|u| + a}. \tag{4.163}$$

Based on the initial conditions (4.154), we have $\dfrac{|u|}{a^*} = 1$ and $\dfrac{a}{a^*} = \sqrt{1.4}$, thus

$$r = \frac{2}{1 + \sqrt{1.4}} \approx 0.916.$$

The approximated polynomial (3.162) on page 155 provides the maximum Courant number for the specific $r$ such that

$$\nu_{\max}(r) \approx 0.907. \tag{4.164}$$

The density distribution at $t_{\text{end}} = 5.0$, superposed to the exact solution of the isothermal Euler equations, is shown in Figure 4.16. Even though the exact solution of the isothermal Euler equations does not contain the $O(\epsilon)$-term, the numerical result is in good agreement with the exact solution. This is because the relaxation time, $\epsilon$, is so small ($\epsilon = 10^{-8}$), thus the $O(\epsilon)$-term is negligible, at least in the eyeball norm.

To disclose the order of accuracy of each method, Richardson extrapolation was adopted for the grid-convergence study. The $L_1$-norm of the density error, $L_1(\rho)$, is shown in Figure 4.17. The plot shows that all three methods are second-order accurate, yet DG(1)–Hancock has an error nearly an order of magnitude lower than HR2/DG(1)–MOL. Note that previously the linear analysis predicted third-order convergence of the DG(1)–Hancock method (4.103d); however, due to the linearization of the source term in space, (2.21) on page 41, the method reduces to second-order accuracy for the nonlinear source in (4.152).

In Figure 4.17 it is shown that the DG(1)–Hancock method again is superior to the other two methods in terms of accuracy; however, the method would not be attractive if it required excessive CPU time to achieve the high accuracy. Thus, we examined the overall efficiency of each method. Again, we defined the efficiency based on the total CPU time to achieve a target error level. CPU time normalized by the CPU time of the DG(1)–MOL method for a specific error level is shown in Figure 4.18. It clearly shows the high efficiency of DG(1)–Hancock compared to HR2/DG(1)–MOL. Such a high efficiency is achieved by a combination of accurate computation typical of the DG spatial discretization and the wide stability range owing to the Hancock temporal discretization.

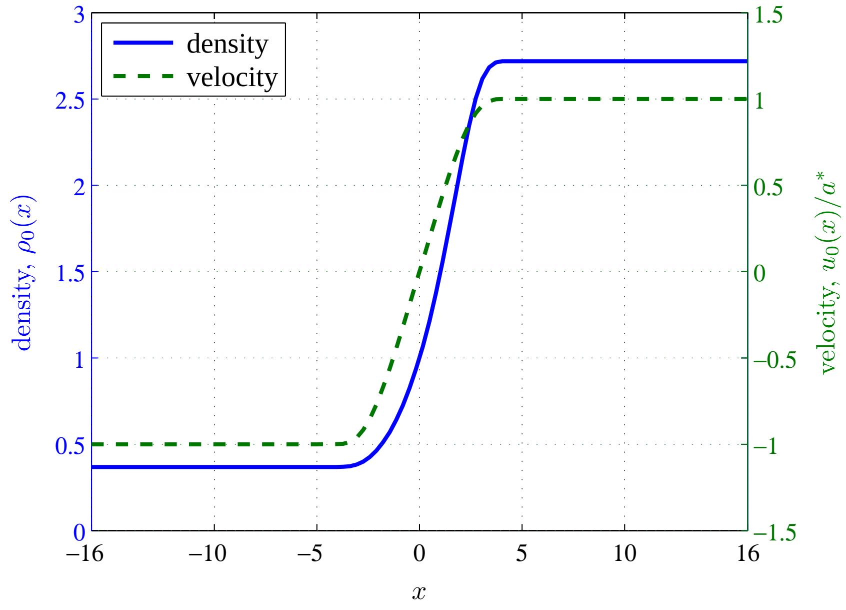


Figure 4.15: The initial distributions of density, $\rho_0(x)$, and normalized velocity, $u_0(x)/a^*$.

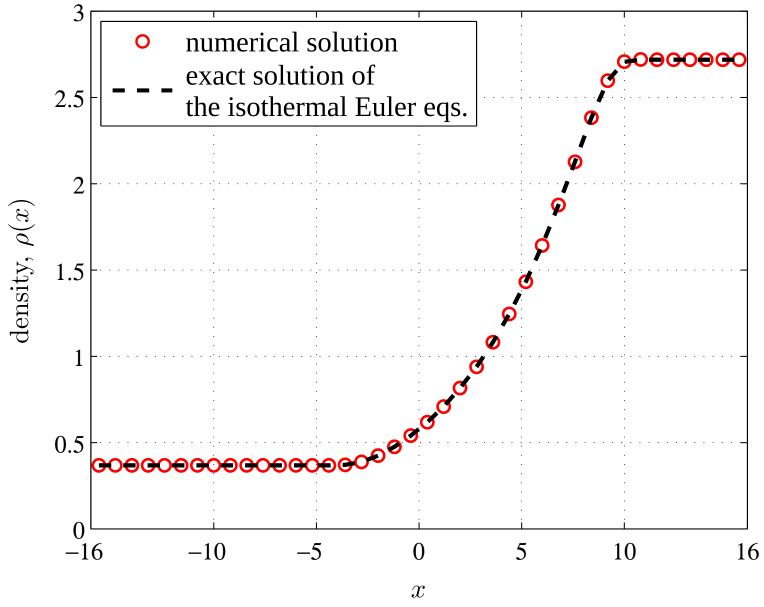


Figure 4.16: The density distribution at $t_{\text{end}} = 5.0$, computed by the DG(1)–Hancock method, is superposed on the exact solution of the isothermal Euler equations.

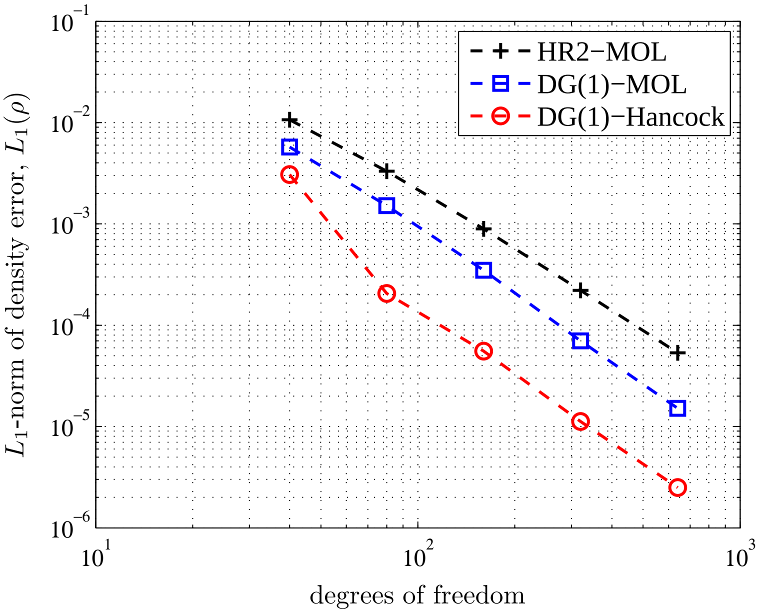


Figure 4.17: $L_1$-norms of density error, $L_1(\rho)$, for three methods are compared, showing the high accuracy of the DG(1)–Hancock method.

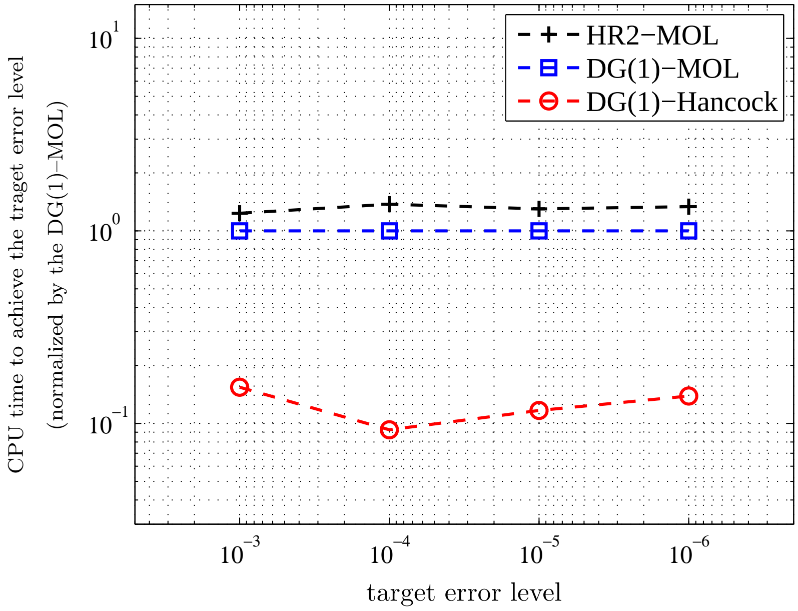


Figure 4.18: Three methods are compared regarding their overall efficiency. The CPU time required to achieve the target error level is normalized by the DG(1)–MOL method. The high efficiency of the DG(1)–Hancock method is observed.

CHAPTER V

# APPLICATION TO EXTENDED HYDRODYNAMICS (10-MOMENT MODEL)

## 5.1 Introduction

In this chapter, the DG(1) and HR2 spatial discretization methods are applied to nonlinear hyperbolic-relaxation equations which describe the motion of fluid, namely, the 10-moment equations. The 10-moment equations are the best known and most studied among models that use multiple moments of the Boltzmann equation [Bro96, Hit00]. Recall the hierarchical relation of the moment approach among other mathematical models (see Figure 1.2 on page 11).

The 10-moment equations can be derived in several ways. Note that different distribution functions could lead to identical macroscopic transport equations. Gombosi describes the 10-moment equations by simplifying a larger set of moment equations: the 20-moment equations derived by Grad's method of moments [Gom94, pp. 223–224]. He also shows that both 20-moment equations and the model obtained by the Chapman–Enskog expansion are identical up to the third order in terms of the relaxation time. Another approach is that of Holway, who replaces the Maxwellian by an ellipsoidal distribution function [Hol65]. More recently, Levermore derived the 10-moment model as a member of a hierarchy of moment closures [Lev96]. He refers to the closure leading to the 10-moment equations as the Gaussian closure.

Subsequent to the theoretical development of moment equations, numerical results were presented by Brown et al. [BRG95, Bro96] and Levermore et al. [LM98, LMN98] for resolving one-dimensional shock structures. Later, Groth presented results for planar Couette flow, in which the shear stress has good agreement with an analytical solution up to a Knudsen number of 10. McDonald and Groth extended the numerical experiments to the diatomic 10-moment or 11-moment equations [MG05], which were originally derived by Hittinger [Hit00, Chapter V]. Suzuki et al. compared numerical results for the 10-moment equations to Navier–Stokes and DSMC results [SvL05]; this chapter is an outgrowth of that work.

## 5.2 10-Moment Model

The 10-moment model is based on a Gaussian velocity distribution (Gaussian closure) [Lev96]. The general form of the Gaussian velocity distribution $\mathcal{G} \in \mathbb{R}_+$ is as follows:

$$\mathcal{G}(\boldsymbol{x}, \boldsymbol{v}, t) = \frac{n(\boldsymbol{x}, t)}{(2\pi)^{3/2}(\det \boldsymbol{\Theta})^{1/2}} \exp\left(-\frac{1}{2}\Theta_{ij}^{-1} c_i c_j\right), \tag{5.1}$$

where

$$\Theta_{ij} = \frac{P_{ij}}{\rho}, \quad i, j \in \{1, 2, 3\} \tag{5.2}$$

is the temperature tensor, $n(\boldsymbol{x}, t)$ is the number density, $\mathbf{c}(\boldsymbol{x}, t)$ the random velocity, and $P_{ij}$ the generalized stress tensor. The model is equivalent to the Navier–Stokes equations without heat conduction; this is sufficiently accurate for the flow problem studied in this chapter, which has an almost isothermal solution.

The 10-moment model is derived as follows. Assume the velocity-distribution function used with the Boltzmann equation is Gaussian, $\mathcal{G}$, multiplying the equation with powers of velocity components, and integrate over all particle velocities.

The Gaussian velocity distribution has the mathematical property that third-order velocity moments are zero, leading to zero heat flux, as well as all higher-order moments, which leads to closure of the set of moment equations. Using the BGK approximation for the collision operator [BGK54], and expressing the equations in vector form in a 3-D Cartesian coordinate system, the 10-moment transport equations assume the form

$$\frac{\partial \mathbf{u}(\boldsymbol{x},t)}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{u})}{\partial y} + \frac{\partial \mathbf{h}(\mathbf{u})}{\partial z} = \frac{1}{\tau}\mathbf{s}(\mathbf{u}), \quad \boldsymbol{x} \in \mathbb{R}^3,\ t > 0, \tag{5.3}$$

where $\mathbf{u} \in \mathbb{R}^{10}$ is the vector of conserved quantities, $\mathbf{f}, \mathbf{g}$, and $\mathbf{h} \in \mathbb{R}^{10}$ are the flux vectors, and $\mathbf{s} \in \mathbb{R}^{10}$ is the source vector for the conservation form of the transport equations. Here, $\tau > 0$ in the source term is a characteristic relaxation time related to viscosity and hydrostatic pressure:

$$\tau = \frac{\mu}{p}, \tag{5.4}$$

with

$$p = \frac{P_{ii}}{3}, \tag{5.5}$$

and the shear stress is defined by

$$\tau_{ij} = p\,\delta_{ij} - P_{ij}. \tag{5.6}$$

The flux and source vectors are given by

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho u_z \\ \rho u_x^2 + P_{xx} \\ \rho u_x u_y + P_{xy} \\ \rho u_x u_z + P_{xz} \\ \rho u_y^2 + P_{yy} \\ \rho u_y u_z + P_{yz} \\ \rho u_z^2 + P_{zz} \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u_x \\ \rho u_x^2 + P_{xx} \\ \rho u_x u_y + P_{xy} \\ \rho u_x u_z + P_{xz} \\ \rho u_x^3 + 3u_x P_{xx} \\ \rho u_x^2 u_y + 2u_x P_{xy} + u_y P_{xx} \\ \rho u_x^2 u_z + 2u_x P_{xz} + u_z P_{xx} \\ \rho u_x u_y^2 + u_x P_{yy} + 2u_y P_{xy} \\ \rho u_x u_y u_z + u_x P_{yz} + u_y P_{xz} + u_z P_{xy} \\ \rho u_x u_z^2 + u_x P_{zz} + 2u_z P_{xz} \end{pmatrix}, \tag{5.7a}$$

$$\mathbf{g} = \begin{pmatrix} \rho u_y \\ \rho u_x u_y + P_{xy} \\ \rho u_y^2 + P_{yy} \\ \rho u_y u_z + P_{yz} \\ \rho u_x^2 u_y + 2u_x P_{xy} + u_y P_{xx} \\ \rho u_x u_y^2 + u_x P_{yy} + 2u_y P_{xy} \\ \rho u_x u_y u_z + u_x P_{yz} + u_y P_{xz} + u_z P_{xy} \\ \rho u_y^3 + 3u_y P_{yy} \\ \rho u_y^2 u_z + 2u_y P_{yz} + u_z P_{yy} \\ \rho u_y u_z^2 + u_y P_{zz} + 2u_z P_{yz} \end{pmatrix}, \tag{5.7b}$$

$$\mathbf{h} = \begin{pmatrix} \rho u_z \\ \rho u_x u_z + P_{xz} \\ \rho u_y u_z + P_{yz} \\ \rho u_z^2 + P_{zz} \\ \rho u_x^2 u_z + 2u_x P_{xz} + u_z P_{xx} \\ \rho u_x u_y u_z + u_x P_{yz} + u_y P_{xz} + u_z P_{xy} \\ \rho u_x u_z^2 + u_x P_{zz} + 2u_z P_{xz} \\ \rho u_y^2 u_z + 2u_y P_{yz} + u_z P_{yy} \\ \rho u_y u_z^2 + u_y P_{zz} + 2u_z P_{yz} \\ \rho u_z^3 + 3u_z P_{zz} \end{pmatrix}, \tag{5.7c}$$

$$\mathbf{s} = - \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ (2P_{xx} - P_{yy} - P_{zz})/3 \\ P_{xy} \\ P_{xz} \\ (2P_{yy} - P_{xx} - P_{zz})/3 \\ P_{yz} \\ (2P_{zz} - P_{xx} - P_{yy})/3 \end{pmatrix}. \tag{5.7d}$$

Alternatively, in tensor notation,

$$\frac{\partial}{\partial t}(\rho) + \frac{\partial}{\partial x_k}(\rho u_k) = 0, \tag{5.8a}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_k}(\rho u_i u_k + P_{ik}) = 0, \tag{5.8b}$$

$$\frac{\partial}{\partial t}(\rho u_i u_j + P_{ij}) + \frac{\partial}{\partial x_k}(\rho u_i u_j u_k + u_i P_{jk} + u_j P_{ik} + u_k P_{ij}) = -\frac{1}{\tau}\left(P_{ij} - \frac{1}{3}P_{kk}\delta_{ij}\right). \tag{5.8c}$$

## 5.3 Numerical Methods and Allowable Time Step

Numerical methods for nonlinear hyperbolic-relaxation equations including fully discrete and semi-discrete methods are described in Chapter II. Among the finite-volume discretization methods, second-order accuracy in space is achieved by introducing linear subcell distributions; for temporal accuracy, the HR2–Hancock method evaluates fluxes and source terms halfway during the time step. The half-time (predictor) step, which includes gradient-limiting, is done with primitive variables $\mathbf{w} \in \mathbb{R}^{10}$ such that

$$\mathbf{w} = (\rho\ u_x\ u_y\ u_z\ P_{xx}\ P_{xy}\ P_{xz}\ P_{yy}\ P_{yz}\ P_{zz})^T\,, \tag{5.9}$$

instead of conserved variables $\mathbf{u}$ to prevent non-physical values such as negative pressures. Here, the Jacobian matrix $\mathbf{M} \in \mathbb{R}^{10\times 10}$ is defined for transformation of variables:

$$\mathbf{M} := \frac{\partial \mathbf{u}}{\partial \mathbf{w}}. \tag{5.10}$$

In the case of the 10-moment equations,

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u_x & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u_y & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u_z & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 \\ u_x^2 & 2\rho u_x & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ u_x u_y & \rho u_y & \rho u_x & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ u_x u_z & \rho u_x & 0 & \rho u_x & 0 & 0 & 1 & 0 & 0 & 0 \\ u_y^2 & 0 & 2\rho u_y & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ u_y u_z & 0 & \rho u_z & \rho u_y & 0 & 0 & 0 & 0 & 1 & 0 \\ u_z^2 & 0 & 0 & 2\rho u_z & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5.11}$$

Finding the allowable time step for a highly nonlinear system of equations on general computational meshes is not straightforward. When systems of one-dimensional conservation laws are considered, the time step is restricted by a CFL stability condition:

$$\lambda \Delta t \leq \Delta x \quad \longrightarrow \quad \tilde{\nu} \leq 1. \tag{5.12}$$

In the case of the moment equations, the presence of two distinct characteristic time scales, the advection time scale and the relaxation time scale, makes the stability analysis even more difficult. In practice, an analogy to the result from a simple 1-D problem may be applied to the multidimensional problem for an explicit method [Lin98, pp. 89–92]; the stability limit for explicit time integration in cell $j$ is approximately given by

$$\Delta t_j \leq \frac{|A_j|}{\dfrac{1}{2} \sum\limits_{e_i \in \partial K_j} |\lambda_{je_i}|^{\max} |e_{i,K_j}| + \dfrac{|A_j|}{\tau_j}}, \tag{5.13}$$

where $|\lambda_{je_i}|^{\max}$ is the largest wave speed on either side of the element face $(j, e_i)$, $\tau_j$ is the relaxation time in cell $j$, and $|e_{i,K_j}|$ is the length of the edge $e_i$ shared by elements $K_j$ and $K_e$. See Figure 2.9 on page 82 for the schematic of two adjacent elements. It shows that the local time step $\Delta t_j$ is determined by the combination of two characteristic times, $\dfrac{\Delta n}{\lambda}$ and $\tau$, where $\Delta n = \dfrac{|A_j|}{|e_{i,K_j}|}$ is the width of element $j$ normal to $e_{i,K_j}$. This criterion is especially restrictive when the flow field is in the near-equilibrium, where the relaxation time is much smaller than the advection time: $\dfrac{\Delta n}{\lambda} \gg \tau$. Our main interest is in wave propagation; however, for an explicit method, the time step has to be of the order of the relaxation time to resolve the correct physics. This stability issue in the stiff regime can be solved by utilizing implicit time integration methods described in Chapter II. For our numerical experiments with the 10-moment equations, we have stayed with fully explicit time integration.

## 5.4 HLLL Riemann Solver for the 10-Moment Model

Among numerical methods for hyperbolic system, those of the Godunov-type have been most successful; these require an algorithm for solving the Riemann problem arising at each cell interface, either exactly or approximately. For a large system of equations it is practical to use an approximate Riemann solver that does not attempt to account for all separate waves through which the cells interact, but lumps the information. See Figure 5.1 for its approximation of waves. Harten, Lax, and Van Leer [HLvL83] described two families of such methods; the latest member is due to Linde [Lin02]. The HLLL Riemann solver uses three waves to cover the domain of influence of the cell interface; it requires only the following knowledge:

- The PDE system is hyperbolic and possesses a convex entropy function;
- maximum and minimum wave speeds are known.

This solver is designed to capture an isolated discontinuity exactly similar to the Roe flux, and do a reasonable job if more waves are present. This simple design criterion allows us to approximate the solution of a Riemann problem by only three waves bracketing two intermediate states. For the 1-D Euler equations, all approximate Riemann solvers based on characteristic decomposition use three waves anyway, but more complicated physical systems such as magnetohydrodynamics, radiation hydrodynamics, and extended-hydrodynamics posses more than three waves, and characteristic-based solvers would need to distinguish all waves in order to provide a detailed approximation. In the three-wave HLL Riemann solver the middle wave speed, representing an isolated discontinuity, is obtained by solving generalized Rankine–Hugoniot conditions instead of using known analytical formulas for the wave speeds. Thus, the algorithm does not require a full analysis of the characteristic wave decomposition for the system of PDE's. As Linde mentions, the family of HLL Riemann solvers can be applied to complex physical systems for which the characteristic wave-decomposition analysis is extremely difficult [Lin02]. In this respect, systems of extended-hydrodynamics equations are excellent candidates. In fact, the eigenstructure of the 10-moment model was already analyzed by Brown et al. [Bro96, BRG95], and analytical results are known. However, its simplicity and the planned application of the algorithm to even higher-order moment models such as the 35-moment model equations [GRGB95, Bro96] made us select the HLLL Riemann solver to compute the cell-interface fluxes.

