SANDIA REPORT

SAND2003-3238
Unlimited Release
Printed September 2003

An Assessment of Semi-Discrete Central Schemes for Hyperbolic Conservation Laws

Mark A. Christon
David I. Ketcheson
Allen C. Robinson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-3238
Unlimited Release
Printed September 2003

An Assessment of Semi-Discrete Central Schemes for Hyperbolic Conservation Laws[1]

Mark A. Christon, David I. Ketcheson and Allen C. Robinson[2]
Computational Physics R & D Department
Sandia National Laboratories
M/S 0819
P. O. Box 5800
Albuquerque, NM 87185-0819

Revised September 8, 2003

Abstract

High-resolution finite volume methods for solving systems of conservation laws have been widely embraced in research areas ranging from astrophysics to geophysics and aero-thermodynamics. These methods are typically at least second-order accurate in space and time, deliver non-oscillatory solutions in the presence of near discontinuities, e.g., shocks, and introduce minimal dispersive and diffusive effects. High-resolution methods promise to provide greatly enhanced solution methods for Sandia's mainstream shock hydrodynamics and compressible flow applications, and they admit the possibility of a generalized framework for treating multi-physics problems such as the coupled hydrodynamics, electro-magnetics and radiative transport found in z-pinch physics. In this work, we describe initial efforts to develop a generalized "black-box" conservation law framework based on modern high-resolution methods and implemented in an object-oriented software framework. The framework is based on the solution of systems of general non-linear hyperbolic conservation laws using Godunov-type central schemes. In our initial

[1]Keywords and Phrases: high-resolution, hyperbolic conservation laws, Godunov, semi-discrete, central schemes

[2]Authors listed in alphabetic order.

efforts, we have focused on central or central-upwind schemes that can be implemented with only a knowledge of the physical flux function and the minimal/maximal eigenvalues of the Jacobian of the flux functions, i.e., they do not rely on extensive Riemann decompositions. Initial experimentation with high-resolution central schemes suggests that contact discontinuities with the concomitant linearly degenerate eigenvalues of the flux Jacobian do not pose algorithmic difficulties. However, central schemes can produce significant smearing of contact discontinuities and excessive dissipation for rotational flows. Comparisons between "black-box" central schemes and the piecewise-parabolic method (PPM), which relies heavily on a Riemann decomposition, shows that roughly equivalent accuracy can be achieved for the same computational cost with both methods. However, PPM clearly outperforms the central schemes in terms of accuracy at a given grid resolution and the cost of additional complexity in the numerical flux functions. Overall we have observed that the finite volume schemes, implemented within a well-designed framework, are extremely efficient with (potentially) very low memory storage. Finally, we have found by computational experiment that second and third-order strong-stability preserving (SSP) time integration methods with the number of stages greater than the order provide a useful enhanced stability region. However, we observe that non-SSP and non-optimal SSP schemes with SSP factors less than one can still be very useful if used with time-steps below the standard CFL limit. The "well-designed" integration schemes that we have examined appear to perform well in all instances where the time step is maintained below the standard physical CFL limit.

Acknowledgment

The NEVADA code framework is being developed at Sandia National Laboratories and provides the baseline software infrastructure for the conservation law framework. The FLASH software used in this work for comparison purposes was developed by the DOE-supported ASCI/Alliances Center for Astrophysical Thermonuclear Flashes at the University of Chicago. The authors also wish to acknowledge Tim Trucano and Tom Smith for their helpful comments during the preparation of this report. This work was supported by the Mathematics, Information, and Computer Science program, as well as the Computer Science and Research Foundation program, at Sandia National Laboratories.

Contents

1 Introduction **11**
1.1 Motivation and Background 11
1.2 The Conservation Law Framework 14
1.3 Central Schemes 14
1.4 Time Integrators 16

2 Formulation and Design Issues **19**
2.1 Reconstruction Methods 20
2.1.1 Constant Reconstruction Methods 21
2.1.2 Linear Reconstruction Methods 21
2.1.3 Quadratic Reconstruction Methods 22
2.2 Numerical Flux Methods 24
2.2.1 Kurganov-Tadmor (KT) 25
2.2.2 Kurganov-Noelle-Petrova (KNP) 25
2.3 Time Integration Methods 25
2.3.1 First-Order Method 26
2.3.2 Second-Order Methods 26
2.3.3 Third-Order Methods 28
2.4 Software Framework Design 30
2.4.1 Memory Requirements 31
2.4.2 Application Developer Interface 35
2.4.3 User Interface 35

3 Accuracy Assessment and Methods Comparison **37**
3.1 Prototypical Applications 37
3.1.1 Scalar Advection 38
3.1.2 Burgers' Equation 38
3.1.3 Euler Equations 38
3.2 Accuracy Assessment 39
3.2.1 Scalar Advection 41
3.2.2 Burgers' Equation 51
3.2.3 Euler Equations 61

3.3 Methods Comparison 70
3.3.1 Comparison Metrics 71
3.3.2 The Sod Shock Tube 72
3.3.3 The Isentropic Vortex 77
3.3.4 The Sedov Blast Wave 83
3.3.5 Numerical Flux Comparison 90
3.3.6 EOS Interface Comparison 90
3.4 Non-Oscillatory Third-Order Reconstruction 90
3.4.1 LT3 Reconstruction may be $O(\Delta x^2)$ 91
3.4.2 KP3 Reconstruction may be $O(\Delta x)$ 92

4 SSP Runge-Kutta Time Integrators 95
4.1 Second-Order Time Integrators 97
4.2 Third-Order Time Integrators 99
4.3 Summary of SSP Results 100

5 Summary 105

Bibliography 107

List of Figures

2.1 Grid layout for reconstruction and flux evaluation. 21

3.1 Solutions for the scalar square wave advection problem at $t = 1.0$ for various reconstruction methods. 44
3.2 Temporal evolution of errors for the square wave advection test for various reconstruction and time integration methods. . 46
3.3 Snapshots of solutions to the square wave advection test for various reconstruction and time integration methods. 48
3.4 Comparison of solution convergence for the Burgers' equation sine wave problem before and after the solution develops a shock. 55
3.5 Convergence rates for Burgers' equation with sinusoidal initial conditions at two post-shock times. 56
3.6 Snapshots of errors and solutions to Burgers' equation with sinusoidal initial conditions using various reconstruction methods with $\Delta x = 1/40$ at $t = 1.5$. 57
3.7 Comparison of density solutions for the Gaussian pulse advection problem using second-order reconstruction and various limiters. 63
3.8 Snapshots of the density solutions, and L^1 errors at $t = 0.4$ for the square pulse advection problem using second-order reconstruction with various limiters. 65
3.9 Snapshot of the density and density error distribution at $t = 0.1644$ for constant, linear, and quadratic reconstruction methods on the Sod shock tube problem. 67
3.10 Snapshot of the pressure and pressure error distribution at $t = 0.1644$ for constant, linear, and quadratic reconstruction methods on the Sod shock tube problem. 68
3.11 Snapshot of the velocity and velocity error distribution at $t = 0.1644$ for constant, linear, and quadratic reconstruction methods on the Sod shock tube problem. 69
3.12 Error vs. cpu time for constant, linear, and quadratic reconstruction methods on the Sod shock tube problem. 70

3.13 Snapshots of the density solutions for the Sod shock tube problem at $t = 0.1644$ on a 128×128 grid for four solution algorithms. 74
3.14 Errors in the density solutions for the Sod shock tube problem at $t = 0.1644$ on a 128×128 grid for four solution algorithms. 75
3.15 Comparison of Sod shock tube problem solution convergence and cost for four different solution methods. 76
3.16 Deterioration of the density solution for the stationary isentropic vortex problem on a 40×40 grid using ALEGRA/ECL and FLASH/PPM. The density profiles along the x-axis are plotted for $t = 0$, 10, 50, and 100. 79
3.17 Deterioration of the density solution for the advecting isentropic vortex problem on a 40×40 grid using ALEGRA/ECL and FLASH/PPM. The density profiles along the x-axis are plotted for $t = 0$, 10, 50, and 100. 80
3.18 Deterioration of the ALEGRA/ECL solution for the stationary (top) and advecting (bottom) isentropic vortex problem on 80×80 grid. The density profiles along the x-axis are plotted for $t = 0$, 10, 50, and 100. 81
3.19 Cutplanes showing snapshots of the pressure field for the Sedov problem at $t = 0.2$. 84
3.20 Density and pressure solutions as a function of distance from center for Sedov problem at $t = 0.2$. 86
3.21 Density and pressure solutions as a function of distance from center for Sedov problem at $t = 0.2$. 87
3.22 Density and pressure solutions as a function of distance from center for Sedov problem at $t = 0.2$. 88
3.23 Density and pressure solutions as a function of distance from center for Sedov problem at $t = 0.2$. 89
3.24 Comparison of Sod shock tube problem solutions obtained with KT and KNP flux methods. 93
3.25 Error vs. cpu time comparison for the in-line ideal gas EOS and general EOS interface accessing ideal gas for solving Sod shock tube problem. 94

4.1 Stability of various second-order time integration methods plotted versus CFL and effective CFL. Symbols indicate theoretical SSP limits for each method. 102
4.2 Solutions using various SSP and non-SSP time integrators. . . 103
4.3 Stability of various 3rd-order time integration methods plotted versus CFL and effective CFL. Symbols indicate theoretical SSP limits for each method. 104

List of Tables

2.1 Low storage coefficients for the third-order optimal SSP method of Spiteri and Ruuth. 30
2.2 Basic memory requirements in words per cell for the second-order methods. 32
2.3 Basic memory requirements in words per cell for the dimension-by-dimension third-order method. 32
2.4 Basic memory requirements in words per cell for the Euler equations. 33

3.1 Errors and convergence rates for the sine wave advection problem using various reconstruction methods and limiters with the third-order SSPRK33 time integrator. 42
3.2 Errors and convergence rates for the sine wave advection problem using matched reconstruction and time integrator methods. 43
3.3 Errors and convergence rates for the square wave advection problem using various reconstruction methods. 45
3.4 Errors and convergence rates for the square wave advection problem at $t = 10$. 49
3.5 Errors and convergence rates for the square wave advection problem at $t = 100$. 49
3.6 Errors and estimated order of convergence for solutions to Burgers' equation at $t = 0.5$ (pre-shock). 52
3.7 Errors and estimated order of convergence for solutions to Burgers' equation at $t = 0$. 53
3.8 Errors and estimated order of convergence for solutions to Burgers' equation at $t = 0.5$ (pre-shock). 54
3.9 L-infinity Errors and estimated order of convergence for solutions to Burgers' equation at $t = 0.5$ (pre-shock). 54
3.10 Errors and estimated order of convergence for solutions to Burgers' equation at $t = 1.5$. 58
3.11 Errors and estimated order of convergence for solutions to Burgers' equation at $t = 2$. 59

3.12 Errors and estimated order of convergence for solutions to Burgers' equation at $t = 3$. 60
3.13 Errors and convergence rates for the Gaussian pulse advection problem using various reconstruction methods. 62
3.14 Errors and convergence rates for the square pulse advection problem using various reconstruction methods. 64
3.15 Global convergence rates for first, second, and third-order methods on the Sod shocktube problem. 66
3.16 Performance of ALEGRA/ECL vs. other codes running the shocktube on a 256x256 mesh to time $t = 0.1644$ s. 73
3.17 Comparison of ALEGRA/ECL vs. ALEGRA/STR density solution and speed for Sedov problem. 85
3.18 Comparison of ALEGRA/ECL vs. ALEGRA/STR pressure solution for Sedov problem. 85
3.19 Symmetry metrics for ALEGRA/STR based on the Sedov problem at $t = 0.2$. 90

4.1 SSP stability coefficients and truncation constants for various RK2 methods. 98
4.2 Limiting SSP CFL factors and effective CFL numbers. 98
4.3 Limiting SSP CFL factors and effective CFL numbers for third-order schemes. 100

Chapter 1

Introduction

Over the past decade, high-resolution methods for solving systems of conservation laws have been widely embraced in research areas ranging from astrophysics to aero-thermodynamics and geophysics. However, issues such as the treatment of general tabular material models, the inclusion of material strength via a stress deviator, the robust treatment of multiple materials with interfaces, and the use of implicit or hybrid time-integration methods to treat problems that exhibit a broad range of time-scales are just beginning to be explored. This work constitutes a first step towards a generalized simulation framework based on high-resolution algorithms for conservation laws to address problems that involve: 1) predominantly hyperbolic systems where strong shocks are present, 2) real material properties, e.g., tabular material models, and 3) multiple materials with material-interfaces.

1.1 Motivation and Background

There are several motivating factors for this research. First, shock hydrodynamics simulation is a core competency for Sandia with multiple programs relying on a sound shock hydrodynamics simulation capability. Second, research in high-resolution methods for solving systems of hyperbolic conservation laws has produced robust algorithms capable of accurately treating near-discontinuities while retaining high-order accuracy on smooth data, making them ideal for treating shock problems. However, it remains to be demonstrated that high-resolution methods for conservation laws are an optimal match for problems that involve real materials with tabular properties and strength, multiple materials with multiple material interfaces, and where multiple and disparate time scales are present.

Many important problems in mathematical physics can be cast in a gen-

eral conservation law form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}_i(\mathbf{u})}{\partial x_i} = \mathbf{G}(\mathbf{u}), \tag{1.1}$$

where $\mathbf{u}$ is the vector of conserved variables, e.g., $\mathbf{u} = \{\rho, \rho u, \rho v, \rho w, E\}^T$ in the case of the Euler equations. The flux function $\mathbf{F}(\mathbf{u})$ defines the relationship between conserved variables and the local flux. The conservation law may be modified by source terms, $\mathbf{G}(\mathbf{u})$, that may be dissipative and possibly stiff, requiring attention to the details of the time integration methods. However, a generalized simulation framework should be able to deal with systems with such source terms in an efficient manner. For example, carefully designed linear multi-step or Runge-Kutta time integrators can yield relaxed stability criteria in the case when additional dissipation is available from $\mathbf{G}(\mathbf{u})$ [3]. A complete "coupled-physics" modeling approach should confront these issues in their entirety.

High-resolution methods for systems of hyperbolic conservation laws are typically at least second-order accurate in space and time, deliver solutions that are non-oscillatory in the presence of near discontinuities, e.g., shocks, and introduce minimal dispersive and diffusive artifacts. Although high-resolution methods for hyperbolic conservation laws have enjoyed great success in the computational fluid dynamics (CFD) arena, they have typically been applied only to flows with simple polytropic gases. In contrast, typical Sandia applications involve the treatment of multiple materials that frequently require a tabular equation-of-state (EOS) and materials with strength, e.g., elastic-plastic materials. The focus of our effort is to apply high-resolution methodologies to more complicated problems to determine what the relevant numerical issues are, and to develop solution algorithms that outperform the current production shock hydrodynamics simulation methodologies in use at Sandia.

