University of New Mexico
UNM Digital Repository

Mathematics & Statistics ETDs Electronic Theses and Dissertations

Summer 7-17-2017

High Order Hermite and Sobolev Discontinuous Galerkin Methods for Hyperbolic Partial Differential Equations

Adeline Kornelus
University of New Mexico

Follow this and additional works at: https://digitalrepository.unm.edu/math_etds

Part of the Applied Mathematics Commons, Mathematics Commons, and the Statistics and Probability Commons

Recommended Citation
Kornelus, Adeline. "High Order Hermite and Sobolev Discontinuous Galerkin Methods for Hyperbolic Partial Differential Equations." (2017). https://digitalrepository.unm.edu/math_etds/114

This Dissertation is brought to you for free and open access by the Electronic Theses and Dissertations at UNM Digital Repository. It has been accepted for inclusion in Mathematics & Statistics ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.

Adeline Kornelus

Candidate

Applied Mathematics

Department of Mathematics and Statistics

This dissertation is approved, and it is acceptable in quality and form for publication.

Approved by the Dissertation Committee:

Prof. Daniel Appelö, Chair

Prof. Stephen Lau, Member

Prof. Jens Lorenz, Member

Prof. Thomas Hagstrom, Member

High Order Hermite and Sobolev Discontinuous Galerkin Methods for Hyperbolic Partial Differential Equations

by

Adeline Kornelus

B.S., Applied Mathematics, California State Polytechnic University
Pomona, 2010

M.S., Applied Mathematics, University of New Mexico, 2013

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Mathematics

The University of New Mexico

Albuquerque, New Mexico

July 2017

Dedication

To my sister, Angeline and my parents, David and Dian,
for their support and encouragement along the way.

Acknowledgments

I would like to thank my advisor, Professor Daniel Appelö, for his support and guidance. I would also like to thank fellow graduate students at UNM, especially Oleksii Beznosov who gave me solid computer support.

I want to express my gratitude to Kevin Burns for proofreading this thesis.

This work was supported in part by NSF Grant DMS-1319054. Any conclusions or recommendations expressed in this thesis are those of the author and do not necessarily reflect the views of NSF.

High Order Hermite and Sobolev Discontinuous Galerkin Methods for Hyperbolic Partial Differential Equations

by

Adeline Kornelus

B.S., Applied Mathematics, California State Polytechnic University
Pomona, 2010
M.S., Applied Mathematics, University of New Mexico, 2013
Ph.D., Mathematics, University of New Mexico, 2017

Abstract

Many real-world problems involving dynamics of solid or fluid bodies can be modeled by hyperbolic partial differential equations (PDEs). Up to this point, only solutions to selected PDEs are available. Many PDEs are physically or geometrically complex, resulting in difficulties computing the analytical solutions. In this thesis, we focus on numerical methods for approximating solutions to hyperbolic PDEs. Long-term simulation for the motion of the body described by the PDE requires a method that is not only robust and efficient, but also produces small error because the error will be propagated and accumulated over the course of the simulation. Therefore, we use the so-called high order methods. In particular, we study Hermite methods and Sobolev Discontinuous Galerkin methods. This thesis describes my contribution to two high order methods, which includes the development of flux-conservative Hermite methods for nonlinear PDEs of conservation law type and Sobolev Discontinuous Galerkin methods for general hyperbolic PDEs.

Contents

List of Figures

List of Tables

Glossary

CFL	Courant-Friedrichs-Lewy
DoF	Degree of freedom
m	The number of derivatives used in the Hermite methods
ODE	Ordinary differential equation
PDE	Partial differential equation
Δt	Grid size in temporal variable
h	Grid size in spatial variable

Chapter 1

Introduction

My dissertation work concerns numerical methods to approximate solutions to hyperbolic partial differential equations (PDEs). Hyperbolic PDEs describe wave propagation and particle transport arising in science and engineering. Some practical applications of wave propagation are echolocation, electromagnetic waves in technology, radar, and seismic natural disasters.

Some nocturnal animals such as microchiropteran bats, toothed whales and dolphins, a few birds, and some species of shrews use echolocation to navigate and forage. In echolocation, the animals emit sound waves to the environment and when the waves hit an object, they induce echoes. The echoes are then used to identify the location of the objects in their vicinity, which is particularly useful when hunting for food.

To improve safety in air traffic, control systems rely on radar to determine the location and velocity of nearby planes or objects. Radar is the reason why most planes manage to arrive safely at their destinations despite the large number of other planes; it also mitigates the effects of bad weather. Thanks to radar, we have fewer plane crashes because the planes are able to maintain a safe distance from

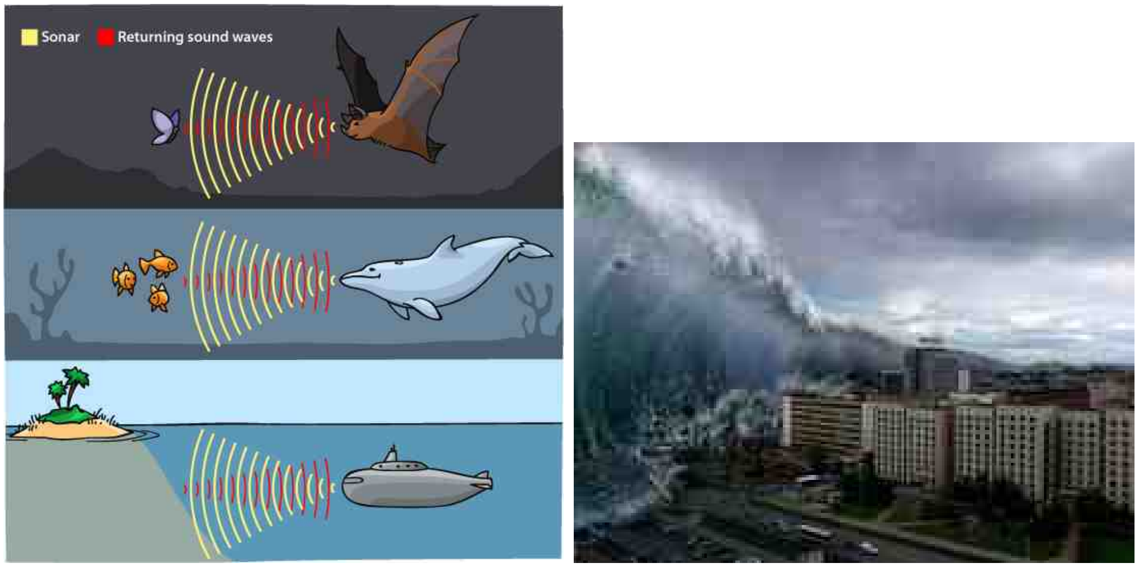


Figure 1.1: Top: bat echocolocation, Courtesy of Arizona Board of Regents/ ASU Ask A Biologist, source: `https://askabiologist.asu.edu/echolocation`, bottom: tsunami wave, Courtesy of Pinterest, source: `https://www.pinterest.com/pin/388436155101247 79`.

other planes, buildings, and mountains.

Electromagnetic waves are used in human's lives, for example in communication and medical science. Radio, telephone and wireless internet all rely on the transfer of information via electromagnetic waves. Electromagnetic waves are also used in the form of x-rays. X-rays are used to examine a patient's body for the purpose of diagnosis, monitoring, or treatment of medical conditions. Since X-rays penetrate tissues and bones, they can be used to produce an image of internal organs or the structure of the human body. Furthermore, X-rays play an important role in cancer therapy.

Waves can also be destructive, for example in earthquakes or tsunamis, which are giant waves caused by earthquakes, volcanic eruptions, or sudden movement in the Earth's tectonic plates under the sea. In particular, my home country, Indonesia, suffered lost lives and homes of thousands of people due to a tsunami. On December 26, 2004, a powerful earthquake near the island of Sumatra triggered a tsunami and damaged the northern part of the island within 10 minutes! According to US Geological Survey Statistics, more than 200,000 people were killed or missing due to the tsunami. I hope that in the future, this kind of tragic incident can be mitigated by designing better emergency warning systems and improved building codes. As scientists, we will always try to develop better numerical methods to simulate wave propagation that in turn can be used to alleviate disasters.

Mathematically, wave propagation is modeled by hyperbolic PDEs. When simulating wave propagation problems over a long period of time, information about the shape and the amplitude of the waves must be transmitted as accurately as possible. The error in approximating the solution committed by the numerical methods will be of the form $C_1\Delta t^{q_1} + C_2 h^{q_2}$, where Δt and h are the temporal and spatial grid sizes, respectively, q_1 and q_2 are the order of accuracy in time and space respectively, C_1 and C_2 are method-specific constants. High order methods (here defined by $\min(q_1, q_2) \geq 3$) are suitable for the long-term simulation since the dispersion error (the error resulting from the inability to numerically propagate a wave at a given frequency at the physically correct speed) and the diffusion error (the error caused by numerical excess damping of a wave at a given frequency) are smaller than low order methods (defined by $\min(q_1, q_2) < 3$). Besides being high order, a good numerical method for wave propagation needs to be able to handle nonuniform wave speeds (corresponding to waves propagating in nonhomogeneous media), and complicated geometry, as the spatial domain associated with the PDE is not always geometrically simple such as a box or a ball.

In the following chapter, we define hyperbolic PDE for a first order (system of) PDEs. For the rest of the dissertation, we assume that the PDEs are hyperbolic.

Chapter 2

Hyperbolic PDEs

We are interested in solving time-dependent initial value boundary problems. In this work, we consider first-order PDEs in the spatial variable $\mathbf{x}$ and the temporal variable t. In one space dimension, it is of the form

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{s}(x,t), \tag{2.1}$$

where $\mathbf{u} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^m$ is a vector of m unknowns that we wish to solve in the PDE, $\mathbf{f}(\mathbf{u})$ is the flux function, $\mathbf{s}(x,t)$ is a source term. We can express Equation (2.1) in quasilinear form

$$\mathbf{u}_t + \mathbf{f}'(\mathbf{u})\mathbf{u}_x = \mathbf{s}(x,t). \tag{2.2}$$

The matrix $\mathbf{f}'(\mathbf{u})$ is called the flux Jacobian matrix, of size m by m. If the matrix $\mathbf{f}'(\mathbf{u})$ has real eigenvalues and is diagonalizable, then the PDE (2.2) is hyperbolic. When the Jacobian matrix $\mathbf{f}'(\mathbf{u})$ has real eigenvalues but is not diagonalizable, the PDE is *weakly* hyperbolic. If some or all eigenvalues of $\mathbf{f}'(\mathbf{u})$ are non-real, then the PDE is not hyperbolic.

We acknowledge that although the second-order wave equation $u_{tt} = c^2 u_{xx}$ is a more well-known equation used to describe wave propagation, we assume that we can rewrite it as a first-order system, by introducing an auxiliary variable $v = u_t$,

$$\begin{cases} u_t &= cv_x \\ v_t &= cu_x. \end{cases} \tag{2.3}$$

Here, c denotes the wave speed. Similarly, for any second order or higher order hyperbolic PDEs, we rewrite it as a system of first order hyperbolic PDEs. Next, we describe a constant coefficient scalar hyperbolic PDE, known as advection or transport equation.

2.1 Advection Equation

The simplest example of a PDE is the scalar ($m = 1$) one-way wave equation, also known as advection equation, given by

$$u_t + cu_x = 0, \tag{2.4}$$

where c is the wave speed, which we take to be constant. This equation has the solution

$$u(x,t) = u_0(x - ct), \tag{2.5}$$

where $u_0(x)$ is the initial data at time $t = t_0$. Intuitively, the solution at $t = t^*$ is a translation of the solution at initial time $t = t_0$ by the distance $c(t^* - t_0)$ and in a direction determined by the sign of c. When c is positive, the solution at initial time $u_0(x)$ is translated to the right, and when c is negative, $u_0(x)$ is translated to the left.

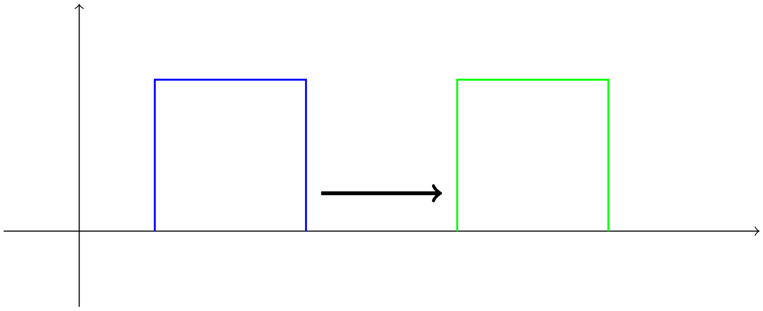


Figure 2.1: The solution to Equation (2.4) with initial data consisting of a square pulse. Here, we assume $c > 0$. The solution at a later time given is given by the green curve and is a translation of initial data given by the blue curve.

2.2 Hyperbolic Systems

For systems of hyperbolic PDEs, the diagonalization of the Jacobian matrix $\mathbf{f}'(\mathbf{u})$ allows us to decouple the waves, where in the new coordinate system, the equations look like (2.4). Since the eigenvalues of $\mathbf{f}'(\mathbf{u})$ are real, we can list them

$$\lambda^1 \leq \lambda^2 \leq ...\lambda^m.$$

Further, since the $m \times m$ matrix $\mathbf{f}'(\mathbf{u})$ is diagonalizable, it has m linearly independent eigenvectors and they form a basis for $\mathbb{R}^m$. The corresponding eigenvectors $w^1, w^2, ..., w^m$ are such that

$$\mathbf{f}'(\mathbf{u})w^i = \lambda^i w^i, i = 1, ..., m.$$

If we put the vectors w as the columns of the matrix W, then W is invertible and we denote its inverse by W^{-1}. We then use W to transform $\mathbf{f}'(\mathbf{u})$ to a diagonal matrix Λ containing the eigenvalues of $\mathbf{f}'(\mathbf{u})$,

$$W^{-1}\mathbf{f}'(\mathbf{u})W = \Lambda, \tag{2.6}$$

i.e. $\Lambda = \mathrm{diag}(\lambda^1, ..., \lambda^{\mathrm{m}})$.

We can then substitute Equation (2.6) into Equation (2.2), to get

$$u_t + W\Lambda W^{-1}u_x = 0. \tag{2.7}$$

Multiplying Equation (2.7) by W^{-1}, we obtain

$$W^{-1}u_t + \Lambda W^{-1}u_x = 0. \tag{2.8}$$

If $\mathbf{f}'(\mathbf{u})$ is a constant matrix, so is W. Then, we have $W^{-1}u_t = (W^{-1}u)_t$ and similarly, $W^{-1}u_x = (W^{-1}u)_x$. Now, letting $q = W^{-1}u$ gives us

$$q_t + \Lambda q_x = 0. \tag{2.9}$$

In other words, we can write Equation (2.9) as m decoupled advection equations

$$(q^i)_t + \lambda^i (q^i)_x = 0,\ i = 1, ..., m,$$

where q^i is i^{th} entry of vector q. The eigenvalues $\lambda^1, ..., \lambda^m$ are called the characteristic speeds because they define the speeds at which the information propagates along the characteristic curves $X(t) = x_0 + \lambda^i t$.

Notice that the equation above is identical to Equation (2.4). Hence, the similarity transformation enables us to conquer and divide the hyperbolic systems into m advection equations that we know how to solve, see Section 2.1. Once all m advection equations are solved, we use the relation $u = Wq$ to obtain u.

2.3 Method of Characteristics

The method of characteristics finds characteristic curves along which the PDE turns into a system of Ordinary Differential Equations (ODEs). The idea is to perform change of variables to reduce the number of independent variables to one. This method always works when $\mathbf{f}'(\mathbf{u})$ is a function of t and u only, and may not work for some problems where $\mathbf{f}'(\mathbf{u})$ depends on the spatial variable(s) explicitly. We will describe the method for scalar hyperbolic PDEs in one space dimension ($m = 1$), since systems of hyperbolic (excluding weakly hyperbolic) PDEs can be decoupled into scalar hyperbolic PDEs using similarity transformation.

In higher dimensions, there is no general method to find the solution to the PDE analytically. However, we can exploit the symmetry of the problem to reduce the number of dimensions by rewriting the equation in polar or spherical coordinates, see [50]. Alternatively, we can use transform methods such as Fourier transform or Laplace transform, see [17].

Let us try to understand how the method of characteristics works from a geometric point of view. We will use the following arguments to derive the method:

- At any point (x^*, t^*, z^*) on the level surface $F(x, t, z) = k$, the gradient vector $\nabla F(x^*, t^*, z^*)$ is normal to the level surface passing through (x^*, t^*, z^*).
- For any surface with normal $\mathbf{n}$, any vector on the surface that is orthogonal to the normal $\mathbf{n}$ is tangent to the surface.

Suppose we have a solution $u(x, t)$ to a scalar hyperbolic PDE, given by Equation (2.2) with $m = 1$. Consider the graph of the function $u(x, t)$ given by the surface S, as illustrated in Figure (2.2), where $z \equiv u(x, t)$. Since the surface S can be rewritten as a zeroeth level surface of $F(x, t, z) = u(x, t) - z$, we have that the gradient vector $\nabla F = (u_x(\hat{x}, \hat{t}), u_t(\hat{x}, \hat{t}), -1)$ is normal to level surface $F = u - z$ at point $(\hat{x}, \hat{t}, \hat{z})$. Finally, writing Equation (2.2) as

$$(u_x(x, t), u_t(x, t), -1) \cdot (f'(u(x, t)), 1, s(x, t)) = 0,$$

we have that $(f'(u(\hat{x}, \hat{t})), 1, s(\hat{x}, \hat{t}))$ is tangent to the surface S at $(\hat{x}, \hat{t}, \hat{u})$.

In the method of characteristics, we are looking for level curves (instead of level surfaces) that are tangent to the vector field $(f'(u), 1, s(x, t))$. We will explain the recipe to find these curves. First, assume that we have curve $\mathcal{C}$ parameterized by an auxiliary variable ξ, i.e. the points on $\mathcal{C}$ are given by $(x(\xi), t(\xi), z(\xi)$, where $z(\xi) = u(x(\xi), t(\xi))$. Then using chain rule, we compute derivative of the solution u with respect to the new variable ξ,

$$\frac{d}{d\xi}u(x(\xi), t(\xi)) = \frac{\partial u}{\partial x}\frac{dx}{d\xi} + \frac{\partial u}{\partial t}\frac{dt}{d\xi}. \tag{2.10}$$

If we can find $u(\xi)$ satisfying

$$\begin{aligned} \frac{dx}{d\xi} &= f'(u), \\ \frac{dt}{d\xi} &= 1, \\ \frac{du}{d\xi} &= s(x(\xi), t(\xi)), \end{aligned} \tag{2.11}$$

then u solves Equation (2.2). In other words, we want the tangent vector to curve $\mathcal{C}$ to satisfy the system of ODEs (2.11) above. Along these curves, the PDE (2.2) turns into a system of ODEs, as the independent variable is ξ only. Using the first two equations of 2.11, we can solve for $x(t) = f'(u)t + \alpha$. These curves $x(t) = f'(u)t + \alpha$ are known as the characteristic curves, along which information propagates. Furthermore, when the source term $s(x, t)$ is zero, the solution becomes constant or unchanged along the characteristic curves.

We will show how the method of characteristics can be used to solve the advection equation given by (2.4). We first write the equation in terms of the characteristic variable ξ, to get $\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} \equiv \frac{dt}{d\xi}\frac{\partial u}{\partial t} + \frac{dx}{d\xi}\frac{\partial u}{\partial x} = \frac{d}{d\xi}u(x(\xi), t(\xi)) = 0$,

$$\begin{aligned} \frac{dx}{d\xi} &= c, \\ \frac{dt}{d\xi} &= 1, \\ \frac{du}{d\xi} &= 0. \end{aligned}$$

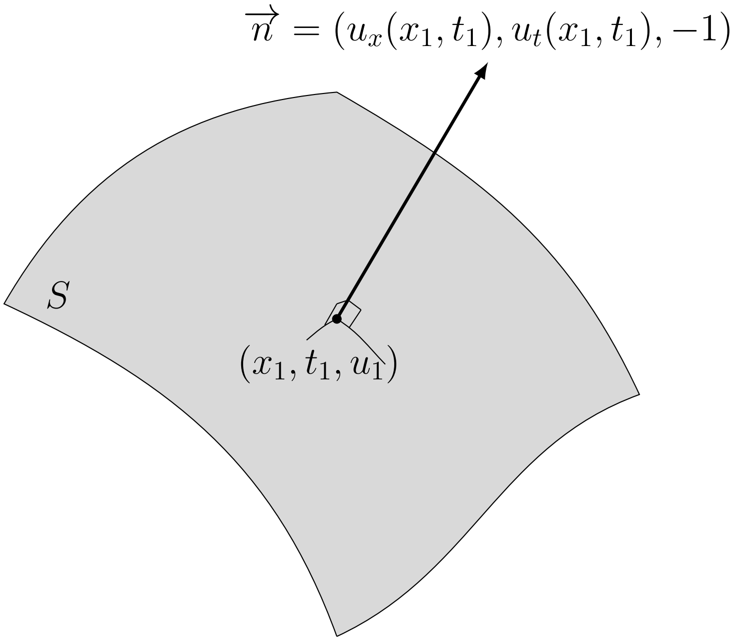


Figure 2.2: Surface $z = u(x,t)$, where u is the solution to the PDE and the normal is $(u_x, u_t, -1)$.

The solution to this system of ODEs is

$$\begin{aligned} x(\xi) &= c\xi + \alpha_1, \\ t(\xi) &= \xi + \alpha_2, \\ u(\xi) &= \alpha_3. \end{aligned}$$

Solving for x in terms of t, we get the characteristic curves $x = ct + \alpha$, where α is some arbitrary constant. That means, for any fixed α, the solution u remains constant along the line $x = ct + \alpha$, hence $u(x,t)$ is a function of $\alpha = x - ct$, i.e. $u(x,t) = \phi(x - ct)$. Since the initial data is given by $u(x,0) = u_0(x)$, we can solve for ϕ, to get $\phi = u_0$ and therefore $u(x,t) = u_0(x - ct)$.

2.4 Domain of Dependence and Range of Influence

Since waves propagate with finite speed, the solution at a point in space-time domain only depends on certain points from previous times (points in the domain of dependence) and can only affect solutions at certain points at later times (points in the range of influence). For constant coefficient linear hyperbolic PDEs, the solution at (x^*, t^*) *only* depends on data at m points at initial time, $(x^* - \lambda^i t^*, 0), i = 1, ..., m$. The set of those points $DD = \{(x^* - \lambda^i t^*, 0)\}, i = 1, ..., m$ is called the domain of dependence. Conversely, any perturbation at (x^*, t^*) *only* affects the solutions at $RI = \{(x^* + \lambda^i \Delta t, t^* + \Delta t)\}, i = 1, ..., m$. RI is called the range of influence. See Figure 2.3 for an illustration of domain of dependence and range of influence. In general, the domain of dependence (similarly, range of influence) is a region in the x-t plane, bounded by the characteristic curves. In the special case of constant linear hyperbolic PDEs, the domain of dependence is bounded by straight lines.

The domain of dependence and range of influence can be understood using the following example. Assume an earthquake occurs at points x_1^*, (assuming time $t = t_1^*$), producing waves that propagate in all directions with constant speed c_1 meters per second. People standing to the right of the epicenter at a distance Δx meters away will feel the shaking effect $\Delta t = \Delta x / c_1$ seconds later as the waves take time Δt seconds to reach these people. Thus, we can say that the ground shaking that these people feel at time $t = t_1^* + \Delta t$ *depends* on the earthquake happening Δt seconds ago, with epicenter at Δx meters away from them. Equivalently, the earthquake *influences* those people standing Δx meters away, Δt seconds later.

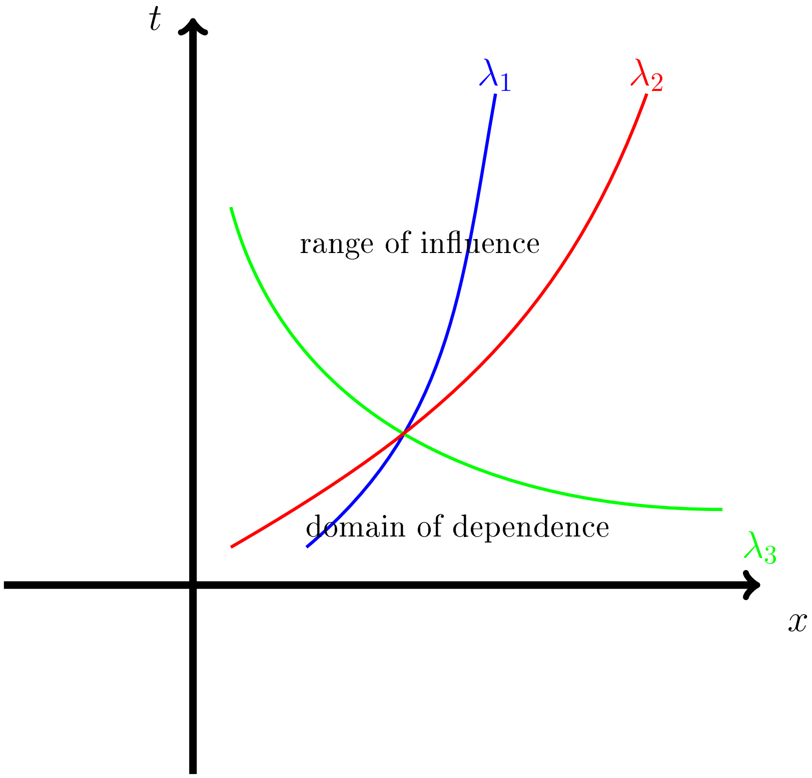


Figure 2.3: Illustration of domain of dependence and range of influence. In this case, the solution on the intersection among the three curves depends on the solution in the domain of dependence and it affects the solution in the range of influence.

2.5 CFL Condition

The domain of dependence motivates the so-called CFL condition that an explicit method must satisfy to be stable. In the advection equation $u_t + cu_x = 0$, the solution at initial time $t = t_0$ is translated with distance $\Delta x = c\Delta t$ to obtain the solution at time $t = t_0 + \Delta t$. The same statement can be made for solution at $t = t_n$. Expressing it the other way around, to compute the solution at $(x_j, t_{n+1} = t_n + \Delta t)$, we need to know the solution at $(x_j - c\Delta t, t_{n+1} - \Delta t = t_n)$. Therefore the numerical method must incorporate the information at $(x_j - c\Delta t, t_n)$ in order to compute the solution at (x_j, t_{n+1}).

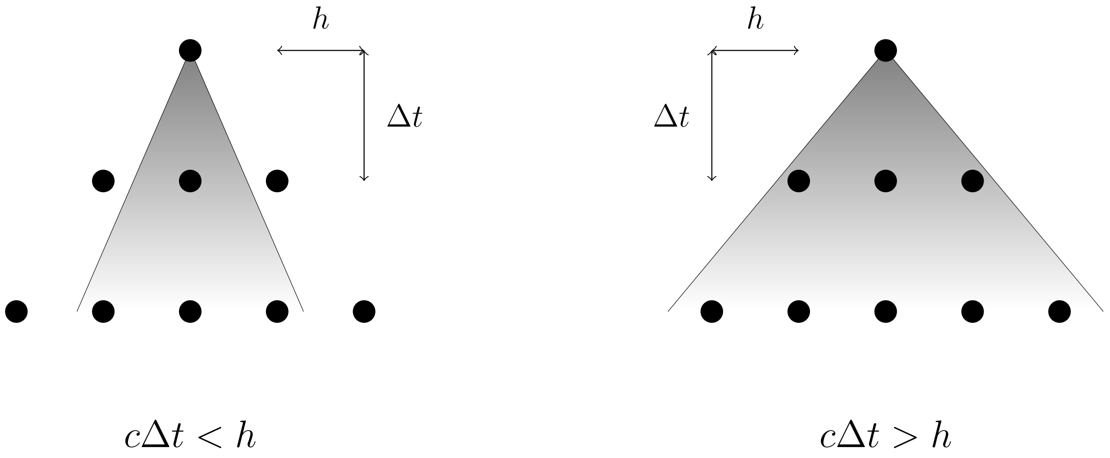


Figure 2.4: The analytic domain of dependence is shaded region and the numerical domain of dependence is given by black dots. On the left where $c\Delta t \leq h$, the numerical method *may* give convergent solutions, and on the right where $c\Delta t > h$, the numerical method cannot converge to the exact solution. This is known as CFL condition.