The middle wave speed $V$ is obtained in the least-square sense,

$$\begin{aligned} V &:= \frac{(\Delta\mathbf{u}, \Delta\mathbf{f})_{\mathrm{P}}}{\|\Delta\mathbf{u}\|_{\mathrm{P}}^2} \\ &= \frac{\Delta\mathbf{u}^T\mathbf{P}\Delta\mathbf{f}}{\Delta\mathbf{u}^T\mathbf{P}\Delta\mathbf{u}} = \frac{\Delta\mathbf{W}^T\Delta\mathbf{f}}{\Delta\mathbf{W}^T\Delta\mathbf{u}}, \end{aligned} \tag{5.14}$$

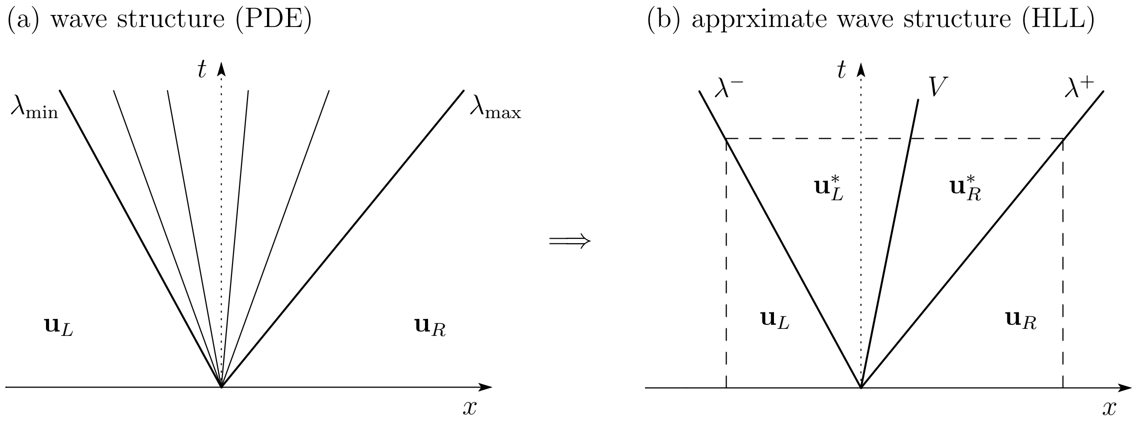


Figure 5.1: The original wave structure (a) is simplified to upper and lower bounding waves plus a middle wave with speed $V$. Conservation is enforced in the space-time domain indicated by the dashed-line box.

where

$$\mathbf{W}(\mathbf{u}) := \frac{\partial S(\mathbf{u})}{\partial \mathbf{u}} \tag{5.15}$$

is a vector of symmetrizing variables (not primitive variables here), formed by taking derivatives of the entropy function $S(\mathbf{u})$. The symmetric positive-definite matrix $\mathbf{P}$ is the Hessian of $S(\mathbf{u})$, hence

$$\mathbf{P}(\mathbf{u}) := \frac{\partial^2 S(\mathbf{u})}{\partial \mathbf{u}^2} = \frac{\partial \mathbf{W}(\mathbf{u})}{\partial \mathbf{u}}. \tag{5.16}$$

The entropy function of the 10-moment model is given as

$$S(\mathbf{u}) = -\rho \left( \frac{1}{3} \ln \frac{\det \mathbf{\Theta}}{\rho^2} \right). \tag{5.17}$$

Straightforward differentiation of the entropy function produces the symmetrizing variables **W** and (for later use) the diagonal entries of the matrix **P**,

$$\mathbf{W}(\mathbf{u}) = \frac{2}{3} \begin{pmatrix} \frac{1}{2}\left(5 - \ln \frac{\det \boldsymbol{\Theta}}{\rho^2} - \mathbf{u}^T \boldsymbol{\Phi}\ \mathbf{u}\right) \\ \boldsymbol{\phi}_1 \mathbf{u} \\ \boldsymbol{\phi}_2 \mathbf{u} \\ \boldsymbol{\phi}_3 \mathbf{u} \\ -\frac{1}{2}\Phi_{11} \\ -\Phi_{12} \\ -\Phi_{13} \\ -\frac{1}{2}\Phi_{22} \\ -\Phi_{23} \\ -\frac{1}{2}\Phi_{33} \end{pmatrix}, \tag{5.18a}$$

$$\operatorname{diag}[\mathbf{P}(\mathbf{u})] = \frac{2}{3\rho} \begin{pmatrix} \frac{1}{2}\left[(\mathbf{u}^T \boldsymbol{\Phi}\ \mathbf{u})^2 + 5\right] \\ (\boldsymbol{\phi}_1 \mathbf{u})^2 + (1 + \mathbf{u}^T \boldsymbol{\Phi}\ \mathbf{u})\Phi_{11} \\ (\boldsymbol{\phi}_2 \mathbf{u})^2 + (1 + \mathbf{u}^T \boldsymbol{\Phi}\ \mathbf{u})\Phi_{22} \\ (\boldsymbol{\phi}_3 \mathbf{u})^2 + (1 + \mathbf{u}^T \boldsymbol{\Phi}\ \mathbf{u})\Phi_{33} \\ \frac{1}{2}\Phi_{11}^2 \\ \Phi_{12}^2 + \Phi_{11}\Phi_{22} \\ \Phi_{13}^2 + \Phi_{11}\Phi_{33} \\ \frac{1}{2}\Phi_{22}^2 \\ \Phi_{23}^2 + \Phi_{22}\Phi_{33} \\ \frac{1}{2}\Phi_{33}^2 \end{pmatrix}, \tag{5.18b}$$

where

$$\Phi_{ij} := \Theta_{ij}^{-1}, \quad i, j \in \{1,2,3\}, \quad \text{or} \quad \mathbf{\Phi} := \mathbf{\Theta}^{-1}, \tag{5.19a}$$

$$\phi_1 := \left(\Phi_{11}, \Phi_{21}, \Phi_{31}\right), \, \phi_2 := \left(\Phi_{12}, \Phi_{22}, \Phi_{32}\right), \, \phi_3 := \left(\Phi_{13}, \Phi_{23}, \Phi_{33}\right), \tag{5.19b}$$

and $\mathbf{u} = (u_x, u_y, u_z)^T$ is the velocity vector. For a Cartesian coordinate system, the inverse of the temperature tensor becomes

$$\mathbf{\Phi} = \frac{1}{\rho^2 \det \mathbf{\Theta}} \begin{pmatrix} P_{yy}P_{zz} - P_{yz}^2 & P_{xz}P_{yz} - P_{xy}P_{zz} & P_{xy}P_{yz} - P_{xz}P_{yy} \\ P_{xz}P_{yz} - P_{xy}P_{zz} & P_{xx}P_{zz} - P_{xz}^2 & P_{xy}P_{xz} - P_{yz}P_{xx} \\ P_{xy}P_{yz} - P_{xz}P_{yy} & P_{xy}P_{xz} - P_{yz}P_{xx} & P_{xx}P_{yy} - P_{xy}^2 \end{pmatrix}, \tag{5.20}$$

where

$$\det \mathbf{\Theta} = \frac{1}{\rho^3} \left( P_{xx}P_{yy}P_{zz} + 2P_{xy}P_{yz}P_{xz} - P_{xx}P_{yz}^2 - P_{yy}P_{xz}^2 - P_{zz}P_{xy}^2 \right). \tag{5.21}$$

Once the symmetrizing variables $\mathbf{W}$ are obtained, the middle wave speed can be computed by (5.14); note that the matrix $\mathbf{P}$ is not explicitly needed here. Then, cell-interface fluxes are obtained by

$$\mathbf{f}_{\mathrm{n}}(\mathbf{u}_i, \mathbf{u}_j) = \frac{\lambda_+ \mathbf{f}_{\mathrm{n}}(\mathbf{u}_i) - \lambda_- \mathbf{f}_{\mathrm{n}}(\mathbf{u}_j)}{\lambda_+ - \lambda_-} + \frac{(1-\alpha)\lambda_-\lambda_+ + \alpha(\lambda_- V_+ + \lambda_+ V_-)}{\lambda_+ - \lambda_-} \Delta\mathbf{u}, \tag{5.22}$$

where

$$\lambda_{\mathrm{max,min}} = \mathbf{n}^T \mathbf{u} \pm \sqrt{3 \mathbf{n}^T \mathbf{\Theta} \; \mathbf{n}}, \tag{5.23a}$$

$$\lambda_+ = \max\bigl(0, V, \lambda_{\mathrm{max}}(\mathbf{u}_i), \lambda_{\mathrm{max}}(\mathbf{u}_j)\bigr), \tag{5.23b}$$

$$\lambda_- = \min\bigl(0, V, \lambda_{\mathrm{min}}(\mathbf{u}_i), \lambda_{\mathrm{min}}(\mathbf{u}_j)\bigr), \tag{5.23c}$$

$$V_+ = \max(0, V), \quad V_- = \min(0, V). \tag{5.23d}$$

Recall $\mathbf{n}$ is an outward unit vector normal to the cell face. The parameter $\alpha \in [\,0, 1\,]$ is an estimation of the relative strength of the middle wave. The computation of $\alpha$

requires knowledge of the matrix $\mathbf{P}$,

$$\alpha := H\big(V\Delta S - \Delta(\mathbf{n}^T\mathbf{u}S)\big)\frac{(\Delta\mathbf{u},\Delta\mathbf{f})_{\mathrm{P}}^2}{\|\Delta\mathbf{u}\|_{\mathrm{P}}^2\|\Delta\mathbf{f}\|_{\mathrm{P}}^2}, \tag{5.24}$$

where $H(x)$ is the Heaviside step function used as a switch to prevent violation of the entropy condition,

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \text{ (discontinuity violates the entropy condition)}, \\ 1 & \text{if } x \geq 0 \text{ (entropy inequality satisfied)}. \end{cases} \tag{5.25}$$

In practice we may reduce $\mathbf{P}$ to its main diagonal, hence the need for (5.18b).

## 5.5 Numerical Results

### 5.5.1 Resolving 1-D Shock Structures

We present some 1-D results from validation studies in which we tried to produce steady shock profiles for various inflow Mach numbers. Assuming a steady state leads to a system of ordinary differential equations (ODE), which can be solved by a standard fourth-order Runge–Kutta method [Bro96, pp. 256–263]. Alternatively, Levermore and Morokoff derive the exact solution of the shock structure in the form of implicit function [LM98]. Solving the implicit formula at quadrature points by a root-finder such as the secant method can provide close-to-exact cell-averaged quantities; this is the technique we have adopted. The resulting analytical solutions are compared with solutions of the PDE's obtained by the finite-volume and discontinuous Galerkin methods described in Chapter II. Note that explicit time-integration methods can be adopted without penalty for all calculations since resolving shock structure requires $\dfrac{\Delta h}{\lambda} \sim \tau$ where $\lambda$ is the maximum wave speed.

Upstream and downstream boundary conditions are assumed to be in equilibrium; given the upstream Mach number, density, and velocity, downstream condi-

|  | $Ma_U$ | $l_U$ [m] | $\rho_U$ | $\rho_D$ | number of cells | computational domain |
|---|---|---|---|---|---|---|
| test1 | 1.1 | $8.549{\times}10^{-9}$ | 1.0 | 1.150 | 200 | $x/\lambda_U \in [-100, 100]$ |
| test2 | 5.0 | $4.362{\times}10^{-9}$ | 1.0 | 3.571 | 200 | $x/\lambda_U \in [-20, 20]$ |

Table 5.1: Initial conditions and problem setup for each test.

tions are determined from the jump-equations. Two different upstream Mach numbers ($Ma_U = 1.1,\ 5.0$) representing weak- and strong-shock cases, are examined. The low value yields a smooth shock structure; for the high value an inviscid jump appears in the structure, inherited from the frozen physics. To avoid constraints by upstream and downstream values, a sufficiently wide computational domain is taken. The computational domain normalized by the upstream mean free path is shown in Table 5.1. We assume the monatomic gas is Argon ($\mathrm{MW}_{\mathrm{Ar}} = 39.948$ kg/kmol) and a power law is used for the viscosity:

$$\frac{\mu}{\mu_{\mathrm{ref}}} = \left(\frac{T}{T_{\mathrm{ref}}}\right)^n, \tag{5.26}$$

where $\mu_{\mathrm{ref}} = 2.125 \times 10^{-5}\ \mathrm{Ns/m}^2,\ T_{\mathrm{ref}} = 273$ K, and $n = \dfrac{13}{18}$ for Argon [Whi91, p. 29][Bro96]. Density distributions are normalized by the upstream and downstream density using the following formula,

$$\hat{\rho} = \frac{\rho - \rho_U}{\rho_D - \rho_U}, \tag{5.27}$$

and shown in Figures 5.2 and 5.3 superimposed on exact solutions. The spatial dimension is normalized by the upstream mean free path derived in gas-kinetic theory using an elastic, hard-sphere model [Bir94],

$$\lambda_U = \frac{16\mu}{5(2\pi\rho p)^{1/2}}. \tag{5.28}$$

The steady numerical solutions (symbols), obtained by running the time-dependent code till convergence, agree well with the exact steady-state solutions (solid line).

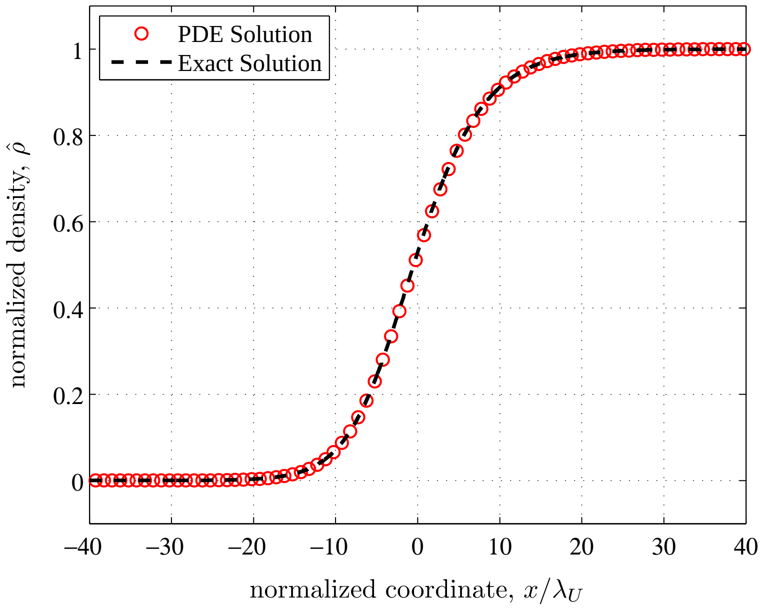


Figure 5.2: Density distribution in steady shock structure for $M_U = 1.1$. The space coordinate is normalized by the upstream mean free path $\lambda_U$.

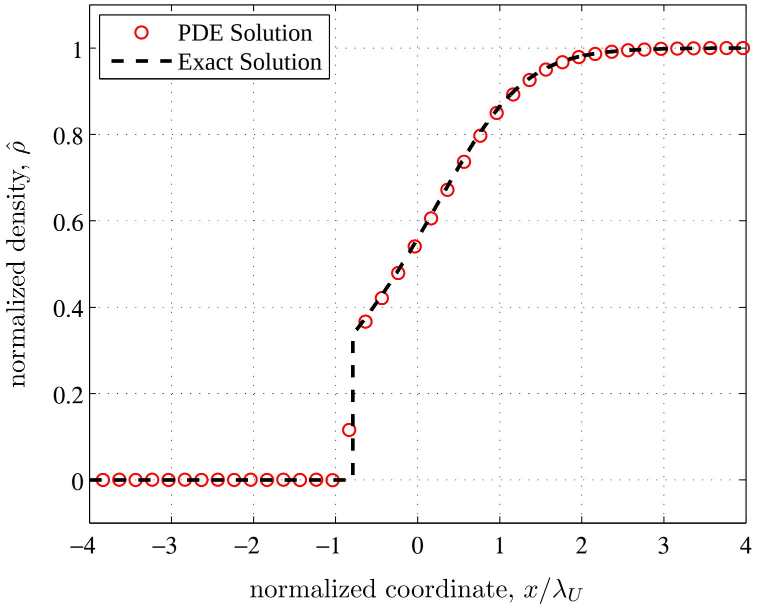


Figure 5.3: Density distribution in steady shock structure for $M_U = 5.0$. A “frozen” shock is followed by a relaxation zone.

In Table 5.2, density errors computed for a sequence of grids when $M_U = 1.1$ are tabulated; their convergence rates demonstrate the second-order spatial accuracy of the HR2–RK2, HR2–Hancock, and DG(1)–RK2 methods, and first-order accuracy for the HR1 method. The Runge–Kutta time-integration methods do not affect the accuracy of the converged steady solutions; the Hancock scheme does yield dependence on the CFL numbers used (not tested). Nevertheless, the error-norms of the HR2–RK2 and HR2–Hancock methods for the same grid are almost identical. The error norms against degrees of freedom and CPU time are also plotted in Figure 5.4. It shows the higher accuracy of the DG(1) spatial discretization over HR2 methods. Owing to the high accuracy of the DG(1) method, it is also the most efficient among all methods.

(a) The HR1–RK1 method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $L_1(\hat{\rho}_{\text{error}})$ | Rate | $L_\infty(\hat{\rho}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 40 | 40 | 1.39e–01 | — | 3.62e–01 | — | 2.55e+00 |
| 80 | 80 | 8.63e–02 | 0.92 | 2.51e–01 | 0.53 | 8.49e+00 |
| 160 | 160 | 5.01e–02 | 0.99 | 1.67e–01 | 0.59 | 3.04e+01 |
| 320 | 320 | 2.74e–02 | 1.01 | 9.94e–02 | 0.75 | 1.15e+02 |

(b) The HR2–RK2 method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $L_1(\hat{\rho}_{\text{error}})$ | Rate | $L_\infty(\hat{\rho}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 40 | 40 | 1.17e–02 | — | 3.92e–02 | — | 5.08e+00 |
| 80 | 80 | 2.89e–03 | 2.27 | 1.19e–02 | 1.72 | 1.66e+01 |
| 160 | 160 | 5.72e–04 | 2.48 | 2.60e–03 | 2.20 | 6.01e+01 |
| 320 | 320 | 1.13e–04 | 2.42 | 5.03e–04 | 2.37 | 2.27e+02 |

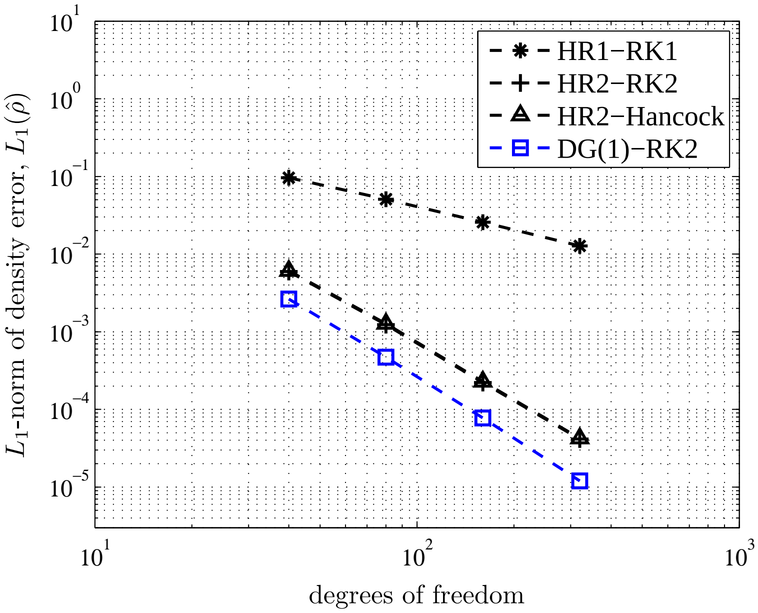
(c) The HR2–Hancock method ($\tilde{\nu} = 0.9$)

| $N_x$ | DOF | $L_1(\hat{\rho}_{\text{error}})$ | Rate | $L_\infty(\hat{\rho}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 40 | 40 | 1.19e–02 | — | 3.99e–02 | — | 2.78e+00 |
| 80 | 80 | 2.97e–03 | 2.26 | 1.23e–02 | 1.70 | 7.93e+00 |
| 160 | 160 | 5.82e–04 | 2.48 | 2.64e–03 | 2.21 | 2.65e+01 |
| 320 | 320 | 1.15e–04 | 2.43 | 5.06e–04 | 2.38 | 9.50e+01 |

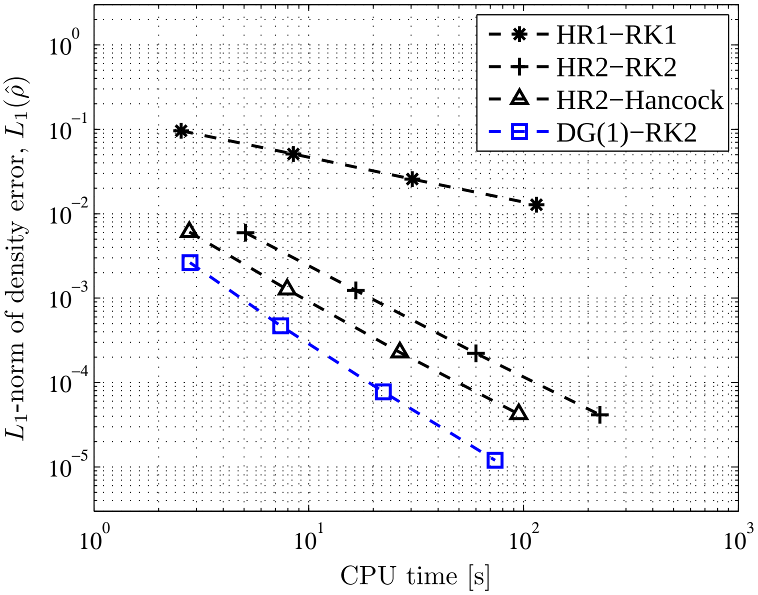
(d) The DG(1)–RK2 method ($\tilde{\nu} = 0.3$)

| $N_x$ | DOF | $L_1(\hat{\rho}_{\text{error}})$ | Rate | $L_\infty(\hat{\rho}_{\text{error}})$ | Rate | CPU time [s] |
|---|---|---|---|---|---|---|
| 20 | 40 | 6.12e–03 | — | 1.92e–02 | — | 2.80e+00 |
| 40 | 80 | 1.09e–03 | 2.48 | 3.53e–03 | 2.44 | 7.40e+00 |
| 80 | 160 | 1.89e–04 | 2.60 | 8.34e–04 | 2.08 | 2.22e+01 |
| 160 | 320 | 2.94e–05 | 2.70 | 1.59e–04 | 2.39 | 7.37e+01 |

Table 5.2: A grid convergence study is performed by solving the 10-moment equations. $L_1$-, $L_\infty$-norms and rates of convergence for steady shock solutions ($M_U = 1.1$) are computed.



(a) $L_1$-norms of error plotted against number of degrees of freedom.



(b) $L_1$-norms of error plotted against CPU time.

Figure 5.4: The $L_1$-norms of errors tabulated in Table 5.2 are plotted in terms of both degrees of freedom and CPU time. The grid convergence study shows that steady shock solutions ($M_U = 1.1$) are second order accurate for HR2 and DG(1) methods.

### 5.5.2 Cosine-Nozzle Flow

Internal nozzle flow is examined as the first 2-D test case. Here, only the result obtained by the HR2–Hancock method is presented. Since there is no stagnation point inside the nozzle, this flow problem is easier than an airfoil problem and serves as a precursor test case. A symmetric cosine-shaped nozzle (Figure 5.5) is used as the computational domain. The throat is located at the origin of the $x$-axis and the total length over which area variation occurs is 0.1 m. There are 0.02 m and 0.08 m long constant-area regions at inlet and outlet. Table 5.3 shows the reservoir conditions. Stagnation temperature $T_0$ and Reynolds number are specified in the reservoir. The Knudsen number is based on the throat width and the reservoir condition. The Reynolds number is defined as [Rot71]

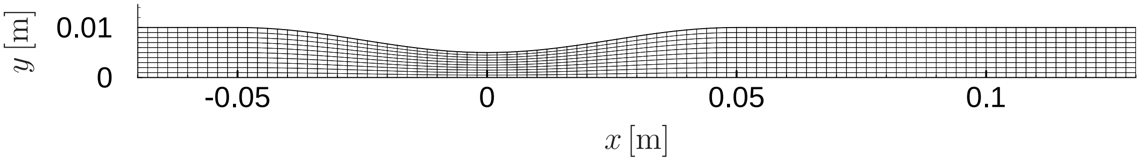


Figure 5.5: Computational grid of cosine-curve nozzle. The number of cells is 100×10.

| $Re$ | $Kn$ | $\rho_0$ [kg/m$^3$] | $p_0$ [Pa] | $T_0$ [K] | $h_0$ [J/kg] | $r_t$ [m] |
|---|---|---|---|---|---|---|
| 100 | 0.014 | $8.229 \times 10^{-4}$ | 51.38 | 300 | $1.561 \times 10^5$ | $5.0 \times 10^{-3}$ |

Table 5.3: Reservoir condition for the nozzle flow

$$Re = \frac{\rho_0 \sqrt{2h_0} r_t}{\mu_0}, \tag{5.29}$$

where $\sqrt{2h_0}$ is an ideal maximum escape speed from the reservoir and $r_t$ is the throat half-width. Equation (5.29) leads to a direct relation between Reynolds number and

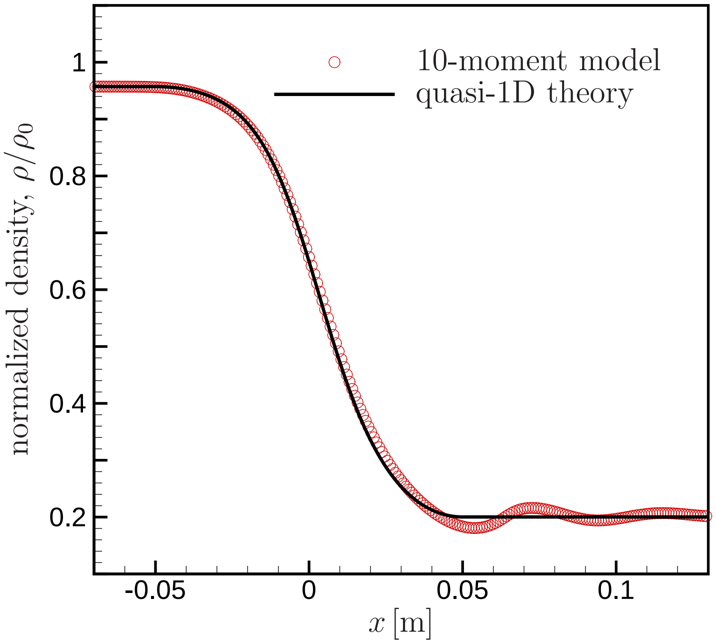


Figure 5.6: Density profile along axis of nozzle.

reservoir pressure,

$$p_0 = \left(\sqrt{\frac{\gamma-1}{2\gamma}RT_0}\,\frac{\mu_0}{r_t}\right) Re. \tag{5.30}$$

In this test case, Argon ($\mu_0 = 2.299 \times 10^{-5}$ N s/m$^2$, $R_{\mathrm{Ar}} = 208.13$ J/kg K) is used and the reservoir pressure satisfies the relation

$$p_0 \simeq 0.5138\, Re. \tag{5.31}$$

Viscosity is computed by Sutherland's law,

$$\frac{\mu}{\mu_{\mathrm{ref}}} = \left(\frac{T}{T_{\mathrm{ref}}}\right)^{3/2} \frac{T_{\mathrm{ref}} + S}{T + S}, \tag{5.32}$$

where $\mu_{\mathrm{ref}} = 2.125 \times 10^{-5}\,\mathrm{N\,s/m^2}$, $T_{\mathrm{ref}} = 273\,\mathrm{K}$, and $S = 144\,\mathrm{K}$ for Argon [Whi91]. Figure 5.6 shows the normalized density profile obtained by the 10-moment model (circles) on the axis direction, compared with quasi-1D theory (solid line). There is a good agreement between the PDE solution and the theoretical density profile.