Turning to the multi-material problem, the numerical treatment of material interfaces can lead to significant errors, especially in terms of global conservation properties – a non-trivial subject both from a theoretical and an implementation point of view. There are several different algorithmic approaches that are possible for a multi-material treatment in a conservation law framework. One possibility is a direct interface treatment, i.e., interface tracking, which typically uses lower dimensional data structures to specify fluid interfaces. While tractable for certain classes of two-dimensional problems, for our purposes, this approach appears to pose tremendous computational obstacles – particularly in three-dimensions.

In contrast, interface reconstruction procedures use local material volume fraction information to determine rules for advection of material between grid cells. This approach is also referred to as a pseudo-concentration method

where the pseudo-concentration (i.e., volume fraction) is evolved as a scalar quantity using an advection equation. To develop a multiple material interface algorithm within a general conservation law framework one additional scalar transport equation is added for each pseudo-concentration, also referred to as a color function. The material interfaces are defined at a fixed level of each pseudo-concentration or color function. A continuous color function methodology should allow for natural treatments of regularizing physics such as surface tension. These methods are one form of a level-set method in which fluid volume fractions or "colors" are used to generate surfaces that correspond to material interfaces.

Level-set technologies have become popular as a way to easily and robustly deal with singular surfaces of various sorts. For example the work of Fedkiw, et al. [9] solves for single fluids separately by filling in "ghost-fluid regions" with appropriate values that are consistent with the interface motion. Such methodologies can be extended to model "shocks" separating fluid states. These schemes can reproduce jump conditions exactly and do not reduce the solution accuracy to first-order at singular surfaces [5]. This basic approach can also be used to properly provide an interface between an Eulerian conservation law scheme and a purely Lagrangian body [2] for coupled Lagrangian-Eulerian calculations. The level-set/ghost-fluid techniques provide one framework for implementation of consistent boundary jump conditions and interface tracking. A review of such technologies has been provided by Abgrall and Karni [1]. Multi-phase (mixed-phase) methodologies also fit into the general conservation law framework of Eq. (1.1), but require special treatment to account for stiff terms and non-hyperbolicity – as an example, see the work by Saurel and Abgrall [37].

Ultimately, a generalized conservation law simulation framework and the concomitant algorithms constructed using high-resolution methods will impact multiple programs at Sandia that rely on a core competency in computational shock hydrodynamics as the foundation for sophisticated simulation capabilities. Example applications include magneto-hydrodynamic z-pinches, neutron generator power supplies, high-velocity ordinance impact, aero-thermodynamic design, blast loading on structures, and ultimately applications such as hybrid Eulerian-Lagrangian earth-penetrator problems.

The remainder of this chapter outlines the conservation law framework and provides a historical perspective of the central schemes and time integrators considered in this work. Chapter 2 addresses the formulation issues for the conservation law framework. In Chapter 3 a detailed methods comparison/evaluation is presented. This chapter includes a summary of several pathologies in the third-order reconstruction methods in §3.4. In Chapter 4, the strong-stability preserving Runge-Kutta time integrators are analyzed. Finally, a summary is presented with concluding remarks on this work in

Chapter 5.

1.2 The Conservation Law Framework

The primary focal point of this work is the development of a general framework, both the mathematical and software, that can support high-resolution algorithms for application to essentially any physics model that can be placed within the generalized mathematical framework for a system of hyperbolic conservation laws.

The discretization approach operates on the formal mathematical framework of hyperbolic conservation laws. Conceptually, the details of the physics are placed in pre-specified routines that provide the flux functions and other necessary information, e.g., maximal eigenvalues. The application developer provides only minimal information, but is required to do some analysis of the system to be solved prior to implementation. Initial efforts have focused on investigating high-resolution semi-discrete methods for systems of conservation laws that do not require extensive Riemann decompositions.

1.3 Central Schemes

This section provides a brief survey of the relevant literature on central or central-upwind schemes for systems of hyperbolic conservation laws. A general overview of hyperbolic conservation laws and the associated numerical solution techniques is given by Leveque [24] (or see Chapter 6 of Leveque [25]).

Central or central-upwind schemes can be easily implemented with only a knowledge of the physical flux functions $\mathbf{F}_i$ and the extremal eigenvalues of the flux-Jacobian $\mathbf{A}_i = \partial \mathbf{F}_i / \partial \mathbf{u}$ associated with a strictly hyperbolic system of conservation laws.

The current generation of Godunov-type central or central upwind schemes may be traced back to the work of Nessyahu and Tadmor [32] in 1990, where they constructed their second-order staggered-grid high-resolution scheme using a Lax-Friedrichs (LxF) solver [23] rather than a Riemann decomposition. The Nessyahu and Tadmor (NT) method avoids the well-known excessive dissipation of the LxF algorithm by relying on high-resolution non-oscillatory reconstruction techniques in conjunction with an averaging over Riemann fans at cell interfaces. Like Rusanov's method [36], also known as the local Lax-Friedrichs method, the NT algorithm, in effect, selects the local viscosity coefficient based on the Riemann problem. However, the NT algorithm provides a framework for obtaining a high-resolution Godunov-like method

without resorting to the use of an explicit Riemann solver and overcomes the first-order accuracy of the LxF method.

In 1996, Liu and Osher [30] developed a third-order accurate central scheme based on a quadratic reconstruction procedure. Note that the formal accuracy will only be observed on smooth data with a reduction to $O(h)$ behavior in regions of non-smooth data. The reconstruction enforces a local maximum principle and guarantees that no new extrema are introduced. Liu and Tadmor [31] introduced a third-order extension of the original staggered NT algorithm for scalar conservation laws. The extension to systems of conservation laws is implemented component-by-component and retains the third-order accuracy of the scalar algorithm.

Jiang, et al. [14] later extended the NT algorithm to multiple space dimensions and demonstrated that the central scheme does not require a spatial splitting. Subsequently, Jiang, et al. [14] introduced a non-staggered central scheme that retains the "Riemann-solver-free" aspect of the original staggered NT algorithm. Here, it was first suggested that central schemes could form the basis for a robust generalized computational framework for systems of conservation laws.

Bianco, et al., [6] developed third and fourth-order staggered schemes based on the NT algorithm, but used essentially non-oscillatory (ENO) reconstruction rather than the MUSCL type employed by Nessayahu and Tadmor. In this work, several Runge-Kutta time integrators were assessed – although none was categorized as strong-stability preserving (SSP). The work by Levy, Puppo and Russo [26] considered a weighted ENO (WENO) reconstruction in conjunction with Runge-Kutta time integrators for third and fourth-order accurate schemes – albeit in only one space dimension.

In 2000, Kurganov and Tadmor [21] (KT) used a more precise averaging over the Riemann fans to obtain a refined central scheme that reduced the numerical viscosity present in the original NT scheme from $O(\Delta x^{2r}/\Delta t)$ to $O(\Delta x^{2r-1})$ where r is the formal order of spatial accuracy for the method. In addition, this formulation yields a consistent semi-discrete method in the limit as $\Delta t \to 0$. In this work, the KT algorithm is implemented with both third and fourth-order Runge-Kutta time integrators and applied to problems with both convex and non-convex flux functions. A third-order semi-discrete version of the KT method was outlined by Kurganov and Levy [17] using the WENO reconstruction from Levy, et al. [26].

Central schemes trade their simplicity and ease of programming for accuracy in the approximation of intermediate wave discontinuities such as contact discontinuities. This can be remedied at the price of extra effort to extract and use additional information from the characteristic fields. Kurganov and Petrova [19] have proposed such a refinement to central upwind schemes to more accurately represent contact waves. Here, a partial characteristic

decomposition is used to obtain a more accurate width of the Riemann fan for averaging, and to more faithfully represent the local propagation speeds – albeit at the cost of simplicity.

In contrast, Lie and Noelle [28] have investigated the use of Harten's artificial compression method (ACM) in the context of central schemes. While some improvements in resolving the linear contact wave can be observed, ACM can not distinguish between near-discontinuities, e.g., near a discrete shock profile, and extrema in smooth regions of a field. The inability to distinguish can result in artificial steepening and the generation of entropy-violating shocks.

Recent work by Linde [29] has suggested a refined two-state HLL Riemann solver to improve the treatment of contact discontinuities. This approach may be useful in the context of a generalized conservation law framework, but requires additional information beyond the extremal eigenvalues of the flux Jacobian.

A non-staggered Godunov-type semi-discrete central scheme was introduced by Kurganov, Noelle and Petrova in 2001 [18]. A non-split (genuinely multi-dimensional) central scheme along with a third-order limiter based on a convex combination of a limited linear reconstruction and a quadratic reconstruction has also been presented by Kurganov and Petrova [20]. A series of computational experiments on two-dimensional Riemann problems using these genuinely multi-dimensional schemes is reported on by Kurganov and Tadmor [22], and demonstrates that, despite their simplicity, the central schemes perform very well in comparison to "Riemann-full" methods. A detailed study of the entropy production of central schemes may be found in the recent work by Puppo [35].

We now turn our attention to time integrators, and note that a semi-discrete formulation for all of the central methods is important since it guarantees that the methods will be robust for the solution of steady state problems. The ensuing discussion provides a brief summary of some of the time integration methods considered in this work.

1.4 Time Integrators

The time integration component of the conservation law framework provides for the integration of a system of ordinary differential equations generated by the spatial discretization. Time integration methods can be developed to satisfy temporal truncation error requirements, and to also optimize for other properties such as dispersive properties (phase and group errors), numerical dissipation, and memory requirements, depending on the particular application constraints that are considered to be most important. The work by Kennedy, et al. [15] and Kennedy and Carpenter [16] provides a good

overview of these technical issues. In general, a time marching method of accuracy compatible with the accuracy of the spatial discretization should be used to integrate forward in time.

It has been proposed that time integration methods that satisfy a strong-stability preserving (SSP) property are preferable for method of lines approaches to solving non-linear hyperbolic conservation laws [12, 13]. SSP methods, sometimes referred to as total variation diminishing (TVD) time discretizations, are high-order time integration algorithms that preserve the strong stability properties of a first-order Euler time integrator. Extended-stage SSP Runge-Kutta (SSP-RK) methods have been developed that have extended the stability limits of the SSP-RK methods as well [40].

Low-storage Runge-Kutta methods in conjunction with high-resolution spatial discretization schemes may be of particular interest for large scale applications of interest to Sandia. One particular time integrator of interest is the third-order low-storage scheme suggested by Williamson [43]. This paper is suggested as a good introduction to the trade-offs that can be made in the design of time integration schemes. We note in passing that Williamson's "preferred" third-order algorithm has been adopted by the FLASH software. This choice is consistent with the FLASH piecewise parabolic spatial discretization [4].

Another important item of note is that Runge-Kutta time integration schemes have the potential of introducing low-order errors at the boundary due to the interaction of Runge-Kutta stages with the boundary conditions – see for example the work by Pathria [34].

Chapter 2

Formulation and Design Issues

This chapter outlines the formulation and framework design issues for the high-resolution central and central-upwind schemes implemented in ALEGRA [7] and the NEVADA [8] code framework. The physics independent conservation law methods described below will eventually become a permanent part of the NEVADA framework and are at present collectively called "ALEGRA/ECL" for Extended Conservation Laws.

Hyperbolic conservation law applications can use the ECL framework to implement a problem-specific solver. The current ECL framework only supports methods that require partial information about the eigenvalues of the flux Jacobian [14, 21, 18] – rather than the entire eigenspectrum. Thus, very limited information about the specific conservation law application is required from the developer. Note that the consideration of exact or approximate Riemann methods that utilize a full set of eigenfunctions or eigenvalues and solution schemes based on exact Riemann solutions is beyond the scope of this report.

For the current work, attention is restricted to problems that can be cast in a homogeneous flux-divergence form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}_i(\mathbf{u})}{\partial x_i} = 0, \tag{2.1}$$

where $1 \leq i \leq N_{dim}$, N_{dim} is the number of space dimensions, and repeated indices imply summation.

The problem is required to be hyperbolic, i.e., the flux Jacobian

$$\mathbf{A}_i = \frac{\partial \mathbf{F}_i(\mathbf{u})}{\partial \mathbf{u}}, \tag{2.2}$$

is required to have both real eigenvalues $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N$ and a complete set of eigenvectors. If the real eigenvalues are distinct then the problem is strictly hyperbolic and a complete set of eigenvectors is guaranteed.

The eigensystem of the flux-Jacobian matrix $\mathbf{A}_i$ plays a key role in the design of high-resolution algorithms and their associated "upwind" behavior. The eigenvalues of the flux-Jacobian are the wave speeds for the physical problem. The right eigenvectors for the flux-Jacobian define the phase-space paths for simple waves while the left eigenvectors define the characteristic equations. In a strictly hyperbolic system of conservation laws, the eigenvalues are all real and distinct. In general, the eigenvalues are categorized as either "genuinely nonlinear" or "linearly degenerate". In general for the Euler equations, the eigenvalues associated with sound speeds are genuinely nonlinear, while the eigenvalue associated with a material velocity is linearly degenerate. Contact discontinuities, i.e. density jumps, correspond to linearly degenerate eigenvalues.

The solution methods implemented in ALEGRA/ECL and discussed in this report are based on the family of Godunov-type solution methods known as central schemes. In central schemes, there are three basic concepts required for updating the average state in each cell of a computational grid. First the solution is reconstructed on cell faces from cell averages, i.e., the reconstruction stage. The second is the definition of the interface flux using the reconstructed left and right values at the cell interface and associated wave speeds. These are essentially spatial operators calculated from the reconstructed conserved variables. The third component of the procedure is the time discretization algorithm for advancing the solution in time. Generally a time integration scheme whose order is matched to the order of the spatial reconstruction method is desired in order to achieve optimal convergence rates. More complicated solver methodologies envisioned for future investigation require additional characteristic information.

2.1 Reconstruction Methods

The semi-discrete methods used here evaluate the flux function at states on the cell boundaries based on a polynomial reconstruction from cell averages. Figure 2.1 shows the grid layout and indexing scheme used for the reconstruction procedure. In the methods discussed below, the reconstructed field values are located at the plus '+' and '−' locations in each cell, where + and − are associated with the right and left sides of each cell *face* respectively. For a given cell at grid-location j, the left face corresponds to $j - 1/2$, and the right face to $j + 1/2$.