In general, in order to compute the exact solution analytically, we need to incorporate information from the *analytical* domain of dependence, defined in Section 2.4. On the other hand, we can approximate the solution using a numerical method, where we discretize the equation and approximate the solution at discrete points. For any discrete point (x_j, t_n), the *numerical* domain of dependence is defined as the set of grid points (depending on the numerical method) used to compute the solution at (x_j, t_n).

Suppose we have an explicit time stepping method which computes the solution at (x_j, t_{n+1}) using the solution at discrete points on the previous time step $t = t_n, t_{n-1}, ...$ Tracing back the dependence to initial time t_0, we get the numerical domain of dependence of (x_j, t_{n+1}). To obtain a convergent solution, as the grid size (h) is refined, the time step size Δt must be chosen in such a way that the numerical domain of dependence includes the analytical domain of dependence.

CFL condition: The numerical domain of dependence must contain the analytical domain of dependence, in the limit as Δt and h approach zero. For explicit methods, we have that for convergence, the time step size must satisfy

$$\left|\frac{\lambda_{\max}\Delta t}{h}\right| \leq C_{\max},$$

where $C_{\max}$ depends on the time discretization and $\lambda_{\max}$ is the maximum eigenvalue of the flux Jacobian matrix $f'(u)$ given in Equation (2.2).

A time stepping method for PDEs evolves solutions at time $t = t_n$ to a later time, let's say $t = t_{n+1}$. Since for hyperbolic PDEs, the solution at a point in the space time domain only depends on the solution at previous time within the neighborbood of that point (not the whole domain), explicit time-stepping methods are usually more efficient to solve hyperbolic PDEs than implicit time-stepping methods. However, when explicit discretization leads to numerical stiffness and hence strict CFL conditions (small $C_{\max}$), implicit time-stepping methods may be preferred. We will discuss the cause and some attempts to improve the time step size (or sometimes we say improve CFL condition) in explicit methods in Chapter 7.

Chapter 3

Conservation Laws

Conservation laws are a class of hyperbolic PDEs that model conservation of physical properties, such as mass, momentum, and energy. In one spatial dimension the PDEs that describe conservation laws can be written as Equation (2.1) with zero source term,

$$\frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}f(u(x,t)) = 0, \tag{3.1}$$

where $u(x,t)$ is the state variable at location x and time t and $f(u)$ is the flux, or the rate of flow, of the state variable u.

The derivation of conservation laws arises from the observation that at any given time t, the rate of change of the total quantity of the state variable u over some small interval $[x_1, x_2]$ must be equal to the net flux $f(u)$ into the interval through the endpoints, with no quantity created or destroyed in the region $[x_1, x_2]$, see Figure 3.1 for illustration. Mathematically, this can be expressed as

$$\frac{d}{dt}\int_{x_1}^{x_2} u\,dx = f(u(x_1,t)) - f(u(x_2,t)). \tag{3.2}$$

If the functions u and $f(u)$ are smooth, then using the Fundamental Theorem of

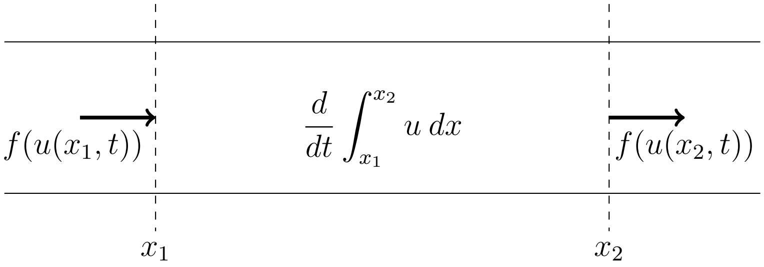


Figure 3.1: Conservation of fluid inside a pipe depends on the flux at the end points of the pipe.

Calculus, we can rewrite Equation (3.2) as

$$\frac{d}{dt}\int_{x_1}^{x_2} u(x,t)\,dx = -\int_{x_1}^{x_2}\frac{\partial}{\partial x}f(u(x,t))dx, \tag{3.3}$$

or equivalently,

$$\int_{x_1}^{x_2}\left[\frac{\partial u(x,t)}{\partial t}+\frac{\partial}{\partial x}f(u(x,t))\right]dx = 0. \tag{3.4}$$

Since this equation is true for arbitrary x_1 and x_2, we obtain the differential form of conservation law, as given in Equation (3.1).

The multi-dimensional conservation law has a similar form, except that it has the flux functions in y and z directions, which we will denote by g and h respectively,

$$\int_{z_1}^{z_2}\int_{y_1}^{y_2}\int_{x_1}^{x_2}\left[\frac{\partial u(x,y,z,t)}{\partial t}+\frac{\partial f(u)}{\partial x}+\frac{\partial g(u)}{\partial y}+\frac{\partial h(u)}{\partial z}\right]dx\,dy\,dz = 0. \tag{3.5}$$

3.1 Solving Conservation Laws with Variable Coefficients or Nonlinearity

In Section 2.3, we saw that when the wave speed c is constant, the characteristics are straight lines and the solution u is constant along the characteristics lines. In this section, we will describe the difference in the characteristics when the PDE has variable coefficients or nonlinearity.

Nonlinear Conservation Laws

The solution to nonlinear conservation laws $u_t + f(u)_x = 0$ can be found using the method of characteristics given in Section 2.3. Since the source term in conservation laws is zero, we have that the solution u is constant along the characteristics curves given by $x - f'(u)t = K$ for some constant K. Hence, the solution is given by

$$u(x,t) = u_0(x - f'(u)t).$$

When the initial data is not constant, the curves are no longer straight lines but they are curves that depend on x and t.

Variable Coefficient Conservation Laws

Similarly, we can still use the method of characteristics, but first we need to write the conservation law

$$u_t + (c(x)u)_x = 0, \tag{3.6}$$

in quasilinear form,

$$u_t + c(x)u_x + c'(x)u = 0. \tag{3.7}$$

The method of characteristics leads to

$$\frac{dx}{d\xi} = c(x), \tag{3.8}$$

$$\frac{dt}{d\xi} = 1, \tag{3.9}$$

$$\frac{du}{d\xi} = -c'(x(\xi))\, u((x(\xi), t(\xi)). \tag{3.10}$$

In this case, the characteristic curves are obtained by solving Equation 3.8 for x. Since $\frac{du}{d\xi} \neq 0$, the solution is not constant along characteristic curves. However, the PDE still reduces to a system of ODEs.

3.2 Weak Formulation

When the solution to Equation (3.1) contains discontinuities, the PDE does not make sense in the classical sense since derivatives do not exist there. The idea behind weak solutions is to require that the solution satisfies the PDE in a weaker sense by removing the derivative in the equation onto a smooth test function. Since the PDE has one derivative in x, we define the test function to be continuously differentiable, with compact support inside some domain $[x_L, x_R] \times [0, T]$,

$$\Omega = \{\phi(x,t) : \mathbb{R} \times [0,\infty) \to \mathbb{R} \big| \phi(x,t) \in C_0^1([x_L, x_R] \times [0,T])\}. \tag{3.11}$$

We then multiply the PDE (3.1) by a test function ϕ to get

$$\begin{aligned} 0 &= \int_0^\infty \int_{-\infty}^\infty (u_t + f(u)_x)\phi(x,t) dxdt \\ &= \int_{-\infty}^\infty \int_0^\infty u_t \phi(x,t) dtdx + \int_0^\infty \int_{-\infty}^\infty f(u)_x \phi(x,t) dxdt \\ &= \int_{-\infty}^\infty \left[\lim_{T\to\infty} (u\phi)\big|_{t=0}^{t=T} - \int_0^\infty u\phi_t dt \right] dx + \\ &\quad \int_0^\infty \left[\lim_{N\to-\infty} \lim_{M\to\infty} (f\phi)\big|_N^M - \int_{-\infty}^\infty f(u)\phi_x dx \right] dt \\ &= -\int_{-\infty}^\infty u(x,0)\phi(x,0)dx - \int_{-\infty}^\infty \int_0^\infty u\phi_t dtdx - \int_0^\infty \int_{-\infty}^\infty f(u)\phi_x dxdt, \end{aligned}$$

because

$$\lim_{T\to\infty} \phi(x,T) = \lim_{N\to-\infty} \phi(N,t) = \lim_{M\to\infty} \phi(M,t) = 0.$$

The weak formulation is then given by

$$\int_{-\infty}^\infty u(x,0)\phi(x,0)dx + \int_0^\infty \int_{-\infty}^\infty (u\phi_t + f(u)\phi_x)\, dxdt = 0. \tag{3.12}$$

We note that since the derivatives appear on the test function, this formulation is valid when u is discontinuous, for example when u contains a shock. We know that if u satisfies the strong formulation (3.1), then it also satisfies the weak formulation

(3.12) since the weak form (integral form) is derived from the strong form (PDE form). Conversely, if u satisfies Equation (3.12) for all test function $\phi \in C_0^1$ and u is smooth, then u is a classical solution to Equation (3.1).

3.3 Discontinuous Solutions

Equation (3.1) admits discontinuous solutions. There are two types of discontinuity in the solution to (3.1):

- Contact discontinuity: a discontinuity that appears on the initial data and gets propagated over time. In this case, the characteristic curves are parallel to the wave in $x - t$ plane, meaning the "nearby" waves propagate at the same speed.
- Shock: a discontinuity that appears due to intersecting characteristic curves. Shocks can form even if the initial data is smooth.

The propagation speed of the discontinuity, v_d can be computed according to Rankine-Hugoniot jump condition

$$(u_R - u_L)v_d = f(u_R) - f(u_L). \tag{3.13}$$

Any weak solution satisfies the Rankine-Hugoniot condition across a jump discontinuity. We note that the Rankine-Hugoniot condition does not guarantee physically correct solutions.

In the presence of shocks, the time at which the characteristics first intersect is called the breaking time, denoted by T_b. At time $t = T_b$, the solution has jump discontinuity, where u_x takes infinite value. After time $t = T_b$, the PDE does not have a classical solution, in the sense that the solution obtained by following the characteristics becomes multi-valued or does not exist. However, the weak formulation 3.12 enables us to compute the (weak) solution after the breaking time.

3.4 Nonuniqueness

The weak formulation (3.12) provides us with the existence of a solution, but the solutions are not necessarily unique. For example, consider the following nonlinear conservation law that models traffic flow,

$$u_t + (c_{\max} u(1-u))_x = 0,$$

with initial data

$$u(x,0) = \begin{cases} u_L, & \text{if } x < 0 \\ u_R, & \text{if } x > 0. \end{cases}$$

The exact solution can be written as [50]

$$u(x,t) = \begin{cases} u_L, & \text{if } x/t \leq c_{\max}(1-2u_L) \\ \frac{1}{2}\left(1 - \frac{x}{c_{\max} t}\right) & \text{if } c_{\max}(1-2u_L) \leq x/t \leq c_{\max}(1-2u_R) \\ u_R, & \text{if } x/t \geq c_{\max}(1-2u_R). \end{cases} \tag{3.14}$$

As mentioned before, this initial value problem has other solutions, among which is the following expansion shock

$$u(x,t) = \begin{cases} u_L, & x/t < \frac{1}{2}(u_L + u_R) \\ u_R, & x/t > \frac{1}{2}(u_L + u_R). \end{cases} \tag{3.15}$$

One way to choose the physically correct solution is by using the vanishing viscosity approach. We introduce a dissipation (viscous) term in Equation (3.1) to smooth the jump discontinuity, preventing u_x from becoming unbounded:

$$u_t + f(u)_x = \nu u_{xx}. \tag{3.16}$$

Away from discontinuities, the term u_{xx} is small so the smoothing effect is negligible. For discontinuous solutions, Equation 3.16 is a better model to the physical system described by Equation (3.1), because in general, physical quantities change continuously with smooth transitions, and a discontinuous solution is just a mathematical representation of a solution that changes rapidly.

In the mathematical model, the discontinuity can be approximated by a sequence of smooth solutions with sharp gradients, obtained by introducing a small amount of dissipation and taking the limit as the dissipation approaches zero. Furthermore, it can be shown that the viscous equation (3.16) has a unique solution for $t > 0$, $\nu > 0$ [50]. As the viscous term vanishes ($\nu \to 0$), the solution to Equation (3.16) is the physically correct weak solution to the corresponding inviscid equation, given by Equation (3.1). Precisely, if u^ν is a solution to Equation (3.16), then

$$\lim_{\nu \to 0} u^\nu = u,$$

where u solves Equation (3.1). Intuitively, we can think of the dissipation term as a small perturbation, and the solution to a well-posed PDE will converge to the correct solution when the perturbation vanishes.

Another way to eliminate the nonuniqueness is by using the fact that the physically correct solution satisfies the entropy condition. The term entropy is derived from the second law of thermodynamics which states that the entropy of a system is nondecreasing over time. We will find that in the second solution in Equation (3.15), the entropy jumps to a lower value across the expansion shock, which violates the entropy condition. In following section, we will discuss entropy conditions that are mathematically formulated to resolve the nonuniqueness problem.

3.5 Entropy Condition

3.5.1 Lax Entropy Condition

Suppose a solution to the scalar conservation law, given by Equation (3.4), has a discontinuity propagating with speed v_d, where the solution to the left and right of the discontinuity is denoted by u_L and u_R respectively. Then, this solution satisfies Lax entropy condition if

$$f'(u_L) > v_d > f'(u_R).$$

In other words, the speed v_d, given by the Rankine-Hugoniot condition (3.13), at which the discontinuity propagates, must be between the characteristic speeds of the wave to the left and right of the discontinuity.

3.5.2 Entropy Inequality

The idea of the entropy inequality is to define a *scalar* function called the entropy function that satisfies the same conservation law as u (with a different *scalar* flux function) when the solution u is smooth, but is not conserved at discontinuity. The entropy function and the entropy flux function are always scalar even when we solve a system of hyperbolic PDEs. Unfortunately, the entropy inequality does not give any information about contact discontinuity, so the discontinuity that we consider here is shock.

When the solution u is smooth, the entropy function $\eta(u)$ and its entropy flux $\psi(u)$ satisfy the conservation law $\nu(u)_t + \psi(u)_x = 0$. We integrate the entropy equation on a small time interval from t_1 to t_2,

$$\int_{t_1}^{t_2} \frac{d}{dt} \int_a^b \eta(u) dx dt = \int_{t_1}^{t_2} \psi(u(x(a), t)) - \psi(u(x(b), t)) dt, \tag{3.17}$$

or

$$\int_a^b \eta(u(x,t_2))dx = \int_a^b \eta(u(x,t_1))dx + \int_{t_1}^{t_2} \psi(u(x(a),t)) - \psi(u(x(b),t))dt, \quad (3.18)$$

Since the entropy is nondecreasing, we have

$$\int_a^b \eta(u(x,t_2))dx \geq \int_a^b \eta(u(x,t_1))dx + \int_{t_1}^{t_2} \psi(u(x(a),t)) - \psi(u(x(b),t))dt. \quad (3.19)$$

Note that the entropy function η is assumed to be concave, $\eta''(u) < 0$, but when the entropy function is convex, $\eta''(u) > 0$, then the inequality in (3.19) is reversed.

Using the same recipe in Chapter 3 to derive the PDE form of the conservation law from the integral form, we obtain the entropy inequality

$$\eta(u)_t + \psi(u)_x \geq 0. \quad (3.20)$$

Let us assume that the solution is smooth, so the inequality above becomes equality. Then, we are going to determine what the entropy flux $\psi(u)$ should be. First, we need to write Equation (3.20) in quasilinear form, assuming that both η and ψ are smooth functions of the solution u,

$$\eta'(u)u_t + \psi'(u)u_x = 0. \quad (3.21)$$

We compare this to the conservation law for u, scaled by $\eta'(u)$,

$$\eta'(u)u_t + \eta'(u)f'(u)u_x = 0, \quad (3.22)$$

to obtain the relation between the entropy function and the entropy flux

$$\psi'(u) = \eta'(u)f'(u). \quad (3.23)$$

In the case of systems of m hyperbolic PDEs, the derivatives $\eta'(u)$ and $\psi'(u)$ are defined as the partial derivatives with respect to each component of u,

$$\eta'(u) = \left[\frac{\partial \eta}{\partial u^1}, \frac{\partial \eta}{\partial u^2}, ..., \frac{\partial \eta}{\partial u^{m-1}}, \frac{\partial \eta}{\partial u^m}\right], \quad (3.24)$$

and similarly

$$\psi'(u) = \left[\frac{\partial \psi}{\partial u^1}, \frac{\partial \psi}{\partial u^2}, \ldots, \frac{\partial \psi}{\partial u^{m-1}}, \frac{\partial \psi}{\partial u^m}\right]. \tag{3.25}$$

Furthermore, Equation (3.23) becomes a system of $m \times m$ equations, and for $m > 2$, the solution is not guaranteed to exist since the system is overdetermined.

It can be shown that the solution u is the physically correct solution to (3.1) iff it satisfies the entropy inequality, see [50, 64] for example. For discontinuous solutions, we need to use the weak formulation of entropy inequality, which can be obtained by following the same recipe as in Section 3.2. Then, we need to show that there is an entropy function (and the associated entropy flux function) such that the solution u satisfies the weak formulation of entropy inequality, for all test functions defined in (3.11). It is not convenient to check the inequality against all test functions in (3.11) or to find/construct a test function such that the entropy inequality does not hold. Guermond, see [25–29], combined the idea of the entropy inequality and the vanishing viscosity to regularize discontinuous solutions to the hyperbolic PDEs, which we will discuss in detail in Chapter 7.

Chapter 4

Spatial Discretizations for Conservation Laws

Much of the contemporary research on numerical methods for PDE is centered around developing methods that

- are stable and accurate,
- are efficient,
- incorporate correct boundary conditions,
- work for discontinuous material (PDE with discontinuous coefficients),
- work for complex geometry, and
- inherit physical properties such as conservation.

A major contribution of this thesis is the derivation, analysis and implementation of flux-conservative Hermite methods for conservation laws as discussed in Chapter

6. Here we briefly review other possible methods for numerical approximation of conservation laws.

Conservation laws describe the conservation of some quantities that change in space and time due to flux. A good numerical method for conservation laws must respect the conservation at the discrete level. For linear conservation laws, a consistent and stable numerical method produces convergent solutions, by the Lax equivalence theorem. However, for nonlinear conservation laws involving shocks, consistency and stability are not enough to guarantee convergence. For nonlinear conservation laws, we also need to address the issue of nonlinear stability and nonuniqueness of solutions. An example where a consistent and stable method produces a wrong solution (where the shock travels at an incorrect speed, therefore the solution violates the entropy condition) can be found in [49]. The problem of convergence to non-physical solutions can be eliminated by using a conservative method.

For hyperbolic PDEs of conservation law type, we can exploit the characteristic structure to construct a conservative method. Godunov, in [20], proposed the so-called reconstruct-evolve-average (REA) algorithm. The preliminary step in the algorithm is to obtain a piecewise constant function U^0 to approximate the solution at initial time. On each cell $C_j = (x_{j-1/2}, x_{j+1/2})$, the approximate solution U_j^0 is the cell average of the initial data $u(x, t_0) = u_0(x)$,

$$U_j^0 = \frac{1}{h} \int_{C_j} u_0(x) dx,$$

where $h = x_{j+1/2} - x_{j-1/2}$. The solution at time $t = t_n$ is evolved using the recipe

1. *Reconstruct* a piecewise polynomial $\hat{u}(x, t_n)$ defined for all x on the domain of the PDE, from the cell averages U_j^n's. For example, we can take the reconstructed solution to be a piecewise constant function where $\hat{u}(x, t_n) = U_j^n$ when $x \in C_j$.

2. *Evolve* the reconstructed solution $\hat{u}(x, t_n)$ exactly using method of characteristic whenever possible. Otherwise we approximate the solution at $t = t_{n+1}$, $\hat{u}(x, t_{n+1})$. To derive the time evolution procedure, consider Equation (3.2) on cell C_j, with $x_1 = x_{j-1/2}$, $x_2 = x_{j+1/2}$,
$$\frac{d}{dt}\int_{C_j} \hat{u}(x,t)dx = f(\hat{u}(x_{j-1/2},t)) - f(\hat{u}(x_{j+1/2},t)).$$
We then integrate both sides with respect to time from $t = t_n$ to $t = t_{n+1}$,
$$\int_{C_j} \hat{u}(x,t_{n+1})dx - \int_{C_j} \hat{u}(x,t_n)dx = \int_{t_n}^{t_{n+1}} f(\hat{u}(x_{j-1/2},t)) - f(\hat{u}(x_{j+1/2},t))dt$$
Dividing by the grid size h, we have
$$\frac{1}{h}\left(\int_{C_j} \hat{u}(x,t_{n+1}) - \hat{u}(x,t_n)dx\right) = \frac{1}{h}\int_{t_n}^{t_{n+1}} f(\hat{u}(x_{j-1/2},t)) - f(\hat{u}(x_{j+1/2},t))dt.$$
Now, writing the left hand side as cell averages, we have
$$\hat{U}_j^{n+1} - \hat{U}_j^n = \frac{1}{h}\int_{t_n}^{t_{n+1}} f(\hat{u}(x_{j-1/2},t)) - f(\hat{u}(x_{j+1/2},t))dt.$$
In some special cases, the integral on the right hand side can be computed exactly. However, when the solution $\hat{u}$ changes with time at $x = x_{j-1/2}$ and $x = x_{j+1/2}$, we can approximate the integral using $\hat{u}$ at $t = t_n$ or using numerical quadrature.

3. *Average* the solution obtained in the evolution step to get new cell averages
$$U_j^{n+1} = \frac{1}{h}\int_{C_j} \hat{u}(x,t_{n+1})dx$$

We repeat the REA algorithm until we reach the desired final time. The evolution step is described in detail in [50, 65].

Finite Volume methods are the simplest Godunov methods where the reconstructed solution is taken to be the cell average. The methods are popular in the field of computational fluid dynamics, and sample codes are available in the software package *Clawpack* for linear and nonlinear hyperbolic PDEs. Despite being conservative, Finite Volume methods are first order, and therefore are not very accurate when used for long-time simulation. To obtain a second order method, a linear piecewise polynomial is used in the reconstruction step. In order to maintain conservation, we need to choose the slope of the reconstructed solution carefully so that the cell average remains unchanged. A different choice of slope leads to Fromm, Beam-Warming, or Lax-Wendroff scheme.

For discontinuous solutions, the high order polynomial reconstruction is known to produce oscillatory behavior. Techniques such as flux limiters, Essentially Non-Oscillatory (ENO) and Weighted ENO (WENO), and artificial viscosity have been developed to improve or eliminate oscillations in the approximate solutions. The flux limiters use two fluxes, corresponding to a low order method and a high order method. The flux limiter determines weights assigned to the low order flux and high order flux in different parts of the solution. The idea is to use a high order method when the solution is smooth and switch to a low order method when the solution is discontinuous. Under the flux limiter category, we find minmod, superbee, monotonized central-difference, minmod- central scheme by Nessyahu and Tadmor [53], van Leer method [66–70], hybrid method by Harten and Zwas [33], and the flux-corrected transport [7, 55, 73]. To construct an even higher order method, we reconstruct the solution using a higher order polynomial, for example a piecewise parabola [13].

Another approach to construct a high order method without oscillations is the ENO/ WENO method. In the ENO method [36], given cell averages U_j^n on k contiguous cells (or k stencil), we can construct a polynomial of degree at most $k-1$ whose cell average in each of the k cells matches with U_j^n. Note that on each cell, we have different k-stencil options to construct the $(k-1)$th degree polynomial. ENO method chooses the k cells so that the resulting polynomial has the least oscillations. Significant amount of work has been put into developing ENO methods, see for example [9, 35, 37, 38, 61, 62]. The WENO method uses all possible k-stencils and assigns more weight to the stencil that produces smooth polynomials, [40, 51].

The main idea of artificial viscosity was first proposed in [54]. Since then, the main research in the artificial viscosity method focuses on designing a robust yet accurate artificial viscosity method, which adds a sufficient amount of viscosity to regularize shocks, but adds a relatively small amount of viscosity elsewhere. Some works of this nature include PDE-based viscosity [41], weak local residual [48], and filtering [72]. The entropy viscosity method, introduced by Guermond and Pasquetti [25], uses the fact that the entropy residual takes the form of a Dirac delta distribution centered at the shock. This property establishes the entropy residual as a natural shock detector, that can be easily extended to higher dimensions and unstructured grids. In essence, the entropy viscosity method provides the discretization method with grid independent shock-capturing capability without compromising the high order of accuracy away from the shocks, where the amount of viscosity is porportional to the truncation error in the discretization of the residual.

Each of the methods described above eliminates or reduces oscillations by extracting information about oscillations that would occur if no further action is taken. Then, the method adjusts the construction of the numerical solution to avoid the appearance of spurious oscillations. In high order methods, the oscillations often cannot be completely suppressed, so we can only hope for almost non-oscillatory solutions. Alternatively, we can remove the oscillations after computing the solutions, which is known as post-processing, [3, 12, 58, 59].

4.1 Hermite Methods

First introduced by Goodrich, Hagstrom, and Lorenz in [21] for hyperbolic initial-boundary value problems, Hermite methods use the solution and its first m derivatives in each coordinate to construct an approximate solution to the PDE. These methods have great explicit time stepping properties, efficient memory consumption and a high computation to communication ratio. They are thus suitable for parallel implementation on distributed, shared or hybrid memory platforms. One of the main ingredients to Hermite methods' efficiency is the use of a Cartesian grid, which becomes a downside when implementing the method on complex geometry and applying boundary conditions. The difficulty for complex geometry and boundary condition can be remedied by hybridizing Hermite methods with another method that handles boundary well, for example a Discontinuous Galerkin method [39]. The hybrid Hermite-DG method is presented in [11].

In the formulation by Goodrich et al, the flux is computed using polynomial representation of the solution at the cell centers, and for nonlinear problems, the number of terms representing the flux function grows and therefore must be truncated. The truncated flux becomes discontinuous at cell interfaces, and as a consequence, conservation is lost. We then developed a new class of flux-conservative Hermite methods. The conservation property represents a significant improvement to the original Hermite methods [21]. The flux-conservative Hermite methods improve the nonlinear stability of the original Hermite methods. For the flux-conservative methods, we proved that the solution and its derivatives are evolved in a conservative manner. Although this conservation property does not guarantee positive state variables, density for example, we still find that the new method is capable of simulating Lax's and Sod's shock tube problems without any stabilization.

4.2 Finite Element Methods (FEM) and Discontinuous Galerkin (DG) Methods

Finite Element methods [16, 63, 75] approximate a solution to a hyperbolic PDE by projecting the solution into a finite dimensional space V_N. The approximate solution is chosen so that the error in the solution is *orthogonal* to the projected space V_N (Galerkin orthogonality). The discretization of FEM is prescribed in the weak or variational form, and the test functions live in the approximation space V_N, therefore can be chosen as basis functions of V_N. The formulation becomes convenient when deriving a energy estimate to establish the stability of the method. Besides the excellent stability results, the boundary implementation of FEM on complex geometry is straightforward, and hence FEM can be implemented on unstructured elements and curved boundaries.

FEM divides the domain of the PDE into a *finite* number of non-overlapping *elements*, hence also derives the name. The grid size h can vary from one element to another. In each element, the solution is approximated by a linear combination of basis functions of V_N. We restrict the approximation space V_N to polynomial spaces since the error analysis in non-polynomial spaces can be complicated and the order of the method becomes harder to compute. In polynomial space, the order of the method, p, is dictated by the number of basis functions N. Convergence in FEM can be obtained with $h-$refinement (dividing the elements into smaller elements), $p-$refinement (increasing the number of polynomial basis, hence the order of the method), or hp refinement (h and p refinements simultaneously), [4, 30].

For time evolution, we consider the Method of Lines (MOL) approach, where we approximate the spatial derivatives in the PDE using spatial discretization of choice, FEM for example, while leaving the temporal derivatives continuous. FEM spatial discretization of Equation 3.1 leads to a system of ODE,

$$M\frac{d\mathbf{c}}{dt} = S\mathbf{c} + F, \tag{4.1}$$

where $\mathbf{c}$ is a vector containing the degrees of freedom in the numerical solution, M is the mass matrix, S is the stiffness matrix, and F is the flux vector. To accommodate communication between elements, the basis functions in FEM are global, meaning that the support of the basis functions expands over several elements. The global nature of the coupling makes the mass matrix M difficult to invert. Discontinuous Galerkin methods, on the other hand, use local basis functions, which result in an easily invertible block diagonal mass matrix M. The locality of the basis functions induces discontinuity in the solution, but a penalty term on the flux is introduced to impose a weak continuity in the solution and hence stabilizes the method. For solving conservation laws, Gassner developed energy stable DG methods [18, 44].