The decay of the (diamond-pattern) waves in the supersonic section indicates that the equation system is dissipative (unlike the Euler equations).

### 5.5.3 NACA0012 Airfoil Flow

Before we investigate flow over a micro-airfoil, which is characteristic of the higher limit of the continuum-transition regime ($Kn \simeq 10^{-1}$), numerical solutions of the 10-moment and Navier–Stokes equations in the near-equilibrium limit are compared. Since the 10-moment equations formally reduce to the Navier–Stokes equations without heat flux in the near-equilibrium limit, we expect that numerical solutions obtained with the two models become comparable. We chose the external flow around a NACA0012 airfoil which is a widely used validation test for Navier–Stokes solvers [BR97, SWL06]. The free-stream conditions are shown in Table 5.4 where the free-stream Reynolds number, $Re_\infty$, is based on the frees-stream velocity, $U_\infty$, and the chord length, $L_{\text{chord}}$. The free-stream static temperature is chosen such that the stagnation temperature is 290 K for a monatomic gas at the given free-stream Mach number. Argon is used as a gas, and Sutherland's law (5.32) is adopted to calculate viscosity. The order of magnitude of the local Knudsen num-

(a) Prescribed conditions

| $Ma_\infty$ | $Re_\infty$ | $\gamma$ (Argon) | $T_\infty$ [K] | $L_{\text{chord}}$ [m] | $\alpha$ |
|---|---|---|---|---|---|
| 0.5 | 5,000 | 5/3 | 267 | 1.0 | 0° |

(b) Resulting free-stream conditions

| $Kn_\infty$ | $\rho_\infty$ [kg/m$^3$] | $U_\infty$ [m/s] | $p_\infty$ [Pa] |
|---|---|---|---|
| $1.62 \times 10^{-4}$ | $6.858 \times 10^{-4}$ | 152 | 38.21 |

Table 5.4: Free stream condition for airfoil flow.

ber in a boundary layer is approximately

$$Kn \sim \frac{Ma_\infty}{Re_\delta} \sim \frac{Ma_\infty}{\sqrt{Re_\infty}} = 7.1 \times 10^{-3}, \tag{5.33}$$

where $\delta$ is the boundary layer thickness [GeH99]. Hence, the flow in the boundary layer is at the lower end of the continuum-transition regime (cf. Table 1.1 on page 10).

The HR2–Hancock method is applied to solve the 10-moment equations, and HR2–RK is used for the Navier–Stokes equations. Both methods employ a $C$-type grid composed by 120 × 76 cells. A typical finite-volume discretization of the Navier–Stokes equations consists of two parts: the inviscid flux obtained by a Riemann solver, and the viscous flux by central differencing. Here, we omit the description of discretization methods for the viscous-flux terms. Details of the implementation, particularly for a quadrilateral grid, can be found in [Fle91, pp. 105–110]. Since our Navier–Stokes solver is explicit, the stable time step becomes a function of both the advective time-scale, $\dfrac{h}{\lambda}$, and the viscous time-scale, $\dfrac{\rho h^2}{\mu}$, where $h$ is a typical cell size. Based on the 1-D linear stability analysis for an advection-diffusion equation, and following the notation used in (5.13) on page 296, the stable time step for the 2-D Navier–Stokes equations can be written as

$$\Delta t_j \leq \min\left[\frac{|A_j|}{c_1} + \frac{c_2}{c_1^2}, \left(\frac{c_1}{|A_j|} + \frac{c_2}{|A_j|^2}\right)^{-1}\right], \tag{5.34}$$

where

$$c_1 = \frac{1}{2}\sum_{e_i \in \partial K_j} |\lambda_{je_i}|^{\max} |e_{i,K_j}|, \tag{5.35a}$$

$$c_2 = \sum_{e_i \in \partial K_j} \frac{\mu_{je_i}|e_{i,K_j}|^2}{\rho_{je_i}}. \tag{5.35b}$$

See also Figure 2.7 on page 68 for the notation.

For a finite-volume method, in order to achieve second-order accuracy, the reconstruction of flow gradients in the each cell is required. Typically, either computing a Green–Gauss contour-integral, or solving a least-square problem provides the gradients; these procedure require all neighboring cells of the cell of interest. When gradients at cells abutting the solid wall are computed, ghost cells are introduced. Along the solid boundary, the non-slip boundary condition is employed. Let $(u_{cj}, v_{cj})$ be the velocity vector associated with the centroid $\boldsymbol{x}_{cj}$ of the cell $i$, $(u_{cg}, v_{cg})$ the velocity vector at the centroid $\boldsymbol{x}_{cg}$ of the ghost cell, and $\mathbf{Q}$ the rotation matrix defined by

$$\mathbf{Q} := \begin{pmatrix} n_x & s_x & 0 \\ n_y & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & n_y & 0 \\ n_y & -n_x & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{5.36}$$

where $\mathbf{n} = (n_x, n_y, 0)$ the outward unit vector normal to the wall, and $\mathbf{s} = (s_x, s_y, 0)$ the unit vector tangential to the wall (see Figure 5.7); then, the non-slip boundary condition leads to the velocity at the centroid of the ghost cell as follows:

$$(u_{cg}, v_{cg}, 0)\mathbf{Q} = -(u_{cj}, v_{cj}, 0)\mathbf{Q} \quad \longrightarrow \quad (u_{cg}, v_{cg}) = (-u_{cj}, -v_{cj}). \tag{5.37}$$

This condition enforces zero velocity at the wall surface: $(u_s, v_s) = \mathbf{0}$. As to the density and pressure, owing to the assumption, $\dfrac{\partial p}{\partial n} \approx 0$, inside the boundary layer, and the adiabatic wall, $\dfrac{\partial T}{\partial n} = 0$, together with the equation of state, $p = \rho R T$, we have

$$\frac{\partial \rho}{\partial n} \approx 0, \quad \frac{\partial p}{\partial n} \approx 0. \tag{5.38}$$

Hence, the density and pressure in the ghost cell can be prescribed such that

$$\bar{\rho}_{cg} = \bar{\rho}_{cj}, \quad \text{and} \quad p_{cg} = p_{cj}. \tag{5.39}$$

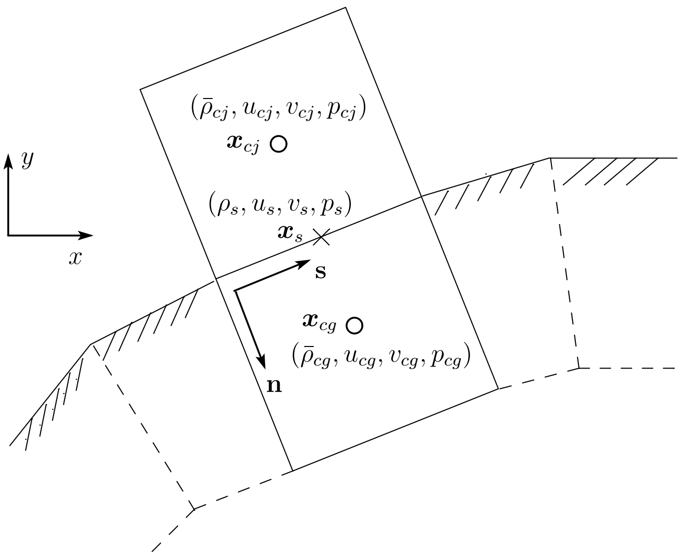

$(\bar{\rho}_{cj}, u_{cj}, v_{cj}, p_{cj})$
$\boldsymbol{x}_{cj}$
$y$
$x$
$(\rho_s, u_s, v_s, p_s)$
$\boldsymbol{x}_s$
$\mathbf{s}$
$\boldsymbol{x}_{cg}$
$(\bar{\rho}_{cg}, u_{cg}, v_{cg}, p_{cg})$
$\mathbf{n}$

Figure 5.7: Ghost cells are introduced to compute gradients of flow quantities.

Note that the density at the centroid of the cell $j$ is identical to the cell-averaged density, whereas pressure is specifically calculated at the centroid based on the conserved quantities. No boundary condition on the temperature is prescribed since the Navier–Stokes equations considered here neglect heat transfer in order to strictly compare with the 10-moment equations. Once an approximated solution-gradient, $\nabla \mathbf{w}_j$, is obtained in cell $j$, the following linear extrapolation,

$$\mathbf{w}_L(\boldsymbol{x}_s) = \mathbf{w}_{cj} + \nabla \mathbf{w}_j \cdot (\boldsymbol{x}_s - \boldsymbol{x}_{cj}), \tag{5.40}$$

provides input values to a Riemann solver, $\mathbf{f}(\mathbf{w}_L, \mathbf{w}_R)$, implemented in the location $\boldsymbol{x}_s$ at the edge. The other input, $\mathbf{w}_R$, can be specified directly by the non-slip boundary condition:

$$\mathbf{w}_R = (\rho_s, u_s, v_s, p_s) = (\bar{\rho}_{cg}, 0, 0, p_{cg}). \tag{5.41}$$

The numerical solutions are presented in terms of the pressure and skin friction coefficients defined by

$$C_p(\tilde{x}) := \frac{p(\tilde{x}) - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2} = \frac{2}{\gamma Ma_\infty^2}\left(\frac{p(\tilde{x})}{p_\infty} - 1\right), \tag{5.42a}$$

$$C_f(\tilde{x}) := \frac{\tau_s(\tilde{x})}{\frac{1}{2}\rho_\infty U_\infty^2}, \tag{5.42b}$$

along the surface of the airfoil; the pressure in the 10-moment model is given by (5.5). Here, the dimensionless $x$-coordinate is defined by $\tilde{x} := \dfrac{x}{L_{\text{chord}}}$, and $\tilde{x} = 0, 1$ correspond to the leading and trailing edges, respectively. The tangential shear stress along the wall, $\tau_s$, can be directly computed with the 10-moment equations by rotating the stress tensor $\mathbf{P}$ in $x$-,$y$-coordinates to $\mathbf{P}'$ in $n$-,$s$-coordinates:

$$\mathbf{P}' = \begin{pmatrix} P_{nn} & P_{ns} & P_{nz} \\ P_{sn} & P_{ss} & P_{sz} \\ P_{zn} & P_{zs} & P_{zz} \end{pmatrix} := \mathbf{Q}^T\mathbf{P}\mathbf{Q}, \tag{5.43}$$

yielding

$$\begin{aligned} \tau_s &= -P_{ns} \\ &= (P_{yy} - P_{xx})n_x n_y + P_{xy}(n_x^2 - n_y^2). \end{aligned} \tag{5.44}$$

In the case of the Navier–Stokes equations, the tangential shear stress is obtained by using the relation (5.6) on page 292, yielding

$$\tau_s = (\tau_{xx} - \tau_{yy})n_x n_y + \tau_{xy}(n_y^2 - n_x^2), \tag{5.45}$$

where the components of the shear stress tensor are obtained by the following Navier–Stokes constitutive laws:

$$\tau_{xx} = \mu\left[2\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right], \tag{5.46a}$$

$$\tau_{yy} = \mu\left[2\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right], \tag{5.46b}$$

$$\tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right). \tag{5.46c}$$

Figure 5.8(a) shows the excellent agreement of pressure coefficients between the 10-moment and Navier–Stokes solutions. The theoretical maximum pressure-coefficient for $(Ma_\infty, \gamma) = (0.5, 5/3)$ is obtained by the isentropic relation:

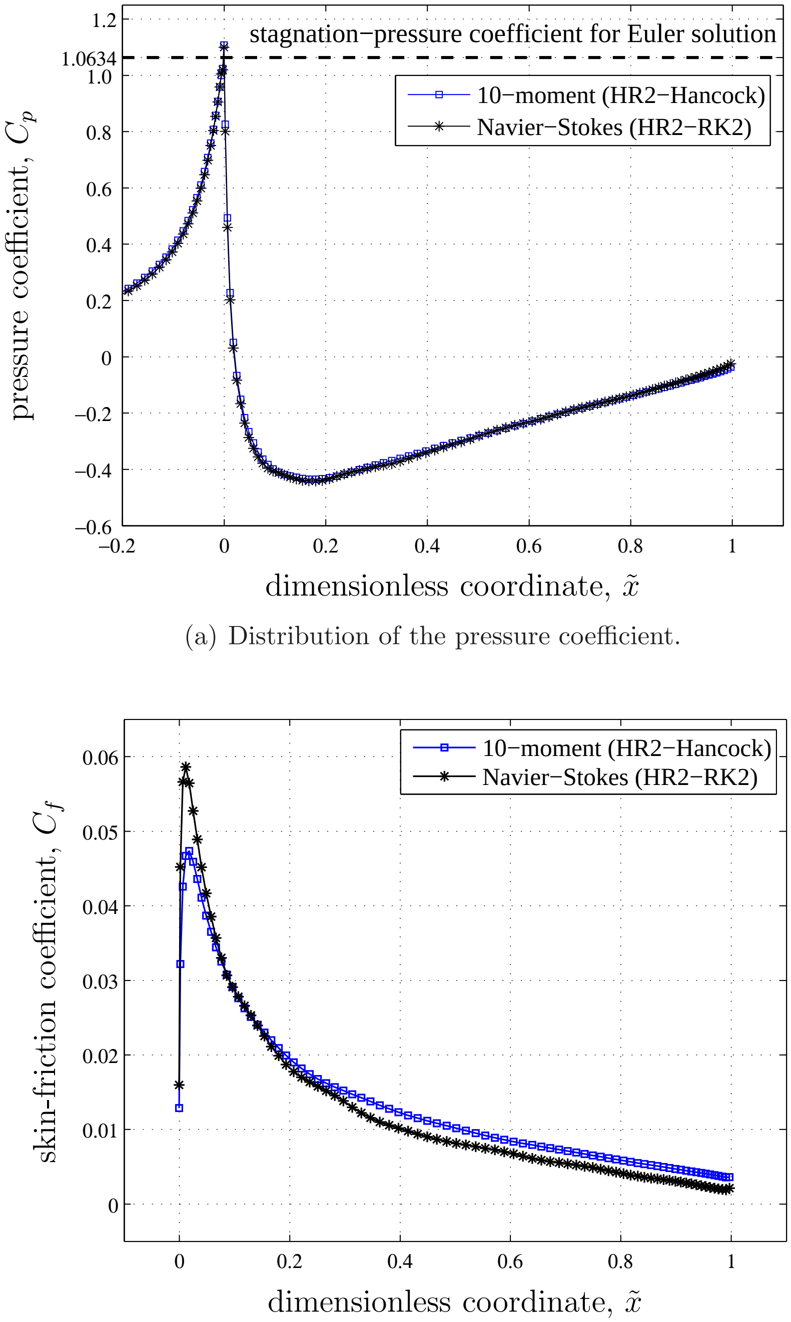
$$\frac{p_0}{p_\infty} \simeq 1.2215 \qquad \longrightarrow \qquad (C_p)_{\max} = 1.0634. \tag{5.47}$$

Both solutions slightly overshoot the theoretical maximum pressure coefficient at the cell adjacent to the leading edge. In Figure 5.8(b), we observe that the 10-moment model predicts a lower peak of the skin-friction coefficient, $(C_f)_{\max}$. This is somewhat surprising since we expect the 10-moment equations to predict the shear stresses more accurate because they are flow variables. This might be traced back to the relatively high Knudsen number in the boundary layer shown in (5.33). Since the flow is not completely in the continuum flow regime ($Kn \leq 10^{-3}$), nonequilibrium effects described by the 10-moment equations in the boundary layer could predict different surface values as computed to the Navier–Stokes equations.

The density profiles along the centerline of the airfoil are shown in Figure 5.9. Both models agree well except near the stagnation point; the Navier–Stokes code predicts slightly higher density values. The isentropic relation provides the theoretical maximum stagnation density,

$$\frac{\rho_0}{\rho_\infty} \simeq 1.1276. \tag{5.48}$$

Unlike the pressure coefficients, the maximum densities predicted by both models stay clearly lower that the isentropic stagnation density. While the pressure approximately remains constant across the boundary layer near the stagnation point, the density is affected by the local dissipation. Insufficient grid resolution near the stagnation could lead to lower predicted density values.



(a) Distribution of the pressure coefficient.

(b) Distribution of the skin-friction coefficient.

Figure 5.8: Dimensionless pressure and shear stress along the NACA0012 airfoil obtained by 10-moment and Navier–Stokes codes ($Ma = 0.5, Re = 5,000, \alpha = 0°$).

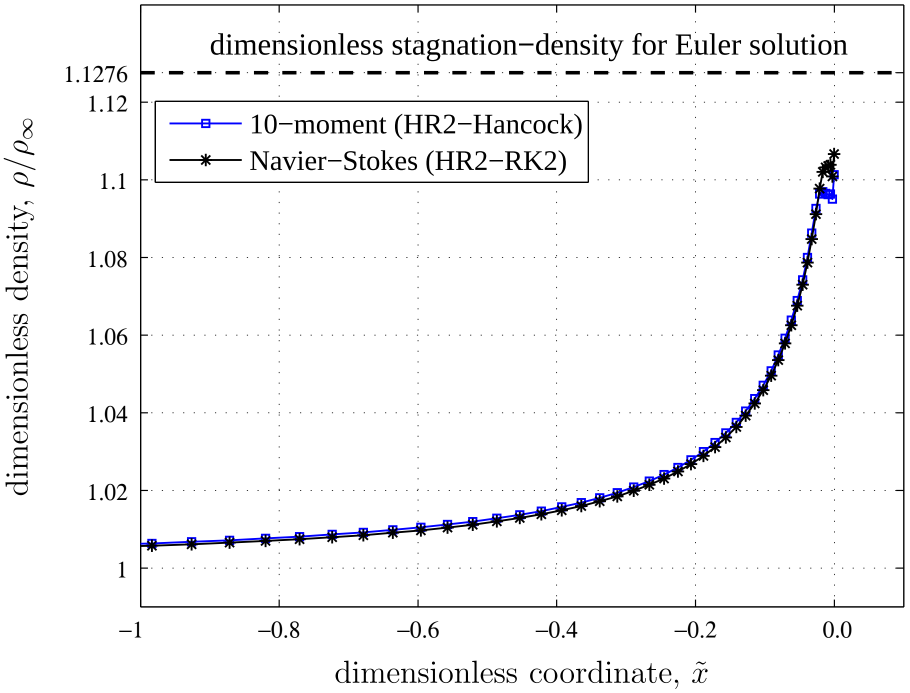


Figure 5.9: Distribution of the dimensionless density along the centerline of interval $\tilde{x} \in [-1, 0]$ is plotted.

### 5.5.4 NACA0012 Micro-Airfoil Flow

Next, the external flow around a NACA0012 micro-airfoil is computed using the 10-moment model. The free-stream initial conditions are shown in Table 5.5. The Knudsen number is based on the chord length of the airfoil and the free-stream conditions. The chord length of the airfoil is 0.04 m and a $C$-type grid is used. The grid geometry is shown in Figure 5.10. Since various results are available using air as

(a) Prescribed conditions

| $Ma_\infty$ | $Re_\infty$ | $\gamma$ (Argon) | $T_\infty$ [K] | $L_{\text{chord}}$ [m] | $\alpha$ |
|---|---|---|---|---|---|
| 0.8 | 73 | 1.4 | 257 | 0.04 | 0° |

(b) Resulting free-stream conditions

| $Kn_\infty$ | $\rho_\infty$ [kg/m$^3$] | $U_\infty$ [m/s] | $p_\infty$ [Pa] |
|---|---|---|---|
| 0.016 | $1.161 \times 10^{-4}$ | 257 | 8.565 |

Table 5.5: Free-stream conditions for micro-airfoil flow.

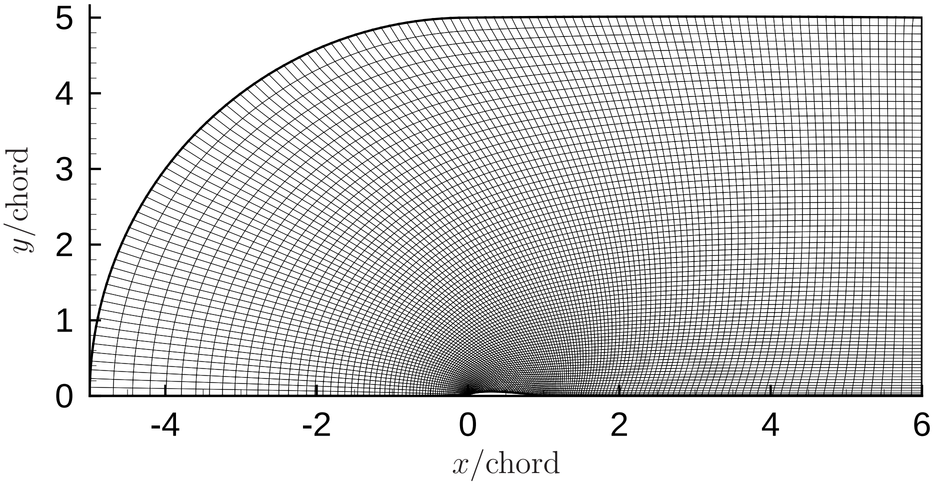


Figure 5.10: Computational grid around NACA0012 micro-airfoil. The coordinate is normalized by the chord length. The number of cells is 120 × 76.

the gas, we also assume the gas is air ($\mathrm{MW}_{\mathrm{Air}} = 28.966$ kg/kmol), even though the 10-moment model assumes a monatomic gas. Viscosity is computed by Sutherland's law (5.32) where $\mu_{\mathrm{ref}} = 1.716 \times 10^{-5}$ Ns/m$^2$, $T_{\mathrm{ref}} = 273$ K, and $S = 111$ K for Air. Hittinger has added rotational degrees of freedom to the system, leading to an 11-moment model for a diatomic gas [Hit00]; we have not used this model. The 10-moment equation implicitly assume $\gamma = 5/3$, rather than $\gamma = 1.4$; for the density results presented below this hardly makes a difference (see below).

In the continuum-transition regime where $Kn \in [10^{-3}, 10^{-1}]$, the flow on the wall has a finite velocity. This slip velocity is approximated by Maxwell's first-order slip boundary condition,

$$u_{\mathrm{gas}} - u_{\mathrm{wall}} = \frac{2-\sigma}{\sigma}\frac{\lambda}{\mu}\tau_s, \tag{5.49}$$

where $\sigma$ is an accommodation coefficient, $\lambda$ is the mean free path, $\mu$ is the viscosity, and $\tau_s$ is the tangential shear stress. The flow toward the wall is assumed to have a Maxwellian velocity distribution. At the airfoil, completely diffuse molecular reflection is assumed in formulating the boundary condition for both methods (achieved by setting $\sigma = 1$). The tangential shear stress is obtained by (5.45) for the 10-moment equations. For instance, the slip velocity on a plane wall parallel to the $x$-direction, $\mathbf{n} = (0, 1)$, is obtained by

$$u_{\mathrm{gas}} - u_{\mathrm{wall}} = \frac{2-\sigma}{\sigma}\frac{\lambda}{\mu}(-P_{xy}). \tag{5.50}$$

Furthermore, inserting the mean free path based on the Chapman–Enskog distribution function [VK86, p. 414],

$$\lambda = \mu\sqrt{\frac{\pi}{2\rho p}}, \tag{5.51}$$

into the above equation leads to the following slip velocity:

$$u_{\mathrm{gas}} - u_{\mathrm{wall}} = \frac{2-\sigma}{\sigma}\sqrt{\frac{\pi}{2}}\frac{(-P_{xy})}{\sqrt{\rho p}}, \tag{5.52}$$

The above slip velocity based on Maxwell's first-order approximation is comparable to the boundary condition recently presented by McDonald and Groth, where a Gaussian distribution function is adopted for incoming particles [MG05]. Their slip velocity for a flat wall is given by

$$u_{\text{gas}} - u_{\text{wall}} = \frac{2-\sigma}{\sigma}\sqrt{\frac{\pi}{2}}\frac{(-P_{xy})}{\sqrt{\rho P_{yy}}}. \tag{5.53}$$

In the case of the Navier–Stokes equations, the tangential shear stress is obtained by inserting constitutive laws (5.46) on page 314 into (5.45); then the the slip velocity based on the velocity gradient at wall becomes

$$u_{\text{gas}} - u_{\text{wall}} = \frac{2-\sigma}{\sigma}\lambda\left(\frac{\partial u_s}{\partial n} + \frac{\partial u_n}{\partial s}\right). \tag{5.54}$$

The above slip boundary condition is the actual form derived from Maxwell's first-order model, but we often encounter a simplified slip-velocity value under the approximation of $\dfrac{\partial u_n}{\partial u_s} \approx 0$, which is valid only for a planar wall. The benefit of this assumption is the simplification of code implementation, especially for a second-order finite-volume method. Implementing the original form (5.54) together with the reconstruction process at cells adjacent to the wall results in an implicit procedure in boundary cells. Thus, for simplicity, even though the wall along the airfoil is curved, we adopt the following simplified slip velocity under the assumption that the curved wall is made of small flat plates:

$$u_{\text{gas}} - u_{\text{wall}} = \frac{2-\sigma}{\sigma}\lambda\frac{\partial u_s}{\partial n}. \tag{5.55}$$

The 10-moment result is shown in Figure 5.11(a) with corresponding results of the Navier–Stokes code (Figure 5.11(b)), the Information Preservation (IP) method [SB02], (Figure 5.11(c)), and a DSMC method [SB02] (Figure 5.11(d)). The numerical results of the 10-moment and Navier–Stokes equations are obtained by the same codes used in the previous section (near-equilibrium flow), but with different free stream and slip boundary conditions.