The error associated with the reconstruction may be defined as the difference between the reconstructed and exact fields,

$$\mathbf{u}^e(x,t) - \mathbf{u}^r(x,t) = \mathbf{O}(\Delta x^n), \tag{2.3}$$

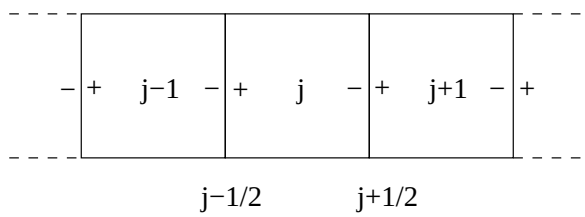


Figure 2.1: Grid layout for reconstruction and flux evaluation.

where $\mathbf{u}^e$ is the exact field and $\mathbf{u}^r$ is the reconstructed field, and the cell-centered average is denoted by an over-bar. The difference between the exact and reconstructed field is $\mathbf{O}(\Delta x^n)$ for an nth-order reconstruction. In our initial efforts, we have restricted our attention to grids with a uniform spacing in order to avoid the additional complexities in the reconstruction methods. In the sections below all min, max and inequality operators acting on vectors are assumed to apply component-by-component.

2.1.1 Constant Reconstruction Methods

For a conserved variable, $\overline{\mathbf{u}}$, the donor-cell reconstruction (DONOR) is

$$\mathbf{u}^-_{j+\frac{1}{2}} = \overline{\mathbf{u}}_j \tag{2.4}$$

$$\mathbf{u}^+_{j-\frac{1}{2}} = \overline{\mathbf{u}}_j. \tag{2.5}$$

The donor reconstruction is a first-order reconstruction method, and is used only as a backup when the high-order reconstruction procedures produce non-physical states. The reconstruction is, in effect, an extrapolation of the cell data to the cell boundaries.

2.1.2 Linear Reconstruction Methods

Each of the following three reconstruction methods is second-order, and uses some form of slope limiting. In each case, the undivided differences are used at grid location j and $j-1$ as

$$\Delta\overline{\mathbf{u}}_j = \overline{\mathbf{u}}_{j+1} - \overline{\mathbf{u}}_j, \tag{2.6}$$

$$\Delta\overline{\mathbf{u}}_{j-1} = \overline{\mathbf{u}}_j - \overline{\mathbf{u}}_{j-1}. \tag{2.7}$$

From these two differences , a limited difference $\delta\mathbf{u}_j$ is determined for each cell. The reconstructed left and right values for each cell are then given by

$$\mathbf{u}^-_{j+\frac{1}{2}} = \overline{\mathbf{u}}_j + \frac{1}{2}\delta\mathbf{u}_j \tag{2.8}$$

$$\mathbf{u}^+_{j-\frac{1}{2}} = \overline{\mathbf{u}}_j - \frac{1}{2}\delta\mathbf{u}_j. \tag{2.9}$$

Harmonic van Leer (VANLEER)

The second-order harmonic van Leer limiter is

$$\delta \mathbf{u}_j = \begin{cases} \frac{\Delta\overline{\mathbf{u}}_j \Delta\overline{\mathbf{u}}_{j-1}}{\Delta\overline{\mathbf{u}}_j + \Delta\overline{\mathbf{u}}_{j-1}} & \text{if } \Delta\overline{\mathbf{u}}_j \Delta\overline{\mathbf{u}}_{j-1} > 0, \\ \mathbf{0} & \text{if } \Delta\overline{\mathbf{u}}_j \Delta\overline{\mathbf{u}}_{j-1} \leq 0. \end{cases} \tag{2.10}$$

Superbee (SUPERBEE)

The second-order superbee limiter is

$$\delta \mathbf{u}_j = \begin{cases} \max(\min(\Delta\overline{\mathbf{u}}_j, 2\Delta\overline{\mathbf{u}}_{j-1}), \min(\Delta\overline{\mathbf{u}}_{j-1}, 2\Delta\overline{\mathbf{u}}_j)) & \text{if } \Delta\overline{\mathbf{u}}_j, \Delta\overline{\mathbf{u}}_{j-1} > 0, \\ \min(\max(\Delta\overline{\mathbf{u}}_j, 2\Delta\overline{\mathbf{u}}_{j-1}), \max(\Delta\overline{\mathbf{u}}_{j-1}, 2\Delta\overline{\mathbf{u}}_j)) & \text{if } \Delta\overline{\mathbf{u}}_j, \Delta\overline{\mathbf{u}}_{j-1} < 0, \\ \mathbf{0} & \text{if } \Delta\overline{\mathbf{u}}_j \Delta\overline{\mathbf{u}}_{j-1} \leq 0. \end{cases} \tag{2.11}$$

Minmod (MINMOD)

The second-order minmod limiter is

$$\delta \mathbf{u}_j = \begin{cases} \min(\omega\Delta\overline{\mathbf{u}}_j, \omega\Delta\overline{\mathbf{u}}_{j-1}, (\Delta\overline{\mathbf{u}}_j + \Delta\overline{\mathbf{u}}_{j-1})/2) & \text{if } \Delta\overline{\mathbf{u}}_j, \Delta\overline{\mathbf{u}}_{j-1} > 0, \\ \max(\omega\Delta\overline{\mathbf{u}}_j, \omega\Delta\overline{\mathbf{u}}_{j-1}, (\Delta\overline{\mathbf{u}}_j + \Delta\overline{\mathbf{u}}_{j-1})/2) & \text{if } \Delta\overline{\mathbf{u}}_j, \Delta\overline{\mathbf{u}}_{j-1} < 0, \\ \mathbf{0} & \text{if } \Delta\overline{\mathbf{u}}_j \Delta\overline{\mathbf{u}}_{j-1} \leq 0, \end{cases} \tag{2.12}$$

where $1 \leq \omega \leq 2$. For $\omega = 2$ we recover the second variant of van Leer's original limiting procedure.

2.1.3 Quadratic Reconstruction Methods

Two quadratic reconstruction methods have been investigated for use in the conservation law framework. Both techniques are implemented in a "dimension-by-dimension" form that permits simple one-dimensional grid updates. These reconstruction methods have been found to be deficient and are, in fact, not provably third-order as claimed. The details of this proof may be found in §3.4.

Liu and Osher (LT3)

This reconstruction is performed dimension-by-dimension and is based on the third-order non-oscillatory reconstruction of Liu and Osher [30]. Liu and Tadmor [31] describe this approach in detail. First, a quadratic reconstruction is used to determine right and left values,

$$\mathbf{q}_j(x_{j\pm\frac{1}{2}}) = \overline{\mathbf{u}}_j + \frac{\Delta\overline{\mathbf{u}}_j - \Delta\overline{\mathbf{u}}_{j-1}}{12} \pm \frac{\Delta\overline{\mathbf{u}}_j + \Delta\overline{\mathbf{u}}_{j-1}}{4}. \tag{2.13}$$

A parameter θ_j is then determined from

$$\theta_j = \begin{cases} \min\{\frac{\mathbf{M}_{j+\frac{1}{2}}-\overline{\mathbf{u}}_j}{\mathbf{M}_j-\overline{\mathbf{u}}_j}, \frac{\mathbf{m}_{j-\frac{1}{2}}-\overline{\mathbf{u}}_j}{\mathbf{m}_j-\overline{\mathbf{u}}_j}, 1\} & \text{if } \Delta\overline{\mathbf{u}}_j, \Delta\overline{\mathbf{u}}_{j-1} > 0, \\ \min\{\frac{\mathbf{M}_{j-\frac{1}{2}}-\overline{\mathbf{u}}_j}{\mathbf{M}_j-\overline{\mathbf{u}}_j}, \frac{\mathbf{m}_{j+\frac{1}{2}}-\overline{\mathbf{u}}_j}{\mathbf{m}_j-\overline{\mathbf{u}}_j}, 1\} & \text{if } \Delta\overline{\mathbf{u}}_j, \Delta\overline{\mathbf{u}}_{j-1} < 0, \\ \mathbf{1} & \text{otherwise,} \end{cases} \tag{2.14}$$

where

$$\mathbf{M}_j = \max(\mathbf{q}_j(x_{j-\frac{1}{2}}), \mathbf{q}_j(x_{j+\frac{1}{2}})) \tag{2.15}$$

$$\mathbf{m}_j = \min(\mathbf{q}_j(x_{j-\frac{1}{2}}), \mathbf{q}_j(x_{j+\frac{1}{2}})) \tag{2.16}$$

$$\mathbf{M}_{j\pm\frac{1}{2}} = \max(\frac{1}{2}(\overline{\mathbf{u}}_j + \overline{\mathbf{u}}_{j\pm1}), \mathbf{q}_{j\pm1}(x_{j\pm\frac{1}{2}})) \tag{2.17}$$

$$\mathbf{m}_{j\pm\frac{1}{2}} = \min(\frac{1}{2}(\overline{\mathbf{u}}_j + \overline{\mathbf{u}}_{j\pm1}), \mathbf{q}_{j\pm1}(x_{j\pm\frac{1}{2}})). \tag{2.18}$$

Finally, the left and right reconstructed field values are computed using a convex combination of the second-order interpolant and the cell-average field values,

$$\mathbf{u}^{\mp}_{j\pm\frac{1}{2}} = \overline{\mathbf{u}}_j + \theta_j(\mathbf{q}_j(x_{j\pm\frac{1}{2}}) - \overline{\mathbf{u}}_j). \tag{2.19}$$

Kurganov and Petrova (KP3)

The KP3 reconstruction is also based on that given by Liu and Osher [30], described above, and uses a convex combination of the same quadratic reconstruction but with a linear interpolant (i.e., a second-order reconstruction) as described by Kurganov and Petrova [20].

Again, a quadratic reconstruction is used to determine the right and left quadratic interpolant values,

$$\mathbf{q}_j(x_{j\pm\frac{1}{2}}) = \overline{\mathbf{u}}_j + \frac{\Delta\overline{\mathbf{u}}_j - \Delta\overline{\mathbf{u}}_{j-1}}{12} \pm \frac{\Delta\overline{\mathbf{u}}_j + \Delta\overline{\mathbf{u}}_{j-1}}{4}, \tag{2.20}$$

using right and left values of the linear interpolant,

$$\mathbf{L}_j(x_{j\pm\frac{1}{2}}) = \overline{\mathbf{u}}_j + \mathbf{s_j}(\mathbf{x_{j\pm\frac{1}{2}}} - \mathbf{x_j}). \tag{2.21}$$

that is obtained using a second-order TVD reconstruction method. The non-oscillatory behavior of the quadratic reconstruction is based, in part, on the specific choice of the slope, $\mathbf{s_j}$.

We note here that the only second-order TVD method currently implemented in ALEGRA/ECL is minmod with $\omega = 1$. The rest of the second-order reconstruction methods are not non-oscillatory, though they yield TVD

semi-discrete schemes when combined with either of the currently implemented numerical flux options discussed in §2.2 as was shown by Tadmor [41].

In order to ensure that the final piecewise quadratic reconstruction remains non-oscillatory, a parameter, θ_j, is chosen as

$$\theta_j = \begin{cases} \min\{\frac{\mathbf{M}_{j+\frac{1}{2}}-\mathbf{L}_j}{\mathbf{M}_j-\mathbf{L}_j}, \frac{\mathbf{m}_{j-\frac{1}{2}}-\mathbf{L}_j}{\mathbf{m}_j-\mathbf{L}_j}, 1\} & \text{if } \Delta\overline{\mathbf{u}}_j, \Delta\overline{\mathbf{u}}_{j-1} > 0, \\ \min\{\frac{\mathbf{M}_{j-\frac{1}{2}}-\mathbf{L}_j}{\mathbf{M}_j-\mathbf{L}_j}, \frac{\mathbf{m}_{j+\frac{1}{2}}-\mathbf{L}_j}{\mathbf{m}_j-\mathbf{L}_j}, 1\} & \text{if } \Delta\overline{\mathbf{u}}_j, \Delta\overline{\mathbf{u}}_{j-1} < 0, \\ \mathbf{1} & \text{otherwise,} \end{cases} \tag{2.22}$$

where

$$\mathbf{M}_j = \max(\mathbf{q}_j(x_{j-\frac{1}{2}}), \mathbf{q}_j(x_{j+\frac{1}{2}})) \tag{2.23}$$

$$\mathbf{m}_j = \min(\mathbf{q}_j(x_{j-\frac{1}{2}}), \mathbf{q}_j(x_{j+\frac{1}{2}})) \tag{2.24}$$

$$\mathbf{M}_{j\pm\frac{1}{2}} = \max(\frac{1}{2}(\mathbf{L}_j + \mathbf{L}_{j\pm1}), \mathbf{q}_{j\pm1}(x_{j\pm\frac{1}{2}})) \tag{2.25}$$

$$\mathbf{m}_{j\pm\frac{1}{2}} = \min(\frac{1}{2}(\mathbf{L}_j + \mathbf{L}_{j\pm1}), \mathbf{q}_{j\pm1}(x_{j\pm\frac{1}{2}})). \tag{2.26}$$

Finally, the left and right reconstructed field values are calculated as

$$\mathbf{u}^{\mp}_{j\pm\frac{1}{2}} = \mathbf{L}_j(x_{j\pm\frac{1}{2}}) + \theta_j(\mathbf{q}_j(x_{j\pm\frac{1}{2}}) - \mathbf{L}_j(x_{j\pm\frac{1}{2}})). \tag{2.27}$$

2.2 Numerical Flux Methods

One-dimensional grid updates of the average conserved field variables are calculated using a conservative update as

$$\frac{d}{dt}\overline{\mathbf{u}}_j(t) = -\frac{\mathbf{H}_{j+\frac{1}{2}} - \mathbf{H}_{j-\frac{1}{2}}}{\Delta x} \tag{2.28}$$

where the numerical fluxes, $\mathbf{H}_{j\pm\frac{1}{2}}$, are given according to some prescription. Two numerical flux methods, the Kurganov-Tadmor (KT) and the Kurganov-Noelle-Petrova (KNP) methods, have been investigated with the central schemes in ALEGRA/ECL. Each flux method relies on estimates of the minimum and maximum wave speeds at the cell interfaces, which are computed using

$$a^N_{j+\frac{1}{2}} = \max\left[\lambda_N\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}(\mathbf{u}^-_{j+\frac{1}{2}})\right), \lambda_N\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}(\mathbf{u}^+_{j+\frac{1}{2}})\right), 0\right], \tag{2.29}$$

and

$$a^1_{j+\frac{1}{2}} = \min\left[\lambda_1\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}(\mathbf{u}^-_{j+\frac{1}{2}})\right), \lambda_1\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}(\mathbf{u}^+_{j+\frac{1}{2}})\right), 0\right], \tag{2.30}$$

where $\mathbf{f}$ represents one spatial component of the flux vector $\mathbf{F}_i$, λ_1 is the minimal eigenvalue of the flux Jacobian, and λ_N is the maximal eigenvalue.