Another contribution of this thesis is the derivation of Sobolev Discontinuous Galerkin methods, where the solution is the best approximation measured in Sobolev norm. It turns out that this formulation yields tame CFL conditions. We acknowledge other efforts in improving the CFL condition, including co-volume mesh based filter by Warburton and Hagstrom [71], modifying the numerical flux [10], and hybridizing Discontinuous Galerkin methods with Finite Difference methods [5].

Chapter 5

Temporal Discretization for Conservation Laws

When solving time-dependent partial differential equations (PDE), most numerical methods first discretize in space to obtain a system of ordinary differential equations (ODE). In this case, solving a PDE can be reduced to simply solving an ODE and the resulting scheme is called method of lines. For nonlinear PDEs with discontinuous solutions, linear stability analysis itself is insufficient to guarantee stability, since linearly stable methods can have solutions with unphysical behavior such as spurious oscillations or overshoots. To measure nonlinear stability in the numerical solution $\mathbf{u}^n = (u_1^n, u_2^n, ...)$ at time $t = t_n$, we use total variation, defined as the sum of the increments in grid values,

$$TV(\mathbf{u}^n) = \sum_j |u_j^n - u_{j-1}^n|. \tag{5.1}$$

In conservation laws,

$$u_t + f(u)_x = 0, \tag{5.2}$$

the weak solution has the *monotonicity preserving* properties, see [19], where the weak solution can not form new local extrema in x, and the value of a local minimum is nondecreasing, while the value of a local maximum is nonincreasing. Therefore, we ask that the numerical solution does not grow in time,

$$TV(\mathbf{u}^{n+1}) \leq TV(\mathbf{u}^{n}), \tag{5.3}$$

where $\mathbf{u}^n$ and $\mathbf{u}^{n+1}$ are numerical solutions at time $t = t_n$ and $t = t_{n+1}$ respectively. A numerical method producing such solutions is said to have a Total Variation Diminishing (TVD) property. To guarantee this property, we discretize $f(u)_x$ in space with a TVD spatial discretization, for example TVD finite difference, or finite element approximation, and we arrive at a ODE of the form

$$u_t = L(u). \tag{5.4}$$

A suitable TVD time discretization is used to evolve solution u in time. Time discretization methods with TVD property are now termed Strong Stability Preserving (SSP) time discretizations, due to the purpose of preserving stability.

When Equation 5.4 is discretized using Forward Euler method, we obtain

$$u^{(n+1)} = u^{(n)} + \Delta t L(u^{(n)}). \tag{5.5}$$

High order Runge Kutta temporal discretizations that maintain the TVD property have been developed, commonly known as Strong Stability Preserving Runge-Kutta (SSPRK). These schemes, developed by Gottlieb and Shu in [23], express an q^{th} stage Runge-Kutta method as

$$u^{(i)} = \sum_{k=0}^{i-1} (\alpha_{ik} u^{(k)} + \Delta t \beta_{ik} L(u^{(k)})), \; i = 1, ..., q, \tag{5.6}$$

$$u^{(0)} = u^n, \; u^{(q)} = u^{n+1}.$$

When all α_{ik} and β_{ik} are nonnegative, the Runge-Kutta method above is a convex combination of Forward Euler operators, but with time step $(\Delta t)\alpha_{ik}/\beta_{ik}$. As a consequence, spatial discretizations that result in TVD schemes when being paired with Forward Euler time stepping can also be combined with SSPRK to obtain higher order in time, possibly under a different CFL condition. The goal of our experiments is to understand when SSPRK is necessary and the benefits of using SSPRK time discretization over non-SSPRK one.

We study the behavior of SSP and non-SSP Runge-Kutta methods for some test problems. We use second-order (one-step) two-stage Runge-Kutta methods(RK(2,2)) for both SSP and non-SSP schemes as in [22]. For all the results presented below, we assume that the term $f(u)_x$ has been discretized, so the problem reduces to an ODE of the form $u_t = L(u)$. We consider a SSPRK(2,2) scheme given by

$$u^{(1)} = u^n + \Delta t L(u^n), \tag{5.7}$$

$$u^{n+1} = \frac{1}{2}u^n + \frac{1}{2}u^{(1)} + \frac{1}{2}L(u^{(1)}), \tag{5.8}$$

and a non-SSPRK(2,2) scheme given by

$$u^{(1)} = u^n - 20\Delta t L(u^n), \tag{5.9}$$

$$u^{n+1} = u^n + \frac{41}{40}u^{(1)} - \frac{1}{40}L(u^{(1)}). \tag{5.10}$$

5.1 Burgers' Equation with Piecewise Constant Initial Data

In this nonlinear initial value problem taken from [23], we consider Burgers' equation with flux function $f(u) = u^2/2$,

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0, \tag{5.11}$$

on the domain $D = [-1, 1]$ with initial data

$$u_0(x) = \begin{cases} 1 & x < 0, \\ -0.5 & x > 0, \end{cases} \tag{5.12}$$

and fixed boundary conditions $u(-1, t) = u(-1, 0)$ and $u(1, t) = u(1, 0)$.

The term $f = \left(\frac{1}{2}u^2\right)$ is discretized on uniform grid consisting of n_x gridpoints with Δx spacing, $x_j = -1 + j\Delta x$, $j = 0, ..., n_x$, where f_j denotes the approximation of $f(x_j)$. We approximate the derivative of flux $f_x = \left(\frac{1}{2}u^2\right)_x$ according to MUSCL scheme,

$$(f_j)_x = \frac{1}{\Delta x}\left(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}\right),$$

where the numerical flux $\hat{f}_{j+1/2}$ is defined by

$$\hat{f}_{j+1/2} = h(u^-_{j+1/2}, u^+_{j+1/2}),$$

with

$$h(u^-, u^+) = \begin{cases} \min_{u^- \leq u \leq u^+} 0.5u^2 & \text{if } u^- \leq u^+, \\ \max_{u^- \geq u \geq u^+} 0.5u^2 & \text{if } u^- > u^+, \end{cases} \tag{5.13}$$

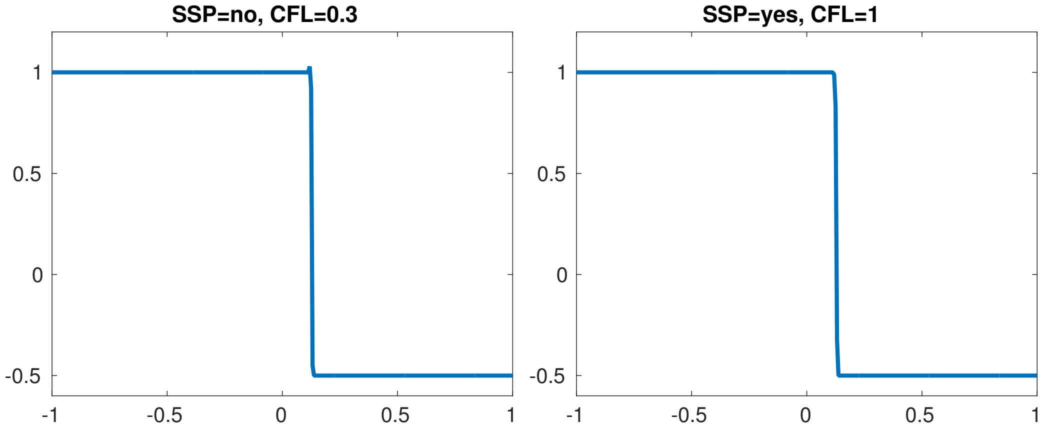
and

$$\begin{aligned} u^-_{j+1/2} &= u_j + \frac{1}{2}\text{minmod}(u_{j+1} - u_j, u_j - u_{j-1}), \\ u^+_{j+1/2} &= u_{j+1} - \frac{1}{2}\text{minmod}(u_{j+2} - u_{j+1}, u_{j+1} - u_j). \end{aligned}$$

Here, the minmod function is defined as

$$\text{minmod}(a, b) = \frac{\text{sign}(a) + \text{sign}(b)}{2}\min(|a|, |b|).$$

The initial data contains a shock at $x = 0$. The shock moves at speed $v_s = 0.25$, which is computed using Rankine-Hugoniot conditions. The exact solution to the



(a) non-SSPRK

(b) SSPRK

Figure 5.1: Comparison of non-SSPRK and SSPRK solutions for Burger's equation with piecewise initial data described in Section 5.1. We were not able to run the non-SSP scheme with higher CFL number than listed above.

initial value problem above is a translation of initial data with distance $v_s T$, where T is the simulation time. The solutions obtained with SSP and non-SSP schemes are compared at time $T = 0.5$ with total number of gridpoints $n_x = 320$. In Figure 5.1, we see overshoot on the solution run with non-SSPRK that we don't observe on the solution obtained with SSPRK. The total variations are plotted in Figure 5.2.

5.2 Burgers' Equation with Sinusoidal Initial Data

In this example taken from [22], we consider Burgers' equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0, \tag{5.14}$$

on the domain $D = [-1, 1]$ with initial data

$$u_0(x) = 0.5 - 0.25 \sin \pi x \tag{5.15}$$

and periodic boundary condition $u(-1, t) = u(1, t)$.

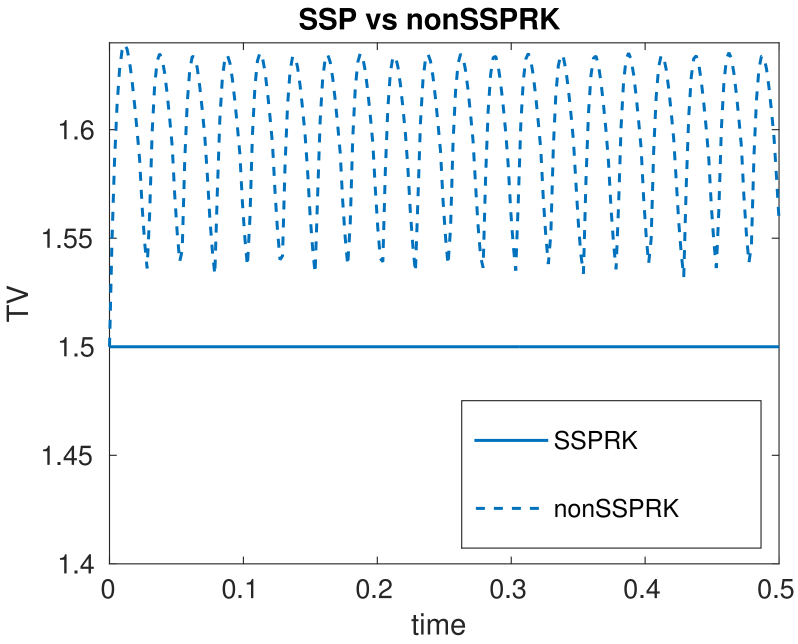


Figure 5.2: Comparison of TV between non-SSPRK and SSPRK for Burger's equation with piecewise initial data described in Section 5.1. The SSPRK results are obtained with CFL= 1 while the non-SSPRK results are obtained with a much lower CFL= 0.3. We were not able to run the non-SSP scheme with higher CFL number than listed above for either case.

The term $\frac{1}{2}(u^2)_x$ is discretized using upwind flux,

$$\frac{1}{\Delta x}\left(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}\right),$$

where the numerical flux $\hat{f}_{j+1/2}$ is defined by

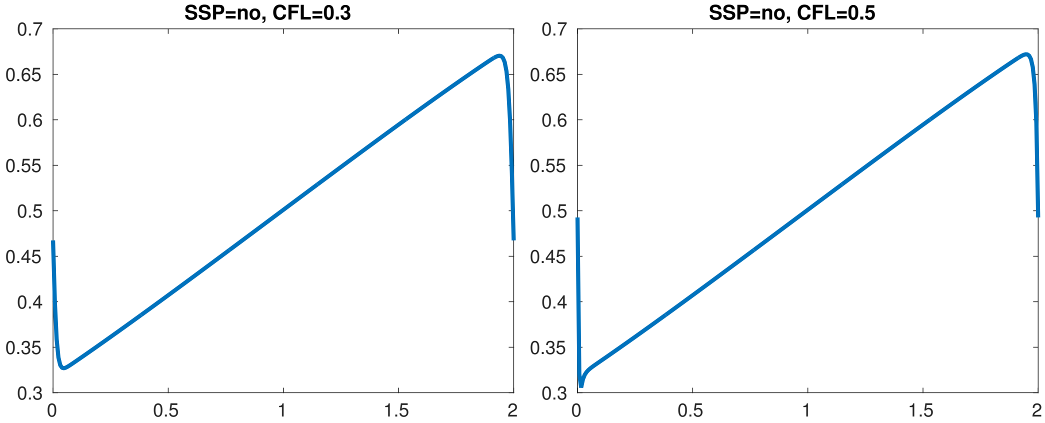
$$\hat{f}_{j+1/2} = (u_j)^2/2.$$

Using Forward Euler to discretize u_t, we arrive at a scheme that looks like

$$u_j^{n+1} = u_j^n - \lambda((u_j^n)^2 - (u_{j-1}^n)^2), \tag{5.16}$$

where $\lambda = 0.5\Delta t/\Delta x$. Now, Harten's Lemma [34] says that if we can write an explicit scheme as

$$u_j^{n+1} = u_j^n + C_{j+1/2}(u_{j+1} - u_j) - D_{j-1/2}(u_j - u_{j-1})$$



(a) non-SSPRK1

(b) non-SSPRK2

Figure 5.3: Comparison of non-SSPRK solutions with different time step size for Burger's equation with sinusoidal initial data described in Section 5.2.

with $C_{j+1/2}, D_{j-1/2} \geq 0$ and $C_{j+1/2} + D_{j-1/2} \leq 1$, then the scheme is TVD. Using this lemma along with the fact that the flux $f(u) = u^2/2$ and hence $f'(u) = u$, we conclude that the scheme given in Equation 5.16 is indeed TVD whenever the CFL condition is satisfied. Thus, we have a TVD scheme when replacing the time stepping method with SSPRK according to the recipe in [23].

The solutions are computed up to time $T = 2$ with $n_x = 256$. In Figure 5.3, we see that reducing the timestep size (equivalently, CFL) in the non-SSP case eliminates undershoot. The total variations are plotted in Figure 5.4. The total variation of non-SSPRK with CFL$= 0.3$ is pretty much identical to the total variation of SSPRK with CFL$= 1$ for this problem. However, when we increase the time step for non-SSPRK, overshoots start to develop and consequently, the total variation increases. The total variation eventually decreases since the solution is supposed to shrink after a while.

5.3 Advection Equation with Discontinuous Data

We consider a linear advection equation,

$$u_t - u_x = 0 \tag{5.17}$$

on domain $D = [-1, 1]$ with initial data

$$u_0(x) = \begin{cases} 0 & x < 0.5, \\ 1 & x > 0.5, \end{cases} \tag{5.18}$$

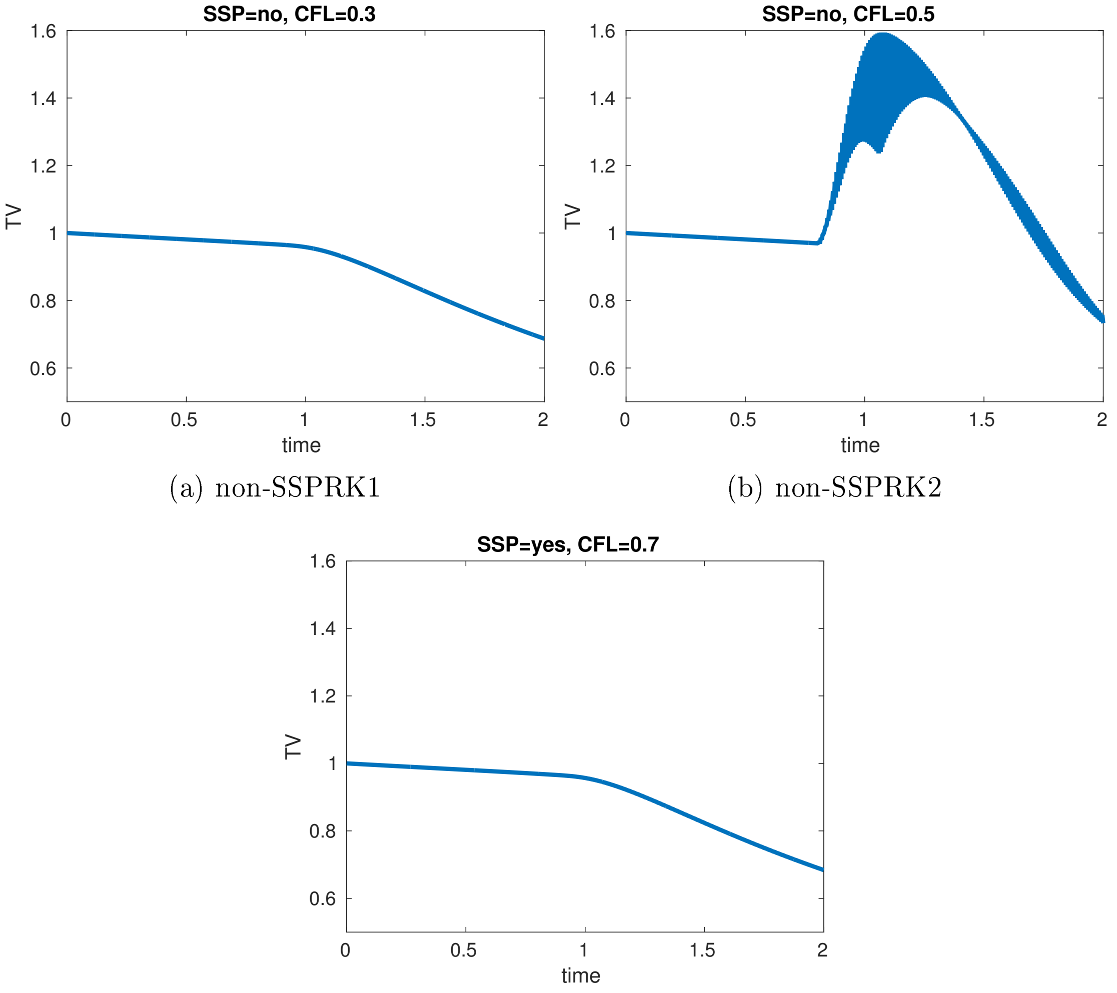
and fixed boundary condition $u(-1,t) = u(-1,0)$ and $u(1,t) = u(1,0)$.

This problem decribes a wave advecting with speed $a = -1$, therefore the wave is moving to the left. The exact solution of the problem described above is $u(x,t) = u_0(x+t)$. We run the simulation before the discontinuity hits the left boundary. The total variation at initial time is $TV^{(0)} = |1-0| = 1$. Since the exact solution at any time t_n is just a translation of initial data to the left by t_n, the total variation stays constant over time.

In Figure 5.5, we present the error in the total variation as a function of time. The plot on the left is performed using CFL= 1 and the one on the right uses CFL= 0.5. We conclude that the difference between SSPRK vs. non-SSPRK is practically nonexistent. In addition to this problem, we tried different initial data, different RK schemes, but SSPRK and non-SSPRK with comparable order of accuracy yield pretty much identical TV. This observation is supported in [22] and turns out to be correct. The reason is that in linear problems, most non-SSPRK schemes can be rearranged as an SSPRK scheme in a way that is not possible in nonlinear problems.

5.4 Summary

For nonlinear problems, non-SSPRK produces solutions with no overshoots/ undershoots. However, when there is enough dissipation in the numerical method itself, SSPRK allows a larger time step thatn non-SSPRK. In linear PDEs, non-SSPRK can be rearranged as SSPRK, so there is no clear advantage of using SSPRK.

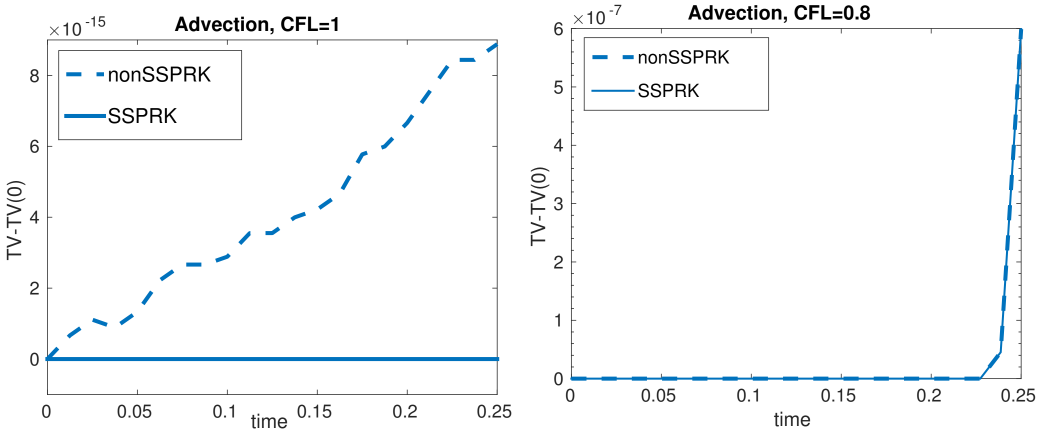


(a) non-SSPRK1

(b) non-SSPRK2

(c) SSPRK

Figure 5.4: Comparison of TV between non-SSPRK and SSPRK for Burger's equation with sinusoidal initial data described in Section 5.2.



(a) max CFL

(b) lower CFL

Figure 5.5: Comparison of TV between non-SSPRK and SSPRK for the advection equation with discontinuous initial data described in Section 5.3.

Chapter 6

Flux-conservative Hermite methods for nonlinear conservation laws

This chapter has been submitted to Journal of Scientific Computing [45] and it is now under revision.

Keywords: Hermite, conservative, high order method, the entropy viscosity method

Abstract: A new class of Hermite methods for solving nonlinear conservation laws is presented. While preserving the high order spatial accuracy for smooth solutions in the existing Hermite methods, the new methods come with better stability properties. Artificial viscosity in the form of the entropy viscosity method is added to capture shocks.

6.1 Introduction

Conservation laws model physical systems arising in traffic flows, aircraft design, weather forecast, and fluid dynamics. Numerical methods for conservation laws ideally conserve quantities like mass or energy, and accurately capture various physical components of the solutions, from small smooth scales to shock waves. The presence of both smooth waves and shock waves, for example in shock-turbulence interaction creates a challenge to the simulation of nonlinear conservation laws.

Shock waves typically appear in solutions to nonlinear conservation laws, and are characterized by very thin regions where the solution changes rapidly. Approximation of shocks and small waves is challenging as the small and large scales need to be solved simultaneously. Historically, low order finite volume and finite difference methods equipped with flux/slope limiters have been used to handle shocks, see for example the textbooks [49, 50]. The drawback with low order methods is that they cannot accurately propagate small scales over long distances and as a result, today the research focus has gravitated towards high order accurate methods with shock capturing capability.

Among high order methods, the weighted essentially non-oscillatory (WENO) method, [40,51,60], has proven to be a method popular among practitioners. WENO methods are still relatively dissipative which may be a drawback for turbulent simulations, [52]. Also, discontinuous Galerkin methods combined with shock capturing, [15, 47], or selectively added artificial viscosity, [43, 56, 76], have received significant interest. The latter approach traces back to the artificial viscosity method by Neumann and Richtmyer, [54] and the popular streamline diffusion method, [8, 41], which computes the viscosity based on the residual of the PDE.

In this work, we advocate the combination of a high order method and selectively added artificial viscosity. Specifically, we show how the entropy viscosity by Guermond and Pasquetti, [25], can be implemented in our new flux-conservative Hermite methods.

First introduced by Goodrich, Hagstrom, and Lorenz in [21] for hyperbolic initial-boundary value problems, Hermite methods use the solution and its first m derivatives in each coordinate to construct an approximate solution to the PDE. The formulation by Goodrich et al. computes the flux at the cell centers, which for nonlinear problems leads to discontinuous fluxes at cell interfaces. This discontinuity results in loss of conservation.

To address the lack of conservation, we develop a new class of Hermite methods, which share the basic features with the method in [21], such as interpolation and time evolution, but differs in the computation of the flux function. In the flux-conservative Hermite methods, the numerical flux is computed at cell edges and then interpolated to cell center for time evolution, hence by construction, the numerical flux is continuous at cell interface. Additionally, for nonlinear problems, it is more efficient to use one step methods than the Taylor series approach in [21], see [31,32]. Here we will use the standard Runge-Kutta method to evolve in time.

The rest of the chapter is organized as follows. In Section 6.2, we derive conservation laws and discrete conservation. Then, in Section 6.3, we describe Hermite methods as first introduced by Goodrich et al. [21], followed by the description of the flux-conservative Hermite methods in Section 6.3.2. For shock capturing capability, we implement the entropy viscosity method, which is explained in Section 6.4. In Section 6.5, we present the results of simulation on Euler's equations, see [46] for results on Burgers' equation.

6.2 Conservation Laws

A scalar conservation law in one space dimension takes the form

$$\frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}f(u(x,t)) = 0, \tag{6.1}$$

where $u(x,t)$ is the state variable at location x and time t and $f(u)$ is the flux, or the rate of flow, of the state variable u.

The derivation of conservation laws comes from the observation that at any given time t, the rate of change of the total quantity of the state variable u over some interval $[a,b]$ must be equal to the net flux $f(u)$ into the interval through the endpoints. Mathematically, this can be expressed as

$$\frac{d}{dt}\int_a^b u\,dx = f(u(a,t)) - f(u(b,t)). \tag{6.2}$$

When approximating the solution to scalar conservation laws given by equation (6.1), the PDE is typically discretized on a grid consisting of N_x cells where $x_0 = a$ and $x_{N_x} = b$. It is desirable that the numerical method satisfies discrete conservation. If the reconstructed solution u_h^j and flux f_h^j on any cell $\mathcal{I}_j = [x_{j-1}, x_j]$ satisfy the condition $f_h^j|_{x=x_j} = f_h^{j+1}|_{x=x_j}$, $j = 1, \ldots, N_x - 1$, we immediately find

$$\begin{aligned}\int_a^b \frac{\partial u_h^j}{\partial t}\,dx &= \sum_{j=1}^{N_x}\int_{x_{j-1}}^{x_j}\frac{\partial u_h^j}{\partial t}\,dx\\ &= \sum_{j=1}^{N_x}\int_{x_{j-1}}^{x_j}\frac{\partial}{\partial x}(-f_h^j)\,dx\\ &= \sum_{j=1}^{N_x}\left(f_h^j|_{x=x_{j-1}} - f_h^j|_{x=x_j}\right)\\ &= f_h^j(u(a)) - f_h^j(u(b)).\end{aligned} \tag{6.3}$$

The property that $f_h^j|_{x=x_j} = f_h^{j+1}|_{x=x_j}$ does not hold for the original Hermite methods, and our goal here is to design a new Hermite method with this property. Before describing our new method, we briefly describe the original method.

6.3 Hermite Methods

A Hermite method of order $2m+1$ approximates the solution to a PDE by an element-wise polynomial that has continuous spatial derivatives up to order m at the element's interfaces. In Hermite methods, the degrees of freedom are function and spatial derivative values, or equivalently the coefficients of the Taylor polynomial at the cell center of each element. The evolution of the degrees of freedom on each element can be performed locally.

6.3.1 Hermite Method in One Dimension

Consider again equation (6.1) on the domain $D = [x_L, x_R]$. Let $\mathcal{G}^p$ and $\mathcal{G}^d$ be the primal grid and the dual grid, defined as

$$\mathcal{G}^p = \left\{x_j = x_L + jh_x,\ j = 0, \dots, N_x\right\}, \tag{6.4}$$

$$\mathcal{G}^d = \left\{x_{j+1/2} = x_L + \left(j + \frac{1}{2}\right) h_x,\ j = 0, \dots, N_x - 1\right\}, \tag{6.5}$$

where $h_x = (x_R - x_L)/N_x$ is the distance between two adjacent nodes. Let u^p and u^d be the approximations to the solution on the primal and dual grids, respectively.