There are clear differences between the solutions, especially upstream of the airfoil. Near the stagnation point the 10-moment approach gives significantly lower density values than the Navier–Stokes, IP, and DSMC approaches, with the former values comparing favorably to the experimental results [ARL87] reproduced in Figure 5.11(e). Specifically, the experiment shows a normalized stagnation density slightly higher than 1.17 (the highest contour value), which the 10-moment result is 1.19, with the density peaking a little distance upstream of the stagnation point. In contrast, the Navier–Stokes density results peak upstream of the stagnation point at 1.31, which both DSMC and IP densities peak in the stagnation point at 1.40. Of course, the 10-moment result is not fully comparable, since it was computed for a monatomic gas instead of a diatomic gas, but for the given upstream Mach number, the difference in stagnation pressure is only 1.1%, based on the isenthalpic-isentropic relation $\big((\rho_0/\rho_\infty)_{\gamma=5/3} = 1.337,\ (\rho_0/\rho_\infty)_{\gamma=1.4} = 1.351\big)$. Thus we conclude that the 10-moment model indicates a density peak of about 1.20, in good agreement with th experiment.

The results of the Navier–Stokes, IP, and DSMC codes raise two questions. Firstly, why are the IP and DSMC results so far from the measured values? Secondly, why are the Navier–Stokes results closer to the measurements than the IP and DSMC results? One would have expected the 10-moment results to lie in between the Navier–Stokes and particle-code results as regards accuracy. We can make several conjectures that might provide answers, but prefer to limit ourselves to commenting on matters over which we have control, i.e., the Navier–Stokes and 10-moment computations.

Significant is here that the Navier–Stokes results are clearly different from the 10-moment or experimental results. Note that, in the near-equilibrium flow presented in the last section, both 10-moment and Navier–Stokes codes provide almost identical density profiles (see Figure 5.9); hence, the discrepancy between the densities near the leading edge can be understood as evidence that the 10-moment equations for the current flow conditions provide a true nonequilibrium solution, which is outside the domain of validity of the Navier–Stokes equations.

In Figure 5.12, the aerodynamic properties of the airfoil obtained by the 10-moment and Navier–Stokes models are presented in terms of the pressure and skin-friction coefficients defined by (5.42). An Euler result is supplemented in the pressure-coefficient plot. Also indicated is the theoretical maximum pressure coefficient for $(Ma_\infty, \gamma) = (0.8, 1.4)$, obtained by isenthalpic-isentropic theory:

$$\frac{p_0}{p_\infty} \simeq 1.524 \quad \longrightarrow \quad (C_p)_{\max} = 1.1704. \tag{5.56}$$

We found it rather surprising result that the pressure coefficient around the leading edge obtained by the Navier–Stokes solver significantly exceed the value for isentropic compression, since that is regarded as the ideal compression process providing the highest possible stagnation pressure. In order to affirm confidence in our coding, an Euler solution was obtained by omitting the viscous flux in the Navier–Stokes code applying full-slip boundary conditions, and using the same free-stream conditions. The result shows that the Eulerian stagnation pressure agrees well with the theoretical maximum value. Earlier we saw that both 10-moment and Navier–Stokes codes provide almost identical maximum pressure coefficients in the near-equilibrium regime (see Figure 5.8(a)), close to the theoretical maximum. In this regime the Navier–Stokes equations with zero slip produce a boundary layer across which the pressure is practically constant, as in the Euler case. In the continuum-transition regime, however, partial slip makes the pressure do work at the bottom of the boundary layer, significantly changing its structure. We believe that this effect allows the unexpectedly high pressures to appear, but have not further pursued this issue.

With the 10-moment model, stagnation pressures above the isenthalpic-isentropic maximum value are not observed. It should be noted that in these equations, due to the inclusion of a full pressure tensor, there is no single total enthalpy to be defined, which the entropy only relates to the determinant of the pressure tensor; see (5.2), (5.17). As a result, there is no procedure to determine a meaningful maximum stagnation pressure.

We found that stagnation pressures computed with the IP and DSMC codes, published elsewhere, significantly overshoot the isenthalpic-isentropic maximum [FBC$^+$01, SB02].

We conclude that the discrepancy between the solutions of the 10-moment and the Navier–Stokes equations for the flow around a micro-airfoil can be attributed to the significant nonequilibrium effect in the stagnation region.

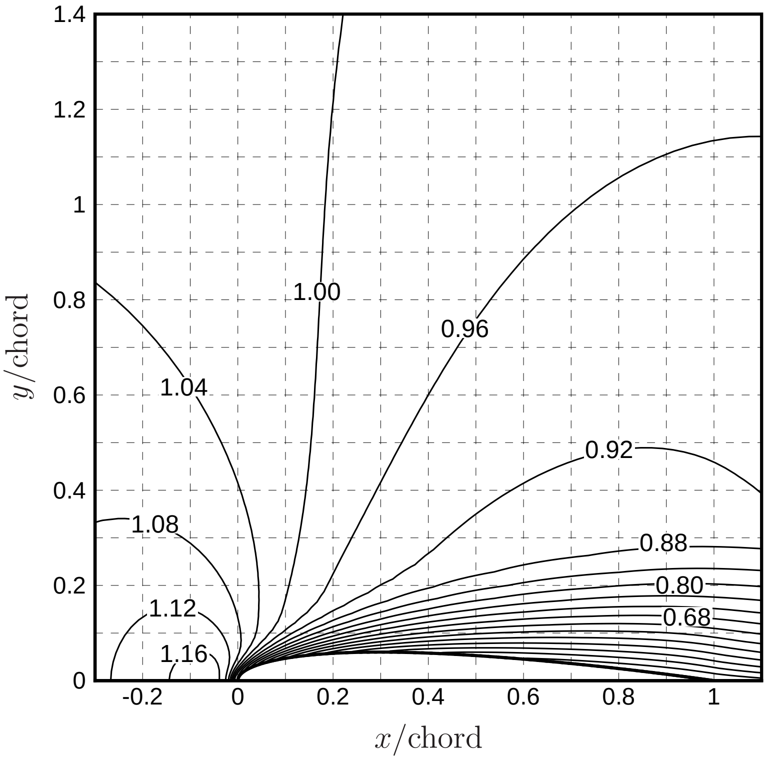
The numerical results for the 10-moment equations obtained by two methods, namely, HR2–Hancock and DG(1)–RK2, agree well except for the skin friction coefficient. This might be due to the convergence stall of the DG(1)–RK2 method; it occurs after the residual drops three order of magnitude. Clearly, convergence-acceleration methods need to be studies in combination with spatial DG discretizations.

The density profiles along the centerline of the airfoil are shown in Figure 5.13.

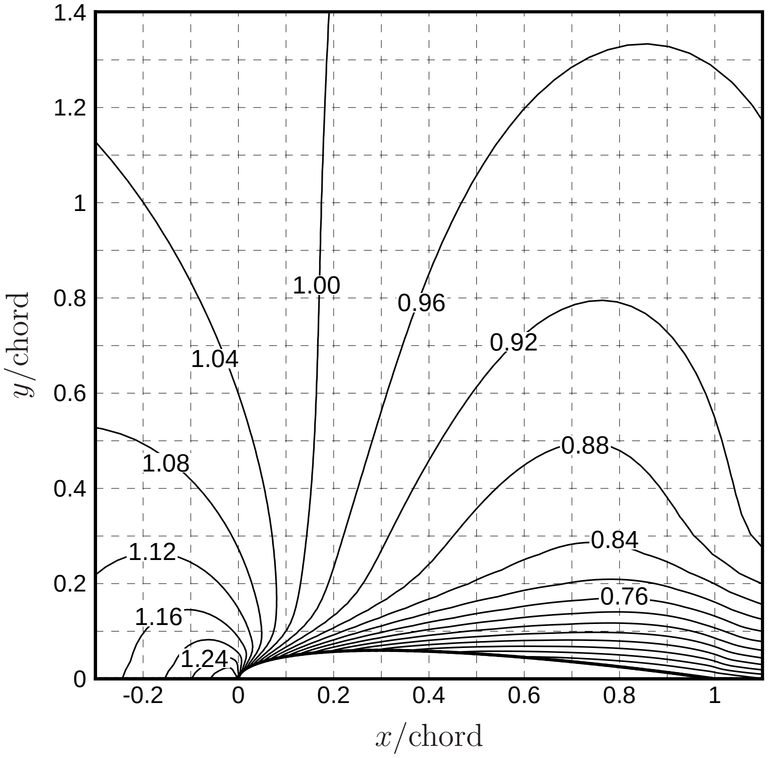
The isentropic relation provides the theoretical stagnation density,

$$\frac{\rho_0}{\rho_\infty} \simeq 1.351. \tag{5.57}$$

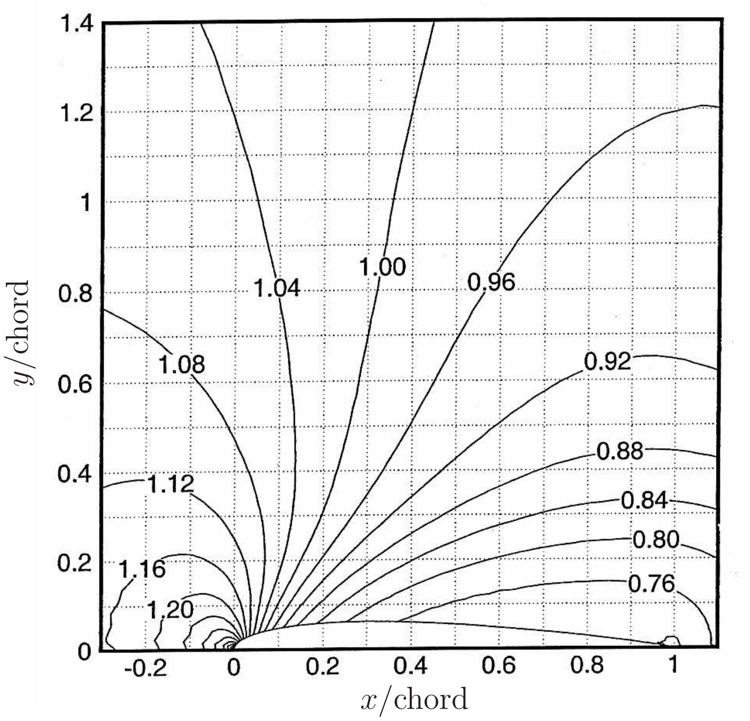
As discussed previously, the maximum density predicted by the 10-moment equations is much lower than the values obtained by the Navier-Stokes equations with slip velocity and by the Euler equations.



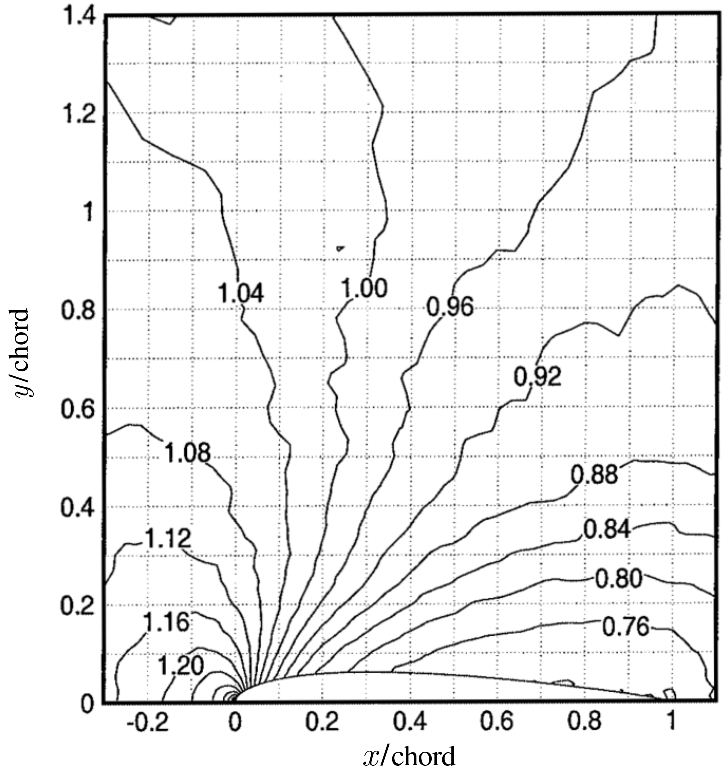
(a) 10-moment model



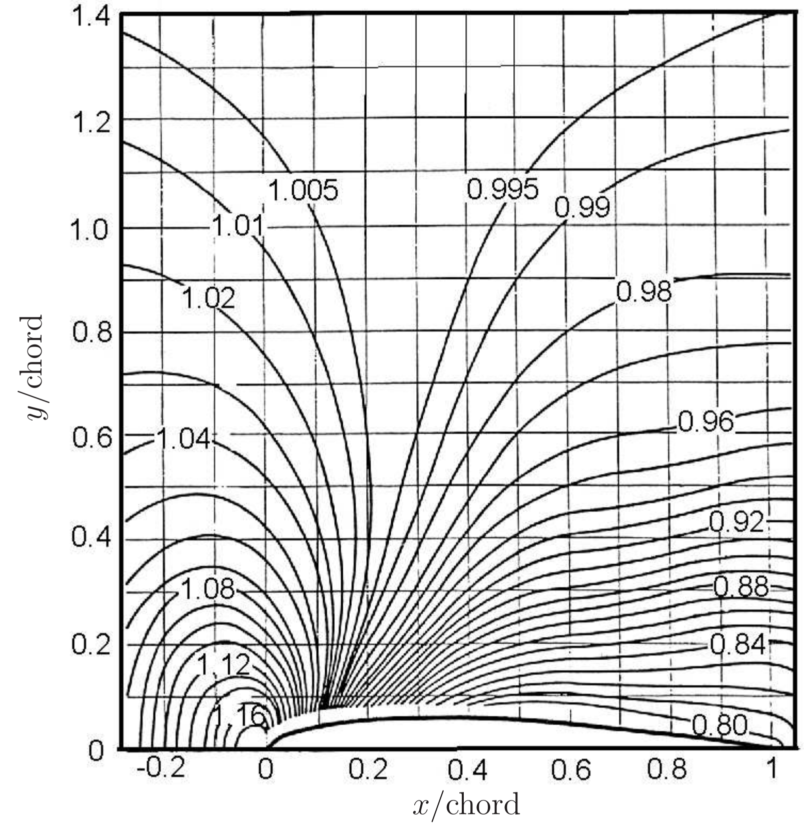
(b) Navier–Stokes equations



(c) IP method

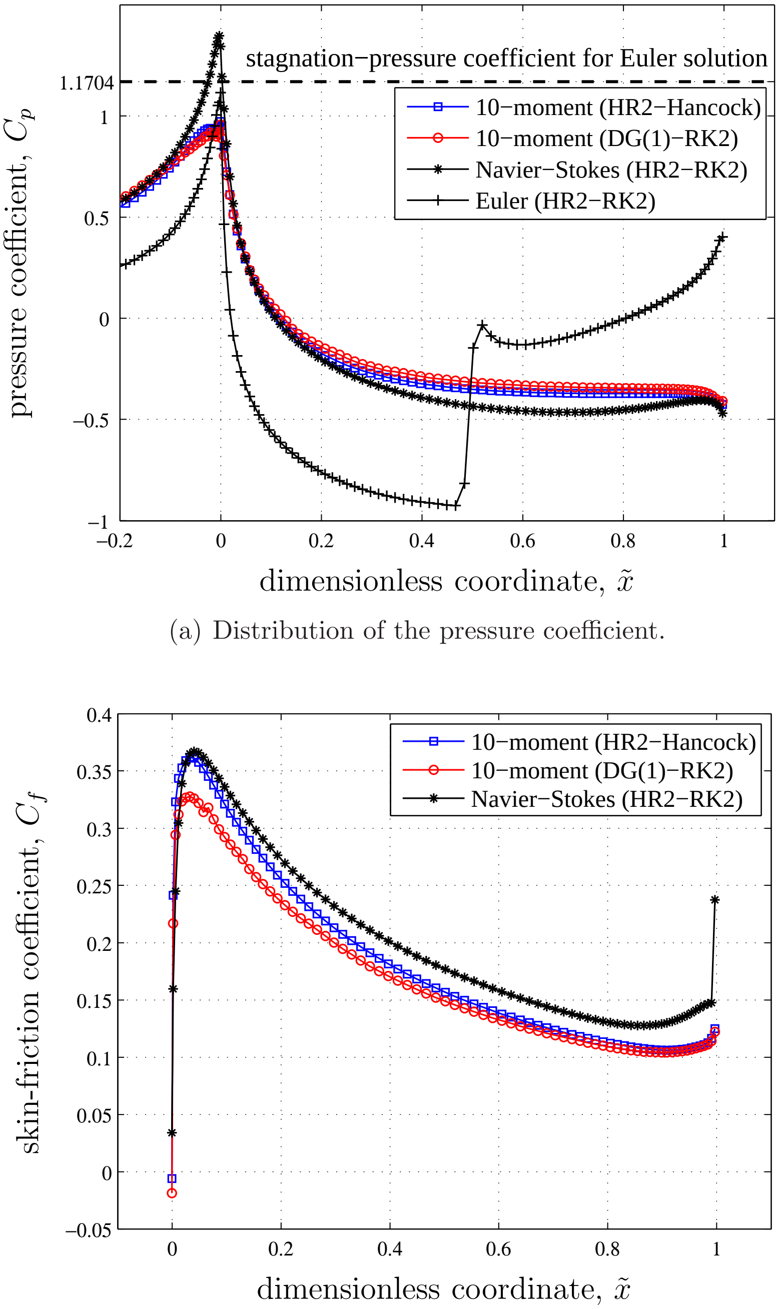


(d) DSMC method



(e) Experiment

Figure 5.11: Density distribution ($\rho/\rho_\infty$) around NACA 0012 micro-airfoil by the 10-moment model, a Navier–Stokes solution, IP, DSMC methods, and experiment.



(a) Distribution of the pressure coefficient.

(b) Distribution of the skin-friction coefficient.

Figure 5.12: Dimensionless pressure and shear stress along the NACA0012 airfoil obtained by 10-moment, Navier–Stokes, and Euler codes ($Ma = 0.8, Re = 73, \alpha = 0°$).

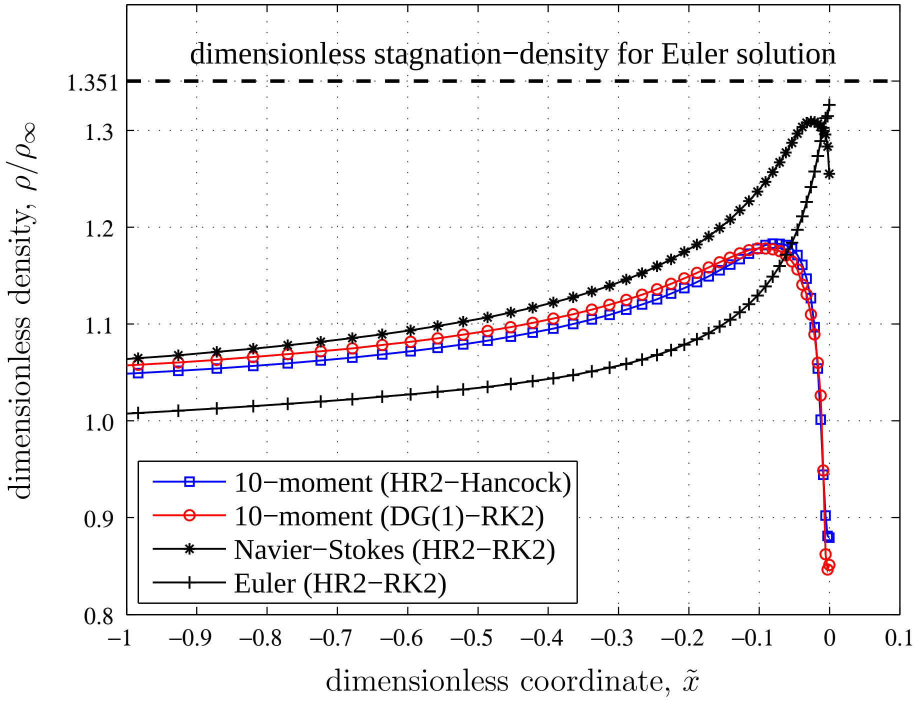


Figure 5.13: Distribution of the dimensionless density along the centerline of interval $\tilde{x} \in [-1, 0]$.

## 5.6 Avoiding Embedded Inviscid Shocks

The appearance of a discontinuity inside a steady viscous shock structure computed with a hyperbolic-relaxation system indicates that the chosen relaxation term is inadequate: it lacks information about the nonlinear characteristic fields of the hyperbolic operator. To gain understanding, consider the 1-D hyperbolic-relaxation system

$$\mathbf{u}_t + \mathbf{A}\mathbf{u}_x = -\frac{\mathbf{s}}{\tau}, \tag{5.58}$$

where the matrix $\mathbf{A}$ is the flux Jacobian. Assume there exists a traveling-wave solution of the form

$$\mathbf{u}(x,t) = \mathbf{u}(z), \quad z = x - ct, \tag{5.59}$$

where $c$ is the shock speed. Inserting this equation into the system yields the ODE

$$(-c\,\mathbf{I} + \mathbf{A})\mathbf{u}' = -\frac{\mathbf{s}}{\tau}. \tag{5.60}$$

Now express $\mathbf{u}'$ and $\mathbf{s}$ in terms of the eigenvectors $\mathbf{r}_k$ of $\mathbf{A}$:

$$\sum_k (\lambda_k - c) C_k' \mathbf{r}_k = -\frac{1}{\tau} \sum_k \sigma_k \mathbf{r}_k; \tag{5.61}$$

here $\lambda_k$ is an eigenvalue of $\mathbf{A}$, $C_k'$ is the corresponding characteristic variable, and $\sigma_k$ is the amplitude of the component along $\mathbf{r}_k$ in the source vector.

Assume the shock is generated in the $l$-th field of characteristics; this means that there is a point inside the shock structure where the difference between characteristic speed and shock speed vanishes:

$$\lambda_l - c = 0. \tag{5.62}$$

We may call this the *sonic* point. If in this point the amplitude $\sigma_l$ does *not* vanish simultaneously, $C'$ can not remain finite, indicating the need to put a discontinuity inside the viscous shock structure.

This is reminiscent of steady inviscid transonic flow in a converging-diverging channel, which is ruled by the *area-velocity relation*:

$$(M^2 - 1)u' = \frac{A'}{A}. \tag{5.63}$$

If the flow reaches the sonic point ($M = 1$) before arriving at the throat ($A' = 0$), $u'$ becomes infinite.

The following detailed example demonstrates how an inviscid embedded shock is created, and, at the same time, how it can be avoided. Consider the hyperbolic-relaxation system

$$u_t + v_x = 0, \tag{5.64a}$$

$$v_t + f^2u^2u_x = -\frac{v - \frac{1}{2}u^2}{\tau}, \quad f > 1; \tag{5.64b}$$

its characteristic speeds are $\pm fu$. For large times $v$ approaches $\frac{1}{2}u^2$, so the first equation tends toward Burgers' equation,

$$u_t + uu_x = \mu u_{xx}; \tag{5.65}$$

the relation between $\mu$ and $\tau$ must still be determined. Note that the equilibrium equation has the characteristic speed $u$, which lies between the characteristic speeds of the hyperbolic system, as required by the so-called *subcharacteristic* condition [Liu87].