The one-dimensional flux updates, as described below, are compatible with a second-order reconstruction and the dimension-by-dimension third-order reconstruction methods. A genuinely multi-dimensional third-order method would require flux evaluations at cell vertices as well as at cell faces as described by Kurganov and Petrova [20].

2.2.1 Kurganov-Tadmor (KT)

The Kurganov-Tadmor numerical flux (KT) uses only the maximum wave speed, $a_{j+\frac{1}{2}} = \max(|a^N_{j+\frac{1}{2}}|, |a^1_{j+\frac{1}{2}}|)$, at the cell interfaces [21]. The numerical flux is

$$\mathbf{H}_{j+\frac{1}{2}} = \frac{\mathbf{F}(\mathbf{u}^-_{j+\frac{1}{2}}) + \mathbf{F}(\mathbf{u}^+_{j+\frac{1}{2}})}{2} - \frac{a_{j+\frac{1}{2}}}{2}(\mathbf{u}^+_{j+\frac{1}{2}} - \mathbf{u}^-_{j+\frac{1}{2}}). \tag{2.31}$$

2.2.2 Kurganov-Noelle-Petrova (KNP)

The central-upwind flux of Kurganov, Noelle and Petrova uses both maximal wave speeds, a^1 and a^N, at cell interfaces [18], i.e.,

$$\mathbf{H}_{j+\frac{1}{2}} = \frac{a^N_{j+\frac{1}{2}}\mathbf{F}(\mathbf{u}^-_{j+\frac{1}{2}}) - a^1_{j+\frac{1}{2}}\mathbf{F}(\mathbf{u}^+_{j+\frac{1}{2}})}{a^N_{j+\frac{1}{2}} - a^1_{j+\frac{1}{2}}} + \frac{a^N_{j+\frac{1}{2}}a^1_{j+\frac{1}{2}}}{a^N_{j+\frac{1}{2}} - a^1_{j+\frac{1}{2}}}(\mathbf{u}^+_{j+\frac{1}{2}} - \mathbf{u}^-_{j+\frac{1}{2}}). \tag{2.32}$$

> **Remark.** It is possible for the denominator to be zero in Eq. (2.32) when $a^N_{j+\frac{1}{2}} = a^1_{j+\frac{1}{2}} = 0$. The algorithm, as implemented in ALEGRA/ECL, checks for this pathology and uses the KT flux method in this situation. In this case, the numerical flux then reduces to the average of the interface fluxes as indicated by Eq. (2.31).

2.3 Time Integration Methods

This section outlines the suite of time integration methods that are provided in the conservation law framework as implemented in ALEGRA/ECL. As described above, the time integrators should be chosen to be compatible with the reconstruction method in order to achieve the desired overall numerical accuracy. For our purposes, an nth-order accurate algorithm for a hyperbolic system of conservation laws indicates that the truncation error in the numerical approximation is $O(\Delta x^n, \Delta t^n)$. Typically we use an nth-order time stepping method with an nth-order reconstruction method.

In ALEGRA/ECL, all of the time integration methods are explicit. Thus, it is necessary to choose time steps based on maximal eigenvalues and suitable stability criteria. The central methods described above may be exercised at a grid CFL (Courant-Freidrichs-Levy) condition given by

$$\frac{\max(|\lambda_1|, |\lambda_N|)\ \Delta t}{h} \leq \frac{1}{2}, \tag{2.33}$$

where h is a measure of the cell size, e.g., $h = \Delta x$ for a uniform Cartesian grid. Thus, the upper limit for stability corresponds to CFL = 1/2.

In addition to the stability constraints, there are issues associated with the interaction between high-order Runge-Kutta time integrators and boundary conditions. This is beyond the scope of the current report, but details on the known boundary condition compatibility issues may be found in the work by Pathria [34].

The time integration methods that are available in ALEGRA/ECL are summarized below.

2.3.1 First-Order Method

The first-order forward-Euler method (EULER) is

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \mathcal{L}(\mathbf{u}^n) \tag{2.34}$$

where $\mathcal{L}(\mathbf{u})$ is the discrete operator for the numerical flux function gradient. In a one-dimensional sense, $\mathcal{L}(\mathbf{u})$ corresponds to the right-hand-side of Eq. (2.28) which is a discrete flux gradient formed by taking a difference of the left and right numerical flux functions. In a multi-dimensional sense, $\mathcal{L}(\mathbf{u})$ represents the discrete flux-divergence based on the differences across cell faces.

2.3.2 Second-Order Methods

The second-order Runge-Kutta (RK) methods available in the conservation law framework are outlined below. A subset of these integrators are the so-called "strong-stability preserving" (SSP) methods. These methods preserve the strong-stability properties of a first-order Euler method over multiple time integration stages – if the associated first-order Euler method has these properties. In the ensuing discussion, we identify SSP RK methods by their order p, number of stages s, and the SSP CFL stability coefficient c. The stability coefficient sets the SSP stable time step relative to the time step of the first-Order Euler method,

$$\Delta t_{SSP} \leq c\ \Delta t_{FE}. \tag{2.35}$$

Additional details on strong-stability preserving Runge-Kutta methods may be found in Chapter 4.

Midpoint Runge-Kutta (RK2)

The second-order midpoint rule Runge-Kutta time integration scheme is

$$\mathbf{u}^{(1)} = \mathbf{u}^n + \frac{1}{2}\Delta t\mathcal{L}(\mathbf{u}^n), \tag{2.36}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\mathcal{L}(\mathbf{u}^{(1)}). \tag{2.37}$$

Minimal Truncation Error Runge-Kutta (MTRK2)

The second-order minimal truncation error Runge-Kutta time integration algorithm is

$$\mathbf{u}^{(1)} = \mathbf{u}^n + \frac{2}{3}\Delta t\mathcal{L}(\mathbf{u}^n), \tag{2.38}$$

$$\mathbf{u}^{n+1} = \frac{5}{8}\mathbf{u}^n + \frac{3}{8}\mathbf{u}^{(1)} + \frac{3}{4}\Delta t\mathcal{L}(\mathbf{u}^{(1)}). \tag{2.39}$$

Gottlieb & Shu's (very) non-SSP Runge Kutta (GSRK2)

This integrator was used by Gottleib and Shu [12] to contrast with the SSP Runge-Kutta integrators. The second-order method is

$$\mathbf{u}^{(1)} = \mathbf{u}^n - 20\Delta t\mathcal{L}(\mathbf{u}^n), \tag{2.40}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \frac{41}{40}\Delta t\mathcal{L}(\mathbf{u}^{(n)}) - \frac{1}{40}\Delta t\mathcal{L}(\mathbf{u}^{(1)}). \tag{2.41}$$

This method is clearly not SSP, as indicated by the negative coefficients multiplying $\mathcal{L}(\mathbf{u})$ (see Chapter 4 for details).

Second-Order SSP Runge-Kutta (SSPRK22)

The optimal second-order strong-stability preserving Runge-Kutta method [13] is

$$\mathbf{u}^{(1)} = \mathbf{u}^n + \Delta t\mathcal{L}(\mathbf{u}^n), \tag{2.42}$$

$$\mathbf{u}^{n+1} = \frac{1}{2}\mathbf{u}^n + \frac{1}{2}\mathbf{u}^{(1)} + \frac{1}{2}\Delta t\mathcal{L}(\mathbf{u}^{(1)}), \tag{2.43}$$

which has an SSP CFL coefficient $c = 1$.

Three-stage, Second-Order SSP Runge-Kutta (SSPRK32)

Spiteri and Ruuth [40] give optimal s-stage, p-order SSP RK methods where $s > p$. These methods are advantageous since the extra stages extend the

effective stability limit. The three-stage integrator is

$$\mathbf{u}^{(1)} = \mathbf{u}^n + \frac{1}{2}\Delta t\mathcal{L}(\mathbf{u}^n), \tag{2.44}$$

$$\mathbf{u}^{(2)} = \mathbf{u}^{(1)} + \frac{1}{2}\Delta t\mathcal{L}(\mathbf{u}^{(1)}), \tag{2.45}$$

$$\mathbf{u}^{n+1} = \frac{1}{3}\mathbf{u}^n + \frac{2}{3}\mathbf{u}^{(2)} + \frac{1}{3}\Delta t\mathcal{L}(\mathbf{u}^{(2)}), \tag{2.46}$$

which has an SSP CFL coefficient $c = 2$.

Four-stage, Second-Order SSP Runge-Kutta (SSPRK42)

The four-stage variant for the SSP RK integrator presented by Spiteri and Ruuth is

$$\mathbf{u}^{(1)} = \mathbf{u}^n + \frac{1}{3}\Delta t\mathcal{L}(\mathbf{u}^n), \tag{2.47}$$

$$\mathbf{u}^{(2)} = \mathbf{u}^{(1)} + \frac{1}{3}\Delta t\mathcal{L}(\mathbf{u}^{(1)}), \tag{2.48}$$

$$\mathbf{u}^{(3)} = \mathbf{u}^{(2)} + \frac{1}{3}\Delta t\mathcal{L}(\mathbf{u}^{(2)}), \tag{2.49}$$

$$\mathbf{u}^{n+1} = \frac{1}{4}\mathbf{u}^n + \frac{3}{4}\mathbf{u}^{(3)} + \frac{1}{4}\Delta t\mathcal{L}(\mathbf{u}^{(3)}), \tag{2.50}$$

which has an SSP CFL coefficient $c = 3$.

2.3.3 Third-Order Methods

Third-Order SSP Runge-Kutta (SSPRK33)

The optimal third-order SSP method [13] is

$$\mathbf{u}^{(1)} = \mathbf{u}^n + \Delta t\mathcal{L}(\mathbf{u}^n), \tag{2.51}$$

$$\mathbf{u}^{(2)} = \frac{3}{4}\mathbf{u}^n + \frac{1}{4}\mathbf{u}^{(1)} + \frac{1}{4}\Delta t\mathcal{L}(\mathbf{u}^{(1)}), \tag{2.52}$$

$$\mathbf{u}^{n+1} = \frac{1}{3}\mathbf{u}^n + \frac{2}{3}\mathbf{u}^{(2)} + \frac{2}{3}\Delta t\mathcal{L}(\mathbf{u}^{(2)}), \tag{2.53}$$

which has an SSP CFL coefficient $c = 1$.

Four-stage, Third-Order SSP Runge-Kutta (SSPRK43)

The four-stage third-order SSP RK method is

$$\mathbf{u}^{(1)} = \mathbf{u}^n + \frac{1}{2}\Delta t\mathcal{L}(\mathbf{u}^n), \tag{2.54}$$

$$\mathbf{u}^{(2)} = \mathbf{u}^{(1)} + \frac{1}{2}\Delta t\mathcal{L}(\mathbf{u}^1), \tag{2.55}$$

$$\mathbf{u}^{(3)} = \frac{2}{3}\mathbf{u}^n + \frac{1}{3}\mathbf{u}^{(2)} + \frac{1}{6}\Delta t\mathcal{L}(\mathbf{u}^{(2)}), \tag{2.56}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{(3)} + \frac{1}{2}\Delta t\mathcal{L}(\mathbf{u}^{(3)}), \tag{2.57}$$

which has an SSP CFL coefficient $c = 2$.

Minimal Truncation Error Runge-Kutta (RK3)

The third-order minimal truncation error Runge-Kutta scheme is

$$\mathbf{u}^{(1)} = \mathbf{u}^n + \frac{1}{2}\Delta t\mathcal{L}(\mathbf{u}^n), \tag{2.58}$$

$$\mathbf{u}^{(2)} = \mathbf{u}^n + \frac{3}{4}\Delta t\mathcal{L}(\mathbf{u}^{(1)}), \tag{2.59}$$

$$\mathbf{u}^{n+1} = \frac{1}{9}\mathbf{u}^n + \frac{4}{9}\mathbf{u}^{(1)} + \frac{4}{9}\mathbf{u}^{(2)} + \Delta t\mathcal{L}(\mathbf{u}^{(1)}). \tag{2.60}$$

Williamson's Low-Storage Runge-Kutta (WLSRK33)

The details on Williamson's one-parameter family of low-storage third-order Runge Kutta methods may be found in [43] The two-state low-storage models can be implemented with only a state $\mathbf{u}$ and a state change $\Delta\mathbf{u}$ as memory locations.

$$\mathbf{u} = \mathbf{u}^n, \tag{2.61}$$

$$\Delta\mathbf{u} = \Delta t\mathcal{L}(\mathbf{u}), \tag{2.62}$$

$$\mathbf{u} = \mathbf{u} + \frac{1}{3}\Delta\mathbf{u}, \tag{2.63}$$

$$\Delta\mathbf{u} = \Delta t\mathcal{L}(\mathbf{u}) - \frac{5}{9}\Delta\mathbf{u}, \tag{2.64}$$

$$\mathbf{u} = \mathbf{u} + \frac{15}{16}\mathbf{u}, \tag{2.65}$$

$$\Delta\mathbf{u} = \Delta t\mathcal{L}(\mathbf{u}) - \frac{153}{128}\Delta\mathbf{u}, \tag{2.66}$$

$$\mathbf{u}^{n+1} = \mathbf{u} + \frac{8}{15}\Delta\mathbf{u}. \tag{2.67}$$

We note that this algorithm is not SSP at any CFL coefficient.