At the initial time $t_n = t_0 + n\Delta t$, we assume that the approximation u^p on the primal grid, the global piecewise polynomial

$$u^p(x, t_n) = \sum_{k=0}^{m} c_k(t_n)\,(x - x_j)^k\,, x \in \mathcal{I}_j^d = [x_{j-1/2}, x_{j+1/2}], \tag{6.6}$$

is known. Starting from time $t = t_n$ on the primal grid $\mathcal{G}^p$, we evolve the solution one full time step by:

Reconstruction by Hermite interpolation: We construct u^d, the global Hermite interpolation polynomial of degree $(2m+1)$, on the dual grid. That is,

$$u^d(x,t_n) = \sum_{k=0}^{2m+1} b_k(t_n)\left(x - x_{j-1/2}\right)^k, x \in \mathcal{I}_j^p = [x_{j-1}, x_j], \tag{6.7}$$

where the coefficients $b_k(t_n)$ are uniquely determined by the interpolation conditions

$$\frac{\partial^k}{\partial x^k} u^d(x_i,t_n) = \frac{d^k}{dx^k} u^p(x_i,t_n), \quad k = 0,\ldots,m, \quad i = j-1, j. \tag{6.8}$$

Time evolution: By rewriting equation (6.1) as $u_t = -f(u)_x$, it is obvious that in order to evolve u^d, we need to compute a polynomial approximation f^d to the flux function $f(u)$. We offer two ways to obtain f^d:

- Modal approach: Perform polynomial operations (addition, substraction, multiplication, and division) on u^d and truncate the resulting polynomial to suitable degree.
- Pseudospectral approach: Compute a local polynomial f_h^* interpolating $f(u^d)$ on a quadrature grid $\mathcal{G}^{ps}$ inside $\mathcal{I}_j^p, j = 1,\ldots,N_x$, and transform f_h^* to a Taylor polynomial f^d.

We differentiate f^d in polynomial sense to get an approximation to the derivative of the flux function $f(u)_x$. We usually use the modal approach unless this option is not applicable. Next, we derive an ODE to evolve u^d, or equivalently the coefficients of the Hermite interpolant $\mathbf{c}(t) = (c_0(t),\ldots,c_{2m+1}(t))^T$, by insisting that the numerical solution u^d satisfy equation (6.1) along with derivatives of (6.1), at the cell centers $x = x_{j-1/2}, j = 1,\ldots,N_x$. The resulting system of ODE for $c_k, k = 0,\ldots,2m+1$, can then be evolved independently on each $\mathcal{I}_j^p$ with any ODE solver. The reconstruction step provides the initial data, $\mathbf{c}(t_n)$.

Repeat on dual grid: At the end of the half time step, we have evolved the degree $2m+1$ polynomial u^d from time $t = t_n$ to $t = t_{n+1/2}$. Before repeating the above process, we truncate $u^d(x, t_{n+1/2})$ by removing the coefficients c_k, $k = m+1, \ldots, 2m+1$. We then repeat the above process (including the truncation) to obtain u^p at time $t = t_{n+1}$, see Figure 6.1 for illustration.

Example: Burgers' Equation

To illustrate the specifics of the time evolution, we consider Burgers' equation, with $f(u) = u^2/2$, approximated by $f_h = u_h^2/2$, where u_h represents the degree $(2m+1)$ interpolant on either of the grids. We can write

$$\begin{aligned}
(u_h)_t + (f_h)_x &= 0, \\
(u_h)_{tx} + (f_h)_{xx} &= 0, \\
(u_h)_{txx} + (f_h)_{xxx} &= 0, \\
\vdots \quad &\quad \vdots
\end{aligned} \tag{6.9}$$

where

$$f_h = \sum_{k=0}^{2m+1} b_k(t)(x - x_{j-1/2})^k.$$

The coefficients b_k are determined by truncated polynomial multiplication, that is $b_k = \frac{1}{2}\sum_{l=0}^{k} c_l c_{k-l}$. Insisting that the approximate solution u_h satisfy equation (6.9) at the cell centers $x = x_{j-1/2}$, we obtain

$$\begin{pmatrix} c_0'(t) \\ c_1'(t) \\ \vdots \\ c_{2m}'(t) \\ c_{2m+1}'(t) \end{pmatrix} = \begin{pmatrix} b_1(t) \\ 2\, b_2(t) \\ \vdots \\ (2m+1)\, b_{2m+1}(t) \\ 0 \end{pmatrix}. \tag{6.10}$$

While equation (6.10) is valid for any cell, the initial data for each cell are different from one cell to another. For a more detailed explanation and open source implementations, see [2, 31, 32].

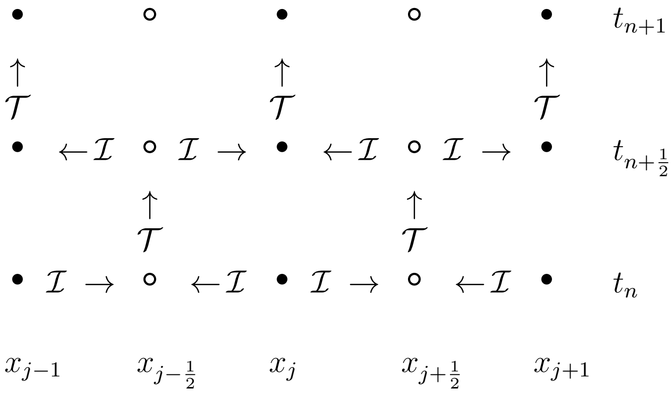


Figure 6.1: Illustration of the numerical process in one dimensional Hermite methods for a full time step. Solid circles represent the primal grid $\mathcal{G}^p$ and open circles represent the dual grid $\mathcal{G}^d$. $\mathcal{I}$ is the Hermite interpolation operator and $\mathcal{T}$ is the time evolution operator.

6.3.2 Flux-Conservative Hermite Methods

The numerical flux f_h obtained by the approach described above, is discontinuous at cell interfaces when the flux function $f(u)$ is nonlinear. Numerically, the discontinuity in the flux induces numerical stiffness. As a result, the time step often needs to be taken very small. To remedy this, we propose new flux-conservative Hermite methods that impose flux continuity by computing the numerical flux at cell edges, and then interpolate the numerical flux to cell center.

To illustrate the difference between the original and flux-conservative Hermite schemes, we plot the numerical flux $f_h = u_h^2/2$ with $m = 3$ and for $N_x = 3$ cells in Figure 6.2. The numerical flux obtained using the original Hermite method, shown as the blue curve, has discontinuities at cell interfaces. On the other hand, the numerical flux obtained by the flux-conservative Hermite method, shown as the black curve, is continuous everywhere.

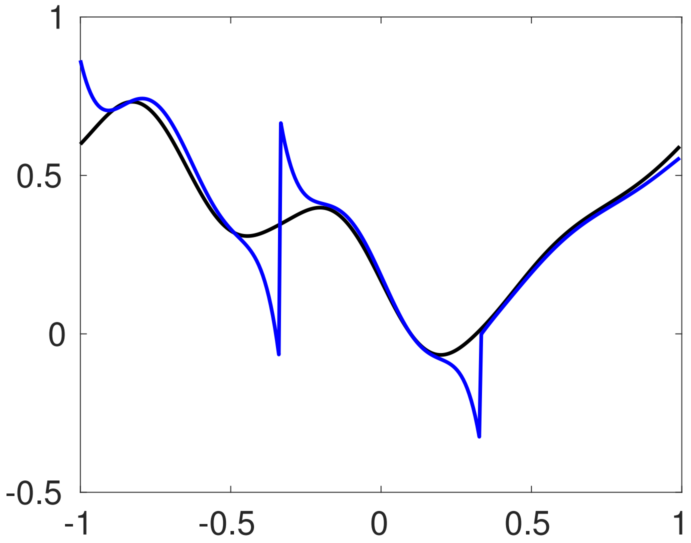


Figure 6.2: Numerical flux for Burgers' equation with random data, obtained by original Hermite (blue), flux-conservative Hermite (black).

The Construction of the New Method

The goal of our construction is to globally conserve the integral of u_h and its m first derivatives over one half time step, with $\Delta\hat{t} = \Delta t/2$.

In the flux-conservative methods, we assume that the solution on the primal grid at time t_n is given by

$$u^p(x, t_n) = \sum_{k=0}^{2m+1} c_k(t_n)\,(x - x_j)^k\,, x \in \mathcal{I}_j^d = [x_{j-1/2}, x_{j+1/2}]. \tag{6.11}$$

Note that the degree of this polynomial is different than in the original method. We assume that the time stepping will be performed by an explicit one step method requiring stage values. The evolution will be carried out at the cell center where the stage values will be the derivative of the Hermite interpolant of the flux. As this interpolant is m times differentiable at the edges, it will result in a conservative discretization. Now, the time evolution of the approximate solution entails the following steps.

Computation of the stage fluxes at the cell edges: For simplicity, assume that we use the classic fourth order Runge-Kutta, then to construct the Hermite interpolants for the four stages we first compute

$$
\begin{aligned}
F_1^p &= f\left(u^p\right), \\
F_2^p &= f\left(u^p + \frac{\Delta\hat{t}}{2}\frac{dF_1^p}{dx}\right), \\
F_3^p &= f\left(u^p + \frac{\Delta\hat{t}}{2}\frac{dF_2^p}{dx}\right), \\
F_4^p &= f\left(u^p + \Delta\hat{t}\frac{dF_3^p}{dx}\right).
\end{aligned}
$$

Note that inside the argument of f, we keep all the coefficients of the polynomials up to degree $2m+1$, while the nonlinearity f itself, which typically is a higher degree polynomial, is truncated to degree $2m+1$.

Reconstruction of solution and fluxes by Hermite interpolation: Next, we construct u^d and F_i^d, $i = 1, \ldots, 4$, the global Hermite interpolation polynomials of degree $(2m+1)$ of the solution and the flux, respectively. Let w^d represent u^d or F_i^d and w^p represent u^p or F_i^p. Then,

$$
w^d(x, t_n) = \sum_{k=0}^{2m+1} d_k(t_n)\left(x - x_{j-1/2}\right)^k, \quad x \in \mathcal{I}_j^p = [x_{j-1}, x_j], \tag{6.12}
$$

where the coefficients $d_k(t_n)$ are uniquely determined by the interpolation conditions (at cell edges)

$$\frac{\partial^k}{\partial x^k} w^d(x_i, t_n) = \frac{d^k}{dx^k} w^p(x_i, t_n), \quad k = 0, \ldots, m, \quad i = j-1, j. \tag{6.13}$$

Time evolution: Let the coefficients of u^d be c_k and the coefficients of F_i^d be $b_k^{(i)}$. Then, again assuming RK4, we have that for $k = 0, \ldots, 2m+1$,

$$c_k(t_n + \Delta\hat{t}) = c_k(t_n) + \frac{\Delta\hat{t}}{6}(k+1)(b_{k+1}^{(1)}(t_n) + 2b_{k+1}^{(2)}(t_n) + 2b_{k+1}^{(3)}(t_n) + b_{k+1}^{(4)}(t_n)).$$

The updated solution on the dual grid is thus

$$u^d(x, t_n + \Delta\hat{t}) = \sum_{k=0}^{2m+1} c_k(t_n + \Delta\hat{t}) \left(x - x_{j-1/2}\right)^k, \quad x \in \mathcal{I}_j^p = [x_{j-1}, x_j].$$

Repeat on dual grid: At the end of the half time step, we have evolved the degree $2m+1$ polynomial u^d. We then repeat the above process to obtain u^p at time $t = t_{n+1}$.

We note that unlike the original method, the number of degrees of freedom that we keep is twice as large, representing an increase in memory requirement. The number of floating point operations, however, to leading order, is the same as for the original method (see the complexity analysis below).

Conservation

We now consider the conservation properties of the above scheme. Due to the fact that the s fluxes used in the stages have m continuous derivatives we immediately find that for periodic problems the following conservation statements hold.

Theorem 1. *Assume we use the flux-conservative Hermite method to evolve $u_t + f(u)_x = 0$ with periodic boundary conditions. Further assume that $u^d(t,x)$ is the* *periodic degree $2m+1$ Hermite interpolating polynomial and that F_i^d, $i = 1, \ldots, s$ are the periodic degree $2m+1$ polynomials Hermite interpolating the fluxes. Further, let the coefficients $c_k(t)$ of u^d on a cell be evolved from time $t = t_n$ to $t = t_n + \Delta\hat{t}$ by the one step method*

$$c_k(t_n + \Delta\hat{t}) - c_k(t_n) + \Delta\hat{t} \sum_{i=1}^{s} \alpha_i (k+1) b_{k+1}^{(i)}(t_n) = 0, \quad k = 0, \ldots, 2m+1,$$

where $b_k^{(i)}$ are the coefficients of F_i^d. Then, the following conservation statement holds.

$$\sum_j \int_{x_{j-1}}^{x_j} \frac{\partial^k}{\partial x^k} u^d(t_n + \Delta\hat{t}, x) dx = \sum_j \int_{x_{j-1}}^{x_j} \frac{\partial^k}{\partial x^k} u^d(t_n, x) dx,$$
$$k = 0, \ldots, 2m+2-s. \quad (6.14)$$

Proof. From the RK time stepping method for conservation laws (6.1) as

$$\frac{u^d(t_{n+1/2}) - u^d(t_n)}{\Delta\hat{t}} = \Delta\hat{t} \sum_i \alpha_i \frac{dF_i^d}{dx}, \quad (6.15)$$

together with the continuity of m first derivatives of each of the F_i^d, the result follows immediately from the update formula. Note also that F_{i+1}^d is one order less accurate than F_i^d due to flux differentiation during stage i. □

To summarize, in the original Hermite scheme, computation of numerical fluxes is performed at the cell center using the interpolated solution. The flux-conservative Hermite scheme requires a computation (and storage) of numerical fluxes at the cell edges and the interpolation of those fluxes to the cell center. Refer to Figure 6.3 for an illustrative comparison between the schemes.

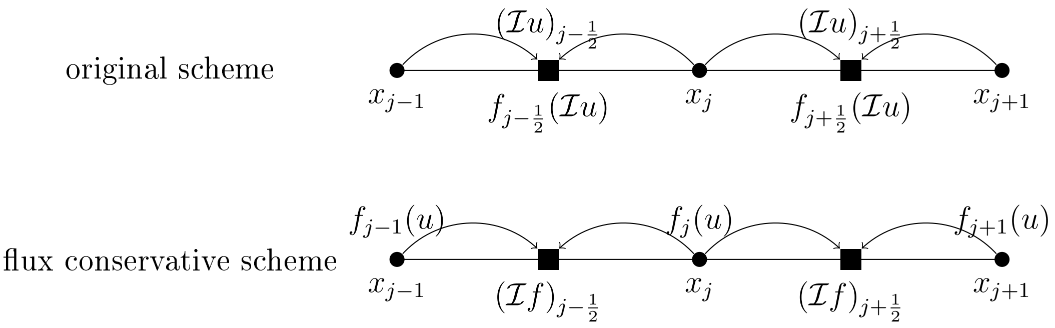

original scheme
$(\mathcal{I}u)_{j-\frac{1}{2}}$
$(\mathcal{I}u)_{j+\frac{1}{2}}$
x_{j-1}
$f_{j-\frac{1}{2}}(\mathcal{I}u)$
x_j
$f_{j+\frac{1}{2}}(\mathcal{I}u)$
x_{j+1}
flux conservative scheme
$f_{j-1}(u)$
$f_j(u)$
$f_{j+1}(u)$
x_{j-1}
$(\mathcal{I}f)_{j-\frac{1}{2}}$
x_j
$(\mathcal{I}f)_{j+\frac{1}{2}}$
x_{j+1}

Figure 6.3: Original vs. Flux Conservative Hermite Methods. Here, we dropped the subscript h in all the computed quantities for compactness.

6.3.3 The Flux-Conservative Hermite Method in Two Dimensions

Now, let us consider a conservation law

$$u_t + (f(u))_x + (g(u))_y = 0, \tag{6.16}$$

on the domain $D = [x_L, x_R] \times [y_B, y_T]$. Let $\mathcal{G}^p$ and $\mathcal{G}^d$ be the primal and dual grids, defined as

$$\mathcal{G}^p = \{(x_i, y_j)\} = (x_L + ih_x, y_B + jh_y),\ i = 0, \ldots, N_x,\ j = 0, \ldots, N_y, \tag{6.17}$$

$$\mathcal{G}^d = \left\{(x_{i+1/2}, y_{j+1/2}) = \left(x_L + \left(i + \frac{1}{2}\right) h_x, y_B + \left(j + \frac{1}{2}\right) h_y\right)\right\}, \\ i = 0, \ldots, N_x - 1,\ j = 0, \ldots, N_y - 1, \tag{6.18}$$

where $h_x = (x_R - x_L)/N_x$ and $h_y = (y_T - y_B)/N_y$ are the distances between two adjacent nodes in x and y directions respectively.

The extension of the flux-conservative method from one dimension is straightforward. Writing equation (6.16) as $u_t = -f(u)_x - g(u)_y$ and letting $u^d(t)$ represent the two-dimensional tensor product Hermite interpolant of the data on the primal grid we can write the RK4 time stepping of $u^d(t_n)$ to time $t = t_n + \Delta\hat{t}$ as

$$\frac{u^d(t_n + \Delta\hat{t}) - u^d(t_n)}{\Delta\hat{t}} = \frac{K_1^d + 2K_2^d + 2K_3^d + K_4^d}{6}. \tag{6.19}$$

The left hand side of equation (6.19) is an approximation to u_t and the right hand side is an approximation to stage values of $-(f(u)_x + g(u)_y)$. Similar to the one dimensional case we have

$$\begin{aligned}
K_1^d &= -(F_1^d)_x - (G_1^d)_y,\\
K_2^d &= -(F_2^d)_x - (G_2^d)_y,\\
K_3^d &= -(F_3^d)_x - (G_3^d)_y,\\
K_4^d &= -(F_4^d)_x - (G_4^d)_y.
\end{aligned}$$

Here, for example, F_i^d is the degree $2m+1$ tensor product polynomial that interpolates $f(u^p + \gamma_i \Delta\hat{t} F_i^p)$ and its m first derivatives in x and y at the four adjacent primal grid-points.

6.3.4 Comparison of Computational Costs

The time evolution portion of the Hermite methods are performed by a one step method with n_K stages, involving computation of the flux function, interpolation of the solution and, in the flux-conservative method, the fluxes, and differentiation of fluxes. For the purpose of this comparison, we assume Burgers' flux function $f(u) = u^2/2$ in 1D or $f(u) = g(u) = u^2/2$ in 2D. Each 1-dimensional Hermite interpolation is equivalent to a multiplication by a $(2m+2)$ by $(2m+2)$ matrix. If we use the recipe above, each 2-dimensional Hermite interpolation corresponds to $2 \times (2m+2)$ one-dimensional interpolations. The factor $(2m+2)$ comes from the the fact that the y dimension brings in $(2m+2)$ interpolations in 1D and the multiplicative factor 2 comes from the fact that we interpolate in y direction on both

	flux computation	**interpolation**
original	$n_K d(2m+2)^{2d}$	$2^{2^{d-1}}(2m+2)^{d+1}$
flux-conservative	$n_K d(2m+2)^{2d}$	$(1+n_K d)2^{2^{d-1}}(2m+2)^{d+1}$

Table 6.1: Comparison of costs in original and flux-conservative Hermite methods, n_K is the number of stages in Runge-Kutta method, d is the spatial dimension.

the left and right edges of the cell. In 3D, we have 4 interpolations in the z direction, each brings in $(2m+2)$ times interpolations in 2D, and so on. We summarize the cost of the method, corresponding to the number of multiplications involved, in Table 6.1. The number of interpolation in the flux-conservative Hermite method is $n_K d\ 2^{2^{d-1}}$ more than the original Hermite method. We note that the flux-conservative scheme can be simplified to just two flux interpolations by adding up the F's and G's, but in this chapter, we interpolate each flux separately. There is also an additional cost of differentiation at cell corners, but it is negligible compared to the cost of interpolation.

6.4 The Entropy Viscosity Method

Given a PDE of the form (6.1), there exists an entropy function $E(u)$ and its corresponding entropy flux function $F(u) = \int E'(u)f'(u)\ du$ such that the entropy residual satisfies

$$r_{EV}(u) \equiv E_t(u) + \nabla \cdot F(u) \leq 0. \tag{6.20}$$

This inequality can be used to select the physically correct solution to (6.1) or (6.16). The direction of the inequality can vary from one problem to another, but the residual takes a nonzero value *only* at shocks. In essence, the entropy viscosity (EV) method uses the fact that the residual approaches a Dirac delta function centered at shocks to construct a nonlinear dissipation. The resulting dissipation is small away from shocks and has a sufficient amount to regularize the shocks. The details of EV for conservation laws are described in detail in [24] but we summarize its most important features here.

Consider the conservation law whose right hand side has been replaced by a viscous term, $u_t + \nabla \cdot f(u) = \nabla \cdot (\nu \nabla u)$, with $\nu = \nu_h(x,t)$ given by

$$\nu_h(x,t) := \min(\nu_{EV}, \nu_{\max}), \tag{6.21}$$

where ν_{EV} is the entropy-based viscosity and $\nu_{\max}$ is a viscosity whose size depends on the largest eigenvalue of the flux function $f(u)$, representing the maximum wave speed. The discretized entropy-based viscosity ν_{EV} is then given by

$$\nu_{EV}(x,t) = \alpha_{EV} C_1(u_h) h^{\beta} |r_{EV}(u_h)|, \tag{6.22}$$

where β is a positive scalar, α_{EV} is a user defined constant and $C_1(u_h)$ is some PDE-specific normalization.

At shocks, the entropy residual approaches Dirac delta function, so we instead use

$$\nu_{\max}(x,t) = \alpha_{\max}\, h \max_{y \in V_x} C_2(u_h(y,t)), \tag{6.23}$$

where $\alpha_{\max}$ is another user defined constant, C_2 serves as the maximum wave speed and V_x is some neighborhood of x. The V_x neighborhood can either be “local”, i.e. containing only a few cells around x, or “global”, i.e. $V_x = D$, where D is the domain of the PDE. In this work, we use global V_x.

In recent papers, the parameter β is chosen to be 2, but we found that this may prevent convergence for moving shocks, see [46] where we also argue that $\beta = 1$ is a more appropriate choice. In essence our argument is as follows. As the entropy residual approaches a Dirac delta distribution, a consistent discretization of the residual with a single shock must satisfy

$$\sum_{j=0}^{N_x-1} h_j (r_{EV})_j = 1, \tag{6.24}$$

where $h_j = x_{j+1} - x_j$. Thus, we expect $(r_{EV})_j \sim h_j^{-1}$ near the shock. When $\beta = 2$, the two terms ν_{EV} and $\nu_{\max}$ are both $\mathcal{O}(h)$. Since the parameters are tuned on a coarse grid and the terms C_1 and C_2 in (6.22)-(6.23) also change with the grid size, the selection of the minimum of ν_{EV} and $\nu_{\max}$ does not necessarily "converge" as the grid gets refined. If instead we choose $\beta = 1$, then $\nu_{EV} = \mathcal{O}(1)$ while $\nu_{\max} = \mathcal{O}(h_j)$, and the particular choice of α_{EV} is thus irrelevant in the limit $h_j \to 0$ as the selection mechanism will eventually select $\nu_{\max}$ at the shocks.

While the explicit formula for C_1 and C_2 varies from one PDE to another, the core of the entropy viscosity method remains the same. The size of the entropy residual gives us a sense of relative distance to the shock, which is then used to take the following actions: near a shock, EV uses sufficiently large dissipation, $\nu_h = \nu_{\max}$, and away from a shock, EV uses entropy-based dissipation, $\nu_h = \nu_{EV}$.

6.5 Numerical Examples

We start by confirming that the rates of convergence, $2m+1$, (in space) are the same for the new flux-conservative method as for the original method.

6.5.1 Convergence for a Smooth Solution

We solve Burgers' equation on the domain $x \in [-\pi, \pi]$ and impose periodic boundary conditions. The initial data is $u(x, 0) = -\sin(x) + 0.3$ and we evolve the solution until time $t = 0.4$. The timestep is chosen as $\Delta t = \mathrm{CFL}\, h_x / \max_x |u(x, 0)|$, with $\mathrm{CFL} = 0.1$.

Table 6.2: Convergence study of smooth solution to Burgers' equation

h_x	$\pi/2$	$\pi/4$	$\pi/8$	$\pi/16$	$\pi/32$
l_∞-err $m=1$	2.30(-01)	5.85(-02)	1.09(-02)	1.42(-03)	1.80(-04)
Rate		1.97	2.43	2.94	2.98
l_∞-err $m=2$	4.85(-02)	5.47(-03)	2.19(-04)	7.25(-06)	1.97(-07)
Rate		3.15	4.64	4.92	5.20
l_∞-err $m=3$	1.09(-02)	6.59(-04)	7.71(-06)	4.70(-08)	2.73(-10)
Rate		4.04	6.42	7.36	7.43

In Table 6.2, we display the maximum error at the final time computed against a reference solution computed using $m = 5$ and $h_x = \pi/64$. We can see from the table that the rates of convergence are consistent with the predicted rate $2m + 1$.

We next present a sequence of experiments displaying the performance of the Hermite-Runge-Kutta-4-Entropy-Viscosity method for Euler's equations (with artificial viscosity).

6.5.2 Euler's equations in One Dimension

We consider Euler's equations which represent conservation of mass, momentum, and energy,

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E+p)u \end{pmatrix}_x = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6.25}$$

Here, ρ is the mass density, ρu is the momentum density, u is the velocity, and E is the internal energy per unit mass. Furthermore, we assume an ideal gas with the equation of state

$$E = \frac{p}{\gamma - 1} + \frac{\rho u^2}{2}, \tag{6.26}$$

where $\gamma = 1.4$ is the adiabatic index and p is the pressure.

To regularize Equation (6.25), we add a viscous term $(\nu(\rho, \rho u, E)_x^T)_x$, where the coefficient ν is obtained using the entropy viscosity method. Thus, the viscous Eulers' equations can be written as

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u - \nu \rho_x \\ \rho u^2 + p - \nu(\rho u)_x \\ (E+p)u - \nu E_x \end{pmatrix}_x = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6.27}$$

We note that an alternative to this simple viscosity would be to use the full Navier-Stokes equations.

Entropy Viscosity (EV) method for 1D Euler's equations

The discretized viscosity coefficient $\nu = \nu_h$ is given in terms of primitive variables $\rho, p,$ and u,

$$\nu_h = \min(\nu_{\max}, \nu_{EV}), \tag{6.28}$$

$$\nu_{EV} = \alpha_{EV}\, h_x\, \rho_h(x,t) |r_{EV}(x,t)|, \tag{6.29}$$

$$\nu_{\max} = \alpha_{\max}\, h_x\, \rho_h(x,t) \max_{y \in D} \left(|u_h(y,t)| + \sqrt{\gamma T_h(y,t)} \right), \tag{6.30}$$

where $T_h = p_h/\rho_h$ is the temperature, h_x is the grid size and

$$r_{EV} = \partial_t S_h + ((uS)_h)_x \geq 0, \tag{6.31}$$

is the entropy residual for the entropy function $S_h(p_h, \rho_h) = \frac{\rho_h}{\gamma - 1} \log\left(\frac{p_h}{\rho_h^\gamma}\right)$ and its corresponding entropy flux $(uS)_h$.

An Improved Entropy Viscosity

The entropy viscosity method discretizes the entropy residual using the numerical solution. In theory, the entropy residual is large at shocks, and zero at contact discontinuities (where no artificial viscosity is needed). However, our experience is that the discretization of the entropy equation may also trigger the maximum viscosity at contact discontinuities. To the left in Figure 6.4, we see a space-time diagram of the entropy residual for Sod's problem in logarithm scale. Note that a relatively large amount of residual is produced at the contact discontinuity.

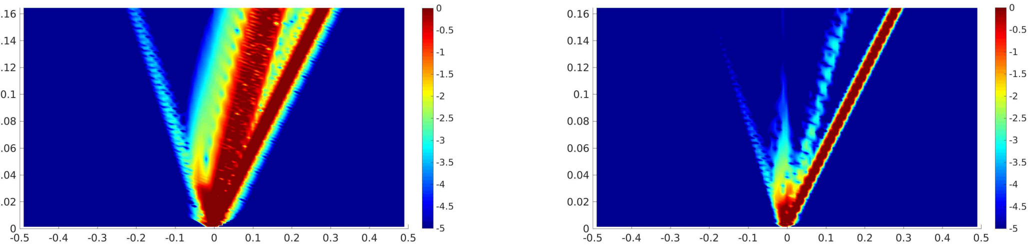


Figure 6.4: Space time diagram of the magnitude of entropy residual $|r_E|$ (left) and $|\Delta u\, r_E|$ (right) for Sod's problem. Blue is small, red is big. Simulations performed with $\nu_S \propto r_E$ (left) and $\nu_S \propto \Delta u\, r_E$ (right). With the new sensor $|\Delta u\, r_E|$, the residual, hence the viscosity is driven to zero along contact discontinuity (thicker red line in the middle disappears with the new sensor).