The traveling-wave solution satisfies

$$-cu' + v' = 0, \tag{5.66a}$$

$$-cv' + f^2u^2u' = -\frac{v - \frac{1}{2}u^2}{\tau}. \tag{5.66b}$$

Eliminating both $v'$ and $v$ from the second equation by using the first equation leads to the ODE

$$\frac{(-c^2+f^2u^2)}{2v_0-2cu_0+c^2-(u-c)^2}du=-\frac{dz}{2\tau}; \tag{5.67}$$

without loss of generality we may choose the sonic point to lie at $z=0$:

$$u_0=c. \tag{5.68}$$

In a Burgers shock the profile is antisymmetric about this point.

In order to find a shock-like solution we must assume

$$2v_0-c^2=U^2, \tag{5.69}$$

where $U$ is half the jump in $u$ across the shock. The ODE now reads

$$\frac{(-c^2+f^2u^2)}{U^2-(u-c)^2}du=-\frac{dz}{2\tau}, \tag{5.70}$$

If the shock is slow enough, the shock structure will include the values $u=\pm c/f$, causing the numerator on the LHS to vanish and $u'$ to become infinite. This will lead to an inviscid shock between those values. In the special case of a steady shock ($c=0$), the profile is continuous, but it still has an infinite slope in the sonic point:

$$u\approx\sqrt[3]{-\frac{3U^2}{2f^2\tau}z}, \quad z \text{ small.} \tag{5.71}$$

For systems relaxing to an equation more complicated than Burgers' equation, the profile is not antisymmetric around the sonic point, causing the inviscid jump to appear even in a steady shock profile.

From (5.70) it is obvious that the LHS factor $-c^2+f^2u^2$ is the culprit; we may explore putting it also on the RHS, suitably normalized, i. e.,

$$\frac{-c^2+f^2u^2}{U^2-(u-c)^2}du=-\frac{-c^2+f^2u^2}{2U^2\tau}dz, \tag{5.72}$$

or

$$\frac{du}{U^2-(u-c)^2} = -\frac{dz}{2U^2\tau}, \tag{5.73}$$

The solution of this ODE, with boundary conditions

$$u_{-\infty} = c+U, \quad u_{\infty} = c-U, \tag{5.74}$$

is

$$u = c - U\tanh\left(\frac{z}{2U\tau}\right), \tag{5.75}$$

which is identical in form to the Burgers shock profile

$$u = c - U\tanh\left(\frac{Uz}{2\mu}\right). \tag{5.76}$$

This shows how to choose the value of $\tau$ in order to relax to a viscosity coefficient $\mu$:

$$\tau = \frac{\mu}{U^2}. \tag{5.77}$$

Thus, the embedded inviscid shock has been completely removed.

In order to show that a time-marching scheme can find the proper asymptotic solution of the hyperbolic-relaxation system, we used a standard second-order upwind-biased finite-volume scheme with a two-stage time-integrator to find the steady solution $(c = 0)$ of the modified system

$$u_t + v_x = 0; \tag{5.78a}$$

$$v_t + f^2u^2u_x = -\frac{\left(v-\frac{1}{2}u^2\right)f^2u^2}{U^2\tau}, \tag{5.78b}$$

with $f = 2.0,\ \tau = 0.1$, and boundary conditions

$$u_{\pm\infty} = \pm U, \quad v_{\pm\infty} = U, \quad U = 1. \tag{5.79}$$

Once the time derivative of the solution has dropped below a certain threshold, the solution is compared to the cell-averaged exact profile. Figure 5.14 shows the numerical results plotted on top of the exact profile, for $\Delta x = 0.075$. There is no trace of an inviscid jump, and the agreement appears to be excellent. For comparison, Figure 5.15 shows the incorrect profile obtained with the original system; it matches the cubic-root solution (5.71).

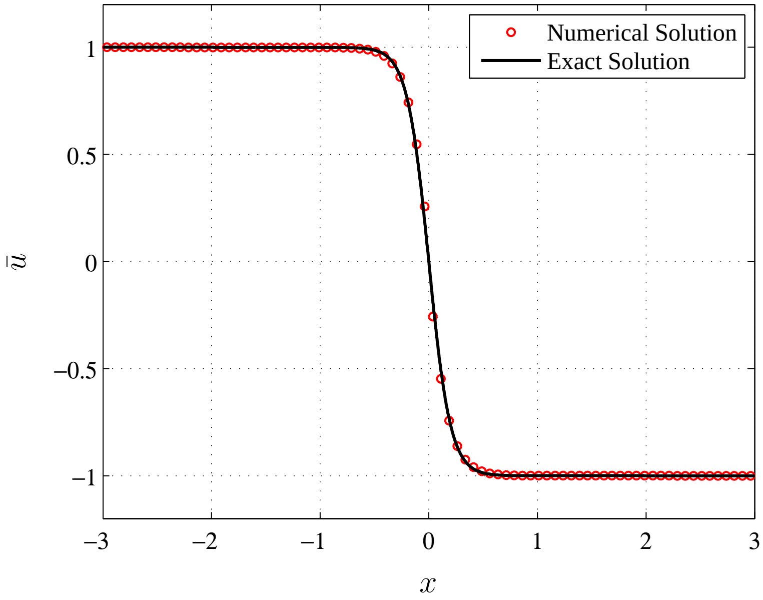


Figure 5.14: Steady Burgers shock profile (line, exact solution cell averaged) and numerical approximation (symbols) obtained with the modified hyperbolic-relaxation system (5.78a); $\tau = 0.1$, $\Delta x = 0.075$.

A grid-refinement study shows second-order convergence of the numerical to the exact solution, see Figure 5.16. Regarding the number of time-steps needed till convergence, this is a function of the cell size for both the hyperbolic-relaxation scheme and a similar scheme applied directly to Burgers' equation. The dependence on the cell-size is not the same, due to the different character of the equations. For $\Delta x \approx 0.1$, with ample coverage of the shock profile, the hyperbolic-relaxation scheme requires fewer time-steps than the direct Burgers scheme, but this advantage

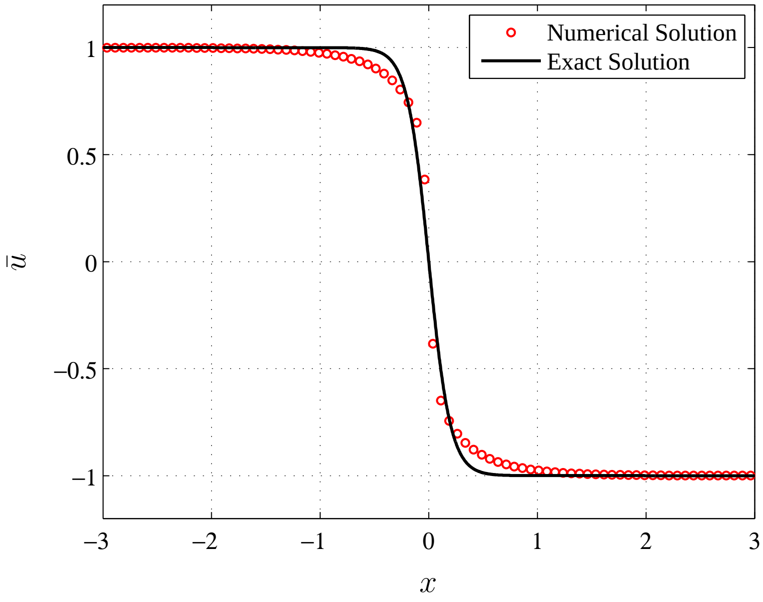


Figure 5.15: Steady Burgers shock profile and numerical approximation obtained with the original hyperbolic-relaxation system (5.64a); $\tau = 0.1$, $\Delta x = 0.075$. The numerical profile is too steep; its derivative in the origin is infinite.

is lost because of its larger computational cost per time step. We expect the two approaches to be comparable in efficiency in real fluid-dynamical applications.

We are currently studying how to include characteristic information on the RHS of the general hyperbolic-relaxation system (5.58), so as to make the inviscid embedded shocks disappear. Based on the previous analysis it is obvious that the matrix $(\mathbf{A} - c\,\mathbf{I})$ and/or its eigenvalues, suitably normalized, will have to enter. In most problems the shock speed $c$ is not *a priori* known, but it can be estimated at each interface by the dominant-wave-speed formula used in the Harten–Lax–Van Leer (HLL) approximate Riemann solver:

$$V_{\text{HLL}} = \frac{\Delta\mathbf{u} \cdot \mathbf{M}\,\Delta\mathbf{f}}{\Delta\mathbf{u} \cdot \mathbf{M}\,\Delta\mathbf{u}}; \tag{5.80}$$

here $\mathbf{M}$ is a suitable positive-definite matrix. Since we intend to use the HLL-solver anyway, the use of $V_{\text{HLL}}$ in the relaxation term comes at zero additional cost.

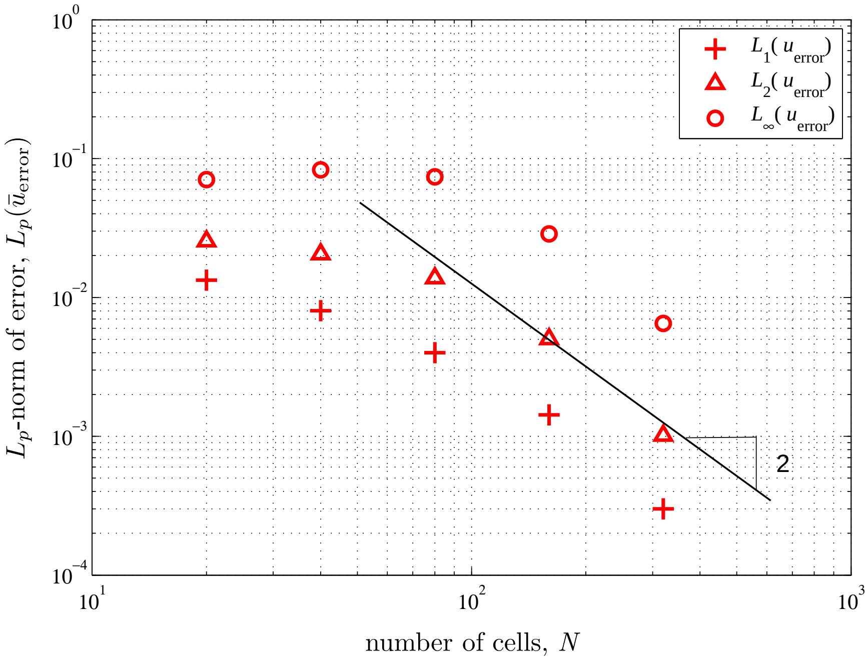


Figure 5.16: Grid convergence of error norms for steady-shock profiles obtained with the hyperbolic-relaxation system with modified source term.

# CHAPTER VI

# CONCLUSIONS

## 6.1 Summary

In this thesis, a step towards a first-order PDE approach to computational fluid dynamics is described. This approach is rather radical; the Navier–Stokes equations are no longer considered as target model equations to solve numerically. Our motivation to move into such an unexplored area is due to the fact that currently available numerical methodologies for advection-dominated compressible flows are not necessarily efficient. Part of the reason is that these methods have trouble remaining just second-order accurate on a distorted, unstructured grid. Furthermore, a numerical method intended to solve the Navier–Stokes equations can not be applied to continuum-transition flows since the model equations themselves are physically invalid in such a flow regime.

In order to advance beyond these issues that plague standard methods for the compressible Navier–Stokes equations, namely, second-order Godunov-type finite-volume methods, two approaches are taken: we adopt first-order PDEs as model equations, and discretize them by a compact numerical method, i.e., the discontinuous Galerkin method.

The major contributions of this thesis are:

- Extension of Huynh's original upwind moment method to systems of hyperbolic equations with stiff relaxation source terms, by utilizing Gauss–Radau quadrature in time. We named the method the DG(1)–Hancock method (Chapter II);

- Formulation of the fully discrete 2-D DG(1)–Hancock method on both quadrilateral and triangular elements (Chapter II);

- Detailed and comparative Fourier analysis of the DG(1)–Hancock method. Dissipation/dispersion errors, order of accuracy, and stability are compared to those of various fully and semi-discrete methods; it is shown that the DG(1)–Hancock method is third-order accurate, with a less restrictive stability condition than semi-discrete methods. The high accuracy and efficiency of the DG(1)–Hancock method are confirmed by numerical tests (Chapter III);

- The discovery that, for the DG(1)–Hancock method combined with the Rusanov flux, the maximum Courant number based on the frozen-wave speed depends on the value of the equilibrium-wave speed. In the 1-D case, its stability range varies from $\frac{1}{3}$ to 1 as the equilibrium wave speed increases from 0 to 1. In the 2-D case, the maximum Courant number lies in the interval of $\frac{1}{3}$ to $\frac{2}{3}$ (Chapter III);

- Proof that the unconditional instability of DG(1) combined with the Lax–Friedrichs flux is caused by the extraneous root. In general, the stability of the DG(1) method combined with the upwind flux is restricted by high-frequency waves (Chapter III);

- Linking the linear analysis of hyperbolic conservation laws to that of hyperbolic-relaxation equations by defining a Courant number based on the frozen wave speed, i.e., $\nu := 1\dfrac{\Delta t}{\Delta x}$, instead of $r\dfrac{\Delta t}{\Delta x}$. It is shown that an upwind method for hyperbolic-relaxation equations in the near-equilibrium limit is equivalent to directly discretizing the reduced equilibrium-equations with the Rusanov flux (Chapter IV);

- Demonstration that the high efficiency of the DG(1)–Hancock method for conservation laws is preserved when hyperbolic-relaxation equations are considered. The original Hancock method loses its efficiency in the stiff regime since the effective flux is no longer upwind in this limit (Chapter IV);

- Explanation of the results of Lowrie and Morel [LM02], whose demonstration of uniform second-order accuracy of DG(1) (independent of $\epsilon$) turns out to depend critically on their assumption that $r = O(\epsilon)$. In our scaling with this limit, it is further shown that DG(1)–Hancock loses its superiority over HR2–Hancock (Chapter IV);

- Numerical computation of the near-equilibrium solution (Navier–Stokes-like limit) of a problem governed by the 1-D Euler equations with heat transfer (Chapter IV);

- Numerical solutions of the 10-moment equations with the HR2–MOL, Hancock, and DG(1)–MOL methods. Computations of steady 1-D shock structures show the high accuracy of the DG(1) spatial discretization. A more practical calculation, of an external flow around a micro-airfoil for which experimental results exist, serves as a benchmark, for comparison to results of a Navier–Stokes solver and particle-based methods. An apparent advantage of the 10-moment model over Navier–Stokes solvers is the treatment of the slip velocity along the wall: it follows directly from the elements of the stress tensor, which are state variables (Chapter V);

- Removal of the inviscid frozen shock (drawback of moment approach) in the solution of hyperbolic-relaxation model system by a modification of the relaxation term (Chapter V).

## 6.2 Future Work

As it was mentioned previously, the first-order PDE approach is relatively new; there are a great number of issues that need to be explored/solved in order to make the approach competitive or even preferable to the conventional methodologies: finite-volume Navier–Stokes solvers and DSMC. Here, we list a few topics for future work:

- First and foremost, constructing robust and physically more realistic models beyond the 10-moment equations is necessary. The 10-moment equations adopted in this thesis are the best known and most robust among sets of moment equations, but the model predicts frozen shocks (jumps) inside viscous shock structures and does not permit heat transfer. Improvement in these areas is critical for reliably computing high-speed flows. Higher-moment systems have been developed and are being numerically studied, but they are not quite robust [Bro96];

- The efficiency and accuracy of the DG(1)–Hancock method on a distorted, unstructured grid needs to be investigated numerically, and possibly complemented with a Fourier analysis on right triangles; our current results are restricted to either rectangular or quadrilateral grids;

- Extending the DG(1)–Hancock method, currently third-order accurate for hyperbolic conservation laws, to higher-order accuracy or possibly arbitrary order is a challenge. In order to achieve this goal, a genuine space-time discretization is necessary; the resulting method is likely to be implicit. Yet, first-order PDE's often lead to a point-implicit method, whereas the Navier–Stokes solvers usually can not avoid global data-dependence;

- Particularly for DG methods, two issues are still remaining: the necessity of more sophisticated limiters that will allow higher than second-order accuracy of multi-dimensional solutions, and reduction of the computational run-time. Resolving these two issues will be a condition for attracting CFD practitioners;

- The method of moments is used to derive moment equations from the Boltzmann equation. Similarly, the method of moments (weak formulation) is adopted to derive a discontinuous Galerkin method for given PDEs. A link between these two procedures, one at the PDE level and the other at the discrete level, might exist, and they may be treated uniformly;

- The treatment of slip velocity on a curved wall is vague, especially for Navier–Stokes solvers. Rigorous analytical and numerical investigations are necessary for both Navier–Stokes and moment approaches;

- Development of the ultimate scheme/flux function which would produce the upwind method in the equilibrium limit is still worth pursuing.

# APPENDICES

# APPENDIX A

# Implementation of Coordinate Transformations

In Chapter II, the multidimensional DG(1)–Hancock method was introduced. The method is formulated such that both quadrilateral and triangular elements can be used. In the final update formulas (2.88) on page 78, three coordinate-dependent terms appear: mass matrix, $\{|A_j|, K_{j1}, K_{j2}, K_{j3}\}$, length of an edge in the surface integral, and volume integral of the flux $\iint\limits_{K_j} (\cdot)\, d\boldsymbol{x}$. While computing an edge length is trivial, the mass matrix and the quadrature for the volume integral depend on the shape of the element. For implementation purposes, specific formulations of these geometrically dependent quantities and the volume integral in the local coordinates are presented in this appendix. The implementation of more complex elements, e.g., curved elements, three-dimensional elements, can be found in [ZTZ05, KS05, Hug00, Li06].

## A.1 Quadrilateral Elements

Figure A.1 shows the schematic of a coordinate transformation from the quadrilateral global element $K_j$ to the square local element $\widehat{K}_j = [-1, 1] \times [-1, 1]$. Any coordinates $\boldsymbol{x} = (x, y)$ in $K_j$ are parameterized by bilinear functions of $\boldsymbol{\xi} = (\xi, \eta)$

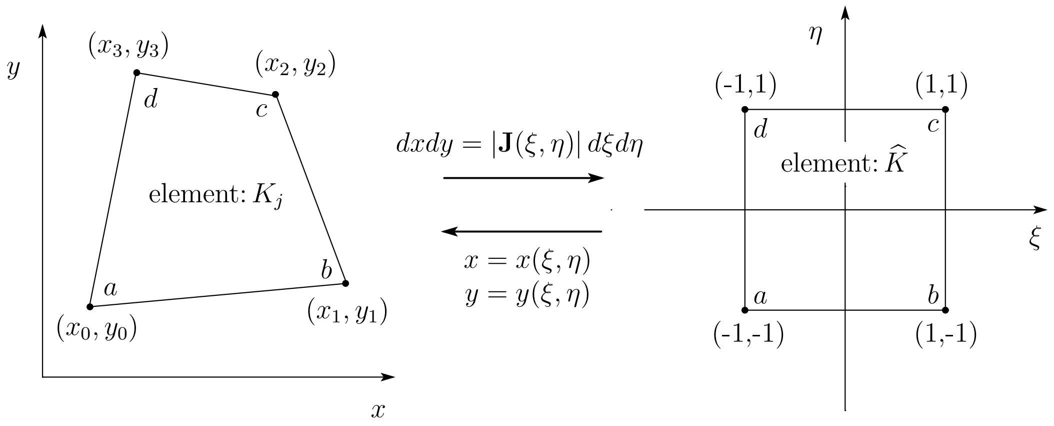


Figure A.1: Coordinate transformation between the global element $K_j$ and the local element $\widehat{K}_j$.

such that

$$\boldsymbol{x}(\boldsymbol{\xi}) = \frac{1-\xi}{2}\frac{1-\eta}{2}\boldsymbol{x}_0 + \frac{1+\xi}{2}\frac{1-\eta}{2}\boldsymbol{x}_1 + \frac{1+\xi}{2}\frac{1+\eta}{2}\boldsymbol{x}_2 + \frac{1-\xi}{2}\frac{1+\eta}{2}\boldsymbol{x}_3, \quad \text{(A.1)}$$

where $\xi, \eta \in [-1, 1]$. The Jacobian matrix $\mathbf{J}$ is defined by

$$\mathbf{J} := \begin{pmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{pmatrix}, \quad \text{(A.2)}$$

then the Jacobian determinant is given by

$$|\mathbf{J}(\xi,\eta)| = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}. \quad \text{(A.3)}$$

Recall that through a coordinate transformation, the integration variables satisfy the following identity:

$$dxdy \equiv |\mathbf{J}(\xi,\eta)|\, d\xi d\eta, \quad \text{(A.4)}$$

hence, spatial integration of a function $g(x, y)$ over the domain $K_j$ is transformed to the quare local domain $\widehat{K}_j$ such that

$$\int\limits_{K_j} g(x,y)\, dxdy = \int\limits_{\widehat{K}_j} g(\xi,\eta)\, |\mathbf{J}(\xi,\eta)|\, d\xi d\eta. \quad \text{(A.5)}$$

By applying the above transformation, components of the mass matrix of a quadrilateral element in the global coordinates are analytically evaluated by the coordinates of its nodes:

$$
\begin{aligned}
|A_j| := \iint\limits_{K_j} dxdy &= \iint\limits_{\widehat{K}} |\mathbf{J}(\xi,\eta)|\, d\xi d\eta \\
&= \frac{1}{2}\left[(x_0 - x_2)(y_1 - y_3) - (x_1 - x_3)(y_0 - y_2)\right], \qquad \text{(A.6)}
\end{aligned}
$$

$$
\begin{aligned}
\iint\limits_{K_j} x\, dxdy &= \iint\limits_{\widehat{K}} x(\xi,\eta)|\mathbf{J}(\xi,\eta)|\, d\xi d\eta \\
&= \frac{1}{6}\big[(x_0 - x_2)(x_1y_1 - x_3y_3) - (x_1 - x_3)(x_0y_0 - x_2y_2) \qquad \text{(A.7)} \\
&\quad + (x_0^2 - x_2^2)(y_1 - y_3) - (x_1^2 - x_3^2)(y_0 - y_2)\big],
\end{aligned}
$$

$$
\begin{aligned}
\iint\limits_{K_j} y\, dxdy &= \iint\limits_{\widehat{K}} y(\xi,\eta)|\mathbf{J}(\xi,\eta)|\, d\xi d\eta \\
&= -\frac{1}{6}\big[(y_0 - y_2)(x_1y_1 - x_3y_3) - (y_1 - y_3)(x_0y_0 - x_2y_2) \qquad \text{(A.8)} \\
&\quad + (y_0^2 - y_2^2)(x_1 - x_3) - (y_1^2 - y_3^2)(x_0 - x_2)\big],
\end{aligned}
$$

$$
\begin{aligned}
\iint\limits_{K_j} x^2\, dxdy = \frac{1}{12}\big[&(x_0^3 - x_2^3)(y_1 - y_3) - (x_1^3 - x_3^3)(y_0 - y_3) \\
&+ (x_0^2 - x_2^2)(x_1y_1 - x_3y_3) - (x_1^2 - x_3^2)(x_0y_0 - x_2y_2) \\
&+ (x_0 - x_2)(x_1^2y_1 - x_3^2y_3) - (x_1 - x_3)(x_0^2y_0 - x_2^2y_2)\big], \qquad \text{(A.9)}
\end{aligned}
$$

$$
\begin{aligned}
\iint\limits_{K_j} y^2\, dxdy = -\frac{1}{12}\big[&(y_0^3 - y_2^3)(x_1 - x_3) - (y_1^3 - y_3^3)(x_0 - x_3) \\
&+ (y_0^2 - y_2^2)(x_1y_1 - x_3y_3) - (y_1^2 - y_3^2)(x_0y_0 - x_2y_2) \\
&+ (y_0 - y_2)(x_1y_1^2 - x_3y_3^2) - (y_1 - y_3)(x_0y_0^2 - x_2y_2^2)\big], \qquad \text{(A.10)}
\end{aligned}
$$

$$\iint\limits_{K_j} xy\,dxdy = \frac{1}{12}\Big[(x_0 - x_2)(x_1 y_1^2 - x_3 y_3^2) - (x_1 - x_3)(x_0 y_0^2 - x_2 y_2^2) \\ - (y_0 - y_2)(x_1^2 y_1 - x_3^2 y_3) + (y_1 - y_3)(x_0^2 y_0 - x_2^2 y_2) \\ + \frac{1}{2}\big((x_0^2 - x_2^2)(y_1^2 - y_3^2) - (x_1^2 - x_3^2)(y_0^2 - y_2^2)\big)\Big]. \tag{A.11}$$

The above geometric quantities (metrics) are evaluated analytically since the coordinate transformation is expressed in bilinear form (A.1).