Optimal Low Storage LSSSP33

Spiteri and Ruuth [40] document a series of s-stage, p-order time integrators that require only 2 memory locations per unknown. The 3-stage third-order

i	A_i	B_i
1	0.	0.92457411523577
2	-2.91549398859489	0.28771294148749
3	0.00000000151682	0.62653829645172

Table 2.1: Low storage coefficients for the third-order optimal SSP method of Spiteri and Ruuth.

scheme was implemented in ALEGRA/ECL and has an SSP CFL coefficient $c = 0.322$.

$$\begin{aligned}
\mathbf{u} &= \mathbf{u}^n, && (2.68)\\
\Delta\mathbf{u} &= \Delta t\mathcal{L}(\mathbf{u}), && (2.69)\\
\mathbf{u} &= \mathbf{u} + B_1\Delta\mathbf{u}, && (2.70)\\
\Delta\mathbf{u} &= \Delta t\mathcal{L}(\mathbf{u}) + A_2\Delta\mathbf{u}, && (2.71)\\
\mathbf{u} &= \mathbf{u} + B_2\Delta\mathbf{u}, && (2.72)\\
\Delta\mathbf{u} &= \Delta t\mathcal{L}(\mathbf{u}) + A_3\Delta\mathbf{u}, && (2.73)\\
\mathbf{u}^{n+1} &= \mathbf{u} + B_3\Delta\mathbf{u}. && (2.74)
\end{aligned}$$

See Table 2.1 for the required coefficients.

2.4 Software Framework Design

The solution methods described above require a software design that is flexible from both a physics and an algorithmic point of view, and that does not sacrifice computational speed. To achieve a reasonable level of computational performance, the data are organized to maximize the floating point operations per memory access, and so that potential memory conflicts and data dependencies are minimized.

In order to achieve this without sacrificing any flexibility in the physics, three major passes are made through the data in each coordinate direction for each Runge-Kutta stage. Data are laid out sequentially in memory according to a standard Fortran ordering, e.g. CV(N,N_x,N_y,N_z) so that the N conserved variables for each cell are local in memory. One can reason that each phase of the algorithm must access (essentially) all of the conserved quantities and if the data are laid out this way only one incoming cache line (instead of N) is required for each memory index.

During the calculation there are three primary computational units that are called at each RK time integration step.

Reconstruction: A reconstruction computation. This phase computes a high-order limited reconstruction of the conserved variables from only nearest neighbors and then computes left and right (LR) states as discussed in §2.1.

Flux Evaluation: This is a virtual function in the C++ language which depends on the particular equations being solved. Inputs are LR states and outputs are LR fluxes and LR minimum and maximum eigenvalues. Future extensions may support curvilinear coordinate systems. In this case the coordinate transformation will enter here as well.

Conservative Update: Inputs are a current state, LR states, LR fluxes, LR minimum and maximum eigenvalues and the state to which the updates will be added. After all directional passes the new state is obtained. Details on the update procedure are discussed in §2.2.

Remark. Future extensions to the conservation law framework require that the flux evaluation and update be modified in order to accommodate exact or approximate Riemann solves, e.g. numerical flux functions which use more detailed characteristic information. This has the added benefit of significantly reducing the memory requirements outlined below.

2.4.1 Memory Requirements

The memory costs for the first implementation of the conservation law framework, presented in terms of words per unit computational cell, are summarized in Table 2.2 for the first and second-order methods, and in Table 2.3 for the third-order methods. Here, N is the number of conserved variables stored per grid cell. As a basis for comparison between the second and third-order methods, Table 2.4 shows the memory cost for a standard Runge-Kutta time integrator for the Euler equations, which require storage for mass density, a momentum vector, and energy.

We note here that the third-order methods in ALEGRA/ECL use a dimension-by-dimension update that does not recover strict third-order accuracy in a genuinely multidimensional sense, which is reflected in the memory estimates. However, third-order accuracy is achieved in a one-dimensional sense for each coordinate direction. Genuinely multidimensional third-order schemes have been developed by Kurganov and Petrova [20] and appear to give improved results over the split or "dimension-by-dimension" schemes. However, such a genuinely multi-dimensional third-order capability has not yet been implemented in ALEGRA/ECL.

The memory requirements reported in Tables 2.2 – 2.4 neglect the memory overhead associated with using a multi-material software framework

(NEVADA), general equation-of-state coding, and parallel I/O staging. Regardless of this, the memory estimates are considered to be non-optimal since, by using judicious memory and algorithmic blocking, only the state memory need be permanent. The remainder can be allocated as essentially scratch on a per-block basis. This memory-use optimization has not yet been implemented in ALEGRA/ECL.

State 0	N
State 1	N
LR state	$2N$
LR Physical Flux	$2N$
LR λ_1	2
LR λ_N	2
Total	$6N + 4$

Table 2.2: Basic memory requirements in words per cell for the second-order methods.

State 1	N
State 2	N
State 3	N
LR state	$2N$
LR quadratic state	$2N$
LR Physical Flux	$2N$
LR λ_1	2
LR λ_N	2
Total	$9N + 4$

Table 2.3: Basic memory requirements in words per cell for the dimension-by-dimension third-order method.

	2D ($N = 4$)	3D ($N = 5$)
second-order	28	34
split third-order	40	49

Table 2.4: Basic memory requirements in words per cell for the Euler equations.

2.4.2 Application Developer Interface

In order to implement physics using the conservation law framework with the Godunov-type central schemes, the following data and computational methods must be provided by the application developer.

- A vector, $\mathbf{u}$, containing N conserved variables.
- The method for evaluating a physical flux vector, $\mathbf{F}_i(\mathbf{u})$.
- The method for computing (or providing) the minimum and maximum eigenvalues of $\frac{\partial \mathbf{F}_i(\mathbf{u})}{\partial \mathbf{u}}$ denoted by $\lambda_1(\frac{\partial \mathbf{F}_i}{\partial \mathbf{u}})$ and $\lambda_N(\frac{\partial \mathbf{F}_i}{\partial \mathbf{u}})$ respectively.
- An ability to initialize the conserved variables in some way. In the applications described here this is done by using the concepts of mass, momentum and energy which are currently fundamental concepts in the NEVADA framework.
- A list of output variables to be derived from the conserved variables and the coding to implement the output of these variables.

2.4.3 User Interface

The following input is required by the user of an application.

- A prescription of initial conditions and boundary conditions (Note: only periodic boundary conditions are supported by ALEGRA/ECL at this time).
- Selection of a compatible set of reconstruction, flux function evaluation and time integration options. In particular the order of time integration and reconstruction should be compatible, i.e., the order of the time integrator should be at least the order of the reconstruction.
- An optional Courant factor between 0 and 1 which will effectively reduce the time step. In general the computation will be run at exactly the maximal stability limit for the marching method chosen or a small fraction below it.

Chapter 3

Accuracy Assessment and Methods Comparison

This chapter presents an assessment of the accuracy of the conservation law solution methods presented in Chapter 2. In §3.1, a series of prototypical conservation law applications are outlined. In §3.2, the results of a preliminary accuracy assessment for a series of prototypical applications are presented, and in §3.3, a multi-methods comparison is discussed.

3.1 Prototypical Applications

In the conservation law framework, applications are constructed solely from the definition of conserved variables, the associated flux functions, and limited information about the eigenspectra of the flux Jacobian. In the initial efforts, three prototypical "physics" applications of increasing complexity were considered. The first is scalar (linear) advection, the second is the well-known inviscid Burgers' equation which has a simple quadratic flux function, and finally, the compressible Euler equations. For the Euler equations, two equation-of-state (EOS) implementations are considered. The first uses an ideal gas EOS that computes the pressure and sonic velocity using in-line coding. The second implements an arbitrary equation-of-state interface in which a pressure and sound speed are computed using a general EOS library that is accessed by the flux function.

In all the applications considered, the physical problem is cast in a conservation law form consistent with Eq. (2.1). However, the definition of the conserved variables and physical flux function changes according to the application.

3.1.1 Scalar Advection

The linear scalar advection equation is

$$\frac{\partial u}{\partial t} + a_x \frac{\partial u}{\partial x} + a_y \frac{\partial u}{\partial y} + a_z \frac{\partial u}{\partial z} = 0, \tag{3.1}$$

where u is the conserved variable, $\mathbf{a} = (a_x, a_y, a_z)$ is the constant advection velocity (and $\nabla \cdot \mathbf{a} = 0$). For this problem, the flux functions are $F_x(u) = a_x\, u$, $F_y(u) = a_y\, u$, and $F_z(u) = a_z\, u$.

3.1.2 Burgers' Equation

Burgers' equation,

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2/2)}{\partial x} + \frac{\partial (u^2/2)}{\partial y} + \frac{\partial (u^2/2)}{\partial z} = 0, \tag{3.2}$$

introduces a quadratic nonlinearity in the dependent variable, u. The conserved variable is u, and the associated flux functions are simply $F_x(u) = F_y(u) = F_z(u) = u^2/2$.

3.1.3 Euler Equations

The Euler equations describe the behavior of an inviscid compressible fluid. Here, they are written in terms of primitive variables in a conservation law form,

$$\frac{\partial}{\partial t}\left\{\begin{array}{c} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{array}\right\} + \frac{\partial}{\partial x}\left\{\begin{array}{c} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ u(E+p) \end{array}\right\} + \frac{\partial}{\partial y}\left\{\begin{array}{c} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v w \\ v(E+p) \end{array}\right\} + \frac{\partial}{\partial z}\left\{\begin{array}{c} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ w(E+p) \end{array}\right\} = 0, \tag{3.3}$$

where (u, v, w) is the velocity, p is the pressure, E is the total energy, and $\mathbf{u}^T = (\rho, \rho u, \rho v, \rho w, E)$ is the vector of conserved variables. The physical flux functions are

$$F_x = \left\{\begin{array}{c} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ u(E+p) \end{array}\right\}, \quad F_y = \left\{\begin{array}{c} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v w \\ v(E+p) \end{array}\right\}, \quad F_z = \left\{\begin{array}{c} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ w(E+p) \end{array}\right\}. \tag{3.4}$$

In order to solve the conservation law system, an equilibrium thermodynamic state, i.e., equation-of-state (EOS), is required so that the pressure and

sound speed can be computed in terms of the density and specific energy, $p = p(\rho, e)$. The specific internal energy is defined as $e = E/\rho - 1/2(u^2+v^2+w^2)$.

The eigenvalue triples for each Jacobian matrix, $\mathbf{A}_i = \partial \mathbf{F}_i(\mathbf{u})/\partial \mathbf{u}$, are $(u-c, u, u+c)$, $(v-c, v, v+c)$, and $(w-c, w, w+c)$ where $c^2 = \partial p/\partial \rho$ at constant entropy, i.e., c is the sound speed. It is assumed that the functions or tables representing p and c can be easily accessed by the flux and time step routines but are possibly very complicated and intricate routines as, for example, in the case of a general tabular EOS.

In the case of an in-line ideal-gas EOS, we have

$$p = (\gamma - 1)\left[E - \frac{\rho}{2}(u^2 + v^2 + w^2)\right], \tag{3.5}$$

and

$$c = \gamma p/\rho. \tag{3.6}$$

Here γ, the ratio of the constant-pressure and constant-volume specific heat, is a parameter that can be stored directly in the derived physics class and used in-line in a very efficient way. In contrast, the use of a generalized EOS interface induces the overhead of a function call, and possibly one or more table lookup operations. This is discussed further in §3.3.

> **Remark.** The reconstruction procedures used to calculate values of the density, momentum and total energy on cell faces do not necessarily combine to yield a physically reasonable positive internal energy state under all conditions. This appears to happen particularly in the case of very strong shocks. The negative internal energy case can be detected and the reconstruction of all variables in a given cell reverted to donor in order to add more dissipation when this occurs. The consequence is that the scheme is forced to first order in space wherever the internal energy becomes negative. This appears to be fairly effective but can still fail in some instances. In such a case it has been found advantageous to also reduce the time step. However, reducing the time step alone appears to be insufficient to work around the strong shock case. Robustness in the face of strong shocks appears to be an area for future research.

3.2 Accuracy Assessment

There are various metrics available that may be used to assess the accuracy of a solution method or, alternatively, to verify its correctness (see for example Oberkampf and Tarragon [33]). For our purposes, we are interested in extracting the leading order terms in the discretization error.

The discretization error is comprised of both spatial and temporal errors. It is defined as the difference between the computed and exact solutions,

$$\mathbf{e}_j = \overline{\mathbf{u}}_j^e - \overline{\mathbf{u}}_j, \tag{3.7}$$

where the exact grid function $\overline{\mathbf{u}}_j^e$ is defined in terms of the exact field solution $\mathbf{u}_j^e$ as

$$\overline{\mathbf{u}}_j^e = \frac{1}{\Delta x_j} \int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{u}_j^e(\xi) d\xi. \tag{3.8}$$

It can be shown that the discretization error is proportional to the truncation error which permits the discretization error to be written in a one-dimensional form as

$$\mathbf{e}_j = \alpha \Delta x^p + \beta \Delta t^q + H.O.T., \tag{3.9}$$

where p corresponds to the order of the spatial discretization, q to the order of the temporal discretization, and $H.O.T.$ indicates higher-order terms.

It is typical to use reconstruction methods that are matched with the time integrator, i.e., $p = q$. In addition, the time step and mesh size are related by the CFL number,

$$\Delta t = \frac{CFL\ \Delta x}{\lambda} \tag{3.10}$$

where λ is the magnitude of the largest eigenvalue of the flux Jacobian. This permits the discretization error to be written as

$$\mathbf{e}_j = \left(\alpha + \beta\ \frac{CFL}{\lambda}\right) \Delta x^p + H.O.T. \tag{3.11}$$

In terms of a global error metric,

$$\|\mathbf{e}\| = \overline{\alpha} \Delta x^p + H.O.T. \tag{3.12}$$

For all of the computations presented below, errors are measured in terms of a discrete L^1 norm,

$$\|\mathbf{e}\| = \sum_j |\overline{\mathbf{u}}_j^e - \overline{\mathbf{u}}_j| \Delta x_j. \tag{3.13}$$

or in terms of a discrete L^∞ norm,

$$\|\mathbf{e}\| = \max_j |\overline{\mathbf{u}}_j^e - \overline{\mathbf{u}}_j|. \tag{3.14}$$

Using a sequence of two grids, the global error metric may be used to extract the order of accuracy p, e.g.,

$$\frac{\|\mathbf{e}_1\|}{\|\mathbf{e}_2\|} = \left(\frac{\Delta x_1}{\Delta x_2}\right)^p, \tag{3.15}$$

where the subscript refers to the grid level. For all calculations presented here, $\Delta x_1/\Delta x_2 = 2$.

For smooth solutions we expect to recover an order of accuracy that is consistent with the formal accuracy of the method, e.g., second-order or $O(\Delta x^2)$ for a formally second-order spatial discretization ($p = 2$). For solutions with non-smooth data, e.g., shocks, we expect to achieve first-order or $O(\Delta x)$ convergence rates measured in the L^1 norm. In addition to the order of accuracy, the magnitude of the measured error is also useful in a multi-methods comparison. In our work, we consider the computational time required to obtain a given discretization error as our overall performance metric. This is the most useful performance metric since the primary goal is to obtain the best accuracy for a given computational cost. See Wesenburg [42] for an example of this performance metric applied to magnetohydrodynamics.