To eliminate this undesired behavior along the contact discontinuity, we use the fact that the velocity of the fluid, u, is a Riemann invariant along the second characteristic field. Since u is continuous at the contact discontinuity but not at a shock, the product of the increment in the velocity Δu_j and $(r_E)_j$ is small at contact discontinuities but still large at shocks. We incorporate the term Δu into the improved entropy viscosity

$$\nu_{EV} = \alpha_{EV}\, h_x\, \rho_h(x,t)|\Delta u||r_{EV}(x,t)|.$$

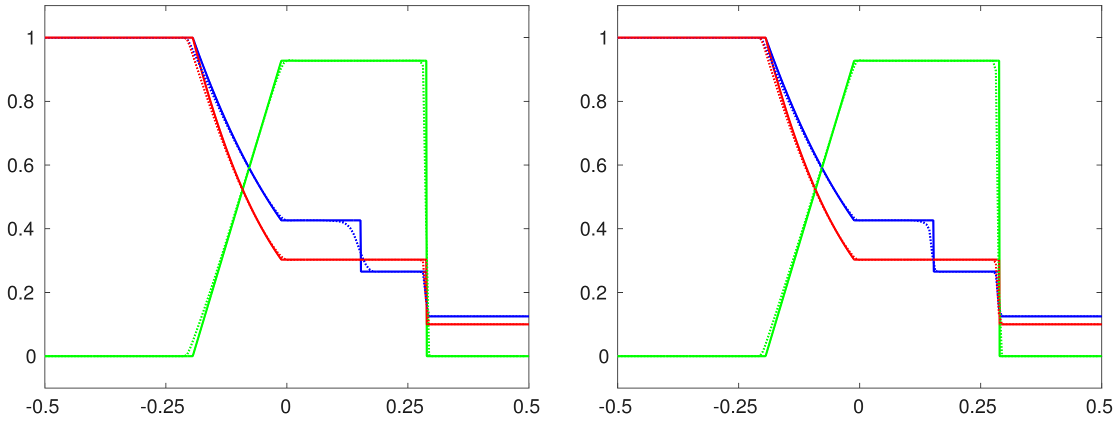


Figure 6.5: Numerical solution (dotted lines) obtained using entropy viscosity proportional to $|r_E|$ (left) and $|\Delta u \, r_E|$ (right) for Sod's problem are plotted against the exact solution (solid lines). We plot density (blue), velocity (green), and pressure (red). Contact discontinuity is sharper using the new sensor $|\Delta u \, r_E|$.

To this end we take ν_E and ν_{max} as a piecewise constant function on each cell. Thus, we compute the discretized density ρ_h, velocity u_h, temperature T_h, entropy function S_h and entropy flux uS_h at cell center in pointwise manner. Now, to get the entropy residual given in (6.31), we compute temporal and spatial derivatives using finite differences. Using the notation $S_h = S_j^n$ to denote the approximate flux function S at $x = x_j$, $t = t_n$, we discretize the term $\partial_t S_j^n$ using second order Backward Difference formula

$$\partial_t S_j^n = \frac{3S_j^n - 4S_j^{n-1} + S_j^{n-2}}{2\Delta t}. \tag{6.32}$$

Similarly, the term $\partial_x (uS)_j^n$ is approximated by the centered finite difference

$$\partial_x (uS)_j^n = \frac{(uS)_{j+1}^n - (uS)_{j-1}^n}{2h_x}. \tag{6.33}$$

h_x	1.25(-02)	6.25(-03)	3.13(-03)	1.56(-03)	7.81(-04)	3.91(-04)
L_1 error	3.44(-03)	2.29(-03)	1.40(-03)	7.96(-04)	4.46(-04)	2.49(-04)
Rate		0.59	0.71	0.81	0.83	0.84

Table 6.3: Convergence study on Euler's equation with stationary shock.

6.5.3 Experiments in One Dimension with Euler's Equations

We now present results obtained using our Hermite-RK4-EV method for a stationary shock, the Lax, the Sod, and the Shu-Osher problem. For experiments where we use more than one resolution, the EV parameters are tuned on the coarsest grid. In the 1D Euler's equations simulations, the timestep is chosen as $\Delta t = \text{CFL}\, h_x / \max_x |(u \pm c)(x,0)|$, where $c = \sqrt{\gamma p/\rho}$ is the speed of sound, with CFL values given in Table 6.3.

Stationary shock problem

By solving the Riemann problem, we decide the states corresponding to a stationary shock. The goal of this experiment is to investigate the stability and accuracy of EV in the presence of shocks. Since small oscillations coming from shocks could potentially pollute the "smooth" regions, this is also a test for how well EV removes numerical artifacts. The computational domain is $D = [-0.5, 0.5]$ with the stationary shock given by

$$(\rho, u, p)(x,t) = \begin{cases} (0.84, 1.08, 0.56) & x < 0, \\ (1, 0.9, 0.71) & x > 0. \end{cases} \tag{6.34}$$

The boundary condition are imposed by setting the solution at the boundary so that it coincides with the solution at initial time. We perform a grid refinement study and report the errors in the density in Table 6.3. We also present the ratio between successive errors. It appears that the rate of convergence for the L_1 error is approaching 7/8.

Lax's and Sod's Shock Tube Problems

Lax's and Sod's problems come from physical experiments in which a gas tube is separated by a membrane into two sections. The gas in each section is uniform in the y and z direction, so the problem is modeled as a 1-dimensional shock tube. The gas in the left section is kept at a different state than the gas in the right section. At time $t = 0$, the membrane is punctured. In the problem setup, the Euler's equations are solved on the domain $D = [-0.5, 0.5]$ with initial data

$$(\rho, u, p)(x, 0) = \begin{cases} (0.445, 0.698, 3.528) & x < 0 \\ (0.5, 0, 0.571) & x > 0 \end{cases} \tag{6.35}$$

for Lax, and

$$(\rho, u, p)(x, 0) = \begin{cases} (1, 0, 1) & x < 0 \\ (0.125, 0, 0.1) & x > 0 \end{cases} \tag{6.36}$$

for Sod. For both problems, we impose fixed boundary condition so that the solution on the boundary is the same as at the initial time. The solution is computed up to time $t = 0.16$ for Lax's problem and time $t = 0.1644$ for Sod's problem.

The solution to Riemann problems such as Lax's and Sod's shock tubes contains 3 waves propagating from the discontinuity at the initial time. The second wave is a contact discontinuity, where the discontinuity is translated over time. The first and third waves are nonlinear, and can take either rarefaction waves or shock waves.

The results for density ρ, velocity u and pressure p are plotted against the exact solution in Figure 6.6. In each plot, we use $N_x = 100$ elements. The entropy viscosity parameters used are given in Table 6.4. In both problems, the shocks are

Problem	CFL	m	α_{EV}	α_{max}
Lax	0.2	3	0.5	0.08
Sod	0.15	3	0.2	0.08
Shu-Osher	0.15	3	0.01	0.05
Stationary shock	0.2	3	10	0.3

Table 6.4: Parameters for examples in 1D Euler's equations.

resolved better than the contact discontinuities. Although the shock strength is only of medium size for both problems, some experts considered these tough test cases for non-characteristic-based high order schemes [61].

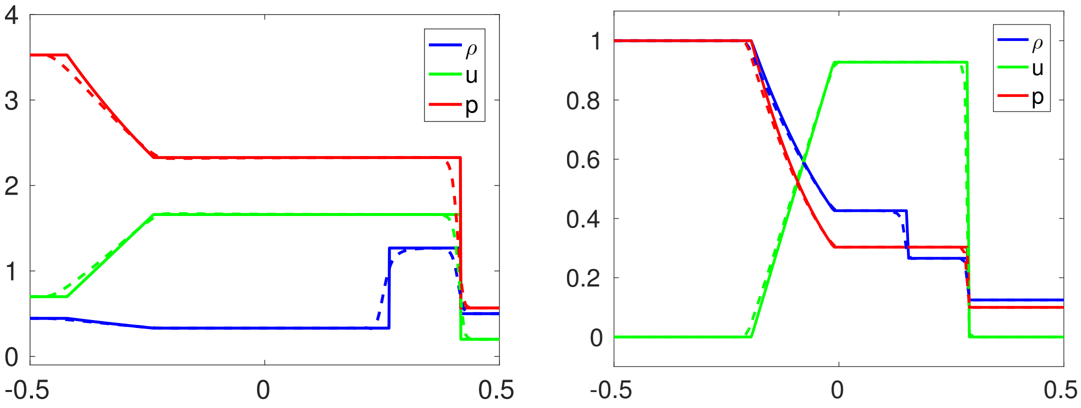


Figure 6.6: Left: Lax shock tube, right: Sod shock tube. Dashed lines are the numerical solutions, solid lines are the exact solutions. Numerical solutions are obtained using $N_x = 100$ elements. The color blue represents density, green represents velocity, and red represents pressure.

The Shu-Osher Problem

The Shu-Osher problem poses difficulties for numerical methods due to the sinusoidal interacting with the shock. Here the domain is $D = [-5, 5]$ with initial data

$$(\rho, u, p)(x, 0) = \begin{cases} (3.86, 2.63, 10.33) & x < -4, \\ (1 + 0.2\sin(5x), 0, 1) & x > -4, \end{cases} \tag{6.37}$$

and with fixed boundary condition so that the solution on the boundary coincides with the solution at the initial time. The solution is computed up to time $t = 1.8$ and compared against a computed solution on a much finer grid. We use $N_x = 80$ to obtain the numerical solution in Figure 6.7, where we interpolate the solution on to a finer grid. The "exact" solution is computed on a grid with $N_x = 1280$. Note that even if we use a coarse grid, we can still get roughly the shape of the solution, especially away from the shock. However, when smooth waves are present (see blue oscillatory line to the left of shock) and too close to the shock, these waves get damped.

6.5.4 Euler's equations in Two Dimensions

The two dimensional inviscid Euler equations are given by

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (E+p)u \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (E+p)v \end{pmatrix}_y = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6.38}$$

Here, ρ is the mass density, ρu and ρv are the momentum density, u and v are the velocity in x and y directions respectively and E is the internal energy per unit mass. Furthermore, we assume an ideal gas with equation of state

$$E = \frac{p}{\gamma - 1} + \frac{\rho(u^2 + v^2)}{2}, \tag{6.39}$$

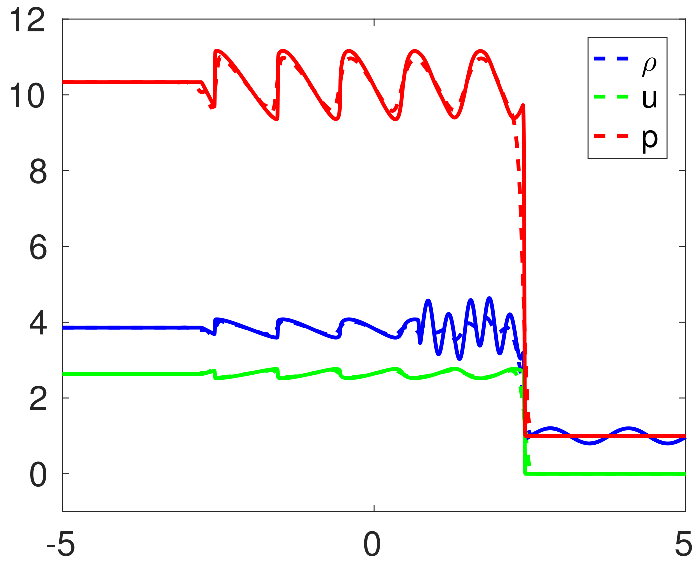


Figure 6.7: Shu-Osher problem. Dashed lines are the numerical solutions, solid lines are the “exact” solutions. Numerical solutions are computed using $N_x = 80$ elements, “exact” solutions are computed on a much finer grid, with $N_x = 1280$ elements. The color blue represents density, green represents velocity, and red represents pressure.

where $\gamma = 1.4$ is the adiabatic index and p is the pressure. For all experiments below, the timestep is chosen as $\Delta t = \text{CFL}\, h_x / \max_x |(u \pm c)(x, 0)|$, where $c = \sqrt{\gamma p/\rho}$ is the speed of sound, with CFL values given in Table 6.5. We then add viscosity term $\nabla \cdot (\nu \nabla (\rho, \rho u, \rho v, E)^T)$ to regularize the inviscid equation.

Problem	CFL	m	α_{EV}	α_{max}
Explosion/implosion	0.2	3	0.1	0.2
Vortex-shock interaction 1	0.2	3	0.01	0.04
Vortex-shock interaction 2	0.2	3	0.05	0.07
Jet	0.2	3	0.03	0.2

Table 6.5: Parameters for examples in 2D Euler’s equations.

6.5.5 Entropy Viscosity Method for Euler's Equations in Two Dimensions

The entropy viscosity is identical to the 1D version given in (6.28), with the exception that it takes the velocity in both directions into account.

$$\nu_h = \min(\nu_{\max}, \nu_{EV}), \tag{6.40}$$

$$\nu_{EV} = \alpha_{EV}\, h\, \rho_h(x,t)(|\Delta u| + |\Delta v|)|r_{EV}(x,t)|, \tag{6.41}$$

$$\nu_{\max} = \alpha_{\max}\, h\, \rho_h(x,t) \max_{y\in D} \left(\sqrt{u_h^2(y,t) + v_h^2(y,t)} + \sqrt{\gamma T_h(y,t)} \right), \tag{6.42}$$

where $T_h = p_h/\rho_h$ is the temperature, $h = \min(h_x, h_y)$ is the grid size, Δu and Δv are the value of the jumps in the velocity in x and y directions respectively, and

$$r_{EV} = \partial_t S_h + ((uS)_h)_x + ((vS)_h)_y \geq 0. \tag{6.43}$$

To discretize the entropy residual r_{EV}, we again use BDF for the time derivative and centered finite differences for the spatial derivatives,

$$\partial_x (uS)^n_{jk} = \frac{(uS)^n_{j+1,k} - (uS)^n_{j-1,k}}{2h_x}, \tag{6.44}$$

$$\partial_y (vS)^n_{jk} = \frac{(vS)^n_{j,k+1} - (vS)^n_{j,k-1}}{2h_y}. \tag{6.45}$$

On the domain $[x_L, x_R]\times[y_B, y_T]$, we use the subscript jk to indicate that the variable attached is evaluated at $x = x_L + jh_x$ and $y = y_B + kh_y$.

Explosion/Implosion Test

First we solve a radially symmetric Riemann problem from Toro [65]. The computational domain is $D = [-1,1] \times [-1,1]$, and the initial data corresponding to an expanding wave is

$$(\rho, u, v, p)(r,t) = \begin{cases} (1,0,0,1) & r < 0.4, \\ (1,0,0,0.1) & r > 0.4. \end{cases} \tag{6.46}$$

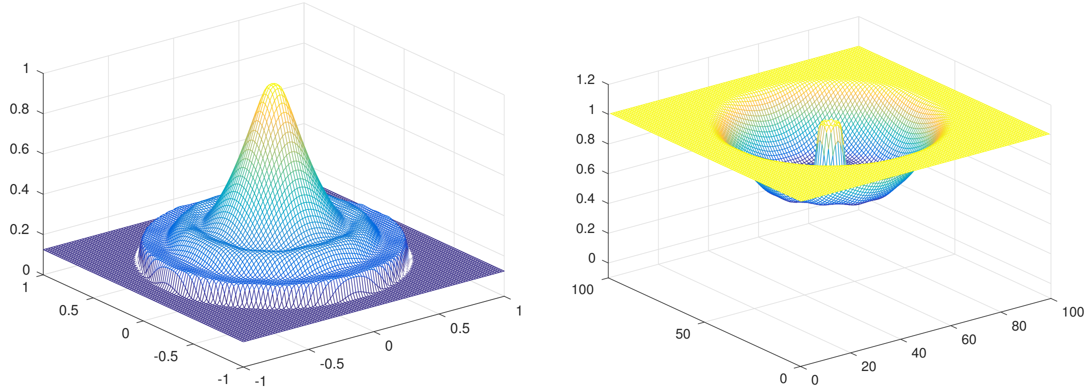


Figure 6.8: Solution to explosion/implosion problem at time $t = 0.25$. To the left: explosion, to the right: implosion. The numerical solutions (circles) are computed using $N_x = N_y = 100$ elements.

For an imploding wave, the initial data is,

$$(\rho, u, v, p)(r, t) = \begin{cases} (1, 0, 0, 1) & r > 0.4, \\ (1, 0, 0, 0.1) & r < 0.4. \end{cases} \tag{6.47}$$

The boundary conditions are imposed by setting the solution on the boundary so that it stays the same as the solution at the initial time. The simulation is performed until time $t = 0.25$, before any waves reach the boundary of the domain. We plot the 2D solution in Figure (6.8). In Figure (6.9), we present a cross section of the density at time $t = 0.25$ with $N_x = N_y = 100$ elements against computed “exact” solution obtained with $N_x = N_y = 400$ elements.

Shock Vortex Interaction

Next we consider the interaction of a shock and a vortex. In general shock-vortex interactions can produce small scales in the form of acoustic waves, and other interesting wave patterns. It has received substantial interest in the literature, see for example [14, 15, 57, 74].

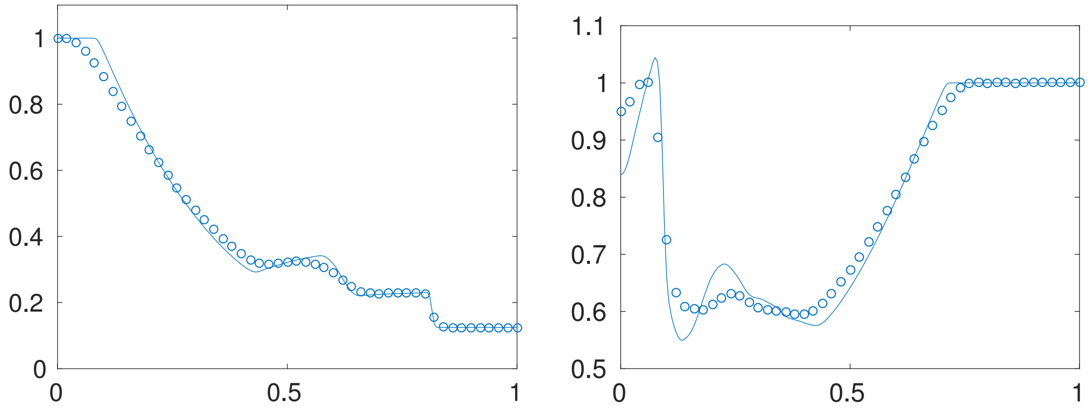


Figure 6.9: Cross section of the density for radially symmetric problem along x axis at time $t = 0.25$. To the left: explosion, to the right: implosion. The numerical solutions (circles) are computed using $N_x = N_y = 100$ elements, "exact" solutions (solid lines) are computed using $N_x = N_y = 400$ elements.

In this experiment, a strong stationary shock with Mach number $2/\sqrt{1.4} \approx 1.69$ collides with a weak vortex with a Mach number $6/2\pi \approx 0.81$. The computational domain is $D = [-9, 3] \times [-4, 4]$ and the initial data is

$$(\rho, u, v, p)(r, t) = \begin{cases} (\rho_{vor}, u_{vor}, v_{vor}, p_{vor}) & x > -4, \\ (2.18, -0.92, 0, 3.17) & x < -4, \end{cases} \tag{6.48}$$

where

$$\rho_{vor} = \left[1 - \frac{(\gamma-1)\beta^2}{8\gamma\pi^2} e^{1-x^2-y^2}\right]^{1/(\gamma-1)} \tag{6.49}$$

$$u_{vor} = 2 - \frac{\beta}{2\pi} y e^{(1-x^2-y^2)/2} \tag{6.50}$$

$$v_{vor} = \frac{\beta}{2\pi} x e^{(1-x^2-y^2)/2} \tag{6.51}$$

$$p_{vor} = \rho^{\gamma}, \tag{6.52}$$

and $\beta = 6$.

As the vortex passes through the shock, the shock is distorted and the vortex is compressed into an elliptical shape. This phenomena is due to the fact that the vortex is relatively weak compared to the shock. The results are consistent with [74]. In Figure 6.10, we compare snapshots of the density Schlieren using two different sets of entropy viscosity parameters, see Table 6.5. Although the schlierens are plotted on the same color scale, notice that the structures are more pronounced in the pictures on the right column.

Fluid Flow in Jet

As a final example we simulate a planar Mach $2/\sqrt{1.4} \approx 1.69$ jet. The domain $D = [-15, 55] \times [-17.5, 17.5]$ is discretized using of 500×250 cells. The initial data is

$$(\rho, u, v, p)(x, y, t) = (1, 0, 0, 1). \tag{6.53}$$

We model the jet nozzle by a simple momentum forcing over a 1×1 patch at the left edge of the computational. The jet is started impulsively causing a relatively strong compression to be generated. This wave sharpens up to a shock wave that is handled by the entropy viscosity as it is propagated from the nozzle and out into a damping absorbing layer of super-grid type, see [1].

In Figures (6.11)-(6.13) we display snapshots of the vorticity, dilatation and density Schlieren. We note that the viscosity we use here is purely for the regularization of shocks so there is no reason to believe that the flow that we compute resembles reality. Nevertheless, the example illustrates the new methods ability to handle rapidly started flows. Also, it is likely that the particular form of the artificial viscosity does not effect the robustness of the method.

6.6 Conclusion

In conclusion we have demonstrated that flux-conservative Hermite methods are suitable for solving nonlinear conservation laws, especially in the presence of shocks. The new methods still converges at a rate of $(2m+1)$ for smooth problems.

The adaptation of the entropy viscosity method to Hermite methods successfully suppresses oscillations near shocks, but we find that our current implementation is quite dissipative when solving the Shu-Osher problem. For contact waves we proposed a modification to the entropy viscosity method which eliminates a large amount of the spurious viscosity at contact discontinuities.

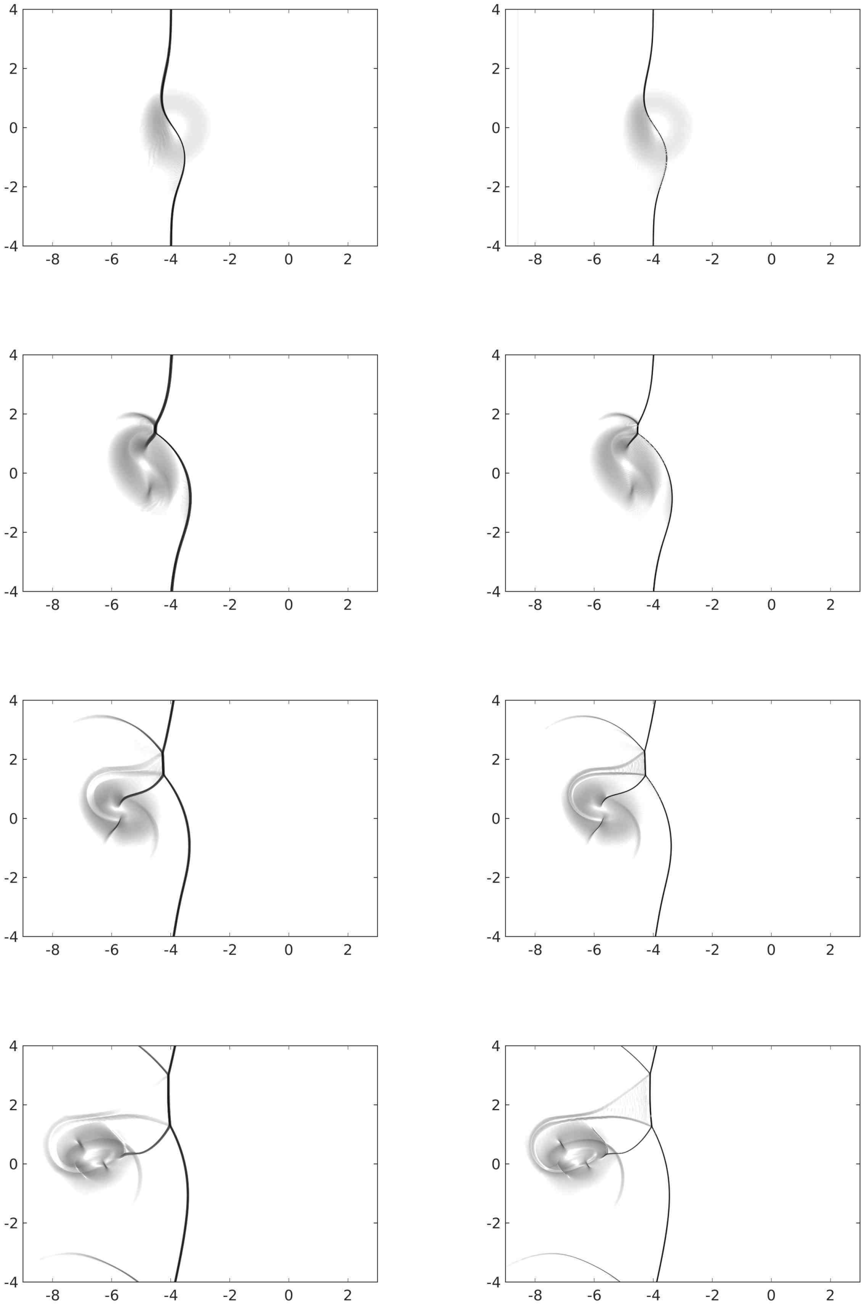


Figure 6.10: The density schlieren at different times, from top to bottom $t \approx 1.97$, $t \approx 2.95$, $t \approx 3.94$ and $t \approx 4.92$. Left: vortex shock interaction 1, right: vortex interaction 2, with parameters given in Table 6.5. The numerical solutions are obtained with $N_x = 720$, $N_y = 480$ elements.

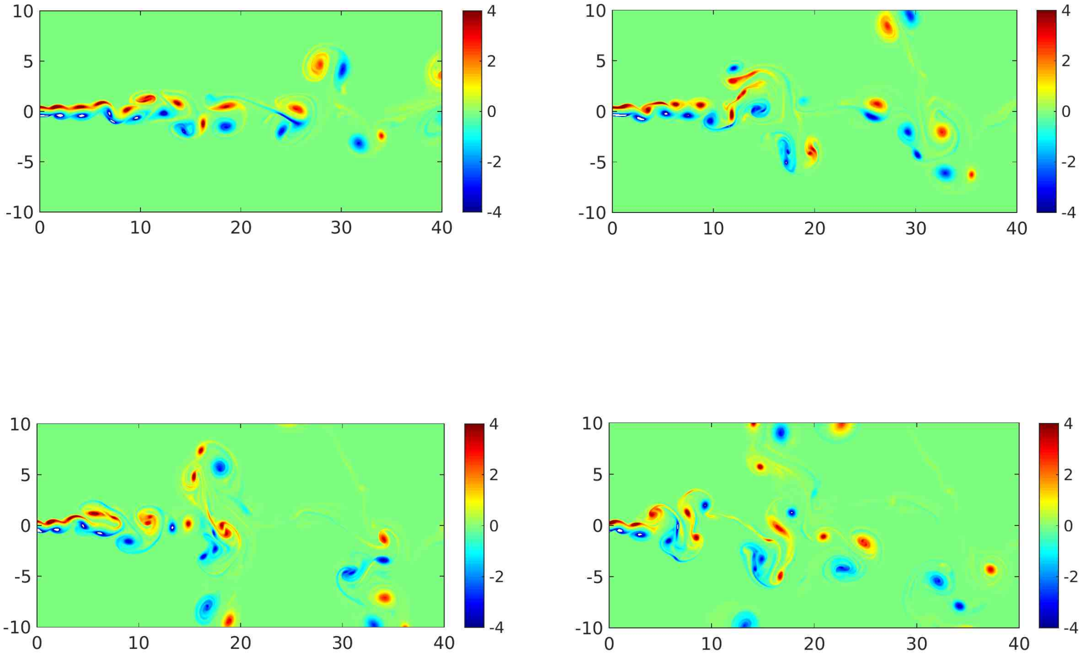


Figure 6.11: The vorticity at different times, from left to right, top to bottom $t = 103.26$, $t = 118.93$, $t = 134.89$, and $t = 150.14$. The numerical solutions are obtained with $N_x = 500$, $N_y = 250$ elements.