Conversely, a quadrature is necessary to evaluate the nonlinear flux tensor $\mathbf{f}(\mathbf{u}(\boldsymbol{x}))$. As described in its derivation, the four-point Gauss quadrature (see Figure 2.8 on page 74) is employed to approximate the spatial integration:

$$\int\limits_{K_j} \mathbf{f}(\mathbf{u}(\boldsymbol{x}))\,d\boldsymbol{x} = \int\limits_{\widehat{K}} \mathbf{f}(\mathbf{u}(\boldsymbol{\xi}))|\mathbf{J}(\boldsymbol{\xi})|\,d\boldsymbol{\xi} \\ \approx \sum_{i=0}^{3} w_i |\mathbf{J}(\boldsymbol{\xi}_i)|\,\mathbf{f}(\mathbf{u}(\boldsymbol{\xi}_i)), \tag{A.12}$$

where the weights are $w_i = \dfrac{1}{4}, i = 0, \ldots, 3$, and the Gauss points where the flux tensor is evaluated are

$$\boldsymbol{\xi}_0 = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right), \quad \boldsymbol{\xi}_1 = \left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right), \tag{A.13a}$$

$$\boldsymbol{\xi}_2 = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right), \quad \boldsymbol{\xi}_3 = \left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right). \tag{A.13b}$$

The solution at the Gauss point $\mathbf{u}(\boldsymbol{\xi}_i)$ is computed by its solution representation (2.58a) on page 61 together with the coordinate transformation from $\boldsymbol{\xi}$ to $\boldsymbol{x}$ given by (A.1).

## A.2 Triangular Elements

At first, it is worthwhile to mention that a triangular element can be subdivided into three quadrilateral elements [Hug00, p. 165]. See Figure A.2.

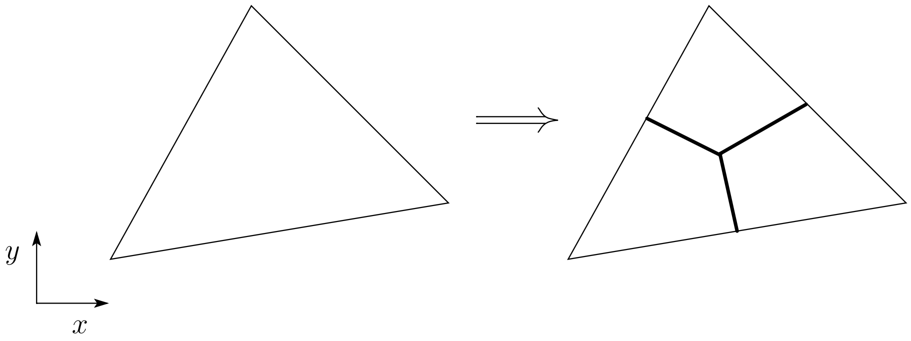


Figure A.2: A Triangular element can be subdivided into three quadrilateral elements.

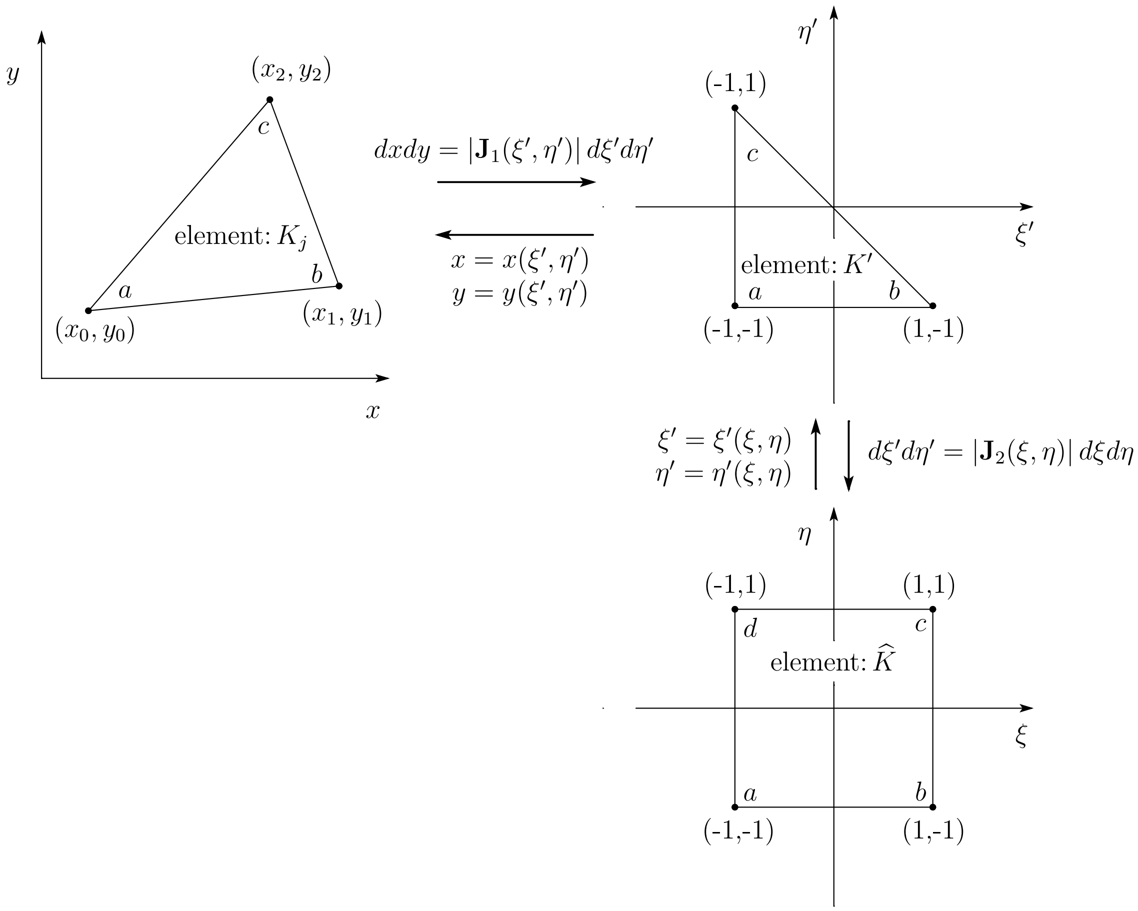


Figure A.3: Coordinate transformation for triangular elements.

Unlike a quadrilateral element, a triangular element requires two consecutive coordinate transformations to map from the global element to the local element. See Figure A.3 for the coordinate transformations. The resulting mapping function is

$$\boldsymbol{x}(\boldsymbol{\xi}) = \frac{1-\xi}{2}\frac{1-\eta}{2}\boldsymbol{x}_0 + \frac{1+\xi}{2}\frac{1-\eta}{2}\boldsymbol{x}_1 + \frac{1+\eta}{2}\boldsymbol{x}_2. \tag{A.14}$$

Following the same procedure as for the quadrilateral element, geometrical quantities are computed by $x$-,$y$-coordinates of the element's nodes:

$$\begin{aligned} |A_j| &:= \iint\limits_{\hat{K}} |\mathbf{J}(\xi,\eta)|\, d\xi d\eta \\ &= \frac{1}{2}\left[(x_1-x_0)(y_2-y_0)-(x_2-x_0)(y_1-y_0)\right], \end{aligned} \tag{A.15}$$

$$\iint\limits_{K_j} x\, dxdy = \frac{|A_j|}{3}(x_0+x_1+x_2), \tag{A.16}$$

$$\iint\limits_{K_j} y\, dxdy = \frac{|A_j|}{3}(y_0+y_1+y_2), \tag{A.17}$$

$$\iint\limits_{K_j} x^2\, dxdy = \frac{|A_j|}{6}\left(x_0^2+x_0x_1+x_1^2+x_0x_2+x_1x_2+x_2^2\right), \tag{A.18}$$

$$\iint\limits_{K_j} y^2\, dxdy = \frac{|A_j|}{6}\left(y_0^2+y_0y_1+y_1^2+y_0y_2+y_1y_2+y_2^2\right), \tag{A.19}$$

$$\iint\limits_{K_j} xy\, dxdy = \frac{|A_j|}{12}\big(x_0(2y_0+y_1+y_2)+x_1(y_0+2y_1+y_2)+x_2(y_0+y_1+2y_2)\big). \tag{A.20}$$

# APPENDIX B

# Fourier Analysis of High-Order Methods

The Fourier analysis of three high-order methods is conducted as a continuation of previous analyses shown in Chapter III. The difference operator and local truncation error of each method are presented in a concise manner.

## B.1 HR3–RK3 method

The third-order high-resolution Godunov method (HR3–MOL) utilizes quadratic reconstructed values as input for the flux function. The input values are given by

$$u_{j+1/2,L}(t) = \bar{u}_j + \frac{\Delta x}{2}\left(\frac{\bar{u}_{j+1} - \bar{u}_{j-1}}{2\Delta x}\right) + \frac{\Delta x^2}{12}\left(\frac{\bar{u}_{j+1} - 2\bar{u}_j + \bar{u}_{j-1}}{\Delta x^2}\right), \tag{B.1a}$$

$$u_{j+1/2,R}(t) = \bar{u}_{j+1} - \frac{\Delta x}{2}\left(\frac{\bar{u}_{j+2} - \bar{u}_j}{2\Delta x}\right) + \frac{\Delta x^2}{12}\left(\frac{\bar{u}_{j+2} - 2\bar{u}_{j+1} + \bar{u}_j}{\Delta x^2}\right). \tag{B.1b}$$

After inserting the cell interface fluxes into the original semi-discrete form, the spatial difference operator becomes

$$\mathrm{N}_{\mathrm{HR3}} = -\frac{1}{12\Delta x}\left[(q-r)(\delta^+)^2 - 2(q-3r)\delta^+ + 2(q+3r)\delta^- + (q+r)(\delta^-)^2\right], \tag{B.2}$$

or, for a Fourier mode,

$$\mathrm{N}_{\mathrm{HR3}} = -\frac{1}{12\Delta x}\left[(q-r)e^{2i\beta} - 4(q-2r)e^{i\beta} + 6q - 4(q+2r)e^{-i\beta} + (q+r)e^{-2i\beta}\right]. \tag{B.3}$$

The spatial discretization in the low-frequency limit has the following discretization error:

$$\lambda_{\mathrm{HR3}} - \lambda_{\mathrm{exact}} = -\frac{q}{12}\,\boxed{\Delta x^3}\,k^4 + O\big(k^5\big)\,, \tag{B.4}$$

thus, the HR3 method is third-order accurate in space. Applying the RK3 method as the time integrator, the local truncation error becomes

$$\mathrm{LTE}_{\mathrm{HR3RK3}} = -\frac{r}{12}\left(\frac{q}{r}\,\boxed{\Delta x^3} + \frac{1}{2}r^3\,\boxed{\Delta t^3}\,\right)k^4 + O\big(k^5\big)\,. \tag{B.5}$$

Thus, the HR3–RK3 method is third-order accurate in space and time.

## B.2 DG(2)–RK3 method

The DG(2)–MOL introduces an extra update equation for $\overline{\Delta^2 u_j}$:

$$\frac{\partial\overline{\Delta^2 u_j}(t)}{\partial t} = -\frac{1}{\Delta x}60\left(\hat{f}_{j+1/2}(t) - \hat{f}_{j-1/2}(t) - r\overline{\Delta u_j}(t)\right). \tag{B.6}$$

The above update equation together with two update equations for $\bar{u}_j(t)$ and $\overline{\Delta u_j}(t)$ given by (3.97) on page 119 are solved simultaneously. Similar to HR3–MOL, the input values for the flux function are obtained by the quadratic solution representation:

$$u_{j+1/2,L}(t) = \bar{u}_j + \frac{1}{2}\overline{\Delta u_j} + \frac{1}{12}\overline{\Delta^2 u_j}, \tag{B.7a}$$

$$u_{j+1/2,R}(t) = \bar{u}_{j+1} - \frac{1}{2}\overline{\Delta u_{j+1}} + \frac{1}{12}\overline{\Delta^2 u_{j+1}}. \tag{B.7b}$$

After inserting the cell interface fluxes into the original semi-discrete form, the spatial difference operator becomes

$$\mathbf{N}_{\mathrm{DG(2)}} = \boldsymbol{\mathcal{A}}^{+}\mathbf{D}^{+} + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{A}}^{-}\mathbf{D}^{-}, \tag{B.8}$$

where

$$\boldsymbol{\mathcal{A}}^{+} = \frac{q-r}{2\Delta x}\begin{pmatrix} 1 & -\dfrac{1}{2} & \dfrac{1}{12} \\ 6 & -3 & \dfrac{1}{2} \\ 60 & -30 & 5 \end{pmatrix}, \quad \boldsymbol{\mathcal{A}}^{-} = \frac{q+r}{2\Delta x}\begin{pmatrix} -1 & -\dfrac{1}{2} & -\dfrac{1}{12} \\ 6 & 3 & \dfrac{1}{2} \\ -60 & -30 & -5 \end{pmatrix}, \tag{B.9a}$$

$$\boldsymbol{\mathcal{C}} = \frac{1}{\Delta x}\begin{pmatrix} 0 & 0 & 0 \\ 0 & -\dfrac{6q}{r} & -1 \\ 0 & 60 & 0 \end{pmatrix}, \quad \mathbf{D}^{\pm} = \delta^{\pm}\mathbf{I}, \tag{B.9b}$$

or, for a Fourier mode,

$$\mathbf{M}_{\mathrm{DG(2)}} = \boldsymbol{\mathcal{A}}^{+}e^{i\beta\mathbf{I}} + \boldsymbol{\mathcal{C}}' - \boldsymbol{\mathcal{A}}^{-}e^{-i\beta\mathbf{I}}, \tag{B.10}$$

where

$$\boldsymbol{\mathcal{C}}' = -\frac{r}{\Delta x}\begin{pmatrix} \dfrac{q}{r} & \dfrac{1}{2} & \dfrac{q}{12r} \\ -6 & \dfrac{3q}{r} & \dfrac{1}{2} \\ \dfrac{60q}{r} & -30 & \dfrac{5q}{r} \end{pmatrix}. \tag{B.11}$$

The asymptotic eigenvalues in the low-frequency limit are obtained by solving the characteristic equation of $\mathbf{N}_{\mathrm{DG(2)}}$. Here, we only present the result with the upwind flux $q = r$:

$$\lambda^{(1)}_{\mathrm{DG(2)}} - \lambda_{\mathrm{exact}} = -\frac{r}{7200}\,\boxed{\Delta x^5}\,k^6 + O\big(k^7\big)\,, \tag{B.12}$$

$$\lambda^{(2),(3)}_{\mathrm{DG(2)}} - \lambda_{\mathrm{exact}} = -\frac{r}{\Delta x}(3 \pm i\sqrt{51}) + O(k)\,. \tag{B.13}$$

The above equations show that the accurate eigenvalue of the DG(2) spatial discretization is fifth-order in space. Also, similar to the DG(1) method, inaccurate eigenvalues damp quickly since the real part of the leading error, $-\dfrac{3r}{\Delta x}$, is negative for both $\lambda^{(2)}_{\mathrm{DG(2)}}$ and $\lambda^{(3)}_{\mathrm{DG(2)}}$. Applying the RK3 method for the time integrator, the local truncation error becomes

$$\mathrm{LTE}_{\mathrm{DG(2)RK3}} = -\frac{r}{24}(r\nu)^3\,\boxed{\Delta x^3}\,k^4 + O\big(k^5\big)\,, \tag{B.14}$$

thus, the DG(2)–RK3 method is third-order in space and time. Note that the above error is introduced purely by the time integration method since the DG(2) spatial discretization is fifth-order accurate. Hence, it is natural to apply the fifth-order time integration method, RK5; then the local truncation error becomes

$$\mathrm{LTE}_{\mathrm{DG(2)RK5}} = -\frac{r}{7200}\left[1 + 50(r\nu)^5\right] \boxed{\Delta x^5}\, k^6 + O\bigl(k^7\bigr)\,, \tag{B.15}$$

and the method becomes fifth-order accurate. This linear analysis suggests that when the DG discretization adopts the solution representation of a polynomial of degree $k$, then the corresponding time integrator is recommended to have the order of accuracy $2k+1$. Another benefit of using higher-order Runge–Kutta method is enlarging the stability domain. However, increasing the number of stages of a time-integration method results in greater computational expense due to the Riemann problem to be solved at each stage.

## B.3 DG(2)–ADER method

The DG(2)–ADER method is obtained by approximating the volume integral of the flux as follows:

$$\frac{1}{\Delta t}\int\limits_{T^n} f(\bar{u}_j) \approx r\left(\bar{u}_j^n - \frac{1}{2}r\Delta t\frac{\overline{\Delta u_j}^n}{\Delta x} + \frac{1}{6}(r\Delta t)^2\frac{\overline{\Delta^2 u_j}^n}{\Delta x^2}\right), \tag{B.16}$$

$$\frac{\Delta x}{\Delta t}\int\limits_{T^n} \frac{\partial f(\bar{u}_j)}{\partial x} \approx r\left(\overline{\Delta u_j}^n - \frac{1}{2}r\Delta t\frac{\overline{\Delta^2 u_j}^n}{\Delta x}\right). \tag{B.17}$$

After inserting the difference forms, the spatial difference operator of the DG(2)–ADER method has the following form:

$$\mathbf{M}_{\mathrm{DG(2)ADER}} = \boldsymbol{\mathcal{A}}^{+}\mathbf{D}^{+} + \boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{A}}^{-}\mathbf{D}^{-}, \tag{B.18}$$

where

$$\boldsymbol{\mathcal{A}}^{+} = \frac{q-r}{2\Delta x}\begin{pmatrix} 1 & -\dfrac{1}{2}(1+r\nu) & \dfrac{1}{12}(1+r\nu)(1+2r\nu) \\ 6 & -3(1+r\nu) & \dfrac{1}{2}(1+r\nu)(1+2r\nu) \\ 60 & -30(1+r\nu) & 5(1+r\nu)(1+2r\nu) \end{pmatrix}, \tag{B.19a}$$

$$\boldsymbol{\mathcal{A}}^{-} = \frac{q+r}{2\Delta x}\begin{pmatrix} -1 & -\dfrac{1}{2}(1-r\nu) & -\dfrac{1}{12}(1-r\nu)(1-2r\nu) \\ 6 & 3(1-r\nu) & \dfrac{1}{2}(1-r\nu)(1-2r\nu) \\ -60 & -30(1-r\nu) & -5(1-r\nu)(1-2r\nu) \end{pmatrix}, \tag{B.19b}$$

$$\boldsymbol{\mathcal{C}} = \frac{1}{\Delta x}\begin{pmatrix} 0 & 0 & 0 \\ 0 & -\dfrac{6q}{r} & -(1-3q\nu) \\ 0 & 60 & -30r\nu \end{pmatrix}, \quad \mathbf{D}^{\pm} = \delta^{\pm}\mathbf{I}, \tag{B.19c}$$

or, for a Fourier mode,

$$\mathbf{M}_{\mathrm{DG(2)ADER}} = \boldsymbol{\mathcal{A}}^{+}e^{i\beta\mathbf{I}} + \boldsymbol{\mathcal{C}}' - \boldsymbol{\mathcal{A}}^{-}e^{-i\beta\mathbf{I}}, \tag{B.20}$$

where

$$\boldsymbol{\mathcal{C}}' = -\frac{r}{\Delta x}\begin{pmatrix} \dfrac{q}{r} & \dfrac{1}{2}(1-q\nu) & \dfrac{1}{12}\left(\dfrac{q}{r} - 3r\nu + \dfrac{2q}{r}(r\nu)^2\right) \\ -6 & 3\left(\dfrac{q}{r} + r\nu\right) & \dfrac{1}{2}\left(1 - 3q\nu - 2(r\nu)^2\right) \\ \dfrac{60q}{r} & -30(1+q\nu) & 5\left(\dfrac{q}{r} + 3r\nu + \dfrac{2q}{r}(r\nu)^2\right) \end{pmatrix}. \tag{B.21}$$

The overall accuracy is derived from the eigenvalues of the amplification matrix of a fully discrete form, $\mathbf{G}_{\mathrm{DG(2)ADER}} = \mathbf{I} + \Delta t\,\mathbf{M}_{\mathrm{DG(2)ADER}}$. Due to the complexity of the derivation, we only present the case of the upwind flux $q = r$. The local truncation error of the accurate mode becomes

$$\mathrm{LTE}^{(1),\mathrm{upwind}}_{\mathrm{DG(2)ADER}} = -\frac{r}{120}r\nu(1-r\nu)\,\boxed{\Delta x^3}\,k^4 + O\big(k^5\big)\,. \tag{B.22}$$

Thus, DG(2)–ADER method is third-order in space and time.

# APPENDIX C

# Asymptotic Expansion of Dimensionless GHHE

## C.1 One-Dimensional Systems

Let us consider the dimensionless 1-D GHHE with frozen-wave time-scaling; removing the $\hat{(\cdot)}$ symbol in (4.16) on page 213 for simplicity yields

$$\partial_t u + \partial_x v = 0, \tag{C.1a}$$

$$\partial_t v + \partial_x u = -\frac{1}{\epsilon}(v - ru), \tag{C.1b}$$

where $|r| \leq 1$ is fixed, and $\epsilon > 0$ can vary. The dimensionless time $t$ and conserved variable $u$ are set to $O(1)$. Compared to the dimensional form (4.1) on page 208, the above system shows frozen wave speeds $a_F = \pm 1$, the equilibrium speed $a_E = r$, and the relaxation time $\tau = \epsilon$. Here, we present the derivation of the reduced equation in the near-equilibrium limit. The derivation is based on the assumption that, in the near-equilibrium, the state vector $\{u, v\}$ can be expanded in terms of a small parameter, the relaxation time $\epsilon$, such that

$$u(x,t) = u_0 + \epsilon u_1 + \epsilon^2 u_2 + \epsilon^3 u_3 + O\big(\epsilon^4\big)\,, \tag{C.2a}$$

$$v(x,t) = v_0 + \epsilon v_1 + \epsilon^2 v_2 + \epsilon^3 v_3 + O\big(\epsilon^4\big)\,. \tag{C.2b}$$

These expansions are called Chapman–Enskog expansions. The above equations are inserted into (C.1), and terms of the same order of $\epsilon$ are gathered. Since $u_i, v_i, i = 0, 1, \ldots$, are not functions of $\epsilon$, the coefficient of any power of $\epsilon$ has to be zero. After some algebra, we have

$$v_0 = ru_0, \tag{C.3a}$$

$$v_1 = ru_1 - (1 - r^2)\partial_x u_0, \tag{C.3b}$$

$$v_2 = ru_2 - (1 - r^2)\partial_x u_1 - 2r(1 - r^2)\partial_{xx} u_0, \tag{C.3c}$$

$$v_3 = ru_3 - (1 - r^2)\partial_x u_2 - 2r(1 - r^2)\partial_{xx} u_1 - (1 - r^2)(5r^2 - 1)\partial_{xxx} u_0, \tag{C.3d}$$

thus, the flux $v$ can be expressed in the conservative variable $u$ such that

$$v = ru - \epsilon\,(1-r^2)\partial_x u - \epsilon^2\,2r(1-r^2)\partial_{xx} u - \epsilon^3\,(1-r^2)(5r^2-1)\partial_{xxx} u + O\big(\epsilon^4\big)\,. \tag{C.4}$$

Inserting the above equation into (C.1a) provides the reduced form of the dimensionless GHHE:

$$\partial_t u + r\partial_x u = \epsilon\,(1-r^2)\partial_{xx} u + \epsilon^2\,2r(1-r^2)\partial_{xxx} u + \epsilon^3\,(1-r^2)(5r^2-1)\partial_{xxxx} u + O\big(\epsilon^4\big)\,. \tag{C.5}$$

Note that this reduced form is valid only when $\epsilon \ll O(1)$ (near-equilibrium limit). As we can see, the equation has the form of an advection-diffusion equation with higher-order dissipation and dispersion terms; the equilibrium advection wave speed is $r$, and the leading dissipation coefficient is $\epsilon\,(1 - r^2)$. Truncating the above equation at $O(\epsilon^2)$ leads to the advection-diffusion equation:

$$\partial_t u + r\partial_x u = \epsilon\,(1 - r^2)\partial_{xx} u. \tag{C.6}$$

## C.2 Two-Dimensional Systems

Following the same procedure as in the 1-D case, the coefficients in the asymptotic expansions of $v$ and $w$ are given by

$$v_0 = ru_0, \qquad v_1 = ru_1 - (1-r^2)\partial_x u_0, \tag{C.7}$$

$$w_0 = su_0, \qquad w_1 = su_1 - (1-s^2)\partial_y u_0. \tag{C.8}$$

Thus, the fluxes can be expressed in the conservative variable $u$ such that

$$v = ru - \epsilon(1-r^2)\partial_x u + O\big(\epsilon^2\big)\,, \tag{C.9}$$

$$w = su - \epsilon(1-s^2)\partial_y u + O\big(\epsilon^2\big)\,. \tag{C.10}$$

Inserting the above equations into the original PDE for $u$ leads to the reduced form of the 2-D GHHE in the near-equilibrium limit ($\epsilon \ll 1$):

$$\partial_t u + r\partial_x u + s\partial_y u = \epsilon\left[(1-r^2)\partial_{xx} u + (1-s^2)\partial_{yy}\right] + O\big(\epsilon^2\big)\,. \tag{C.11}$$

# APPENDIX D

# Jin–Levermore's Semi-Discrete High-Resolution Godunov Method

Jin and Levermore already realized that the upwind flux in a Godunov method for the original hyperbolic-relaxation equations does not retain the upwind property in the near-equilibrium limit [JL96]. In other words, the discretization in the near-equilibrium limit is equivalent to discretizing the advection-diffusion equation directly with a somewhat more dissipative flux function. Later, it was shown that the corresponding flux is the first HLL1 flux function [HSvL05].