In the ensuing discussion, we report on an accuracy assessment of the algorithms available in ALEGRA/ECL for a variety of one-dimensional problems. Unless specifically stated otherwise, we used the following "matched" combinations of reconstruction method and time integrator: EULER time integrator with DONOR reconstruction, SSPRK22 with MINMOD ($\omega = 1.0$), VANLEER (harmonic version), and SUPERBEE, and finally SSPRK33 with the LT3 and KP3 reconstruction methods. Additionally, in order to avoid large overshoots near discontinuities (which may lead to non-physical values) in the LT3 and KP3 reconstructions, we used the restriction suggested by Kurganov and Petrova (see Eq. (4.4) in [20]) at extrema, with $\alpha = 10$. The reconstruction methods and their associated acronyms, e.g., DONOR, are outlined in §2.1, and the time integrators and their associated acronyms are detailed in §2.3. We use h and Δx interchangeably in the discussion of the one-dimensional results. All computational studies were run with $CFL = 0.475$.

3.2.1 Scalar Advection

The scalar advection equation, Eq. (3.1), is considered here with a variety of initial conditions – both smooth and non-smooth. For simplicity, a unit advective velocity is prescribed, $a_x = 1$. As a basis for comparison, the exact solution at a given time, $t = \tau$, is $u^0(x - a\tau)$, where $u^0 = u(x, 0)$ represents the initial conditions.

Scalar Advection: Periodic Sine Wave

The initial conditions for this problem consist of a sinusoidal variation in the dependent variable,

$$u(x, 0) = \sin(x) \quad 0 < x < 2\pi. \tag{3.16}$$

The calculation is carried out to a time of $t = 1.0$ and then compared to the exact solution projected onto the discrete grid, i.e., the exact grid function.

For this case, smooth initial conditions are prescribed and the convergence rate is extracted from a series of calculations on grids of increasing resolution using a fixed CFL number ($CFL = 0.475$) to specify the time step. We expect the extracted convergence rates to match the formal order of accuracy. For these calculations, the SSPRK33 time integrator was used in an attempt to minimize any effects of time-truncation error. In this case, the contribution of the temporal errors to the global discretization error is one order smaller than that of the spatial errors. Thus, the extracted convergence rates are indicative of the spatial convergence rates.

Table 3.1 shows the L^1 error and order of accuracy for constant, linear and quadratic reconstruction methods. In each case, the extracted convergence rates correspond to the formal order of accuracy, e.g., $O(\Delta x^3)$ for the quadratic methods. While all the linear methods appear to yield convergence rates consistent with their formal order of accuracy, the SUPERBEE limiter yields error levels that are nearly a factor of two larger than with the MINMOD or VANLEER limiters – the underlying cause for the larger error is not clear at this time.

	DONOR		MINMOD		VANLEER	
Cells	L^1	Order	L^1	Order	L^1	Order
512.	4.894E-02	–	3.327E-04	–	2.972E-04	–
1024.	2.451E-02	0.998	7.636E-05	2.123	7.146E-05	2.056
2048.	1.226E-02	0.999	1.762E-05	2.116	1.704E-05	2.068

	SUPERBEE		LT3		KP3	
Cells	L^1	Order	L^1	Order	L^1	Order
512.	5.953E-04	–	1.494E-06	–	1.494E-06	–
1024.	1.497E-04	1.992	1.868E-07	3.000	1.868E-07	3.000
2048.	3.755E-05	1.995	2.335E-08	3.000	2.335E-08	3.000

Table 3.1: Errors and convergence rates for the sine wave advection problem using various reconstruction methods and limiters with the third-order SSPRK33 time integrator.

In order to determine the affect of the higher-order time integrator on the extracted convergence rates, the same set of tests was repeated using a time integrator matched to the reconstruction method. That is EULER and first-order reconstruction, SSPRK22 and second-order reconstruction, etc. The results of this study are shown in Table 3.2. For the MINMOD limiter, there is a small, but detectable reduction in the observed rate of convergence

relative to the rates shown in Table 3.1. In addition, the measured error levels for the second-order methods are all comparable. The LT3 and KP3 methods yield identical results. This is a consequence of the fact that the quadratic reconstruction used by the two methods are identical, including the treatment of extrema. However, for this problem, limiting resulting in values of $\theta \neq 1$ does not occur; hence third-order convergence rates are achieved.

	DONOR		MINMOD		VANLEER	
Cells	L^1	Order	L^1	Order	L^1	Order
512.	2.579E-02	–	9.039E-04	–	3.867E-04	–
1024.	1.290E-02	0.999	2.476E-04	1.868	9.389E-05	2.042
2048.	6.446E-03	1.001	6.891E-05	1.845	2.285E-05	2.039

	SUPERBEE		LT3		KP3	
Cells	L^1	Order	L^1	Order	L^1	Order
512.	6.081E-04	–	1.298E-06	–	1.298E-06	–
1024.	1.514E-04	2.005	1.622E-07	3.000	1.622E-07	3.000
2048.	3.839E-05	1.980	2.028E-08	3.000	2.028E-08	3.000

Table 3.2: Errors and convergence rates for the sine wave advection problem using matched reconstruction and time integrator methods.

Scalar Advection: Periodic Square Wave

In order to assess the convergence rates of these methods on non-smooth initial data, we consider a square wave as the initial condition,

$$u(x,0) = \begin{cases} u(x,0) = 0.5 & 0 < x < 0.5, \\ u(x,0) = -0.5 & 0.5 < x < 1. \end{cases} \tag{3.17}$$

For this test case, all computations are carried out to $t = 1$ time units for comparison with the analytic solution. Due to the discontinuous initial data, we do not expect to achieve convergence rates greater than one with any of the methods in ALEGRA/ECL.

Figure 3.1 shows snapshots of the solutions obtained using 64 cells at $t = 1$. It is clear that the DONOR method introduces significant artificial diffusion as evidenced by the severe smearing of the square wave. The MINMOD and VANLEER limiters also introduce significant smearing relative to the SUPERBEE limiter. Surprisingly, the SUPERBEE result is sharper than both quadratic reconstruction methods, LT3 and KP3.

Table 3.3 shows the L^1 errors and the observed convergence rates for the various reconstruction methods. The DONOR method achieves a convergence rate of $O(\Delta x^{1/2})$, while SUPERBEE achieves a first-order rate $O(\Delta x)$.

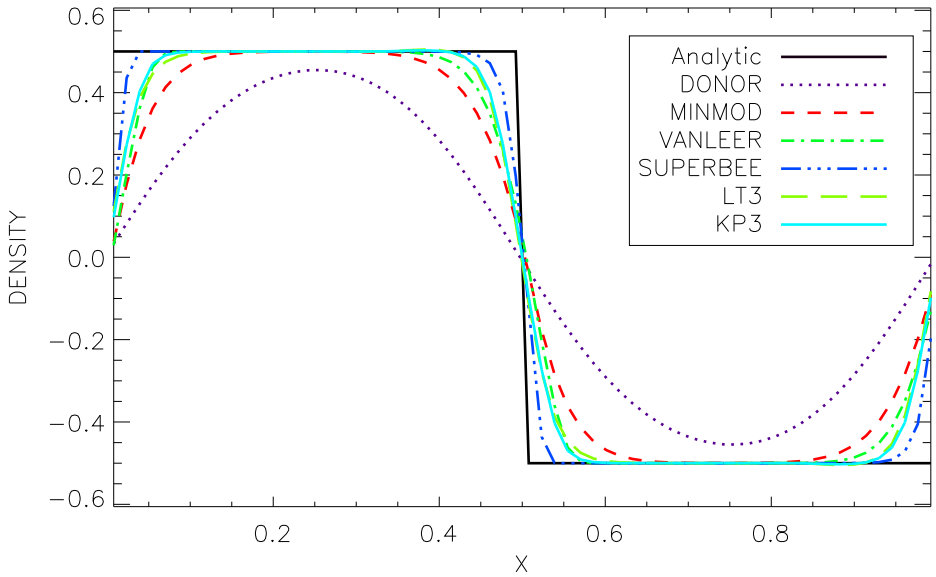


Figure 3.1: Solutions for the scalar square wave advection problem at $t = 1.0$ for various reconstruction methods.

All the other linear and quadratic methods achieve sub-linear convergence rates. The $O(\Delta x^{1/2})$ convergence rate observed for the DONOR method may be explained by considering the convergence rate for a modified advection equation that includes a constant artificial viscosity. In this case, Leveque [24] (see pg. 121) has shown that that the convergence rate will be proportional to $\sqrt{\Delta x t}$. As will be seen in subsequent calculations, this is in contrast to non-linear problems where $O(\Delta x)$ errors appear to be confined to the shock front by converging characteristics.

The results show that for linear problems with non-smooth initial data, the numerical dissipation can significantly degrade the solution accuracy over time. At this time, the sub-linear convergence rates for the MINMOD and VANLEER limiters, and the quadratic reconstruction methods, have not been resolved. The SUPERBEE limiter does not exhibit the reduced convergence associated with the other limiters. Further studies are warranted to understand the reasons for the reduced convergence rates with these methods.

Time Discretization Errors

The results presented in Table 3.3 are based on discrete solutions at a fixed point in time, i.e., at $t = 1$. If the temporal errors are significant relative to the spatial errors, i.e., if $\beta\ CFL/c \gg \alpha$ in Eq. (3.11), an evaluation of

	DONOR		MINMOD		VANLEER	
Cells	L^1	Order	L^1	Order	L^1	Order
64.	1.992E-01	–	8.426E-02	–	5.933E-02	–
128.	1.410E-01	0.499	5.369E-02	0.650	3.638E-02	0.706
256.	9.970E-02	0.500	3.412E-02	0.654	2.235E-02	0.703
512.	7.051E-02	0.500	2.165E-02	0.657	1.377E-02	0.698

	SUPERBEE		LT3		KP3	
Cells	L^1	Order	L^1	Order	L^1	Order
64.	2.744E-02	–	5.216E-02	–	4.856E-02	–
128.	1.378E-02	0.994	3.169E-02	0.719	2.913E-02	0.737
256.	6.895E-03	0.999	1.919E-02	0.724	1.732E-02	0.750
512.	3.449E-03	0.999	1.155E-02	0.733	1.028E-02	0.752

Table 3.3: Errors and convergence rates for the square wave advection problem using various reconstruction methods.

the discretization error at a single point in time may not give an accurate representation of a method's overall behavior.

In order to understand the effect of time-integration errors on the discretization error and observed convergence rates, we repeated the square wave advection problem, but extended the simulation time to $t = 1000.0$ using a grid with 64 cells. These calculations required 27000 time steps with $CFL = 0.475$. The L^1 error was calculated at intervals of $\Delta t = 0.05$ time units. Five tests cases were run. The first three each used the KT flux and VANLEER reconstruction while the time integrator was varied. The last two tests used the EULER time integrator with DONOR reconstruction and SSPRK22 time integration with SUPERBEE reconstruction. The Results are plotted in Figure 3.2.

The SSPRK22/VANLEER and SSPRK33/VANLEER yield discretization errors that grow as $\sqrt[3]{t}$ for $0 \leq t \leq 200$. Similarly, the EULER/DONOR scheme yields discretization error that grows as $\sqrt{t}$ for $0 \leq t \leq 10$. However, for all three of these methods, the error asymptotes to a value of $1/2$ – corresponding to the square wave being completely dissipated. Similar behavior is expected for any of the second- or third-order Runge-Kutta methods. The similarity in the temporal variation of the discretization error for the second-order and third-order time integrators (using VANLEER reconstruction) suggests that the observed error is principally spatial and accumulates in time. In addition, the observed dissipative nature of these schemes is clearly shown by the fact that the square wave is eventually damped completely. This makes them ineffective for simulations over long time periods.

Of course, the rate of damping is significantly larger for the EULER/DONOR method than the second and third-order methods, i.e., it takes nearly a factor of 10 times as long to damp the signal out.

The EULER/VANLEER and SSPRK22/SUPERBEE combinations yield discretization errors that, although oscillatory, do not grow in time. We note in passing that, if $CFL = 1.0$, the errors for the EULER integrator are identically zero at all times, i.e., the exact solution is recovered at each time step. It would appear that both methods are nearly neutrally dissipative in time. In the case of the EULER/VANLEER method, the artificial viscosity introduced by the reconstruction method appears to just balance the under-diffusive nature of the EULER time integrator. A similar, but less well understood behavior seems to occur with the SSPRK22 time integrator and the SUPERBEE reconstruction method.

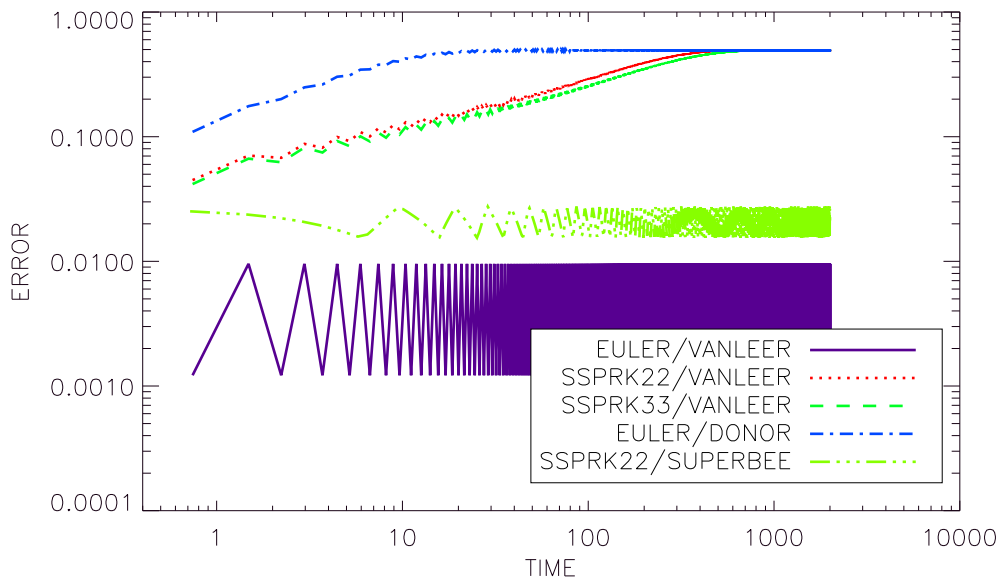
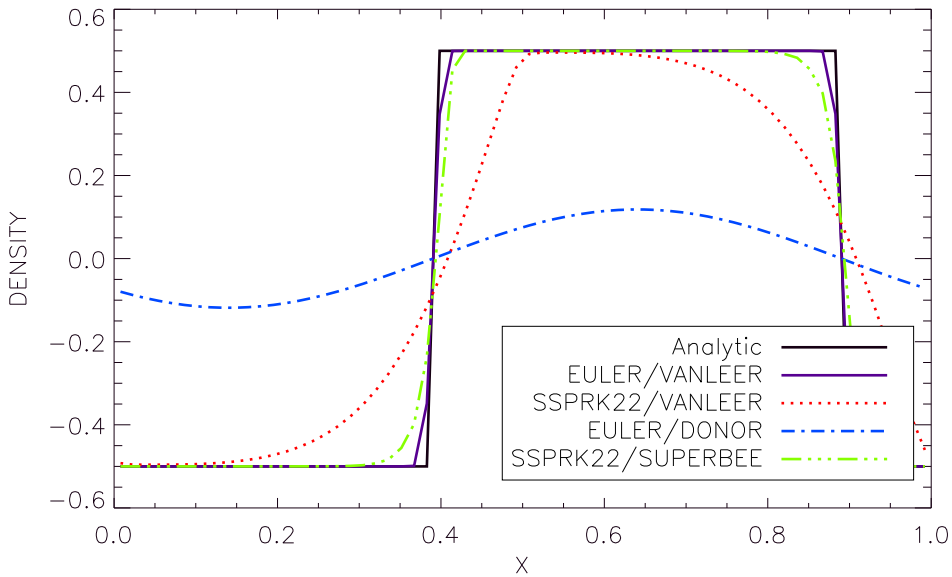


Figure 3.2: Temporal evolution of errors for the square wave advection test for various reconstruction and time integration methods.