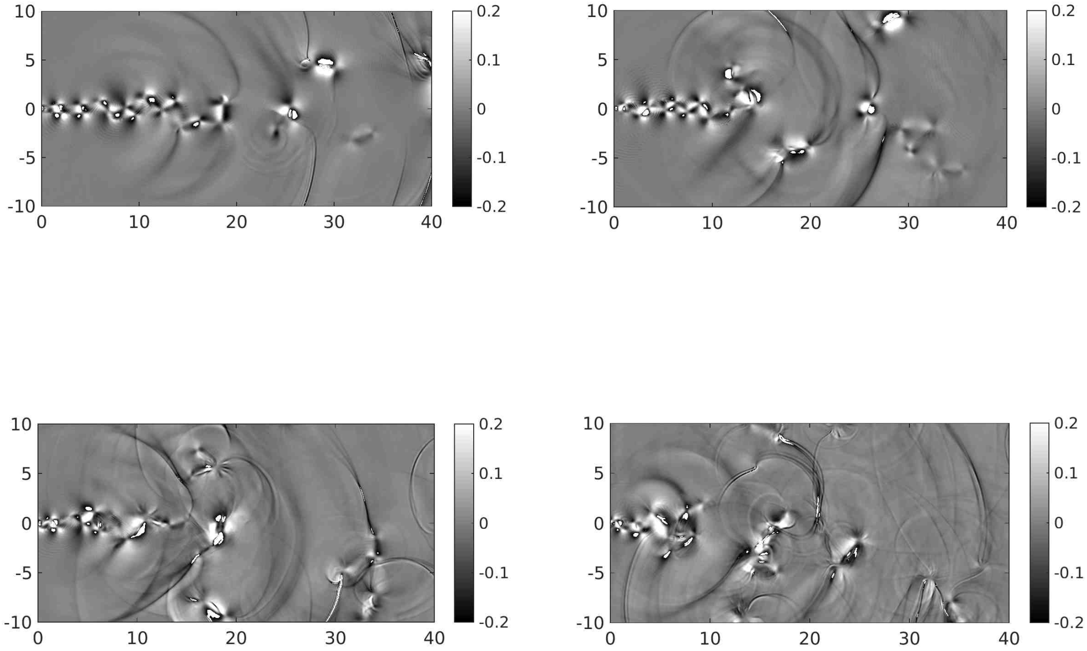


Figure 6.12: The dilatation at different times, from left to right, top to bottom $t = 103.26$, $t = 118.93$, $t = 134.89$, and $t = 150.14$. The numerical solutions are obtained with $N_x = 500$, $N_y = 250$ elements.

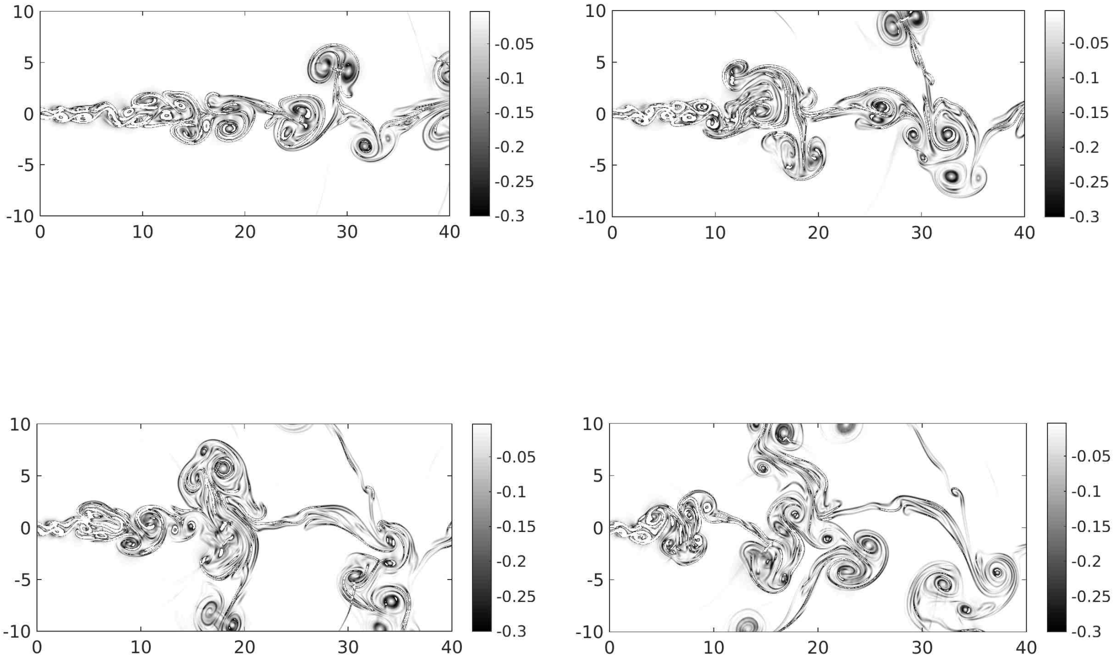


Figure 6.13: The density schlieren at different times, from left to right, top to bottom $t = 103.26$, $t = 118.93$, $t = 134.89$, and $t = 150.14$. The numerical solutions are obtained with $N_x = 500$, $N_y = 250$ elements.

Chapter 7

On the scaling of entropy viscosity in high order methods

This chapter has been accepted for publication as part of ICOSAHOM 2016 proceeding. To appear in Springer Lecture Notes in Computational Science and Engineering (Book 119) [46].

Abstract: In this work, we outline the entropy viscosity method and discuss how the choice of scaling influences the size of viscosity for a simple shock problem. We present examples to illustrate the performance of the entropy viscosity method under two distinct scalings.

7.1 Introduction

Hyperbolic partial differential equations (PDE) are used to model various fluid flow problems. In the special case of 1-dimensional linear constant coefficient scalar hyperbolic problems, the solutions to these PDE are simply a translation of the initial data. However, for nonlinear problems the solution may deform, and as a result, shock waves can form even if the initial data is smooth [50].

In computational fluid dynamics, it is desirable that numerical methods capture shock waves and maintain a high accuracy for smooth waves. Low order methods have sufficient numerical dissipation to regularize shock waves but obtaining accurate solutions in smooth regions can become expensive. On the other hand, high order methods are capable of achieving high accuracy at a reasonable cost. Their low numerical dissipation enables such accuracy, but on the downside, it limits their ability to regularize shock waves.

Various techniques have been implemented to capture shocks while maintaining high accuracy, at least away from shocks. There are two major classes of shock capturing techniques: shock detection techniques, where we find slope limiters [50], Essentially Non-Oscillatory (ENO) and Weighted ENO (WENO) [60], and artificial viscosity techniques, where we find filtering [56,72], the PDE-based viscosity method [41], the entropy viscosity method [24], among others.

In this work, we focus on the entropy viscosity method. In essence, the entropy viscosity method provides shock capturing without compromising the high accuracy away from the shock. An important advantage of this method is that it generalizes very easily to higher dimensions and unstructured grids.

As a model problem, we consider Burgers' equation

$$u_t + f(u)_x = 0, \tag{7.1}$$

where $f = \frac{u^2}{2}$. Physically correct solutions to (7.1) can be singled out by requiring that they satisfy an entropy inequality such as

$$r_{EV} = E_t + F_x \equiv \left(\frac{u^2}{2}\right)_t + \left(\frac{u^3}{3}\right)_x \leq 0. \tag{7.2}$$

The entropy residual, r_{EV}, is zero wherever u is smooth. If the solution u contains a shock, then the entropy residual takes the form of a negative Dirac distribution centered at the location of the shock, x_s, i.e. $r_{EV} = -C\,\delta(x - x_s)$. The property that the entropy residual is unbounded at a shock was first used by Guermond and Pasquetti in [24], as a way to selectively introduce viscosity. The artificial viscosity, ν, proposed in [24], defined as the minima of two viscosities

$$\nu = \min(\nu_{\max}, \nu_{EV}), \tag{7.3}$$

becomes the coefficient of the viscous term in the viscous Burgers' equation,

$$u_t + f(u)_x = (\nu u_x)_x. \tag{7.4}$$

Here, $\nu_{\max}$ is the Lax-Friedrich viscosity whose size depends on discretization and the largest eigenvalue (in absolute value), λ_{LF}, of the flux Jacobian, $\frac{Df(u)}{Du}$. The second viscosity ν_{EV} is proportional to the magnitude of the entropy residual (in fact, a discretization of the entropy residual) and will thus be zero (or small after discretization) away from discontinuities. In theory, the entropy residual becomes unbounded at a shock, numerically however, the entropy residual r_{EV} remains bounded with the size of the residual depending on the discretization size. As we will see below, this subtle difference has consequences for how to choose the scaling of the viscosity terms in the entropy viscosity method.

On a grid with step size h, the second viscosity ν_{EV} can be expressed as

$$\nu_{EV}(x) = \alpha_{EV} h^{\beta} |r_{EV}(x)|, \tag{7.5}$$

with a parameter α_{EV} that requires tuning. In recent papers on entropy viscosity method, see e.g. [26–29, 76], the parameter β is chosen to be 2, but the original paper [25] uses $\beta = 1$. It is unclear to us why the later works prefer $\beta = 2$. Here, we will present analysis and computational results that suggest the original scaling $\beta = 1$ is a more natural choice. We note that the entropy residual is typically scaled by $\|E - \overline{E}\|_\infty$, with the over-bar indicating a spatial average, but as this quantity is roughly constant in the problems presented here, we omit it for simplicity and without consequence.

The rest of the chapter is organized as follows. In Section 7.2, we describe different discretizations of (7.4) that we consider in this chapter; in Section 7.3, we present an analysis of how the entropy viscosity ν depends on the two viscosities, ν_{EV} and $\nu_{\max}$, under different scaling for a model problem. In Section 7.4, we then conduct experiments with the entropy viscosity method where β takes on values 1 or 2 and compare the results.

7.2 Numerical methods

We will consider the discretization of (7.4) by our conservative Hermite method [45], a standard discontinuous Galerkin (dG) method [39] and a simple finite volume type discretization [50]. For all the discretizations we let the domain $x_L \leq x \leq x_R$ be discretized by the regular grid $x_i = x_L + i\,h$, $i = 0, \ldots, n$, $h = (x_R - x_L)/n$.

The degrees of freedom for the finite volume method are cell averages centered at the grid points. For the Hermite method, the degrees of freedom are the coefficients of node centered Taylor polynomials of degree m and for the dG method, they are the $(m+1)$ coefficients of element-wise (we take an element to be $\Omega_i = [x_{i-1}, x_i]$) expansions in Legendre polynomials. For smooth solutions the spatial accuracy of the Hermite method is $2m+1$ and $m+1$ for the dG method.

All three methods use the classic fourth order Runge-Kutta method to evolve the semi-discretizations in a method-of-lines fashion.

In the Hermite method, we evaluate the fluxes and their derivatives at the nodes (element edges) for the four stages in the RK method. Precisely, for the first stage we compute the slope $f_1^h = \frac{1}{2}\mathcal{T}[(u_1^h)^2] - \frac{\nu}{h}\frac{du_1^h}{dx}$ for the Taylor polynomial $u_1^h = u^h$ approximating the solution at the first stage. Here $\mathcal{T}[(u_1^h)^2]$ is the truncated polynomial multiplication of u_1^h with itself and $\frac{du_1^h}{dx}$ is the derivative of the polynomial. At the next stage, the solution is $u_2^h = u^h + \frac{(\Delta t/2)}{2}\frac{df_1^h}{dx}$, the slope is $f_2^h = \frac{1}{2}\mathcal{T}[(u_2^h)^2] - \frac{\nu}{h}\frac{du_2^h}{dx}$ and so on. Once the stage slopes f_s^h, $s = 1, \ldots, 4$ and their spatial derivatives are known, we perform a Hermite interpolation to the element centers of the solution and the spatial derivatives of the stage slopes. These are then used to evolve the element centered Hermite interpolant of u^h to $t = t_n + \Delta t/2$. As the Hermite interpolant is of higher degree than the original Taylor polynomial, we conclude a half-step by truncating it to the appropriate degree. To advance the solution a full time step, the half-step process is repeated starting from the element centers.

To handle the artificial viscosity in the dG method, we use the approach of Bassi and Rebay [6] with a Lax-Friedrichs flux for the advective term and alternating fluxes for the viscous term. The nonlinear terms are constructed explicitly and de-aliased by over-integration [42].

For the finite volume method, we let $u_i \approx u(x_i)$ be a grid function approximating the solution and $f_{i+\frac{1}{2}} = f_{i+\frac{1}{2}}(u_i, u_{i+1})$ be an approximation to the flux at $x_{i+\frac{1}{2}}$. To compute the time derivatives, we use the spatial approximation

$$\frac{du_i}{dt} \approx \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{h}, \tag{7.6}$$

where

$$f_{i+\frac{1}{2}}(u_i, u_{i+1}) = \frac{1}{2}\left(\frac{u_i + u_{i+1}}{2}\right)^2 - \left(\frac{\nu_i + \nu_{i+1}}{2}\right)\frac{u_{i+1} - u_i}{h}. \tag{7.7}$$

When $\nu_i = 0$, the above discretization is linearly stable (when paired with a suitable time-stepping method) but is not non-linearly stable, and we thus add artificial viscosity to stabilize it.

For all three discretizations, we approximate the time derivative of the entropy function, E_t, by a backward difference. This approach is explicit as we use the current solution to compute E at the current time before evolving the solution in time. The residual (and hence the viscosity) is kept on each element / grid-point over each step.

To approximate the entropy flux derivative F_x using the Hermite method, we compute the derivative of the truncated polynomial multiplication $\mathcal{T}[u^h\mathcal{T}[(u^h)^2]]$ at the node. For the dG method, we evaluate the flux F on a Legendre-Gauss-Lobatto (LGL) grid and differentiate it to get an approximation for F_x. The residual on an element is taken to be the maximum of the absolute value of the residual on the LGL grid. In the finite volume method F_x is approximated by

$$\frac{dF_i}{dx} = \frac{F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}}{h}, \text{ where, } F_{i+\frac{1}{2}} = \frac{1}{3}\left(\frac{u_i + u_{i+1}}{2}\right)^3.$$

We note that more sophisticated discretizations of the entropy residual could be considered. In particular, a higher order approximation to r_{EV} would result in a higher rate of convergence for smooth solutions, but as we are mainly concerned with the scaling β, we did not pursue such discretizations here. In fact, in our experience, the results concerning the choice of scaling are not affected by the order of the accuracy of the approximation to r_{EV}. This will be discussed in Section 7.3.

We also define $\nu_{\max}$ to be the classical Lax-Friedrich viscosity, which for Burgers' equation takes the form

$$\nu_{\max} = \alpha_{\max} h \max |u|, \tag{7.8}$$

where the maximum is taken globally.

Finally, for the purpose of comparison we also present some results computed using the sub-cell resolution smoothness sensor of Persson and Peraire, [56]. The smoothness sensor compares the L_2 energy content of the highest (Fourier or expansion) mode with the total L_2 energy on an element and then maps its ratio (which is an indicator of the smoothness) into the size of the artificial viscosity. Precisely, if the approximate dG solution on an element is $u^h = \sum_{k=0}^{m} \hat{u}_k P_k$, with P_k being an orthogonal basis, we compute the smoothness as $s = \log_{10}(\|\hat{u}_m P_m\|^2/\|u^h\|^2)$ and the viscosity as

$$\nu = \begin{cases} 0 & s < s_0 - \kappa, \\ \varepsilon_0 h & s > s_0 + \kappa, \\ \frac{\varepsilon_0 h}{2}\left(1 + \sin\left(\frac{\pi(s-s_0)}{2\kappa}\right)\right) & \text{otherwise.} \end{cases}$$

When applied to the Hermite method, we first project the Taylor polynomials centered at two adjacent grid-points into an orthogonal Legendre expansion on the element defined by the grid-points and then proceed as above.

7.3 Impact of the h-scaling on the selection mechanism

To study how the selection mechanism depends on the shock speed and the size of the jump, consider a solution of the Burgers' equation consisting of a Heaviside function H with left state u_l and right state u_r, given by

$$u(x,t) = u_l + \Delta u H\left(x - v_s t\right). \tag{7.9}$$

This corresponds to a shock of size $|\Delta u| = |u_r - u_l|$ moving with speed $v_s = 0.5(u_l + u_r)$. Solutions of the form (7.9) always have a negative Δu value since Lax entropy condition for Burgers' equation dictates $u_l = f'(u_l) > v_s > f'(u_r) = u_r$.

For simplicity, we use the short hand notation H for $H\left(x - v_s t\right)$. A direct computation

$$
\begin{aligned}
u_t + \left(\frac{u^2}{2}\right)_x &= (u_l + \Delta u H\left(x - v_s t\right))_t \\
&\quad + \frac{1}{2}(u_l^2 + 2u_l \Delta u H\left(x - v_s t\right) + (\Delta u)^2 H\left(x - v_s t\right))_x \\
&= -v_s \Delta u H' + \left(u_l \Delta u + \frac{(\Delta u)^2}{2}\right) H' \\
&= \left(-\frac{u_l + u_r}{2}(\Delta u) H'\right) + \left((\Delta u) u_l H' + \frac{(\Delta u)^2}{2} H'\right) \\
&= -\Delta u \left(\frac{u_l + u_l + \Delta u}{2}\right) H' + \Delta u \left(\frac{2u_l + \Delta u}{2}\right) H' \\
&= 0,
\end{aligned}
$$

shows that (7.9) is a solution of (7.1). Further, it can be shown that the entropy residual (7.2) for (7.9) is

$$
r_{EV} = \frac{(\Delta u)^3}{12} H'(x - x_s) = \frac{(\Delta u)^3}{12} \delta(x - x_s). \tag{7.10}
$$

That is, the size of the entropy residual grows with the cube of Δu.

Now, by the properties that define the Dirac delta function δ, we have

$$
\int_{-\infty}^{\infty} \delta(x) dx = 1. \tag{7.11}
$$

Thus, a consistent discretization of the Dirac delta function $\delta_0, ..., \delta_n$ on a grid $x_0, ..., x_n$ must obey the condition

$$
\sum_{j=0}^{n-1} h_j \delta_j = 1, \tag{7.12}
$$

where $h_j = x_{j+1} - x_j$. For any approximation with a finite width stencil, we must have $\delta_j \sim h_j^{-1}$ and we thus expect r_{EV} to behave like $(\Delta u)^3/h$ on a uniform grid. We therefore proceed with the analysis using the discrete approximation $r_{EV} = (\Delta u)^3/h$. Using this expression for r_{EV}, we estimate the viscosity ν by the minimum of

$$
\nu_{EV} = \alpha_{EV} h^{\beta-1} |(\Delta u)^3| \text{ and } \nu_{\max} = \alpha_{\max} h \max(|u_l|, |u_r|). \tag{7.13}
$$

Table 7.1: Size of μ_E and $\mu_{\max}$ for different size of shock speed (v_s) with respect to the size of the jump (Δu) in the entropy viscosity method.

Case	ν_{EV}	$\nu_{\max}$
$\lvert v_S\rvert \ll \lvert\Delta u\rvert$	$\alpha_{EV} h^{\beta-1}\lvert\Delta u\rvert^3$	$\alpha_{\max} h\lvert v_s\rvert$
$\lvert v_s\rvert \approx \lvert\Delta u\rvert$	$\alpha_{EV} h^{\beta-1}\lvert\Delta u\rvert^3$	$2\alpha_{\max} h\lvert v_s\rvert$
$\lvert v_s\rvert \gg \lvert\Delta u\rvert$	$\alpha_{EV} h^{\beta-1}\lvert\Delta u\rvert^3$	$0.5\alpha_{\max} h\lvert\Delta u\rvert$

The comparison between the size of ν_{EV} and $\nu_{\max}$ in various scenarios is reported in Table 7.1. If $\beta = 2$, then the two viscosities ν_{EV} and $\nu_{\max}$ scale as h. For a problem with multiple shocks, the homogeneity in h-scaling introduces an additional difficulty in determining α_{EV}. Should it be chosen based on the largest or smallest shock? What if new shocks appear during the course of the computation? To avoid answering these questions, we instead consider $\beta = 1$. Now $\nu_{EV} = \mathcal{O}(1)$ while $\nu_{\max} = \mathcal{O}(h)$, and the particular choice of α_{EV} is thus irrelevant since as $h \to 0$, the selection mechanism will eventually select $\nu_{\max}$ at the shocks. We will provide an example to illustrate the two-shock dilemma in Section 7.4.3.

7.4 Experiments

In this section, we describe the experiments and present a convergence study in L_2 norms, and also study the effects of the scaling in the entropy viscosity method on the convergence under grid refinement. For all the examples we solve Burgers' equation and vary the initial data. In each problem, we report the L_2-errors (the L_1-errors behave quantitatively similar).

The solutions are obtained using the following methods: H1 and H2 refer to Hermite-entropy viscosity method for $\beta = 1$ and $\beta = 2$ respectively, DG1 and DG2 refer to dG-entropy viscosity method for $\beta = 1$ and $\beta = 2$ respectively, FV1 and FV2 refer to finite volume-entropy viscosity method with $\beta = 1$ and $\beta = 2$ respectively, DGP and HP refer to dG and Hermite method with smoothness sensor respectively.

The size of the time step is chosen close to the stability limit, which in the cases considered here results in the error being dominated by the spatial discretization.

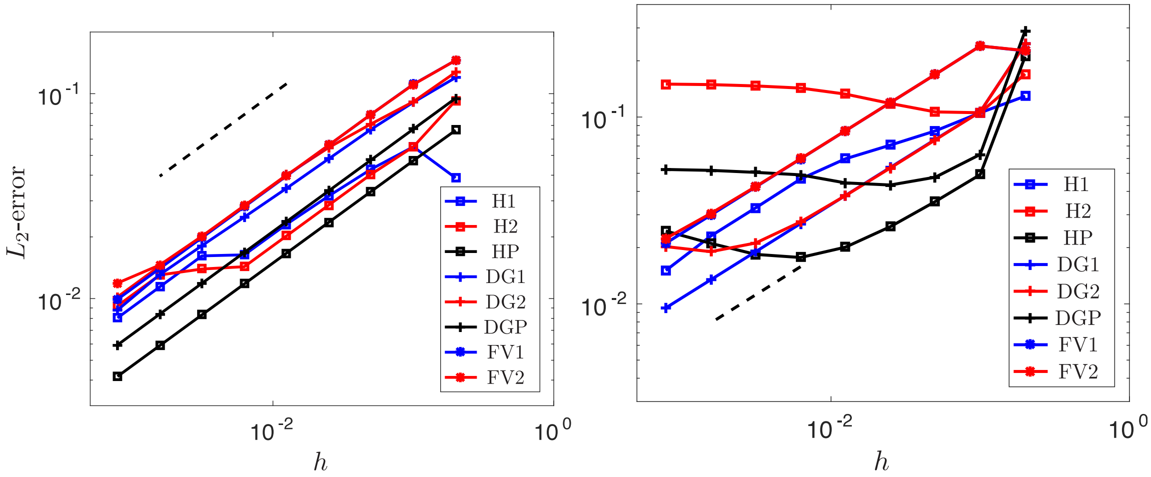


Figure 7.1: Convergence of the different methods for stationary (left) and moving (right) shocks.

7.4.1 A single shock

In this example, we compute the solution to (7.1) on the domain $D = [-1, 1]$ with the initial data imposed as the exact solution

$$u(x,t) = \begin{cases} -0.5 + v_s, & x \in [-1, v_s t), \\ 0.5 + v_s, & x \in [v_s t, 1], \end{cases} \tag{7.14}$$

at time $t = 0$. Here v_s is the shock speed which we choose to be either $v_s = 0$ corresponding to a stationary shock or $v_s = 0.1$ corresponding to a moving shock.

We solve until time $t = 1$ for the two different shock speeds and perform a grid refinement study using a dG method of order 5, a Hermite method of order 9, and the Finite Volume method, all using the classical fourth order Runge-Kutta time stepping. For the Hermite method, we fix $(\max|u|)\Delta t/h = 0.3$, for the dG method, the time step is set as $\Delta t/h = 0.0625$ and for the Finite Volume method, the time step is set according to $(\max|u|)\Delta t/h = 0.9$.

The L_2 norm of errors in the numerical solution u_h are plotted against the different grid sizes for different methods, see Figure 7.1. In the stationary shock experiment, FV1 and FV2 use $(\alpha_{EV}, \alpha_{\max}) = (0.7, 0.5)$ and $(10, 0.5)$ respectively, DG1 and DG2 use $(\alpha_{EV}, \alpha_{\max}) = (1, 0.25)$ and $(10, 0.25)$ respectively, H1 and H2 use $(\alpha_{EV}, \alpha_{\max}) = (1, 0.4)$ and $(10, 0.4)$ respectively, DGP and HP use $(s_0, \kappa, \epsilon_0) = (-1, 2, 0.5)$ and $(\log_{10}(1/256), 1, 0.125)$ respectively.

The parameters for moving shock experiment are $(\alpha_{EV}, \alpha_{\max}) = (0.7, 0.5)$ and $(10, 0.5)$ for FV1 and FV2 respectively, $(\alpha_{EV}, \alpha_{\max}) = (1, 0.25)$ and $(10, 0.25)$ for DG1 and DG2 respectively, $(\alpha_{EV}, \alpha_{\max}) = (1, 0.4)$ and $(10, 0.4)$ for H1 and H2 respectively, $(s_0, \kappa, \epsilon_0) = (2\log_{10}(1/256), 1, 0.5)$ and $(\log_{10}(1/256), 1, 0.125)$ for DGP and HP respectively.

To the left in Figure 7.1, we display convergence results for the stationary shock. In this case, the results indicate that all methods produce convergent solutions with roughly the same rates of convergence. The rate of convergence is limited by the smoothness of the solution but as can be seen in the same figure, the error levels are lower for the higher order methods. It is interesting to note that the smallest errors are observed for the computations using the smoothness-based sensor.

The results for the moving shock, displayed to the right in Figure 7.1, are quite different. Now, for the high order methods, we observe convergence only when we use the entropy viscosity with $\beta = 1$. When we use the entropy viscosity with $\beta = 2$ or when we use the smoothness based sensor, the errors clearly saturate as the grid is refined. The errors for the low order Finite Volume method are still reduced with

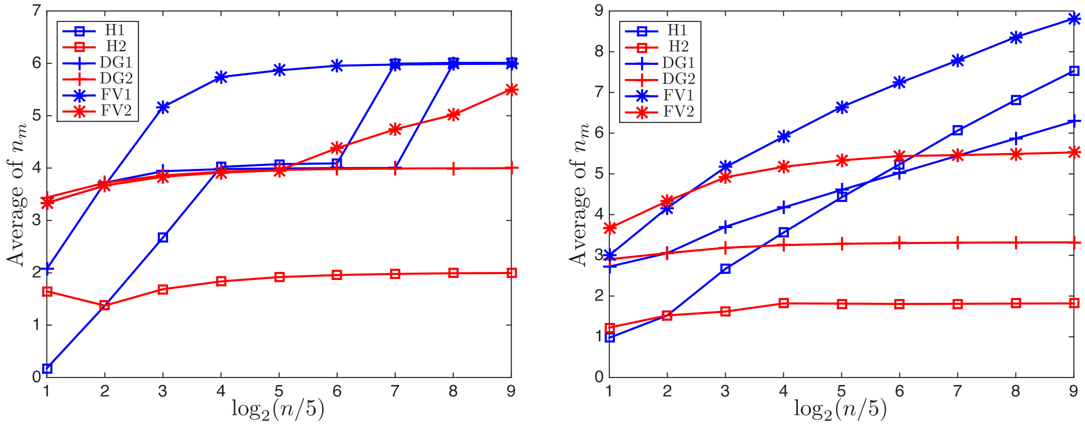


Figure 7.2: Average in time of n_m, number of elements using Lax-Friedrich viscosity $\nu_{\max}$, versus the number of elements (n). Left: stationary shock, right: moving shock.

the grid size, independent of the scaling in the entropy viscosity method.

To understand why the convergence results obtained with $\beta = 1$ and $\beta = 2$ in the moving shock example do not agree, we study where the Lax-Friedrich viscosity $\nu_{\max}$ is activated in the vicinity of the shock. We know that when the viscosity is chosen to be just the Lax-Friedrich type viscosity, then under a suitable Courant number, the solution will converge to the correct vanishing viscosity solution of the conservation law [50].