To remedy the excessive numerical dissipation in the near-equilibrium limit due to a non-upwind flux, Jin and Levermore came up with the idea to interlace two flux functions: one is upwind for the frozen system, and the other is upwind for the equilibrium system. The simplest way is by a linear combination: introduce the parameter $a \in [0, 1]$ such that

$$\hat{f} = a\hat{f}_{\text{frozen}} + (1 - a)\hat{f}_{\text{equilibrium}}, \tag{D.1}$$

where

$$a \sim \frac{\epsilon}{\Delta x}. \tag{D.2}$$

They chose the parameter $a$ such that in the frozen limit ($\epsilon \gg 1$), it approaches unity, so that the original upwind flux $\hat{f}_{\text{frozen}}$ is applied. Conversely, in the near-equilibrium limit ($\epsilon \ll 1$), a small $a$ turns off the frozen flux, and the equilibrium flux $\hat{f}_{\text{equilibrium}}$, which is upwind for the reduced equation, becomes dominant. Note that this modification is only applied to the flux of the conserved quantity, i.e., to the first equation of the GHHE.

Here, the method is applied to the 1-D GHHE, and a Fourier analysis is conducted. The semi-discrete form is identical to the HR2–MOL method,

$$\frac{\partial \bar{\mathbf{u}}_j(t)}{\partial t} = -\frac{1}{\Delta x}\left(\hat{\mathbf{f}}_{j+1/2} - \hat{\mathbf{f}}_{j-1/2}\right) + \frac{1}{\epsilon}\mathbf{s}\left(\bar{\mathbf{u}}_j\right), \tag{D.3}$$

yet, the interface flux is computed by the following hybrid flux:

$$\hat{\mathbf{f}}_{j+1/2} = \begin{pmatrix} \hat{v}_{j+1/2} \\ \hat{u}_{j+1/2} \end{pmatrix} = \begin{pmatrix} a\hat{f}_{1,\text{frozen}} + (1-a)\left[\hat{f}_{\text{equi}} - \epsilon(1-r^2)\dfrac{u_{j+1}-u_j}{\Delta x}\right] \\ \hat{f}_{2,\text{frozen}} \end{pmatrix}. \tag{D.4}$$

The frozen flux is obtained by the system

$$\begin{pmatrix} u \\ v \end{pmatrix}_t + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}_x = 0, \quad \longrightarrow \quad \begin{pmatrix} \hat{f}_{1,\text{ frozen}} \\ \hat{f}_{2,\text{ frozen}} \end{pmatrix} = \mathbf{A}^+\mathbf{u}_L + \mathbf{A}^-\mathbf{u}_R, \tag{D.5}$$

and the equilibrium flux by

$$u_t + ru_x = 0 \quad \longrightarrow \quad \hat{f}_{j+1/2,\text{equi}} = ru_{j+1/2,L}. \tag{D.6}$$

Following the usual procedure for a Fourier analysis, the truncation errors in the low-frequency limit is given by

$$\lambda^{(1)}_{\text{HR2–JL}} - \lambda^{\text{GHHE}}_{\text{exact}} = -\frac{ir}{12}\Delta x^2 k^3 - \left[\frac{1}{8}\big(a + (1-a)r\big)\Delta x^3 + \frac{\epsilon(1-r^2)}{12}(3a-1)\Delta x^2\right]k^4 \\ + O\big(\epsilon^2 k^3, \epsilon^3 k^4, k^5\big), \tag{D.7a}$$

$$\lambda^{(2)}_{\text{HR2–JL}} - \lambda^{\text{GHHE}}_{\text{exact}} = -\frac{1}{\epsilon} + ir(1+a)k + O\big(k^2\big). \tag{D.7b}$$

The above result without $k^3$-term corresponds to Eq. (3.17) in Jin and Levermore [JL96, p. 461], however, the term $-\frac{1}{8}(a+r)\Delta x^3$ is missing there.

As to the parameter $a$, they suggest to take

$$a := \tanh\left(\frac{\epsilon}{\Delta x}\right). \tag{D.8}$$

In the near-equilibrium limit, the above equation can be expanded with respect to $\epsilon \ll 1$ such that

$$a = \frac{\epsilon}{\Delta x} + O\big(\epsilon^3\big), \tag{D.9}$$

thus, the truncation error of the dominant eigenvalue becomes

$$\begin{aligned}\lambda^{(1)}_{\text{HR2–JL}} - \lambda^{\text{GHHE}}_{\text{exact}} = -\frac{ir}{12}\Delta x^2 k^3 - \Bigg[\frac{r}{8}\boxed{\Delta x^3} + \frac{\epsilon(1-r)(1-2r)}{24}\Delta x^2 \\ + \frac{\epsilon^2(1-r^2)}{4}\Delta x\Bigg] k^4 + O\big(\epsilon^2 k^3, \epsilon^3 k^4, k^5\big).\end{aligned} \tag{D.10}$$

Despite Jin and Leremore's claim that the numerical dissipation ($k^4$-term) is proportional to $\epsilon$, thus there is no grid size restriction, the above equation shows that there is a term independent of $\epsilon$, which they omitted. Hence, the method needs to satisfy the following condition,

$$\frac{r}{8}\Delta x^3 k^4 \ll \epsilon(1-r^2)k^2, \tag{D.11}$$

in order for the physical dissipation to be dominant. Solving for $\Delta x$ leads to

$$\Delta x \ll 2\left[\frac{\epsilon(1-r^2)}{rk^2}\right]^{1/3} = 2\left(\frac{1}{k^2\, Pe}\right)^{1/3}. \tag{D.12}$$

This threshold mesh size is less restrictive than (4.52) on page 221 for the original HR2–MOL, which was based on the pure frozen flux. Owing to the inclusion of the equilibrium flux in (D.4), the equilibrium wave speed $|r| \le 1$ disappears from the numerator.

# APPENDIX E

# Dispersion Analysis for the 2-D GHHE

Following the dispersion analysis of Hittinger for the 1-D GHHE model system [Hit00, pp. 35–47], we assume a harmonic solution,

$$\mathbf{u}(x, y, t) = \Re\{\mathbf{v}(t) \exp\bigl(-i(k_x x + k_y y)\bigr)\}, \tag{E.1}$$

where $(k_x, k_y)$ are wave numbers in $(x, y)$. Inserting this into (4.107) on page 237 leads to an ordinary differential equation in terms of $\mathbf{v}(t)$,

$$\frac{d\mathbf{v}(t)}{dt} = [i(k_x\mathbf{A} + k_y\mathbf{B}) + \mathbf{Q}]\,\mathbf{v}(t). \tag{E.2}$$

The solution of this is

$$\mathbf{v}(t) = \exp\left[it(k_x\mathbf{A} + k_y\mathbf{B} - i\mathbf{Q})\right]\hat{\mathbf{u}}_0, \tag{E.3}$$

where $\hat{\mathbf{u}}_0 \in \mathbb{C}^3$ is a constant vector. The characteristic polynomial of the exponent is

$$\det\left[k_x\mathbf{A} + k_y\mathbf{B} - i\mathbf{Q} - \omega\mathbf{I}\right] = 0, \tag{E.4a}$$

$$\downarrow$$

$$(i - \omega)\left(\omega^2 - |k|^2 - i\left[\omega + (rk_x + sk_y)\right]\right) = 0, \tag{E.4b}$$

where $|k| = \sqrt{k_x^2 + k_y^2}$, and its solutions are

$$\omega_1 = i, \tag{E.5a}$$

$$\omega_{2,3} = \frac{i}{2}\left(1 \pm \sqrt{1 + 4\left[i(rk_x + sk_y) - |k|^2\right]}\right). \tag{E.5b}$$

So long as $rk_x + sk_y \neq 0$ and $|k|^2 \neq 1/4$, all three eigenvalues are distinct. In such a case, this matrix is diagonalizable, with right $\mathbf{R}$ and left $\mathbf{L}$ eigenvector matrices such that

$$k_x\mathbf{A} + k_y\mathbf{B} - i\mathbf{Q} = \mathbf{R\Omega L}, \tag{E.6}$$

where $\mathbf{\Omega} = \mathrm{diag}(w_1, w_2, w_3)$ and, with $\Delta k = (k_x - k_y) - i(r - s)$, $\mathbf{L} = \mathbf{R}^{-1}$,

$$\mathbf{R} = \begin{pmatrix} 0 & k_x + k_y & k_x + k_y \\ k_x & \omega_2 - \frac{k_y}{\omega_3}\Delta k & \omega_3 - \frac{k_y}{\omega_2}\Delta k \\ -k_y & \omega_2 + \frac{k_x}{\omega_3}\Delta k & \omega_3 + \frac{k_x}{\omega_2}\Delta k \end{pmatrix}, \tag{E.7a}$$

$$\mathbf{L} = \frac{1}{|k|^2}\begin{pmatrix} 0 & \left(1 + \frac{k_x\Delta k}{\omega_2\omega_3}\right) & -\left(1 - \frac{k_y\Delta k}{\omega_2\omega_3}\right) \\ -\frac{\omega_3}{\omega_2 - \omega_3} & \frac{k_x}{\omega_2 - \omega_3} & \frac{k_y}{\omega_2 - \omega_3} \\ \frac{\omega_2}{\omega_2 - \omega_3} & -\frac{k_x}{\omega_2 - \omega_3} & -\frac{k_y}{\omega_2 - \omega_3} \end{pmatrix}. \tag{E.7b}$$

The solution of the ODE for $\mathbf{v}(t)$ can be written as

$$\mathbf{v}(t) = \exp\left[it(\mathbf{R\Omega L})\right]\mathbf{u}_0 = \mathbf{R}\exp\left(it\mathbf{\Omega}\right)\mathbf{L}\mathbf{u}_0. \tag{E.8}$$

Inserting this into (E.1) gives the general solution at time $t$:

$$\mathbf{u}(x, y, t) = \Re\{\mathbf{R}\exp\left(it\mathbf{\Omega}\right)\mathbf{L}\mathbf{u}_0\exp(-i[k_x x + k_y y])\}, \tag{E.9}$$

with the initial condition $\mathbf{u}_0$, which we define by (4.145) on page 266.

# BIBLIOGRAPHY

# BIBLIOGRAPHY

[AG87] S. Allmaras and M. Giles. A second order flux split scheme for the unsteady 2-D Euler equations on arbitrary meshes. In *8th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawai; USA, June 9–11, 1987. AIAA Paper 1987-1119.

[AH99] R. K. Agarwal and D. W. Halt. A compact high-order unstructured grids method for the solution of Euler equations. *International Journal for Numerical Methods in Fluids*, 31(1):121–147, 1999.

[All89] S. R. Allmaras. *A Coupled Euler/Navier–Stokes Algorithm for 2-D Unsteady Transonic Shock/Boundary-Layer Interaction*. PhD thesis, Massachusetts Institute of Technology, 1989.

[AM07] R. Abgrall and F. Marpeau. Residual distribution schemes on quadrilateral meshes. *Journal of Scientific Computing*, 30(1):131–175, 2007.

[AR97] M. Arora and P. L. Roe. Characteristic-based numerical algorithms for stiff hyperbolic relaxation systems. In P. Kutler, J. Flores, and J.-J. Chattot, editors, *Fifteenth International Conference on Numerical Methods in Fluid Dynamics: Proceedings of the Conference Held in Monterey, CA, USA, 24-28 June 1996*, Lecture Notes in Physics, Volume 490. Springer-Verlag, Berlin, 1997.

[AR98] M. Arora and P. L. Roe. Issues and strategies for hyperbolic problems with stiff source terms. In V. Venkatakrishnan, M. D. Salas, and S. R. Chakravarthy, editors, *Barriers and Challenges in Computational Fluid Dynamics*, ICASE/LaRC Interdisciplinary Series in Science and Engineering, pages 139–154. Kluwer Academic Publishers, Dordrecht, 1998.

[Ari01] V. V. Aristov. *Direct Methods for Solving the Boltzmann Equation and Study of Nonequilibrium Flows*. Fluid Mechanics and Its Applications, Volume 60. Kluwer Academic Publishers, Dordrecht, 2001.

[ARL87] J. Allègre, M. Raffin, and J. C. Lengrand. Experimental flowfields around NACA 0012 airfoils located in subsonic and supersonic rarefied air streams. In M. O. Bristeau, R. Glowinski, J. Periaux, and H. Viviand, editors, *Numerical Simulation of Compressible Navier–Stokes Flows: A GAMM-Workshop*, Notes on Numerical Fluid Mechanics, Volume 18, pages 59–68. Friedrick Vieweg & Sohn, Braunschweig, 1987.

[ARS97] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2-3):151–167, 1997.

[ARW95] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.

[AS96] H. L. Atkins and C.-W. Shu. Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations. In *2nd AIAA and CEAS Aeroacoustics Conference*, State College, Pennsylvania; USA, May 6–8, 1996. AIAA Paper 1996-1683.

[AS98] H. L. Atkins and C.-W. Shu. Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations. *AIAA Journal*, 36(5):775–782, 1998.

[AYB01] R. K. Agarwal, K.-Y. Yun, and R. Balakrishnan. Beyond Navier–Stokes: Burnett equations for flows in the continuum-transition regime. *Physics of Fluids*, 13(10):3061–3085, 2001.

[Bal04] R. Balakrishnan. An approach to entropy consistency in second-order hydrodynamic equations. *Journal of Fluid Mechanics*, 503:201–245, 2004.

[Bar90] T. J. Barth. On unstructured grids and solvers. Technical report, von Karman Institue for Fluid Dynamics, 1990. Computational Fluid Dynamics, Lecture Series 1990-03.

[Bar93] T. J. Barth. Recent developments in high order $k$-exact reconstruction on unstructured meshes. In *31st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada; USA, Jan. 11–14, 1993. AIAA Paper 1993-0668.

[BB95] M. Borrel and B. Berde. Moment approach for the Navier–Stokes equations. In *12th AIAA Computational Fluid Dynamics Conference*, San Diego, Califorinia; USA, June 19–22, 1995. AIAA Paper 1995-1663.

[BB98] B. Berde and M. Borrel. Numerical experiments on the accuracy of a discontinuous Galerkin method for the Euler equations. *Aerospace Science and Technology*, 2(5):279–288, 1998.

[BDF94] R. Biswas, K. D. Devine, and J. E. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14(1-3):255–283, 1994.

[BF90] T. J. Barth and O. P. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. In *28th AIAA Aerospace Sciences Meeting*, Reno, Nevada; USA, Jan. 8–11, 1990. AIAA Paper 1990-0013.

[BGK54] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3):511–525, 1954.

[Bir63] G. A. Bird. Approach to translational equilibrium in a rigid sphere gas. *Physics of Fluids*, 6(10):1518–1519, 1963.

[Bir94] G. A. Bird. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford University Press, New York, 1994.

[BJ89] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *27th AIAA Aerospace Sciences Meeting*, Reno, Nevada; USA, 1989. AIAA-1989-0366.

[BO91] K. Bey and J. T. Oden. A Runge–Kutta discontinuous finite element method for high speed flows. In *10th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawai; USA, 1991. AIAA Paper 1991-1575.

[Bob82] A. V. Bobylev. The Chapman–Enskog and Grad methods for solving the Boltzmann equation. *Soviet Physics Doklady*, 27(1):29–31, 1982.

[Bon99] D. L. Bonhaus. A higher order accurate finite element method for viscous compressible flows. In *37th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada; USA, Jan. 11–14, 1999. AIAA Paper 1999-0780.

[BR97] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 131(2):267–279, 1997.

[BRG95] S. L. Brown, P. L. Roe, and C. P. T. Groth. Numerical solution of a 10-moment model for nonequilibrium gasdynamics. In *12th AIAA Computational Fluid Dynamics Conference*, San Diego, California; USA, June 19–22, 1995. AIAA Paper 1995-1677.

[Bro96] S. L. Brown. *Approximate Riemann Solvers for Moment Models of Dilute Gases*. PhD thesis, The University of Michigan, 1996.

[BS97] F. Bereux and L. Sainsaulieu. A Roe-type Riemann solver for hyperbolic systems with relaxation based on time-dependent wave decomposition. *Numerische Mathematik*, 77(2):143–185, 1997.

[BS02] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, New York, second edition, 2002.

[Bur36] D. Burnett. The distribution of molecular velocities and the mean motion in a non-uniform gas. *Proceedings of the London Mathematical Society*, 40(2):382–435, 1936.

[BY89] P. Bar-Yoseph. Space-time discontinuous finite element approximations for multi-dimensional nonlinear hyperbolic systems. *Computational Mechanics*, 5(2):145–160, 1989.

[BYE90] P. Bar-Yoseph and D. Elata. An efficient $L_2$ Galerkin finite element method for multi-dimensional non-linear hyperbolic systems. *International Journal for Numerical Methods in Engineering*, 29(6):1229–1245, 1990.

[Cap05] G. Capdeville. A new category of Hermitian upwind schemes for computational acoustics. *Journal of Computational Physics*, 210(1):133–170, 2005.

[Cap06] G. Capdeville. A new category of Hermitian upwind schemes for computational acoustics – II. Two-dimensional aeroacoustics. *Journal of Computational Physics*, 217(2):530–562, 2006.

[Cap07] G. Capdeville. Fifth-order Hermitian schemes for computational linear aeroacoustics. *International Journal for Numerical Methods in Fluids*, 55(9):815–865, 2007.

[CC70] S. Chapman and T. G. Cowling. *The Mathematical Theory of Non-Uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*. Cambridge University Press, Cambridge, third edition, 1970.

[CC87] G. Chavent and B. Cockburn. Consistance et stabilite des schemas LRG pour les lois de conservation scalaires. Technical Report RR-0710, INRIA, July 1987.

[CC89] G. Chavent and B. Cockburn. The local projection $P^0$-$P^1$-discontinuous-Galerkin finite element method for scalar conservation laws. *Mathematical Modelling and Numerical Analysis*, 23(4):565–592, 1989.

[CdLBL97] R. Carpentier, A. de La Bourdonnaye, and B. Larrouturou. On the derivation of the modified equation for the analysis of linear numerical methods. *Mathematical Modelling and Numerical Analysis*, 31(4):459–470, 1997.

[CDPR95] J.-C. Carette, H. Deconinck, H. Paillere, and P. L. Roe. Multidimensional upwinding: its relation to finite elements. *International Journal for Numerical Methods in Fluids*, 20(8-9):935–955, 1995.

[CFL28] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[CFL67] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11(2):215–234, 1967.

[CG96] M. H. Carpenter and D. Gottlieb. Spectral methods on arbitrary grids. *Journal of Computational Physics*, 129(1):74–86, 1996.

[Cha90] S.-C. Chang. A critical analysis of the modified equation technique of Warming and Hyett. *Journal of Computational Physics*, 86(1):107–126, 1990.

[CHS90] B. Cockburn, S. Hou, and C.-W. Shu. The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Mathematics of Computation*, 54(190):545–581, 1990.

[CJR97] R. E. Caflisch, S. Jin, and G. Russo. Uniformly accurate schemes for hyperbolic systems with relaxation. *SIAM Journal on Numerical Analysis*, 34(1):246–281, 1997.

[CLL94] G.-Q. Chen, C. D. Levermore, and T.-P. Liu. Hyperbolic conservation laws with stiff relaxation terms and entropy. *Communications on Pure and Applied Mathematics*, 47(6):787–830, 1994.

[CLS89] B. Cockburn, S.-Y. Lin, and C.-W. Shu. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, 1989.

[CP95] W. J. Coirier and K. G. Powell. An accuracy assessment of Cartesian-mesh approaches for the Euler equations. *Journal of Computational Physics*, 117(1):121–131, 1995.

[CS89] 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.

[CS91] B. Cockburn and C.-W. Shu. The Runge–Kutta local projection $P^1$-discontinuous-Galerkin finite element method for scalar conservation laws. *Mahematical Modelling and Numerical Analysis*, 25(3):337–361, 1991.

[CS98] B. Cockburn and C.-W. Shu. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998.

[CS01] B. Cockburn and C.-W. Shu. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[CW84] P. Colella and P. R. Woodward. The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, 54(1):174–201, 1984.

[DH03] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems.* John Wiley & Sons, Chichester, 2003.

[DH05] D. L. Darmofal and R. Haimes. Towards the next generation in CFD. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada; USA, Jan. 10–13, 2005. AIAA Paper 2005-0087.

[DL99] M. Delanaye and Y. Liu. Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids. In *14th AIAA Computational Fluid Dynamics Conference*, Norfolk, Virginia; USA, June 28–July 1, 1999. AIAA Paper 1999-3259.

[DM05] M. Dumbser and C.-D. Munz. ADER discontinuous Galerkin schemes for aeroacoustics. *Comptes Rendus Mecanique*, 333(9):683–687, 2005.

[DM06] M. Dumbser and C.-D. Munz. Building blocks for arbitrary high order discontinuous Galerkin schemes. *Journal of Scientific Computing*, 27(1-3):215–230, 2006.

[DRS93] H. Deconinck, P. L. Roe, and R. Struijs. A multidimensional generalization of Roe's flux difference splitter for the Euler equations. *Computers & Fluids*, 22(2-3):215–222, 1993.

[DvL03] C. Depcik and B. van Leer. In search of an optimal local Navier–Stokes preconditioner. In *16th AIAA Computational Fluid Dynamics Conference*, Orlando, Florida; USA, June 23–26, 2003. AIAA Paper 2003-3703.

[DZ93] D. L. De Zeeuw. *A Quadtree-Based Adaptively-Refined Cartesian-Grid Algorithm for Solution of the Euler Equations*. PhD thesis, The University of Michigan, 1993.

[Eka05] J. A. Ekaterinaris. High-order accurate, low numerical diffusion methods for aerodynamics. *Progress in Aerospace Sciences*, 41(3-4):192–300, 2005.

[Eu92] B. C. Eu. *Kinetic Theory and Irreversible Thermodynamics*. John Wiley & Sons, New York, 1992.

[FBC$^+$01] J. Fan, I. D. Boyd, C.-P. Cai, K. Hennighausen, and G. V. Candler. Computation of rarefied gas flows around a NACA 0012 airfoil. *AIAA Journal*, 39(4):618–625, 2001.

[FFS03] M. Feistauer, J. Felcman, and I. Straškraba. *Mathematical and Computational Methods for Compressible Flow*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2003.

[Fle91] C. A. J. Fletcher. *Computational Techniques for Fluid Dynamics, Volume 1: Fundamental and General Techniques*. Springer Series in Computational Physics. Springer-Verlag, Berlin, second edition, 1991.

[Foc73] J. D. Foch, Jr. On higher order hydrodynamic theories of shock structure. In E. G. D. Cohen and W. Thirring, editors, *The Boltzmann equation: Theory and Applications, Proceedings of the International Symposium "100 Years Boltzmann Equation" in Vienna, 4th–8th September 1972*, pages 123–140. Springer-Verlag, Wien, 1973.

[Fri98] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, 144(1):194–212, 1998.

[Fuj05] K. Fujii. Progress and future prospects of CFD in aerospace — Wind tunnel and beyond. *Progress in Aerospace Sciences*, 41(6):455–470, 2005.

[GeH99] M. Gad-el-Hak. The fluid mechanics of microdevices — The Freeman scholar lecture. *Journal of Fluids Engineering*, 121(1):5–33, 1999.

[GM85] J. Goodman and A. Majda. The validity of the modified equation for nonlinear shock waves. *Journal of Computational Physics*, 58(3):336–348, 1985.

[Gom94] T. I. Gombosi. *Gaskinetic Theory*. Cambridge Atmospheric and Space Science Series. Cambridge University Press, Cambridge, 1994.

[Gra49] H. Grad. On the kinetic theory of rarefied gases. *Communications on Pure and Applied Mathematics*, 2(4):331–407, 1949.

[Gra52] H. Grad. The profile of a steady plane shock wave. *Communications on Pure and Applied Mathematics*, 5(3):257–300, 1952.

[GRGB95] C. P. T. Groth, P. L. Roe, T. I. Gombosi, and S. L. Brown. On the nonstationary wave structure of a 35-moment closure for rarefied gas dynamics. In *26th AIAA Fluid Dynamics Conference*, San Diego, California; USA, June 19–22, 1995. AIAA Paper 1995-2312.