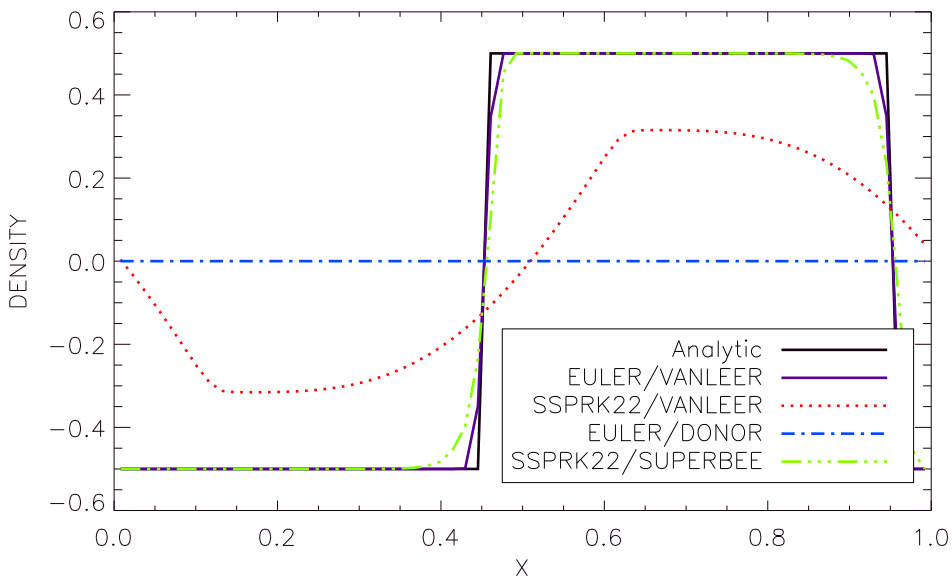
With the time-dependent nature of the discretization error out of the way, attention is turned to the following question. Will the observed rate of convergence vary with time? In order to begin to understand the answers to this question, we show that the extracted rates of convergence do not vary with time – as long as the signal remains finite, and the error can be accurately computed. Figure 3.3 shows snapshots of the square wave at $t \approx 10$ and $t \approx 100$ for the EULER/DONOR, EULER/VANLEER, SSPRK22/VANLEER and SSPRK22/SUPERBEE methods. The excessively dissipative nature of

the EULER/DONOR method is clearly shown with the signal completely damped at $t \approx 100$. The SSPRK22/VANLEER method exhibits both dispersive (phase) and dissipative errors.

With the exception of the EULER/VANLEER and SSPRK22/SUPERBEE schemes, it is clear that the measured discretization errors will increase with time. However, as long as the signal remains finite at a level where the discretization errors may be accurately measured, the observed convergence rates remain unchanged. This may be seen in Tables 3.4 – 3.5 which show the discretization error measured in the L^1 norm and convergence rates measured at $t = 10$ and $t = 100$ respectively. In comparison to the convergence rates measured at $t = 1$ and reported in Table 3.3, the observed convergence rates at the later times appear to be relatively unchanged. This reflects the fact that, as long as the signal remains finite, the discretization error may accumulate with time, but the observed convergence rates remain essentially unchanged. We note that the test case involving EULER/DONOR is already severely dissipated at $t = 10$, and is completely dissipated on the coarser grids by $t = 100$, resulting in essentially a constant error in Table 3.5.



(a) Solutions at $t \approx 10$.



(b) Solution errors at $t \approx 100$.

Figure 3.3: Snapshots of solutions to the square wave advection test for various reconstruction and time integration methods.

	DONOR		MINMOD		VANLEER	
Cells	L^1	Order	L^1	Order	L^1	Order
64.	4.815E-01	–	1.101E-01	–	1.179E-01	–
128.	4.133E-01	0.220	6.801E-02	0.694	7.299E-02	0.692
256.	3.125E-01	0.403	4.242E-02	0.681	4.533E-02	0.687
512.	2.230E-01	0.487	2.662E-02	0.672	2.823E-02	0.683

	SUPERBEE		LT3		KP3	
Cells	L^1	Order	L^1	Order	L^1	Order
64.	2.757E-02	–	9.770E-02	–	8.671E-02	–
128.	1.379E-02	0.999	5.870E-02	0.735	5.185E-02	0.742
256.	6.890E-03	1.002	3.672E-02	0.677	3.131E-02	0.728
512.	3.448E-03	0.999	2.322E-02	0.661	1.904E-02	0.718

Table 3.4: Errors and convergence rates for the square wave advection problem at $t = 10$.

	DONOR		MINMOD		VANLEER	
Cells	L^1	Order	L^1	Order	L^1	Order
64.	5.000E-01	–	2.502E-01	–	2.606E-01	–
128.	5.000E-01	0.000	1.447E-01	0.790	1.515E-01	0.782
256.	4.998E-01	0.001	9.119E-02	0.666	9.485E-02	0.676
512.	4.914E-01	0.024	5.758E-02	0.663	5.946E-02	0.674

	SUPERBEE		LT3		KP3	
Cells	L^1	Order	L^1	Order	L^1	Order
64.	2.760E-02	–	1.714E-01	–	1.629E-01	–
128.	1.379E-02	1.001	1.075E-01	0.673	9.920E-02	0.716
256.	6.897E-03	0.999	7.491E-02	0.521	6.096E-02	0.703
512.	3.445E-03	1.002	4.720E-02	0.666	3.763E-02	0.696

Table 3.5: Errors and convergence rates for the square wave advection problem at $t = 100$.

3.2.2 Burgers' Equation

Burgers' equation, Eq. (3.2), retains the simplicity of a scalar conservation law while introducing the effects of non-linearity, and provides a valuable test vehicle for assessing the conservation law framework as implemented in ALEGRA/ECL. In order to assess the accuracy of the constant, linear, and quadratic reconstruction methods and the time integration methods in a nonlinear setting, we examined the error in a periodic solution of Burgers' equation at various levels of mesh refinement.

Burgers' Equation: Pre-shock Periodic Sine Wave

The initial conditions used for the pre-shock calculations consist of a sinusoidal perturbation about a mean value,

$$u(x,0) = 0.5 + \sin(x) \quad 0 < x < 2\pi. \tag{3.18}$$

All calculations were performed using periodic boundary conditions. Initial conditions were specified so that the exact averages of the initial sinusoidal profile were preserved on the grid.

The solution for this problem develops a shock profile at time $t = 1$. The first series of calculations was terminated at time $t = 0.5$ to ensure that the solution remains smooth. This problem is identical to the first numerical example presented by Kurganov, Noelle and Petrova [18]. The convergence rates are calculated using the L^1 error norm as defined in Eq. (3.13). The discretization error and extracted convergence rates are reported in Table 3.6, and shown graphically in Figure 3.4(a). These results indicate that the first and second-order methods yield convergence rates consistent with the formal order of accuracy for each method. In contrast, the third-order LT3 algorithm appears to be just approaching an asymptotic convergence rate on the fine grid (1280 cells), while the KP3 method shows a decreasing rate with increasing resolution. This is clearly shown by the knee at $1/h = 160$ as shown in Figure 3.4(a).

In order to better understand this behavior, a series of studies at higher resolution with $40 \leq 1/\Delta x \leq 20480$ were undertaken. In order to eliminate the possibility that errors were introduced through the initial conditions, the rate of convergence of the initial data with respect to mesh size was measured. Here, the error between the cell-averages and the analytic initial conditions are computed. Fourth order convergence rates are expected and observed because the computed analytic cell averages are based on two-point Gaussian quadrature. The results of these computations are shown in Table 3.7. The expected convergence rate is observed in the initial conditions up to $\Delta x = 1/2560$. Beyond this mesh resolution, the computed error is comparable to the machine precision and the convergence rate is no longer accurately

	DONOR		MINMOD		VANLEER	
Cells	L^1	Order	L^1	Order	L^1	Order
40.	1.103E-01	–	2.965E-02	–	1.676E-02	–
80.	5.850E-02	0.915	8.068E-03	1.878	4.266E-03	1.974
160.	3.032E-02	0.948	2.146E-03	1.911	1.033E-03	2.047
320.	1.554E-02	0.965	5.740E-04	1.902	2.560E-04	2.012
640.	7.849E-03	0.985	1.523E-04	1.914	6.225E-05	2.040
1280.	3.952E-03	0.990	4.066E-05	1.905	1.518E-05	2.036

	SUPERBEE		LT3		KP3	
Cells	L^1	Order	L^1	Order	L^1	Order
40.	2.318E-02	–	1.535E-03	–	1.702E-03	–
80.	6.269E-03	1.886	2.251E-04	2.769	2.105E-04	3.015
160.	1.636E-03	1.938	3.097E-05	2.862	2.641E-05	2.995
320.	4.227E-04	1.952	4.205E-06	2.880	7.972E-06	1.728
640.	1.064E-04	1.990	5.594E-07	2.910	3.012E-06	1.404
1280.	2.691E-05	1.984	7.641E-08	2.872	9.306E-07	1.695

Table 3.6: Errors and estimated order of convergence for solutions to Burgers' equation at $t = 0.5$ (pre-shock).

estimated. Thus, it is clear that errors are not being introduced by the initial data.

Subsequently, the computational experiments for the pre-shock case were repeated using the higher-resolution grids. The results of the study are presented in terms of the L^1 and L^∞ norms in Tables 3.8 and 3.9 respectively. In the L^1 norm, both methods exhibit erratic convergence rates that, in general, appear to be decreasing with increasing mesh resolution. In comparison, in the L^∞ norm, LT3 exhibits a second-order rate and KP3 degrades to first-order. These tests were repeated without the modification at extrema used in [20], with similar results.

This unexpected behavior prompted further analysis of the LT3 and KP3 methods as presented by Liu and Tadmor [31] and Kurganov and Petrova [20]. After careful analysis, we have concluded that these reconstructions are, in fact, not provably third order on smooth solutions. A complete discussion of the analysis may be found in §3.4.

Burgers' Equation: Post-shock Periodic Sine Wave

Additional tests were performed with termination times $t > 1$, so that the solution would involve a shock, i.e., non-smooth data. Snapshots of the

	LT3		KP3	
Cells	L^1	Order	L^1	Order
40.	5.642E-07	–	5.642E-07	–
80.	3.524E-08	4.001	3.524E-08	4.001
160.	2.202E-09	4.000	2.202E-09	4.000
320.	1.376E-10	4.000	1.376E-10	4.000
640.	8.602E-12	4.000	8.602E-12	4.000
1280.	5.376E-13	4.000	5.376E-13	4.000
2560.	3.364E-14	3.998	3.364E-14	3.998
5120.	2.337E-15	3.847	2.337E-15	3.847
10240.	5.927E-16	1.979	5.927E-16	1.979
20480.	5.762E-16	0.041	5.762E-16	0.041

Table 3.7: Errors and estimated order of convergence for solutions to Burgers' equation at $t = 0$.

solution at $t = 1.5$ for a grid containing 40 cells are shown in Figure 3.6(a), and the computed error distribution for each solution method is shown in Figure 3.6(b).

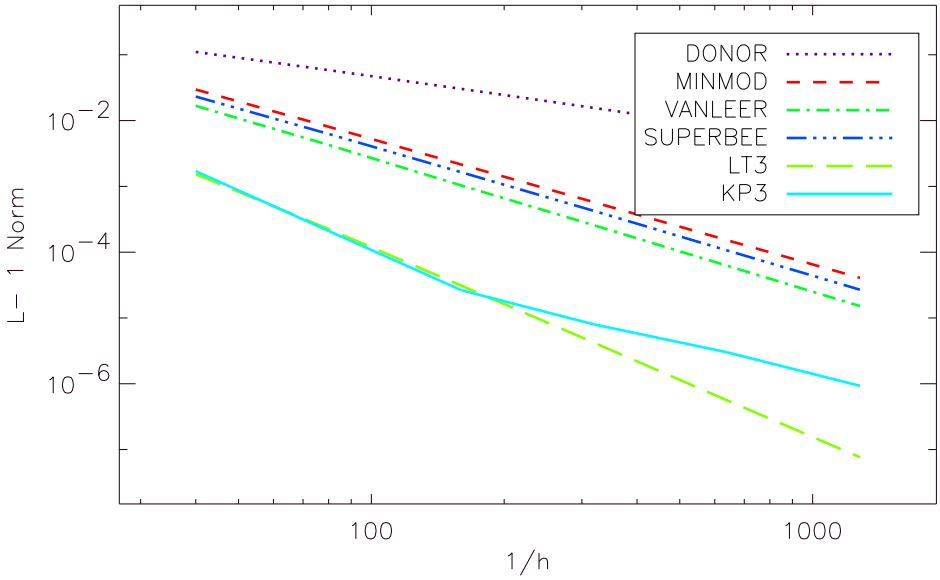
A comparison of convergence before and after the shock is shown in Figure 3.4, and similar plots for two later times are given in Figure 3.5. The same results are tabulated in Tables 3.10 - 3.12. The post-shock convergence rates are highly erratic, although the overall trend is clearly convergent. The reason for the large scatter is unknown.