It seems that the Lax-Friedrich viscosity is necessary in some neighborhood of the shock, and the size of this neighborhood becomes an important factor in the convergence of the solution to the moving shock problem. In Figure 7.2, we plot the average (in time) of the number of elements n_m which use the Lax-Friedrich viscosity $\nu_{\max}$ as a function of total number of elements n for the stationary shock (left) and for the moving shock (right). We see that n_m is roughly constant for both $\beta = 1$ and $\beta = 2$ in the stationary shock. In the moving shock problem, n_m stays constant for $\beta = 2$ as in the stationary shock, but grows slowly for $\beta = 1$ (note the log-scale). While the growth in n_m is irrelevant in the convergence in the stationary shock example, it seems to play an important role in determining the convergence in the moving shock example.

7.4.2 Sinusoidal to N wave

Next, we consider the smooth 2-periodic initial data

$$u(x,0) = -\sin(\pi x) + 0.5, \tag{7.15}$$

which develops into a single N wave.

In Figure 7.3, we present the L_2 norm of the errors at $t = 0.1$ before the shock forms (left) and at $t = 1$ after the shock forms (right). The spatial and temporal discretization of the PDE itself is performed with a high order method, so rate of convergence that we observe in Figure 7.3 is limited by either the discretization of the artificial viscosity or the smoothness of the solution, whichever is more restrictive.

For this N-wave experiment, $FV1$ and $FV2$ use $(\alpha_{EV}, \alpha_{\max}) = (2, 0.5)$ and $(20, 0.5)$ respectively, $DG1$ and $DG2$ use $(\alpha_{EV}, \alpha_{\max}) = (0.1, 0.125)$ and $(1, 0.125)$ respectively, $H1$ and $H2$ use $(\alpha_{EV}, \alpha_{\max}) = (0.4, 0.4)$ and $(5, 0.4)$, DGP and HP use $(s_0, \kappa, \epsilon_0) = (2\log_{10}(1/256), 2, 0.05)$ and $(\log_{10}(1/256), 1, 0.125)$ respectively.

The discretization of the entropy residual r_{EV} is only first order due to the use of backward-Euler, so we expect the entropy-based viscosity ν_{EV} to be $(\beta+1)^{th}$ accurate, i.e. 2^{nd} order when $\beta = 1$ or 3^{rd} order when $\beta = 2$. This analysis agrees with the convergence plot to the left in Figure 7.3. To the right, we observe the same phenomena as in the moving shock example described in Section 7.4.1. We also note that the shock in this sinusoidal wave is also moving.

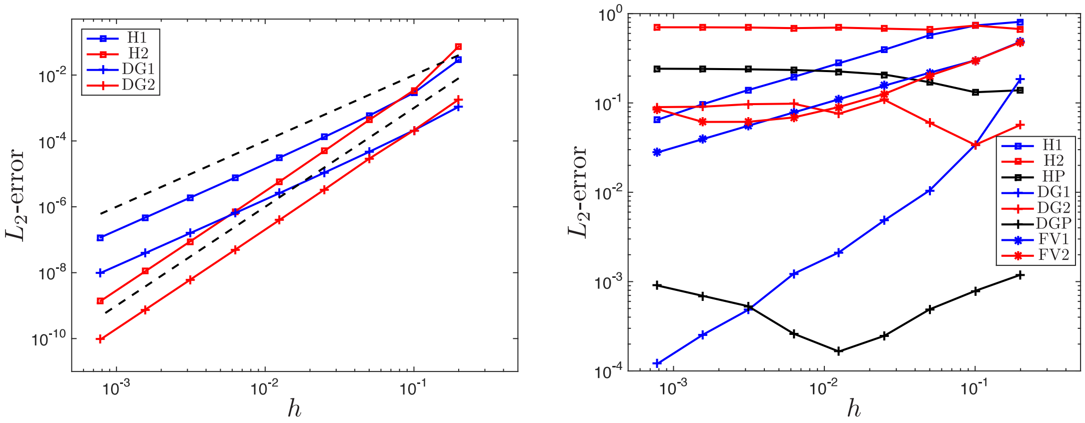


Figure 7.3: Convergence of the different methods for a smooth initial data, left: before shock forms, right: after shock forms. The dashed lines are h^2 and h^3.

7.4.3 Shocks of different size

To complement the analysis in Section 7.3, we next consider a problem with a big shock and a small shock on the same simulation. According to the analysis, the entropy viscosity will capture the small shock when $\beta = 1$, but not when $\beta = 2$. In this setup, we start with an existing shock of size $\Delta u_1 = 0.5$ and a small sinusoidal wave that develops into an N-wave of size $\Delta u_2 = 0.2$. Thus, we consider Burgers' equation on $[-1, 5]$ with initial data

$$u(x,0) = \begin{cases} 0 & x \in [-1,-0.5), \\ -0.1\sin(2\pi x) & x \in [-0.5,0.5), \\ 0 & x \in [0.5,4.5), \\ -0.5 & x \in [4.5,5], \end{cases} \tag{7.16}$$

and fixed boundary condition $u(-1,t) = 0$ and $u(5,t) = -0.5$.

The solution initially consists of a shock and a smooth sine wave, which are placed far away from each other so they never interact. Over time, the sinusoidal wave develops into a N-wave. In Figure 7.5, we present the numerical solutions at time $t = 2$ for different grid resolutions, obtained with a Hermite method of order 9 and dG method of order 5. In these plots, we can see that the shock is resolved for both values of β, however, the N-wave comes with some overshoots when $\beta = 2$ for all the finer grid resolutions, see Figure 7.5.

For this two-shock experiment, $DG1$ and $DG2$ use $(\alpha_{EV}, \alpha_{\max}) = (0.5, 0.25)$ and $(10, 0.25)$ respectively, $H1$ and $H2$ use $(\alpha_{EV}, \alpha_{\max}) = (1, 0.125)$ and $(50, 0.125)$ respectively.

Because the magnitude of this N-wave is small, the entropy residual at the N-wave is relatively small compared to that at the existing shock. On one hand, $\beta = 1$ results are free from overshoots when the grid is refined, but $\beta = 2$ results do have overshoots, see Figure 7.4-7.5.

7.5 Conclusion

In summary, we have performed a convergence study for Burgers' equation with various initial data. We demonstrated that the entropy viscosity method with $\beta = 2$ does not produce convergent results (fixing the parameters α_{EV} and $\alpha_{\max}$) in the cases where the shock is moving or more than one shock is present. Therefore, we recommend readers to use $\beta = 1$; to achieve desired accuracy or better rate of convergence, use a higher order approximation of the residual.

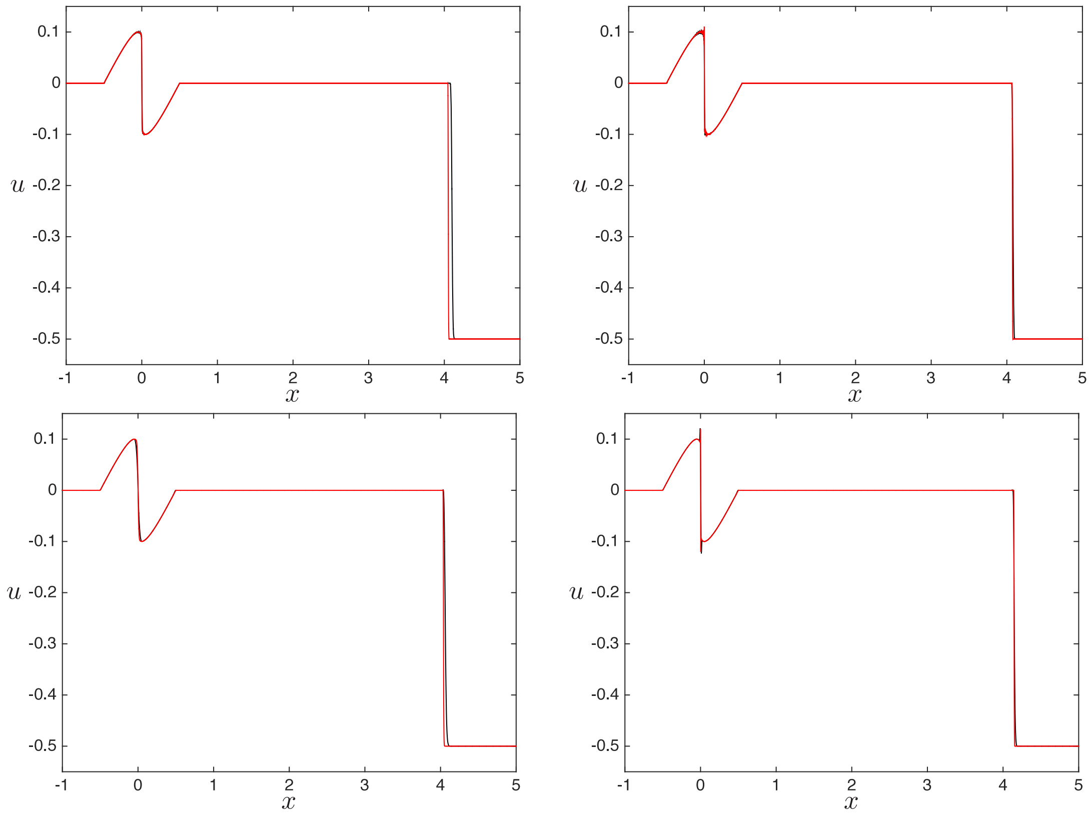


Figure 7.4: Effect of the choice of scaling on a small perturbation near a larger shock. The results in the left and right column are for $\beta = 1$ and $\beta = 2$ respectively. The upper figures display the results for the dG method and the lower figures display the results for the Hermite method. The black curve is for a simulation using 320 elements and the black uses 2560.

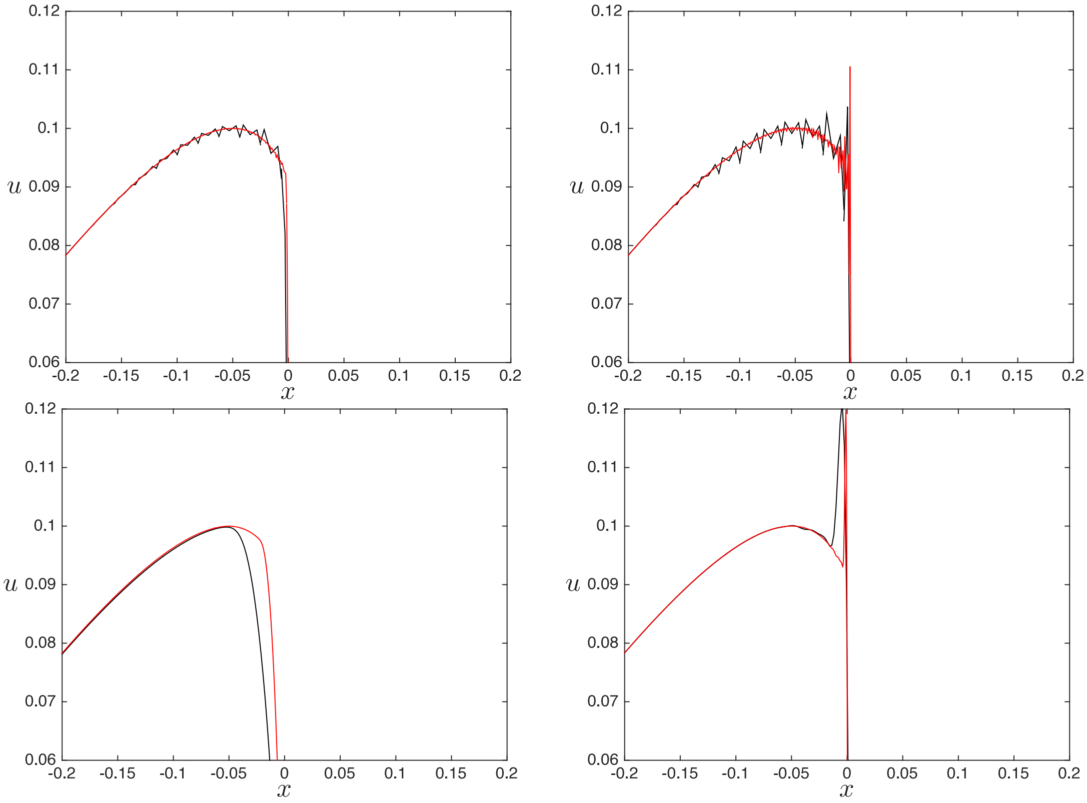


Figure 7.5: Effect of the choice of scaling on a small perturbation near a larger shock. Same as in Figure 7.4 but zoomed in.

Chapter 8

Sobolev Discontinuous Galerkin Methods

We consider solving conservation laws of the type $u_t + f(u)_x = 0$ using a Discontinuous Galerkin (DG) method. The problem with the current DG formulation is that to obtain a high order spatial approximation, the time step must be reduced dramatically. To understand the cause of small time step size, consider solving the PDE $u_t + f(u)_x = 0$ above using modal DG. The numerical solution u_h is expressed as a polynomial expansion

$$u_h = \sum_{n=0}^{N} c_n p_n,$$

where p_n is a polynomial of degree n and c_n is its coefficient in the expansion. A high order DG method can be obtained using large N. The spatial derivative of solution in the PDE is then approximated by the spatial derivative of the approximate solution u_h. On Figure 8.1, we plot Legendre polynomials of degree 0 to 5, along with their derivatives. Observe that the derivatives get large towards the endpoints.

Naturally, we use the derivative of these polynomials to approximate the derivative of the solution to hyperbolic PDEs. We know that the spatial dependence for

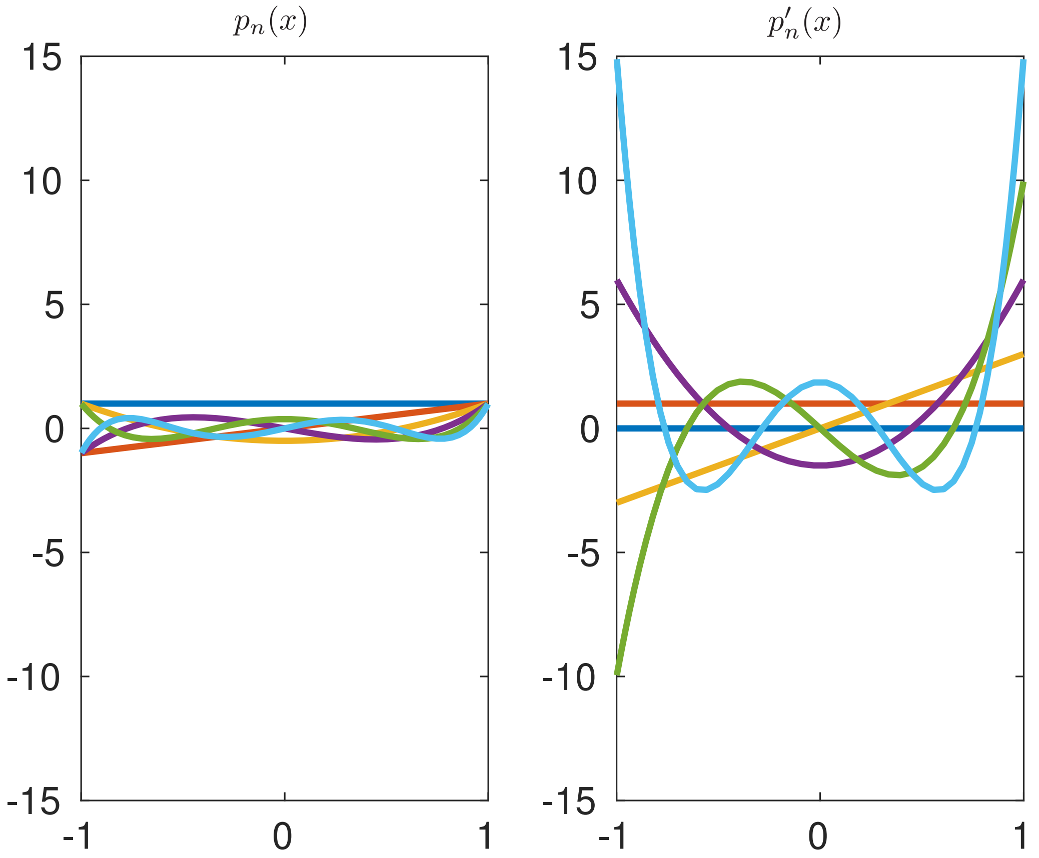


Figure 8.1: Plot of Legendre polynomials and their derivatives. Legendre polynomials are commonly used as a a basis for polynomial approximation in $[-1, 1]$. The size of the Legendre polynomials is normalized to unity. The derivatives get large near the endpoints for high degree polynomials.

solution to a wave propagation problem is of the form $u = e^{inx}$, with derivative

$$\frac{d}{dx}e^{inx} = in\, e^{inx}$$

and hence the size of the derivative is n times the size of the solution

$$\left|\frac{du}{dx}\right| \leq n\|u\|_\infty.$$

We associate one mode in Fourier space (which looks like e^{inx}) with a mode p_n

in the polynomial expansion of u_h. The derivative of p_n satisfies Markov-Bernstein inequalities on $x \in [-1, 1]$,

$$\left| \frac{d}{dx} p_n(x) \right| \leq \min \left(n^2, \frac{n}{\sqrt{1-x^2}} \right) \|p_n\|_\infty.$$

Note that near the endpoints $x = \pm 1$, the size of the derivative grows like n^2 times the size of the function p_n. For large n, the size of the approximate derivative is much larger than the actual derivative. Furthermore, the formulation of DG method evaluates derivatives throughout the element including the endpoints, and as a consequence, the spectral radius of the spatial discretization in DG grows very fast with n and in turn causes numerical stiffness. The numerical stiffness induces restrictive CFL condition, or small time step.

In this new method, we use an even number of degrees of freedom, $2q$. In general, our methods are built from $2l/s$ distinct variational equations. The variable s indicates the increase in number of derivatives applied in between each variational equation. For example, a method which uses 10 degrees of freedom can use $s = 1, 2,$ or 5 and we denote these methods as degree 10-stride methods, where stride can take on values 1,2,5. As a short hand notation, we use SOB-DG(10,1) etc.

To illustrate the SOB-DG method, consider SOB-DG(6,2) with 6 degrees of freedom and stride $s = 2$. In the standard DG method, we would assign all test functions to the variational formulation,

$$\int_{\mathcal{D}^q} \psi_j \left(\frac{\partial u_h^q}{\partial t} + a \frac{\partial u_h^q}{\partial x} \right) \, dx = \int_{\partial \mathcal{D}^q} \psi_j (a u_h^q - (au)^*) \cdot \hat{\mathbf{n}} \, dr, \tag{8.1}$$

$$\text{Test functions: } \psi_j = 1, x, x^2, x^3, x^4, x^5.$$

In the SOB-DG, we distribute the test functions to different variational equations. In SOB-DG(6,2), we have $dof/s = 3$ variational equations, given by Equation (8.2-

8.4),

$$\int_{\mathcal{D}^q} \psi_j \left(\frac{\partial u_h^q}{\partial t} + a \frac{\partial u_h^q}{\partial x} \right) dx = \int_{\partial \mathcal{D}^q} \psi_j (a u_h^q - (au)^*) \cdot \hat{\mathbf{n}} \, dr, \tag{8.2}$$

Test functions: $\psi_j = 1, x$

$$\int_{\mathcal{D}^q} \frac{\partial^2 \psi_j}{\partial x^2} \frac{\partial^2}{\partial x^2} \left(\frac{\partial u_h^q}{\partial t} + a \frac{\partial u_h^q}{\partial x} \right) dx = \int_{\partial \mathcal{D}^q} \frac{\partial^2 \psi_j}{\partial x^2} \left(\frac{\partial^2 (a u_h^q - (au)^*)}{\partial x^2} \right) \cdot \hat{\mathbf{n}} \, dr, \tag{8.3}$$

Test functions: $\psi_j = x^2, x^3$, but notice $\psi_j'' \in \text{span}\{1, x\}$

$$\int_{\mathcal{D}^q} \frac{\partial^4 \psi_j}{\partial x^4} \frac{\partial^4}{\partial x^4} \left(\frac{\partial u_h^q}{\partial t} + a \frac{\partial u_h^q}{\partial x} \right) dx = \int_{\partial \mathcal{D}^q} \frac{\partial^4 \psi_j}{\partial x^4} \left(\frac{\partial^4 (a u_h^q - (au)^*)}{\partial x^4} \right) \cdot \hat{\mathbf{n}} \, dr, \tag{8.4}$$

Test functions: $\psi_j = x^4, x^5$, but notice $\psi_j'''' \in \text{span}\{1, x\}$

8.1 Eigenvalue Analysis

We report the eigenvalues of the spatial discretization operator of SOB-DG for advection equation. In Figure 8.2, we plot the eigenvalues of SOB-DG with fixed degrees of freedom, $dof = 12$, with stride varies from $s = 1$ to $s = 12$. Notice that the increase in the spectral radii is monotone with s. In Figure 8.3, we fix the stride while varying the degrees of freedom. As illustrated in the figure, the spectra look identical for a fixed stride, independent of the number of degrees of freedom.

8.2 Stability Results

We will prove stability of the method for the advection equation $u_t + au_x = 0$. In this proof, we can't get the energy estimate to be nonpositive, but we can only obtain boundedness.

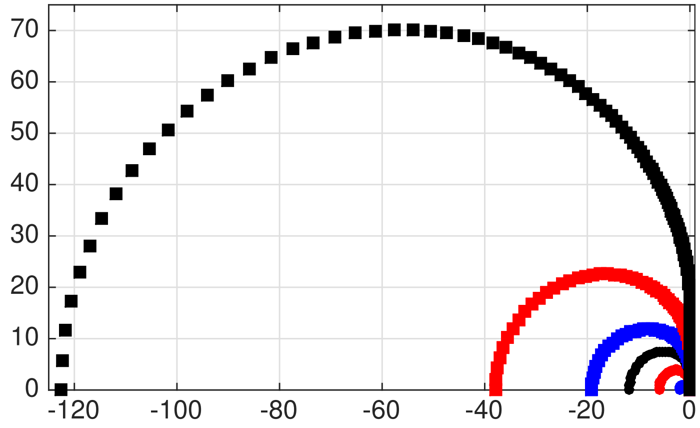


Figure 8.2: Spectrum of SOB-DG with fixed dof, different s ranging from $s = 1$ to $s = 12$

Definition The Sobolev space of index (l, p), where l is a nonngative integer and $p \geq 1$ is defined by

$$W^{l,p}(\Omega) = \{u : D^{\alpha}u \in L^{p}(\Omega) \text{ for all } |\alpha| \leq l\}$$

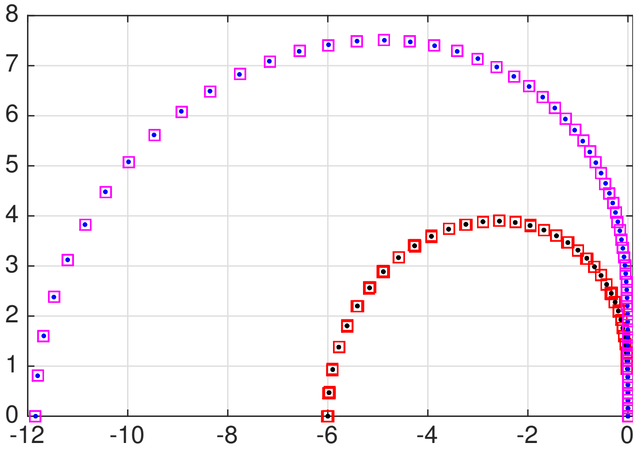


Figure 8.3: Spectrum of SOB-DG is independent of the number of degrees of freedom. Here, (dof, s): (12,2), red squares, (10,2), black dots, (12,3), purple squares, and (6,3), blue dots

.

Lemma 2. *Let $S = span\{\phi_k, \phi_{k+1}, ...\}$ be a polynomial space on domain $\Omega \subset \mathbb{R}$ where ϕ_k is a polynomial of degree k. Then, for any $u \in S$, the (Sobolev) seminorm $|u|_{m,p,\Omega}$ defined by*

$$|u|^p_{m,p,\Omega} = \|D^m u\|^p_{0,p,\Omega} = \int_\Omega |D^m u|^p dx \tag{8.5}$$

is a norm on S, given that $m \leq k$.

Proof. To show that $|u|^p_{k,p,\Omega}$ is a norm, we only need to show that $|u|_{k,p,\Omega} = 0$ implies $u \equiv 0$. Since S consists of continuous functions, $(\int_\Omega |D^m u|^p dx)^{1/p}$ is zero only when $D^m u \equiv 0$ (results from L^p spaces). Suppose $u \in S$. Then, we can write $u = \sum_{j=k}^\infty c_j \phi_j$. Hence, $D^m u = \sum_{j=k}^\infty c_j D^m \phi_j$. Since $m \leq k$, we decrease the degree of each ϕ_j by m, but none of them vanishes. As a consequence, to obtain $D^m u \equiv 0$, we must have $c_j = 0$ for all $j \geq k$, due to linear independency of polynomials of distinct degree. □

We prove the stability for $u_t + u_x = 0$ using 4 dofs. The formulation is as follows

$$\int_{\mathcal{D}^q} \left(\frac{\partial u_h^q}{\partial t} \Psi_i + \frac{\partial u_h^q}{\partial x} \Psi_i \right) dx = \int_{\partial \mathcal{D}^q} (u_h^q - u^*) \Psi_i \cdot \hat{\mathbf{n}}, \tag{8.6}$$

$$\Psi_i = P_0, P_1$$

$$\int_{\mathcal{D}^q} \left(D_x^2 \frac{\partial u_h^q}{\partial t} D_x^2 \Psi_i + D_x^2 \frac{\partial u_h^q}{\partial x} D_x^2 \Psi_i \right) dx = \int_{\partial \mathcal{D}^q} D_x^2 (u_h^q - u^*) D_x^2 \Psi_i \cdot \hat{\mathbf{n}} \tag{8.7}$$

$$\Psi_i = P_2, P_3.$$

Based on the variational formulation, we decompose the solution u_h^q into two parts, $u_h^q = u^I + u^{II}$, with $u^I \in \text{span}\{P_0, P_1\}$, $u^{II} \in \text{span}\{P_2, P_3\}$, where the superscript q has been supressed for compactness. Then, $D_x^2 u_h^q = D_x^2 u^{II}$.

$$\int_{\mathcal{D}^q} D_x^2 \frac{\partial u_h^q}{\partial t} D_x^2 u_h^q + D_x^2 \frac{\partial u_h^q}{\partial x} D_x^2 u_h^q dx = D_x^2 (u_h^q - u^*) D_x^2 u_h^q |_{x_L^q}^{x_R^q}. \tag{8.8}$$

This gives us

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\|D_x^2u_h^q\|^2+\frac{1}{2}(D_x^2u_h^q)^2|_{x_L^q}^{x_R^q}&=D_x^2(u_h^q-u^*)D_x^2u_h^q|_{x_L^q}^{x_R^q}\\ \frac{1}{2}\frac{d}{dt}\|D_x^2u_h^q\|^2&=D_x^2(u_h^q-u^*)D_x^2u_h^q-\frac{1}{2}(D_x^2u_h^q)^2|_{x_L^q}^{x_R^q}\end{aligned}\tag{8.9}$$

Now, using similar format of the flux as standard dG (but for D_x^2u instead of u), we obtain that the right hand side is bounded by 0. Adding up the contribution from all cells, we have that the total energy in $D_x^2u_h$,

$$|u|_{m=2,p=2,\mathcal{D}}=\|D_x^2u_h\|_{\mathcal{D}}=\sum_q\|D_x^2u_h^q\|\tag{8.10}$$

decreases over time. For u^I, we use Equation 8.6 and recall that the test function Ψ_i only lives in space of linear function, so in the energy estimate, we replace it by u^I to get

$$\begin{aligned}\int_{\mathcal{D}^q}\left(\frac{\partial u_h^q}{\partial t}u^I+\frac{\partial u_h^q}{\partial x}u^I\right)dx&=(u_h^q-u^*)u^I|_{x_L^q}^{x_R^q}\\ \int_{\mathcal{D}^q}\left(\frac{\partial(u^I+u^{II})}{\partial t}u^I+\frac{\partial(u^I+u^{II})}{\partial x}u^I\right)dx&=(u^I+u^{II}-u^*)u^I|_{x_L^q}^{x_R^q}\\ \int_{\mathcal{D}^q}\left(\frac{\partial u^I}{\partial t}u^I+\frac{\partial u^{II}}{\partial t}u^I+\frac{\partial u^I}{\partial x}u^I+\frac{\partial u^{II}}{\partial x}u^I\right)dx&=(u^I-u^*)u^I+u^{II}u^I|_{x_L^q}^{x_R^q}\\ \frac{1}{2}\frac{\partial}{\partial t}\|u^I\|_{D^q}^2+\int_{\mathcal{D}^q}\left(\frac{\partial u^{II}}{\partial t}u^I-u^{II}(u^I)_x\right)dx&=(u^I-u^*)u^I|_{x_L^q}^{x_R^q}-\frac{1}{2}(u^I)^2|_{x_L^q}^{x_R^q}\end{aligned}\tag{8.11}$$

Notice that the right hand side of the above equation looks like the boundary contribution in standard dG, so by choosing an appropriate numerical flux u^*, we can bound the right hand side by zero. Furthermore, the integral on the left hand side can be moved to right hand side and then using Cauchy-Schwartz inequality, we obtain

$$\frac{d}{dt}\|u^I\|_D^2\leq 2\sum_{q=1}^{Q}\left(\|u_x^I\|_{\mathcal{D}^q}\|u^{II}\|_{\mathcal{D}^q}+\|u_t^{II}\|_{\mathcal{D}^q}\|u^I\|_{\mathcal{D}^q}\right).$$

Noting that the energy in $D_x^2u^{II}$ decreases (see Equation 8.10)and by using analysis in modal DG, we have that the energy in u^{II} decreases as well, so the energy in u^I is bounded.