[GSS86] D. F. Griffiths and J. M. Sanz-Serna. On the scope of the method of modified equations. *SIAM Journal on Scientific and Statistical Computing*, 7(3):994–1008, 1986.

[GST01] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, 2001.

[HA92] D. W. Halt and R. K. Agarwal. Compact higher order characteristic-based Euler solver for unstructured grids. *AIAA Journal*, 30(8):1993–1999, 1992.

[HEOC87] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71(2):231–303, 1987.

[Hir68] C. W. Hirt. Heuristic stability theory for finite-difference equations. *Journal of Computational Physics*, 2(4):339–355, 1968.

[Hir89] C. Hirsch. *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization.* Wiley Series in Numerical Methods in Engineering. John Wiley & Sons, Chichester, 1989.

[Hit00] J. A. Hittinger. *Foundations for the Generalization of the Godunov Method to Hyperbolic Systems with Stiff Relaxation Source Terms.* PhD thesis, The University of Michigan, 2000.

[HLvL83] A. Harten, P. D. Lax, and B. van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, 1983.

[HNW93] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems.* Springer Series in Computational Mathematics 8. Springer-Verlag, Berlin, second revised edition, 1993.

[HO87] A. Harten and S. Osher. Uniformly high-order accurate nonoscillatory schemes. I. *SIAM Journal on Numerical Analysis*, 24(2):279–309, 1987.

[HOEC86] A. Harten, S. Osher, B. Engquist, and S. R. Chakravarthy. Some results on uniformly high-order accurate essentially nonoscillatory schemes. *Applied Numerical Mathematics*, 2(3-5):347–377, 1986.

[Hol64] L. H. Holway, Jr. Existence of kinetic theory solutions to the shock structure problem. *Physics of Fluids*, 7(6):911–913, 1964.

[Hol65] L. H. Holway, Jr. Kinetic theory of shock structure using an ellipsoidal distribution function. In J. H. de Leeuw, editor, *Rarefied Gas Dynamics, Proceedings of the Fourth International Symposium on Rarefied Gas Dynamics*, volume 1, pages 193–215, New York, 1965. Academic Press.

[Hol96] M. Holt. Review of Godunov methods. Technical Report NASA Contractor Report 198322, ICASE Report No. 96-25, Institute for Computer Applications in Science and Engieering, NASA Langley Research Center, 1996.

[HR99] J. Hittinger and P. Roe. On uniformly accurate upwinding for hyperbolic systems with relaxation. In R. Vilsmeier, F. Benkhaldoun, and D. Hänel, editors, *Finite Volumes for Complex Applications II: Problems and Perspectives*, pages 357–366. Hermes Science Publications, Paris, 1999.

[HR04] J. A. F. Hittinger and P. L. Roe. Asymptotic analysis of the Riemann problem for constant coefficient hyperbolic systems with relaxation. *Zeitschrift für Angewandte Mathematik und Mechanik*, 84(7):452–471, 2004.

[HS99] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1):97–127, 1999.

[HSvL05] J. A. F. Hittinger, Y. Suzuki, and B. van Leer. Investigation of the discontinuous Galerkin method for first-order PDE approaches to CFD. In *17th AIAA Computational Fluid Dynamics Conference*, Toronto, Ontario; Canada, June 6–9, 2005. AIAA Paper 2005-4989.

[Hug00] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Dover Publications, Mineola, 2000.

[Huy03] H. T. Huynh. Analysis and improvement of upwind and centered schemes on quadrilateral and triangular meshes. In *16th AIAA Computational Fluid Dynamics Conference*, Orlando, Florida; USA, June 23–26, 2003. AIAA Paper 2003-3541.

[Huy06a] H. T. Huynh. An upwind moment scheme for conservation laws. In C. Groth and D. W. Zingg, editors, *Computational Fluid Dynamics 2004: Proceedings of the Third International Conference on Computational Fluid Dynamics, ICCFD3, Toronto, 12–16 July 2004*, pages 761–766. Springer-Verlag, Berlin, 2006.

[Huy06b] H. T. Huynh. An upwind moment scheme for conservation laws (unpublished version), 2006.

[Huy07] H. T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida; USA, June 25–28, 2007. AIAA 2007-4079.

[HW96] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer Series in Computational Mathematics 14. Springer-Verlag, Berlin, second revised edition, 1996.

[Jam91] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In *10th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawai; USA, June 24–26, 1991. AIAA Paper 1991-1596.

[Jam01] A. Jameson. A perspective on computational algorithms for aerodynamic analysis and design. *Progress in Aerospace Sciences*, 37:197–243, 2001.

[Jam04] A. Jameson. Aerodynamics. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 3, pages 325–406. John Wiley & Sons, Chichester, 2004.

[Jin95] S. Jin. Runge–Kutta methods for hyperbolic conservation laws with stiff relaxation terms. *Journal of Computational Physics*, 122(1):51–67, 1995.

[Jin99] S. Jin. Efficient Asymptotic-Preserving (AP) schemes for some multiscale kinetic equations. *SIAM Journal on Scientific Computing*, 21(2):441–454, 1999.

[JL96] S. Jin and C. D. Levermore. Numerical schemes for hyperbolic conservation laws with stiff relaxation terms. *Journal of Computational Physics*, 126(2):449–467, 1996.

[Joh87] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge, 1987.

[JP86] C. Johnson and J. Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of Computation*, 46(173):1–26, 1986.

[JS96] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.

[KAA$^+$07] V. I. Kolobov, R. R. Arslanbekov, V. V. Aristov, A. A. Frolova, and S. A. Zabelok. Unified solver for rarefied and continuum flows with adaptive mesh and algorithm refinement. *Journal of Computational Physics*, 223(2):589–608, 2007.

[KS05] G. E. Karniadakis and S. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, second edition, 2005.

[KvdVvdV06] C. M. Klaij, J. J. W. van der Vegt, and H. van der Ven. Space-time discontinuous Galerkin method for the compressible Navier–Stokes equations. *Journal of Computational Physics*, 217(2):589–611, 2006.

[KvLW05] B. Kleb, B. van Leer, and B. Wood. Matching multistage schemes to viscous flow. In *17th AIAA Computational Fluid Dynamics Conference*, Toronto, Ontario; Canada, June 6–9, 2005. AIAA Paper 2005-4708.

[Lam91] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. John Wiley & Sons, New York, 1991.

[Lan98] C. B. Laney. *Computational Gasdynamics*. Cambridge University Press, New York, 1998.

[Lax54] P. D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7(1):159–193, 1954.

[Leo94] B. P. Leonard. Note on the von Neumann stability of explicit one-dimensional advection schemes. *Computer Methods in Applied Mechanics and Engineering*, 118(1-2):29–46, 1994.

[Lev96] C. D. Levermore. Moment closure hierarchies for kinetic theories. *Journal of Statistical Physics*, 83(5-6):1021–1065, 1996.

[LeV02] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2002.

[Li06] B. Q. Li. *Discontinuous Finite Elements in Fluid Dynamics and Heat Transfer*. Computational Fluid and Solid Mechanics. Springer-Verlag, London, 2006.

[Lin98] T. J. Linde. *A Three-Dimensional Adaptive Multifluid MHD Model of the Heliosphere*. PhD thesis, The University of Michigan, 1998.

[Lin02] T. Linde. A practical, general-purpose, two-state HLL Riemann solver for hyperbolic conservation laws. *International Journal for Numerical Methods in Fluids*, 40(3-4):391–402, 2002.

[Liu87] T.-P. Liu. Hyperbolic conservation laws with relaxation. *Communications in Mathematical Physics*, 108(1):153–175, 1987.

[LM89] E. W. Larsen and J. E. Morel. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes II. *Journal of Computational Physics*, 83(1):212–236, 1989.

[LM98] C. D. Levermore and W. J. Morokoff. The Gaussian moment closure for gas dynamics. *SIAM Journal on Applied Mathematics*, 59(1):72–96, 1998.

[LM99a] R. B. Lowrie and J. E. Morel. Discontinuous Galerkin for hyperbolic systems with stiff relaxation. In B. Cockburn, G. E. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, Lecture Notes in Computational Science and Engineering, pages 385–390. Springer-Verlag, Berlin, 1999.

[LM99b] R. B. Lowrie and J. E. Morel. Discontinuous Galerkin for stiff hyperbolic systems. In *14th AIAA Computational Fluid Dynamics Conference*, Norfolk, Virginia; USA, June 28–July 1, 1999. AIAA Paper 1999-3307.

[LM02] R. B. Lowrie and J. E. Morel. Methods for hyperbolic systems with stiff relaxation. *International Journal for Numerical Methods in Fluids*, 40(3-4):413–423, 2002.

[LMM87] E. W. Larsen, J. E. Morel, and W. F. Miller, Jr. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes. *Journal of Computational Physics*, 69(2):283–324, 1987.

[LMN98] C. D. Levermore, W. J. Morokoff, and B. T. Nadiga. Moment realizability and the validity of the Navier–Stokes equations for rarefied gas dynamics. *Physics of Fluids*, 10(12):3214–3226, 1998.

[LOC94] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212, 1994.

[Low96] R. B. Lowrie. *Compact Higher-Order Numerical Methods for Hyperbolic Conservation Laws*. PhD thesis, The University of Michigan, 1996.

[LR74] P. LeSaint and P. A. Raviart. On a finite element method for solving the neutron transport equation. In C. De Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89–123. Academic Press, New York, 1974.

[LRR00] S. F. Liotta, V. Romano, and G. Russo. Central schemes for balance laws of relaxation type. *SIAM Journal on Numerical Analysis*, 38(4):1337–1356, 2000.

[LRvL95] R. B. Lowrie, P. L. Roe, and B. van Leer. A space-time discontinuous Galerkin method for the time-accurate numerical solution of hyperbolic conservation laws. In *12th AIAA Computational Fluid Dynamics Conference*, San Diego, California; USA, June 19–22, 1995. AIAA Paper 1995-1658.

[LRvL98] R. B. Lowrie, P. L. Roe, and B. van Leer. Properties of space-time discontinuous Galerkin. Technical Report LA-UR-98-5561, Los Alamos National Laboratory, 1998.

[LSTZ07] Y. Liu, C.-W. Shu, E. Tadmor, and M. Zhang. Central discontinuous Galerkin methods on overlapping cells with a nonoscillatory hierarchical reconstruction. *SIAM Journal on Numerical Analysis*, 45(6):2442–2467, 2007.

[LVW06] Y. Liu, M. Vinokur, and Z. J. Wang. Spectral difference method for unstructured grids I: Basic formulation. *Journal of Computational Physics*, 216(2):780–801, 2006.

[LW64] P. D. Lax and B. Wendroff. Difference schemes for hyperbolic equations with high order of accuracy. *Communications on Pure and Applied Mathematics*, 17(3):381–398, 1964.

[Mav07] D. J. Mavriplis. Unstructured mesh discretizations and solvers for computational aerodynamics. In *18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida; USA, June 25–28, 2007. AIAA Paper 2007-3955.

[MG05] J. G. McDonald and C. P. T. Groth. Numerical modeling of micron-scale flows using the Gaussian moment closure. In *35th AIAA Fluid Dynamics Conference and Exhibit*, Toronto, Ontario; Canada, June 6–9, 2005. AIAA Paper 2005-5035.

[Mor06] K. Morinishi. Numerical simulation for gas microflows using Boltzmann equation. *Computers & Fluids*, 35(8-9):978–985, 2006.

[MR98] I. Müller and T. Ruggeri. *Rational Extended Thermodynamics.* Springer Tracts in Natural Philosophy, volume 37. Springer-Verlag, New York, second edition, 1998.

[Myo01] R. S. Myong. A computational method for Eu's generalized hydrodynamic equations of rarefied and microscale gasdynamics. *Journal of Computational Physics*, 168(1):47–72, 2001.

[Nat98] R. Natalini. Recent results on hyperbolic relaxation problems. In H. Freistühler, editor, *Analysis of Systems of Conservation Laws*, Monographs and Surveys in Pure and Applied Mathematics, Volume 99, pages 128–198. Chapman & Hall/CRC, Boca Raton, 1998.

[NP00] G. Naldi and L. Pareschi. Numerical schemes for hyperbolic systems of conservation laws with stiff diffusive relaxation. *SIAM Journal on Numerical Analysis*, 37(4):1246–1270, 2000.

[NvL03] H. Nishikawa and B. van Leer. Optimal multigrid convergence by elliptic/hyperbolic splitting. *Journal of Computational Physics*, 190(1):52–63, 2003.

[OGvA02] C. Ollivier-Gooch and M. van Altena. A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics*, 181(2):729–752, 2002.

[OOC98] E. S. Oran, C. K. Oh, and B. Z. Cybyk. Direct simulation Monte Carlo: Recent advances and applications. *Annual Review of Fluid Mechanics*, 30:403–441, 1998.

[Par98] L. Pareschi. Characteristic-based numerical schemes for hyperbolic systems with nonlinear relaxation. *Rendiconti del Circolo Matematico di Palermo (2)*, 57:375–380, 1998.

[Pat84] A. T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of Computational Physics*, 54(3):468–488, 1984.

[Pem93a] R. B. Pember. Numerical methods for hyperbolic conservation laws with stiff relaxation I. Spurious solutions. *SIAM Journal on Applied Mathematics*, 53(5):1293–1330, 1993.

[Pem93b] R. B. Pember. Numerical methods for hyperbolic conservation laws with stiff relaxation II. Higher-order Godunov methods. *SIAM Journal on Scientific Computing*, 14(4):824–859, 1993.

[Pet91] T. E. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM Journal on Numerical Analysis*, 28(1):133–140, 1991.

[PR03] L. Pareschi and G. Russo. High order asymptotically strong-stability-preserving methods for hyperbolic systems with stiff relaxation. In T. Y. Hou and E. Tadmor, editors, *Hyperbolic Problems: Theory, Numerics, Applications: Proceedings of the Ninth International Conference on Hyperbolic Problems*, pages 241–251. Springer, New York, 2003.

[PR05] L. Pareschi and G. Russo. Implicit-explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation. *Journal of Scientific Computing*, 25(1):129–155, 2005.

[QDS05] 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(42-44):4528–4543, 2005.

[Ram94] J. D. Ramshaw. Numerical viscosities of difference schemes. *Communications in Numerical Methods in Engineering*, 10(11):927–931, 1994.

[RH73] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[RH01] P. L. Roe and J. A. F. Hittinger. Toward Godunov-type methods for hyperbolic conservation laws with stiff relaxation. In E. T. Toro, editor, *Godunov Methods: Theory and Applications*, pages 725–744. Kluwer Academic/Plenum Publishers, New York, 2001.

[RL02] W. J. Rider and R. B. Lowrie. The use of classical Lax–Friedrichs Riemann solvers with discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 40(3-4):479–486, 2002.

[RM67] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial-Value Problems*. Interscience Publishers, New York, second edition, 1967.

[RNK02] P. L. Roe, H. Nishikawa, and K. Kabin. Toward a general theory of local preconditioning. In *32nd AIAA Fluid Dynamics Conference and Exhibit*, St. Louis, Missouri; USA, June 24–26, 2002. AIAA Paper 2002-2956.

[Roe81] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.

[Roe82] P. L. Roe. Fluctuations and signals — a framework for numerical evolution problems. In K. W. Morton and M. J. Baines, editors, *Numerical Methods for Fluid Dynamics*, pages 219–257. Academic Press, New York, 1982.

[Roe86] P. L. Roe. Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics. *Journal of Computational Physics*, 63(2):458–476, 1986.

[Roe05a] P. L. Roe. Computational fluid dynamics — retrospective and prospective. *International Journal of Computational Fluid Dynamics*, 19(8):581–594, 2005.

[Roe05b] P. L. Roe. Principles of Computational Fluid Dynamics, February 2, 2005. Course Note.

[Rot71] D. E. Rothe. Electron-beam studies of viscous flow in supersonic nozzles. *AIAA Journal*, 9(5):804–811, 1971.

[Rus62] V. V. Rusanov. The calculation of the interaction of non-stationary shock waves and obstacles. *USSR Computational Mathematics and Mathematical Physics*, 1(2):304–320, 1962.

[Rus02] G. Russo. Central schemes and systems of balance laws. In A. Meister and J. Struckmeier, editors, *Hyperbolic Partial Differential Equations: Theory, Numerics and Applications*, pages 59–114. Friedrich Vieweg & Sohn Verlag, Braunschweig, 2002.

[Sal07] M. D. Salas. A review of hypersonics aerodynamics, aerothermodynamics and plasmadynamics activities within NASA's fundamental aeronautics program. In *39th AIAA Thermophysics Conference*, Miami, Florida; USA, June 25–28, 2007. AIAA 2007-4264.

[SB02] Q. Sun and I. D. Boyd. A direct simulation method for subsonic, microscale gas flows. *Journal of Computational Physics*, 179(2):400–425, 2002.

[Sch91] W. E. Schiesser. *The Numerical Method of Lines: Integration of Partial Differential Equations.* Academic Press, San Diego, 1991.

[Sha93] M. Sh. Shavaliyev. Super-Burnett corrections to the stress tensor and the heat flux in a gas of Maxwellian molecules. *Journal of Applied Mathematics and Mechanics*, 57(3):573–576, 1993.

[Shu87] C.-W. Shu. TVB uniformly high-order schemes for conservation laws. *Mathematics of Computation*, 49(179):105–121, 1987.

[Shu03] C.-W. Shu. High-order finite difference and finite voume WENO schemes and discontinuous Galerkin methods for CFD. *International Journal of Computational Fluid Dynamics*, 17(2):107–118, 2003.

[SO88] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

[Str05] H. Struchtrup. *Macroscopic Transport Equations for Rarefied Gas Flows: Approximation Methods in Kinetic Theory.* Interaction of Mechanics and Mathematics Series. Springer-Verlag, Berlin, 2005.

[SvL05] Y. Suzuki and B. van Leer. Application of the 10-moment model to MEMS flows. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada; USA, Jan. 10–13, 2005. AIAA Paper 2005-1398.

[SvL06] Y. Suzuki and B. van Leer. A discontinuous Galerkin method with Hancock-type time integration for hyperbolic systems with stiff relaxation source terms. In *The Fourth International Conference on Computational Fluid Dynamics, ICCFD4*, Ghent; Belgium, July 10–14, 2006.

[SvL07] Y. Suzuki and B. van Leer. An analysis of the upwind moment scheme and its extension to systems of nonlinear hyperbolic-relaxation equations. In *18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida; USA, June 25–28, 2007. AIAA 2007-4468.

[SW04] Y. Sun and Z. J. Wang. Evaluation of discontinuous Galerkin and spectral volume methods for scalar and system conservation laws on unstructured grids. *International Journal for Numerical Methods in Fluids*, 45(8):819–838, 2004.

[SWL06] Y. Sun, Z. J. Wang, and Y. Liu. Spectral (finite) volume method for conservation laws on unstructured grids VI: Extension to viscous flow. *Journal of Computational Physics*, 215(1):41–58, 2006.

[TAP97] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer.* Series in Computational and Physical Processes in Mechanics and Thermal Sciences. Taylor & Francis, Philadelphia, second edition, 1997.

[Tch06] F. G. Tcheremissine. Solution to the Boltzmann kinetic equation for high-speed flows. *Computational Mathematics and Mathematical Physics*, 46(2):315–329, 2006.

[TMN01] E. F. Toro, R. C. Millington, and L. A. M. Nejad. Towards very high order Godunov schemes. In E. F. Toro, editor, *Godunov Methods: Theory and Applications*, pages 907–940. Kluwer Academic/Plenum Publishers, New York, 2001.

[TS04] M. Torrilhon and H. Struchtrup. Regularized 13-moment equations: shock structure calculations and comparison to Burnett models. *Journal of Fluid Mechanics*, 513:171–198, 2004.

[TSL00] R. Tyson, L. G. Stern, and R. J. LeVeque. Fractional step methods applied to a chemotaxis model. *Journal of Mathematical Biology*, 41(5):455–475, 2000.

[Tur99] E. Turkel. Preconditioning techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 31:385–416, 1999.

[UVGC00] F. J. Uribe, R. M. Velasco, and L. S. García-Colín. Bobylev's instability. *Physical Review E*, 62(4):5835–5838, 2000.

[VAKJ03] V. Venkatakrishnan, S. R. Allmaras, D. S. Kamenetskii, and F. T. Johnson. Higher order schemes for the compressible Navier–Stokes equations. In *16th AIAA Computational Fluid Dynamics Conference*, Orlando, Florida; USA, June 23–26, 2003. AIAA Paper 2003-3987.

[Var62] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, 1962.

[vAvLR82] G. D. van Albada, B. van Leer, and W. W. Roberts, Jr. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics*, 108(1):76–84, 1982.

[vdVvdV02] J. J. W. van der Vegt and H. van der Ven. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows I. General formulation. *Journal of Computational Physics*, 182(2):546–585, 2002.

[Ven95] V. Venkatakrishnan. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics*, 118(1):120–130, 1995.

[VK86] W. G. Vincenti and C. H. Kruger, Jr. *Introduction to Physical Gas Dynamics.* Krieger Publishing Company, Malabar, Florida, 1986.

[vL69] B. van Leer. Stabilization of difference schemes for the equations of inviscid compressible flow by artificial diffusion. *Journal of Computational Physics*, 3(4):473–485, 1969.

[vL77] B. van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics*, 23(3):276–299, 1977.

[vL79] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979.

[vL06] B. van Leer. Upwind and high-resolution methods for compressible flow: From donor cell to residual-distribution schemes. *Communications in Computational Physics*, 1(2):192–205, 2006.

[vLD99] B. van Leer and D. Darmofal. Steady Euler solutions in $O(N)$ operations. In E. Dick, K. Riemslagh, and J. Vierendeels, editors, *Multigrid Methods VI: Proceedings of the Sixth European Multigrid Conference Held in Gent, Belgium, September 27–30, 1999*, Lecture Notes in Computational Science and Engineering, Volume 14, pages 24–33, Berlin, 1999. Springer-Verlag.

[vLLR91] B. van Leer, W.-T. Lee, and P. L. Roe. Characteristic time-stepping or local preconditioning of the Euler equations. In *10th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawai; USA, June 24–27, 1991. AIAA Paper 1991-1552.

[vLLvR07] B. van Leer, M. Lo, and M. van Raalte. A discontinuous Galerkin method for diffusion based on recovery. In *18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida; USA, June 25–28, 2007. AIAA Paper 2007-4083.

[vLN05] B. van Leer and S. Nomura. Discontinuous Galerkin for diffusion. In *17th AIAA Computational Fluid Dynamics Conference*, Toronto, Ontario; Canada, June 6–9, 2005. AIAA Paper 2005-5108.

[vNR47] J. von Neumann and R. D. Richtmyer. On the numerical solution of partial differential equations of parabolic type. Technical Report LA-657, Los Alamos Scientific Laboratory, December 25, 1947.

[vNR63] J. von Neumann and R. D. Richtmyer. On the numerical solutions of partial differential equations of parabolic type. In A. H. Taub, editor, *John von Neumann Collected Works, Volume V: Design of Computers, Theory of Automata and Numerical Analysis*, pages 652–663. Pergamon Press Book, New York, 1963.

[Wag92] W. Wagner. A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation. *Journal of Statistical Physics*, 66(3-4):1011–1044, 1992.

[Wan02] Z. J. Wang. Spectral (finite) volume method for conservation laws on unstructured grids. Basic formulation. *Journal of Computational Physics*, 178(1):210–251, 2002.

[Wan07] Z. J. Wang. High-order methods for the Euler and Navier–Stokes equations on unstructured grids. *Progress in Aerospace Sciences*, 43:1–41, 2007.

[WB76] R. F. Warming and R. M. Beam. Upwind second-order difference schemes and applications in aerodynamic flows. *AIAA Journal*, 14(9):1241–1249, 1976.

[WC48] C. S. Wang Chang. On the theory of the thickness of weak shock waves. Technical report, Departemnt of Engineering Research, The University of Michigan, August 19, 1948. APL/JHU CM-503, UMH-3-F.

[Wei95] W. Weiss. Continuous shock structure in extended thermodynamics. *Physical Review E*, 52(6):R5760–R5763, 1995.

[WH74] R. F. Warming and B. J. Hyett. The modified equation approach to the stability and accuracy analysis of finite-difference methods. *Journal of Computational Physics*, 14(2):159–179, 1974.

[Whi91] F. M. White. *Viscous Fluid Flow*. McGraw-Hill Series in Mechanical Engineering. McGraw-Hill, New York, second edition, 1991.

[ZMC93] X. Zhong, R. W. MacCormack, and D. R. Chapman. Stabilization of the Burnett equations and application to hypersonic flows. *AIAA Journal*, 31(6):1036–1043, 1993.

[ZS05] 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(4-5):581–592, 2005.

[ZTSP03] O. C. Zienkiewicz, R. L. Taylor, S. J. Sherwin, and J. Peiró. On discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering*, 58(8):1119–1148, 2003.

[ZTZ05] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier Butterworth-Heinemann, Burlington, sixth edition, 2005.