8.3 Numerical Experiments

In this experiment, we look at different Sobolev DG methods for computing the solution to transport equation $u_t + au_x = 0$. Here, each element carries $2l$ degrees of freedom. In general, our methods are built from $2l/s$ distinct variational equations. The variable s indicates the increase in number of derivatives applied in between each variational equation. For example, a method which uses 10 degrees of freedom can use $s = 1, 2,$ or 5 and we denote these methods as degree 10-stride methods, where stride can take on values 1,2,5. As a short hand notation, we use SOB-DG(10,1) etc.

The first method we test, denoted by SOB-DG($2l$,l), uses an additional equation, which requires that the l^{th} derivative of the PDE also satisfies the variational form. Precisely, the method takes the form

$$\int_{\Omega^q}\left(\frac{\partial u_h^q}{\partial t}+\frac{\partial a u_h^q}{\partial x}\right)\Psi_i dx = \int_{\partial\Omega^q}(au_h^q-(au)^*)\Psi_i\cdot\hat{\mathbf{n}},$$
$$i = 0, ..., l-1,$$
$$\int_{\Omega^q}\left(\frac{\partial^l}{\partial x^l}\frac{\partial u_h^q}{\partial t}+\frac{\partial^l}{\partial x^l}\frac{\partial a u_h^q}{\partial x}\right)\frac{\partial^l\Psi_i}{\partial x^l}dx = \int_{\partial\Omega^q}\left(\frac{\partial^l a u_h^q}{\partial x^l}-\frac{\partial^l (au)^*}{\partial x^l}\right)\frac{\partial^l\Psi_i}{\partial x^l}\cdot\hat{\mathbf{n}} \quad (8.12)$$
$$i = l, ..., 2l-1.$$

The second method we test, denoted by SOB-DG($2l$,2), uses $(l-1)$ additional equations, which require that even derivatives of the PDE also satisfy the variational form. This method can be formulated as

$$\int_{\Omega^q}\left(\frac{\partial^{2r}}{\partial x^{2r}}\frac{\partial u_h^q}{\partial t}+\frac{\partial^{2r}}{\partial x^{2r}}\frac{\partial a u_h^q}{\partial x}\right)\frac{\partial^{2r}\Psi_i}{\partial x^{2r}}dx = \int_{\partial\Omega^q}\left(\frac{\partial^{2r} a u_h^q}{\partial x^{2r}}-\frac{\partial^{2r} (au)^*}{\partial x^{2r}}\right)\frac{\partial^{2r}\Psi_i}{\partial x^{2r}}\cdot\hat{\mathbf{n}} \quad (8.13)$$

$$i = 2r, 2r+1, r = 0, 1, ..., l-1.$$

To study the order of accuracy in 1 dimension, we solve the transport equation

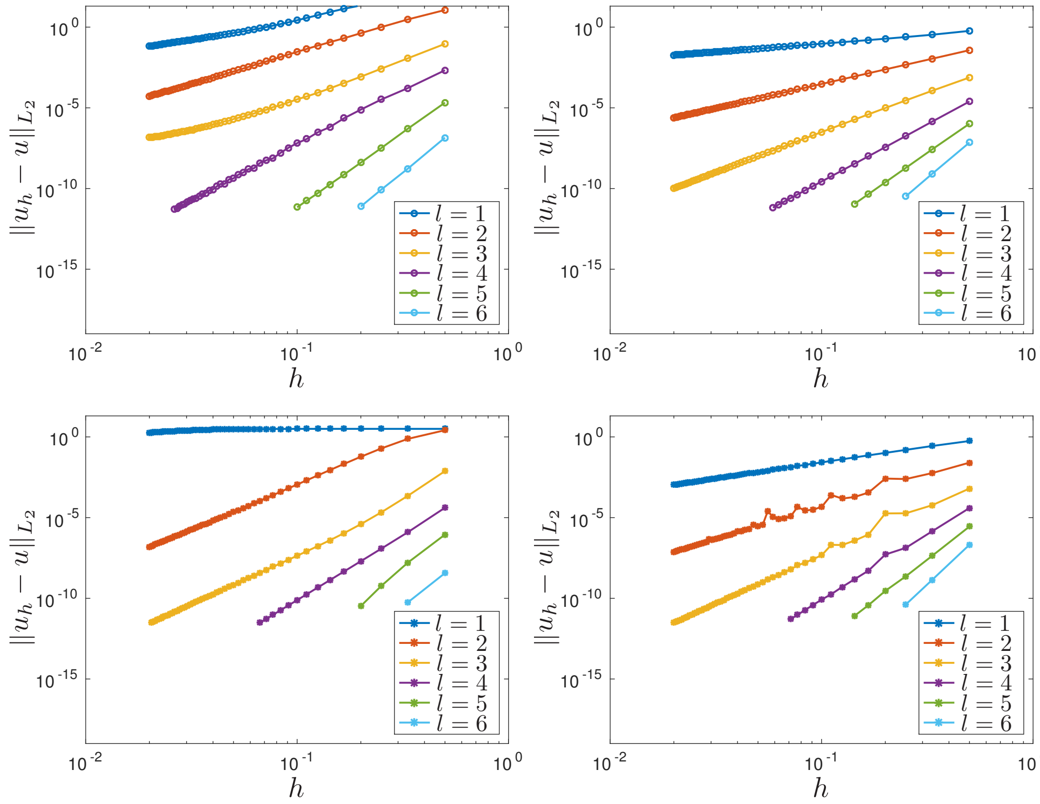


Figure 8.4: Plots of the error in u as a function of h in 1D advection equation. On the top, from left to right, SOB-DG($2l$,l) method and SOB-DG($2l$,2) method with central flux, on the bottom, from left to right, SOB-DG($2l$,l) method and SOB-DG($2l$,2) method with upwind flux.

$u_t + au_x = 0$ on $[-1, 1]$, $t > 0$ with periodic boundary conditions and with initial data $u(x, 0) = \sin(\pi x)$. The discretization is performed on a uniform grid $x_j = -1+jh, j = 0, ..., N, h = 2/N$. The time evolution is done using Taylor time stepping method, to enable less restrictive time stepping size. In Figure 8.4, we report the L_2 error in the solution u_h after 10 temporal periods. Estimated rates of convergence can be found in Table 8.1.

To assess the efficiency of different SOB-DG methods, we solve $u_t + u_x = 0$ on

l	1	2	3	4	5	6
SOB-DG($2l$,l) central	2.41	3.90	3.96	6.88	9.30	10.69
SOB-DG($2l$,2) central	1.96	3.89	5.70	7.53	9.44	11.40
SOB-DG($2l$,l) upwind	0.18	5.46	6.34	8.09	11.07	10.35
SOB-DG($2l$,2) upwind	2.77	4.88	7.98	10.54	12.95	15.23

Table 8.1: Rates of convergence obtained from a linear least square fit to the data in Figure 8.4.

(s, Q)	$\mathcal{E}(t = 1)$	$\mathcal{E}(t = 10)$	$\mathcal{E}(t = 100)$
(2,14)	9.18(-8)	1.07(-4)	6.14(-1)
(2,28)	5.76(-11)	2.05(-9)	1.98(-4)
Rate	10.6	15.7	11.6
(6,8)	3.47(-9)	3.73(-9)	5.96(-9)
(6,16)	1.14(-12)	1.56(-12)	2.51(-12)
Rate	11.6	11.2	11.2
(12,5)	2.13(-8)	2.09(-8)	2.09(-8)
(12,10)	5.12(-12)	7.02(-12)	3.01(-11)
Rate	12.0	11.5	9.4

Table 8.2: Maximum errors in SOB-DG over space and time in 1D advection equation. Here, the number of elements Q has been scaled for matching computation cost among different methods. Here, s: stride, $\mathcal{E}$: error.

the periodic domain $x \in \mathcal{D} = [0, 2]$, with initial data $u = \sin(\pi x)$. We fix the degrees of freedom to be 12, and we also use upwind flux throughout the simulation for all cases. We find that all SOB-DG produce a 12th order method, see Table 8.2.

8.4 Extension to Higher Dimensions

The idea in SOB-DG can be extended to two dimensions using tensor product. In general, we use $(2l)^2$ degrees of freedom. To illustrate how the SOB-DG works in 2D, we write the equations when we have $l = 2$, corresponding to 16 degrees of freedom, with stride $s = 2$. For two dimensional advection $u_t + u_x + u_y = 0$, we have 4 variational equations,

$$\int_{D^q} \psi_j \left(\frac{\partial u_h^q}{\partial t} + \frac{\partial u_h^q}{\partial x} + \frac{\partial u_h^q}{\partial y} \right) dxdy = \int_{\partial D^q} \psi_j (u_h^q - u^*) \cdot \hat{\mathbf{n}} dr, \tag{8.14}$$

Test functions: $\psi_j(x,y) = \phi_j(x)\zeta_j(y),\ \phi_j(x) = 1, x,\ \zeta_j(y) = 1, y$

$$\int_{D^q} \frac{\partial^2 \psi_j}{\partial x^2} \frac{\partial^2}{\partial x^2} \left(\frac{\partial u_h^q}{\partial t} + \frac{\partial u_h^q}{\partial x} + \frac{\partial u_h^q}{\partial x} \right) dxdy = \int_{\partial D^q} \frac{\partial^2 \psi_j}{\partial x^2} \left(\frac{\partial^2 (u_h^q - u^*)}{\partial x^2} \right) \cdot \hat{\mathbf{n}} dr, \tag{8.15}$$

Test functions: $\psi_j(x,y) = \phi_j(x)\zeta_j(y),\ \phi_j(x) = x^2, x^3,\ \zeta_j(y) = 1, y$

$$\int_{D^q} \frac{\partial^2 \psi_j}{\partial y^2} \frac{\partial^2}{\partial y^2} \left(\frac{\partial u_h^q}{\partial t} + \frac{\partial u_h^q}{\partial x} + \frac{\partial u_h^q}{\partial y} \right) dxdy = \int_{\partial D^q} \frac{\partial^2 \psi_j}{\partial y^2} \left(\frac{\partial^2 (u_h^q - u^*)}{\partial y^2} \right) \cdot \hat{\mathbf{n}} dr, \tag{8.16}$$

Test functions: $\psi_j(x,y) = \phi_j(x)\zeta_j(y),\ \phi_j(x) = 1, x,\ \zeta_j(y) = y^2, y^3$

$$\int_{D^q} \frac{\partial^4 \psi_j}{\partial x^2 \partial y^2} \frac{\partial^4}{\partial x^2 \partial y^2} \left(\frac{\partial u_h^q}{\partial t} + \frac{\partial u_h^q}{\partial x} + \frac{\partial u_h^q}{\partial y} \right) dxdy = \int_{\partial D^q} \frac{\partial^4 \psi_j}{\partial x^2 \partial y^2} \left(\frac{\partial^4 (u_h^q - u^*)}{\partial x^2 \partial y^2} \right) \cdot \hat{\mathbf{n}} dr, \tag{8.17}$$

Test functions: $\psi_j(x,y) = \phi_j(x)\zeta_j(y),\ \phi_j(x) = x^2, x^3,\ \zeta_j(y) = y^2, y^3$

In the following experiment, we solve $u_t + u_x + u_y = 0$, on the periodic domain $x \in D = [-1, 1]$. The initial data is $u_0(x, y) = \sin(\pi x)\sin(\pi y)$. We use $l = 2$, corresponding to 4th order method in the spatial direction and we also use central flux. In Table 8.3, we present a convergence study on the error under grid refinement. We obtain a fourth order convergence.

(s, Q)	$\mathcal{E}(t=1)$	Rate
(2,10)	6.13(-4)	
(2,20)	4.40(-5)	3.78
(2,40)	2.91(-6)	3.92
(2,80)	1.84(-7)	3.98
(2,160)	1.15(-8)	3.99

Table 8.3: Convergence study for 2D advection equation.

Standard DG	**Sobolev DG**
High order	High order
"Strong" stability in L_2 norm	"Weaker" stability in Sobolev norm
Unstructured grids	Unstructured grids
CFL: $N\Delta t/\Delta x \leq C$	CFL: $s\Delta t/\Delta x \leq C$
Handle boundary well	Derivatives at boundary?

Table 8.4: Comparison between Standard DG and Sobolev DG.

8.5 Comparison between Standard and Sobolev DG

In this section, we summarize the strengths and weaknesses of the two Discontinuous Galerkin methods. Note that the CFL condition in the Sobolev DG now depends on the stride, instead of the number of degrees of freedom (equivalently, the number of basis functions).

Chapter 9

Conclusions and Future Work

This thesis focuses on the development and analysis of high order numerical methods for long simulation of wave propagation problems. The majority of this work deals with hyperbolic partial differential equations of conservation law type. We saw earlier in Chapter 4 that a conservative method guarantees convergence of numerical solutions to conservation laws. Moreover, to approximate discontinuous solutions, the numerical method should produce oscillation-free solutions, as these oscillations are not physical and may affect the stability of the method.

Numerical methods under the category of Method of Lines discretize in space while leaving the temporal discretization continuous. In Chapter 5, we implemented Runge-Kutta schemes with Total Variation Diminishing (TVD) property, or commonly known as Strong-Stability-Preserving Runge-Kutta (SSPRK). These schemes were developed to evolve the solution to conservation laws while eliminating overshoots and undershoots, and hence maintaining the TVD property. We found that for linear problems, there is no significant difference in the solutions evolved by the SSPRK and non-SSPRK schemes. However, for nonlinear problems, the SSPRK schemes are more efficient at producing oscillation-free solutions than the non-SSPRK schemes.

In Chapter 6, we developed a new class of Hermite methods, flux-conservative Hermite methods, for solving nonlinear conservation laws. While preserving the high order spatial accuracy for smooth solutions in the existing Hermite methods, the new methods come with better stability properties. The adaptation of the entropy viscosity method to Hermite methods successfully suppresses oscillations near shocks, but we find that our current implementation is quite dissipative when solving shocks near high frequency sinusoidal waves. For contact waves we proposed a modification to the entropy viscosity method which eliminates a large amount of the spurious viscosity at contact discontinuities.

In Chapter 7, we implemented the Entropy Viscosity (EV) method to different high order methods to eliminate oscillations that otherwise appear near discontinuity. We then discussed how the scaling in the method influences the size of viscosity, hence the quality of the solution. The EV method involves two tuning parameters, α_{EV} and $\alpha_{\max}$, which we fixed when performing grid refinement study. We demonstrated that under the first scaling ($\beta = 2$), the solutions do not converge when the shock is moving or more than one shock is present. On the other hand, under the second scaling ($\beta = 1$), the solutions do converge, whether in stationary, moving, or multiple shocks situations.

In Chapter 8, we developed a new class of Discontinuous Galerkin Methods, Sobolev Discontinuous Galerkin methods (Sobolev DG), where the error is minimized in Sobolev norm. It turns out that the new formulation allows for a larger time step. However, the error in the Sobolev DG grows faster than the error in the standard DG.

Chapter 10

Future Work

In terms of temporal discretizations, we plan to do the following:

1. Experiment with other non-SSPRK schemes not mentioned in the paper(s) and compare them with SSPRK schemes.
2. Perform a study to obtain efficient TVD spatial discretizations for any PDE, possibly in a system of equations or in higher dimensions.
3. Test the schemes on examples not considered in any literature before.

In terms of spatial discretizations, we plan to do the following:

1. Analyze the stability of hybrid Hermite-Sobolev-DG methods.
2. Implement a stable hybridization of Hermite-Sobolev-DG methods.
3. Extend Sobolev DG methods to handle boundary.
4. Explore the possibility of adopting the entropy viscosity technique for constructing nonreflecting boundary conditions.

References

[1] D. Appelö and T. Colonius. A high-order super-grid-scale absorbing layer and its application to linear hyperbolic systems. *Journal of Computational Physics*, 228(11):4200–4217, 2009.

[2] D. Appelö and T. Hagstrom. CHIDES: The Charles Hermite Interpolation Differential Equation Solver. http://www.chides.org.

[3] Rick Archibald, Anne Gelb, Sigal Gottlieb, and Jennifer Ryan. One-sided post-processing for the discontinuous Galerkin method using ENO type stencil choosing and the local edge detection method. *Journal of Scientific Computing*, 28(2):167–190, 2006.

[4] Ivo Babuška and Manil Suri. The *hp* version of the finite element method with quasiuniform meshes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 21(2):199–238, 1987.

[5] Jeffrey W Banks and Thomas Hagstrom. On Galerkin difference methods. *Journal of Computational Physics*, 313:310–327, 2016.

[6] Francesco Bassi and Stefano 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.

[7] Jay P Boris and David L Book. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 11(1):38–69, 1973.

[8] Alexander N. Brooks and Thomas J.R. Hughes. Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199 – 259, 1982.

[9] Jay Casper and HL Atkins. A finite-volume high-order ENO scheme for two-dimensional hyperbolic systems. *Journal of Computational Physics*, 106(1):62–76, 1993.

[10] Noel Chalmers, Lilia Krivodonova, and Ruibin Qin. Relaxing the CFL number of the discontinuous Galerkin method. *SIAM Journal on Scientific Computing*, 36(4):A2047–A2075, 2014.

[11] X. Chen, D. Appelö, and T. Hagstrom. A hybrid Hermite–discontinuous Galerkin method for hyperbolic systems with application to Maxwell's equations. *Journal of Computational Physics*, 257, Part A:501 – 520, 2014.

[12] Bernardo Cockburn, Mitchell Luskin, Chi-Wang Shu, and Endre Süli. Enhanced accuracy by post-processing for Finite Element methods for hyperbolic equations. *Mathematics of Computation*, 72(242):577–606, 2003.

[13] Phillip Colella and Paul R Woodward. The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, 54(1):174–201, 1984.

[14] B Denet, L Biamino, G Lodato, L Vervisch, and P Clavin. Model Equation for the Dynamics of Wrinkled Shockwaves: Comparison with DNS and Experiments. *Combustion Science and Technology*, 187(1-2):296–323, 2015.

[15] Michael Dumbser, Olindo Zanotti, Raphaël Loubère, and Steven Diot. A posteriori subcell limiting of the discontinuous Galerkin Finite Element method for hyperbolic conservation laws. *Journal of Computational Physics*, 278:47–75, 2014.

[16] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of Finite Elements*, volume 159. Springer Science & Business Media, 2013.

[17] Lawrence C. Evans. *Partial Differential Equations.* American Mathematical Society, 1998.

[18] Gregor J Gassner. A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT Finite Difference methods. *SIAM Journal on Scientific Computing*, 35(3):A1233–A1253, 2013.

[19] James Glimm and Peter D Lax. Decay of solutions of systems of hyperbolic conservation laws. *Bulletin of the American Mathematical Society*, 73(1):105, 1967.

[20] SK Godunov. A Finite Difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Sbornik: Mathematics*, 47(8-9):357–393, 1959.

[21] J. Goodrich, T. Hagstrom, and J. Lorenz. Hermite Methods for Hyperbolic Initial-Boundary Value Problems. *Mathematics of computation*, 75(254):595–630, 2006.

[22] Sigal Gottlieb, David I Ketcheson, and Chi-Wang Shu. High order strong stability preserving time discretizations. *Journal of Scientific Computing*, 38(3):251–289, 2009.

[23] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing Runge-Kutta schemes. *Mathematics of computation of the American Mathematical Society*, 67(221):73–85, 1998.

[24] J.-L. Guermond and R. Pasquetti. Entropy Viscosity method for higher-order approximations of conservation laws. *Lecture Notes in Computational Science and Engineering*, 76:411 – 418, 2011.

[25] Jean-Luc Guermond and Richard Pasquetti. Entropy-based nonlinear viscosity for Fourier approximations of conservation laws. *Comptes Rendus Mathematique*, 346(13):801–806, 2008.

[26] Jean-Luc Guermond and Richard Pasquetti. Entropy Viscosity method for high-order approximations of conservation laws. In *Spectral and High Order Methods for Partial Differential Equations*, pages 411–418. Springer, 2011.

[27] Jean Luc Guermond, Richard Pasquetti, and Bojan Popov. Entropy Viscosity for conservation equations. In *V European Conference on Computational Fluid Dynamics (Eccomas CFD 2010)*, 2010.

[28] Jean-Luc Guermond, Richard Pasquetti, and Bojan Popov. Entropy Viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230(11):4248–4267, 2011.

[29] Jean-Luc Guermond, Richard Pasquetti, and Bojan Popov. From suitable weak solutions to Entropy Viscosity. *Journal of Scientific Computing*, 49(1):35–50, 2011.

[30] B Guo and I Babuška. The hp version of the Finite Element method. *Computational Mechanics*, 1(1):21–41, 1986.

[31] T. Hagstrom and D. Appelö. Experiments with Hermite Methods for Simulating Compressible Flows: Runge-Kutta Time-stepping and Absorbing Layers. In *13th AIAA/CEAS Aeroacoustics Conference*, number 2007-3505, 2007.

[32] T. Hagstrom and D. Appelö. Solving PDEs with Hermite Interpolation. In *Springer Lecture Notes in Computational Science and Engineering*, 2015.

[33] A Harten and G Zwas. Self-adjusting hybrid schemes for shock computations. *Journal of Computational Physics*, 9(3):568–583, 1972.

[34] Ami Harten. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics*, 49(3):357–393, 1983.

[35] Ami Harten. ENO schemes with subcell resolution. *Journal of Computational Physics*, 83(1):148–184, 1989.

[36] Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of computational physics*, 71(2):231–303, 1987.

[37] Ami Harten and Stanley Osher. Uniformly high-order accurate nonoscillatory schemes. I. *SIAM Journal on Numerical Analysis*, 24(2):279–309, 1987.

[38] Ami Harten, Stanley Osher, Björn Engquist, and Sukumar R Chakravarthy. Some results on uniformly high-order accurate essentially nonoscillatory schemes. *Applied Numerical Mathematics*, 2(3-5):347–377, 1986.

[39] Jan S Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applicationss Galerkin Methods: Algorithms, Analysis, and Applications*, volume 54. Springer, New York, 2008.

[40] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.

[41] Claes Johnson, Anders Szepessy, and Peter Hansbo. On the convergence of shock-capturing streamline diffusion Finite Element methods for hyperbolic conservation laws. *Mathematics of Computation*, 54(189):107–129, 1990.

[42] Robert M Kirby and George Em Karniadakis. De-aliasing on Non-uniform grids: algorithms and applications. *Journal of Computational Physics*, 191(1):249–264, 2003.

[43] Andreas Klöckner, Tim Warburton, and Jan S Hesthaven. Viscous shock capturing in a time-explicit discontinuous Galerkin method. *Mathematical Modelling of Natural Phenomena*, 6(3):57–83, 2011.

[44] David A Kopriva and Gregor J Gassner. An energy stable discontinuous Galerkin spectral element discretization for variable coefficient advection problems. *SIAM Journal on Scientific Computing*, 36(4):A2076–A2099, 2014.

[45] Adeline Kornelus and Daniel Appelö. Flux-conservative Hermite methods for nonlinear conservation laws. *Journal of Scientific Computing*, in revision.

[46] Adeline Kornelus and Daniel Appelö. On the scaling of Entropy Viscosity in high order methods. In *Springer Lecture Notes in Computational Science and Engineering*, in press.

[47] Lilia Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226(1):879–896, 2007.

[48] Alexander Kurganov and Yu Liu. New adaptive artificial viscosity method for hyperbolic systems of conservation laws. *Journal of Computational Physics*, 231(24):8114–8132, 2012.

[49] Randall .J. LeVeque. *Numerical methods for conservation laws.* Birkhäuser Verlag, Basel, 2nd edition, 1992.

[50] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*, volume 31. Cambridge University Press, 2002.

[51] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994.

[52] M.P. Martín, E.M. Taylor, M. Wu, and V.G. Weirs. A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence. *Journal of Computational Physics*, 220(1):270–289, 2006.

[53] Haim Nessyahu and Eitan Tadmor. Non-oscillatory central differencing for hyperbolic conservation laws. *Journal of computational physics*, 87(2):408–463, 1990.

[54] J. Von Neumann and R.D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21:232–237, 1950.

[55] Elaine S Oran and Jay P Boris. *Numerical simulation of reactive flow.* Cambridge University Press, 2005.

[56] P.O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous galerkin methods. In *44 th AIAA Aerospace Sciences Meeting and Exhibit*, pages 1–13, 2006.

[57] Audrey Rault, Guillaume Chiavassa, and Rosa Donat. Shock-vortex interactions at high Mach numbers. *Journal of Scientific Computing*, 19(1-3):347–371, 2003.

[58] Jennifer Ryan, Chi-Wang Shu, and Harold Atkins. Extension of a post processing technique for the discontinuous galerkin method for hyperbolic equations with application to an aeroacoustic problem. *SIAM Journal on Scientific Computing*, 26(3):821–843, 2005.

[59] Jennifer Ryan, Chi-Wang Shu, et al. On a one-sided post-processing technique for the discontinuous galerkin methods. *Methods and Applications of Analysis*, 10(2):295–308, 2003.

[60] Chi-Wang Shu. *Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws.* Springer, 1998.

[61] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

[62] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii. *Journal of Computational Physics*, 83(1):32–78, 1989.

[63] Gilbert Strang and George J Fix. *An analysis of the Finite Element method*, volume 212.

[64] James William Thomas. *Numerical partial differential equations: conservation laws and elliptic equations*, volume 33. Springer Science & Business Media, 2013.

[65] Eleuterio F Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction.* Springer Science & Business Media, 2013.

[66] Bram Van Leer. Towards the ultimate conservative difference scheme I. The quest of monotonicity. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, pages 163–168. Springer, 1973.

[67] Bram Van Leer. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *Journal of computational physics*, 14(4):361–370, 1974.

[68] Bram Van Leer. Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow. *Journal of Computational Physics*, 23(3):263–275, 1977.

[69] Bram Van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of computational physics*, 23(3):276–299, 1977.

[70] Bram 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.

[71] Timothy Warburton and Thomas Hagstrom. Taming the CFL number for discontinuous galerkin methods on structured meshes. *SIAM Journal on Numerical Analysis*, 46(6):3151–3180, 2008.

[72] HC Yee and Björn Sjögreen. Development of low dissipative high order filter schemes for multiscale Navier–Stokes/MHD systems. *Journal of Computational Physics*, 225(1):910–934, 2007.

[73] Steven T Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of computational physics*, 31(3):335–362, 1979.

[74] Shuhai Zhang, Yong-Tao Zhang, and Chi-Wang Shu. Multistage interaction of a shock wave and a strong vortex. *Physics of Fluids (1994-present)*, 17(11):116101, 2005.

[75] Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, Olgierd Cecil Zienkiewicz, and Robert Lee Taylor. *The Finite Element method*, volume 3. McGraw-Hill London, 1977.

[76] Valentin Zingan, Jean-Luc Guermond, Jim Morel, and Bojan Popov. Implementation of the entropy viscosity method with the discontinuous Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 253:479–490, 2013.