	LT3		KP3	
Cells	L^1	Order	L^1	Order
40.	1.535E-03	–	1.702E-03	–
80.	2.251E-04	2.769	2.105E-04	3.015
160.	3.097E-05	2.862	2.641E-05	2.995
320.	4.205E-06	2.880	7.972E-06	1.728
640.	5.594E-07	2.910	3.012E-06	1.404
1280.	7.641E-08	2.872	9.306E-07	1.695
2560.	1.157E-08	2.724	2.587E-07	1.847
5120.	2.585E-09	2.162	6.861E-08	1.915
10240.	4.826E-10	2.421	1.907E-08	1.847
20480.	1.232E-10	1.970	5.275E-09	1.854

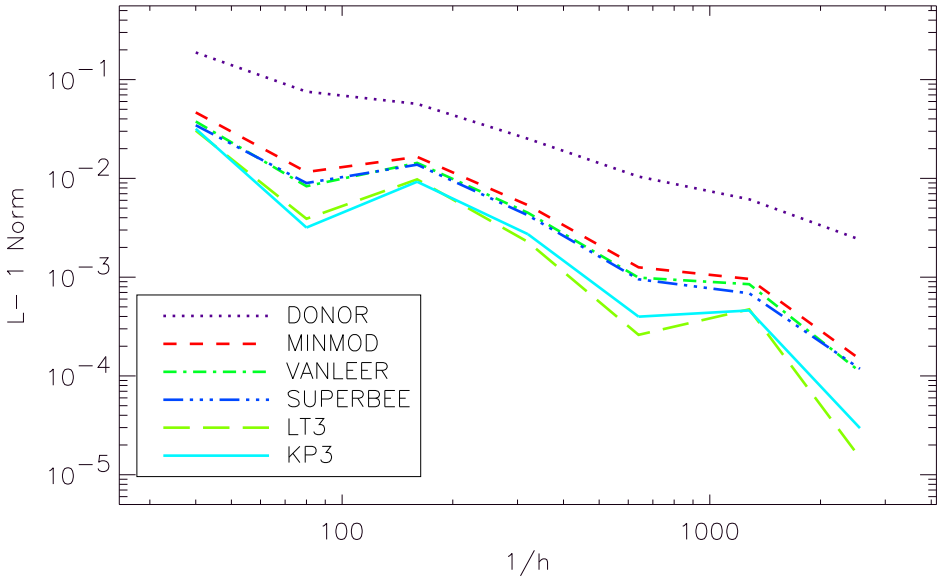
Table 3.8: Errors and estimated order of convergence for solutions to Burgers' equation at $t = 0.5$ (pre-shock).

	LT3		KP3	
Cells	L^∞	Order	L^∞	Order
40.	1.673E-03	–	1.880E-03	–
80.	3.586E-04	2.222	3.178E-04	2.565
160.	7.158E-05	2.325	8.884E-05	1.839
320.	1.608E-05	2.154	4.552E-05	0.965
640.	3.480E-06	2.208	3.056E-05	0.575
1280.	7.392E-07	2.235	1.402E-05	1.125
2560.	1.577E-07	2.229	6.578E-06	1.091
5120.	3.342E-08	2.238	3.170E-06	1.053
10240.	7.218E-09	2.211	1.554E-06	1.029
20480.	1.527E-09	2.241	7.314E-07	1.087

Table 3.9: L-infinity Errors and estimated order of convergence for solutions to Burgers' equation at $t = 0.5$ (pre-shock).

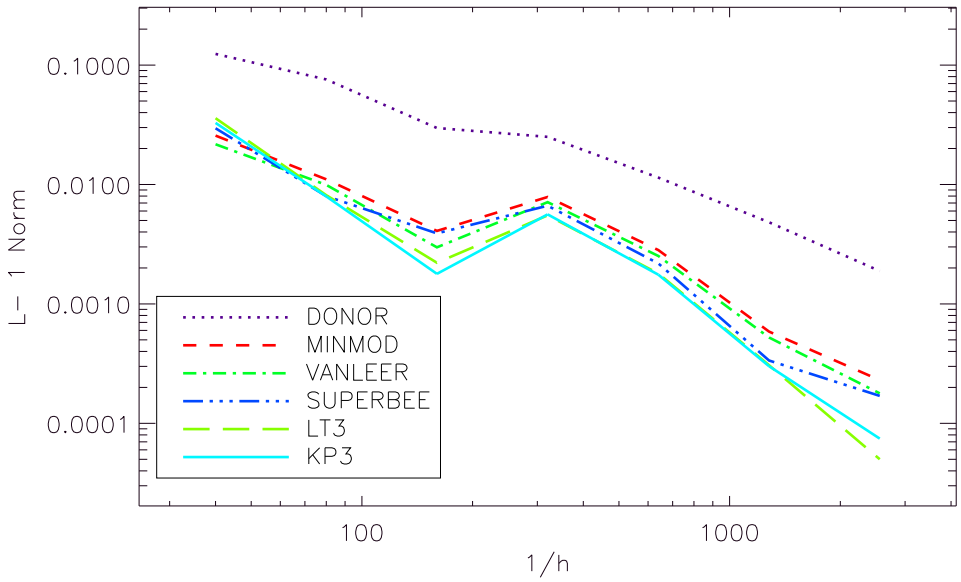


(a) Convergence for the smooth solution at $t = 0.5$.

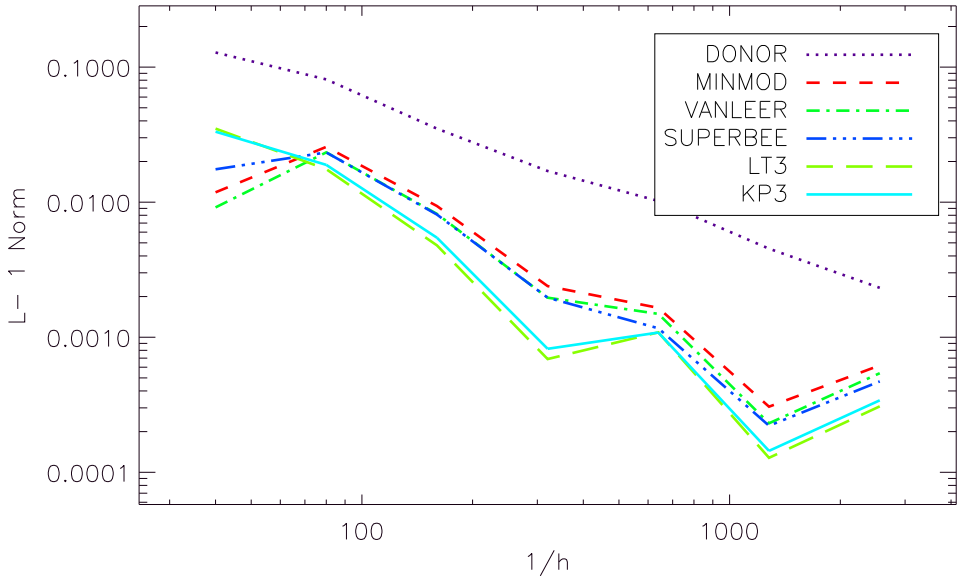


(b) Convergence for the shocked solution at $t = 1.5$.

Figure 3.4: Comparison of solution convergence for the Burgers' equation sine wave problem before and after the solution develops a shock.

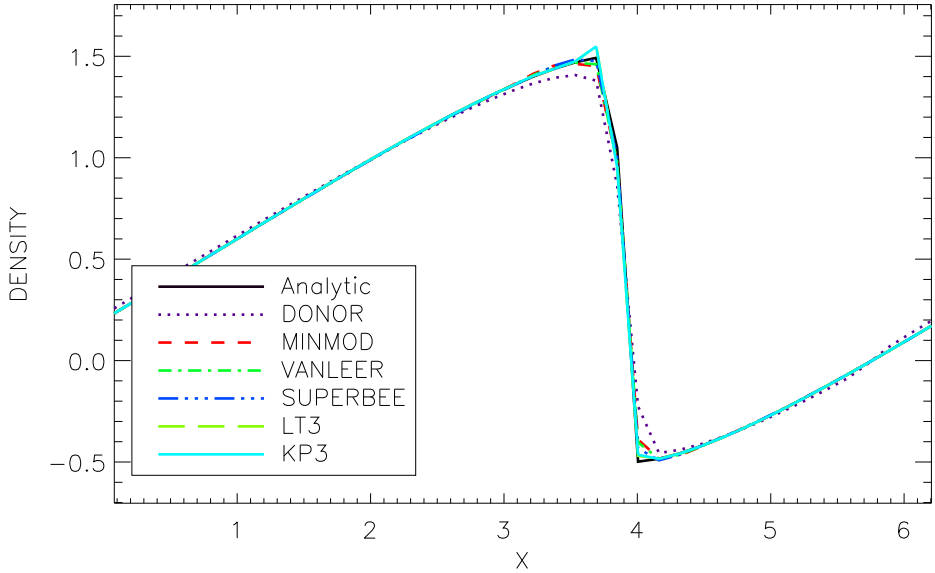


(a) Convergence at $t = 2$.

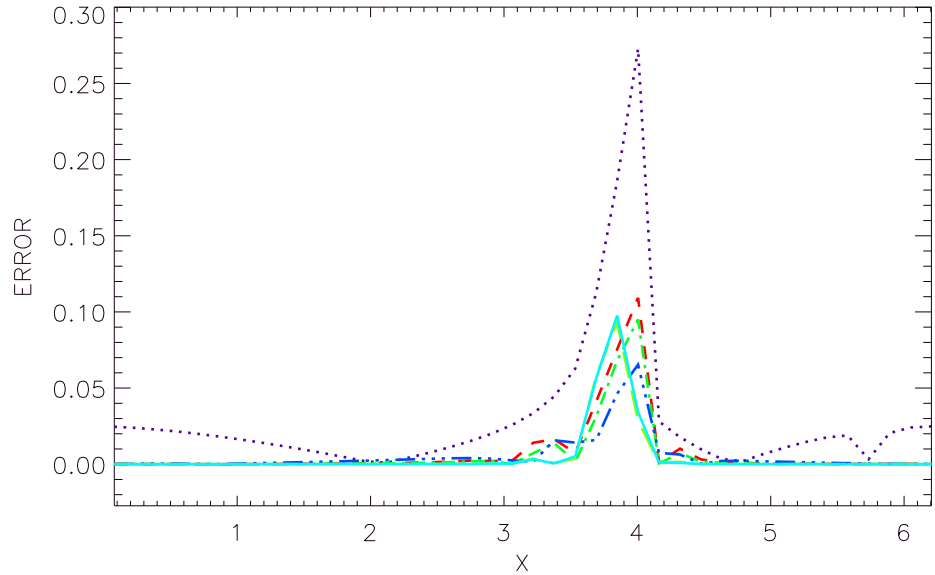


(b) Convergence at $t = 3$.

Figure 3.5: Convergence rates for Burgers' equation with sinusoidal initial conditions at two post-shock times.



(a) Solutions



(b) Errors

Figure 3.6: Snapshots of errors and solutions to Burgers' equation with sinusoidal initial conditions using various reconstruction methods with $\Delta x = 1/40$ at $t = 1.5$.

	DONOR		MINMOD		VANLEER	
Cells	L^1	Order	L^1	Order	L^1	Order
40.	1.879E-01	–	4.656E-02	–	3.768E-02	–
80.	7.553E-02	1.315	1.160E-02	2.005	8.333E-03	2.177
160.	5.680E-02	0.411	1.652E-02	-0.510	1.435E-02	-0.784
320.	2.520E-02	1.172	5.362E-03	1.623	4.491E-03	1.676
640.	1.043E-02	1.273	1.257E-03	2.093	9.889E-04	2.183
1280.	6.137E-03	0.765	9.589E-04	0.391	8.486E-04	0.221
2560.	2.401E-03	1.354	1.478E-04	2.698	1.106E-04	2.940

	SUPERBEE		LT3		KP3	
Cells	L^1	Order	L^1	Order	L^1	Order
40.	3.450E-02	–	3.044E-02	–	3.175E-02	–
80.	8.963E-03	1.944	3.903E-03	2.963	3.176E-03	3.322
160.	1.377E-02	-0.620	9.777E-03	-1.325	9.250E-03	-1.542
320.	4.235E-03	1.701	2.276E-03	2.103	2.727E-03	1.762
640.	9.518E-04	2.154	2.605E-04	3.127	3.981E-04	2.776
1280.	6.860E-04	0.472	4.730E-04	-0.860	4.604E-04	-0.210
2560.	1.181E-04	2.538	1.483E-05	4.995	2.966E-05	3.956

Table 3.10: Errors and estimated order of convergence for solutions to Burgers' equation at $t = 1.5$.

	DONOR		MINMOD		VANLEER	
Cells	L^1	Order	L^1	Order	L^1	Order
40.	1.239E-01	–	2.566E-02	–	2.172E-02	–
80.	7.581E-02	0.709	1.104E-02	1.216	9.921E-03	1.131
160.	2.972E-02	1.351	4.084E-03	1.435	2.977E-03	1.736
320.	2.506E-02	0.246	7.857E-03	-0.944	7.118E-03	-1.257
640.	1.141E-02	1.135	2.821E-03	1.478	2.520E-03	1.498
1280.	4.826E-03	1.242	5.861E-04	2.267	5.235E-04	2.267
2560.	1.886E-03	1.356	2.301E-04	1.349	1.782E-04	1.555

	SUPERBEE		LT3		KP3	
Cells	L^1	Order	L^1	Order	L^1	Order
40.	2.956E-02	–	3.591E-02	–	3.277E-02	–
80.	7.994E-03	1.886	8.134E-03	2.142	7.947E-03	2.044
160.	3.895E-03	1.037	2.213E-03	1.878	1.786E-03	2.154
320.	6.644E-03	-0.770	5.513E-03	-1.317	5.619E-03	-1.653
640.	2.180E-03	1.607	1.798E-03	1.616	1.755E-03	1.678
1280.	3.360E-04	2.698	3.070E-04	2.550	3.011E-04	2.544
2560.	1.697E-04	0.985	5.001E-05	2.618	7.447E-05	2.015

Table 3.11: Errors and estimated order of convergence for solutions to Burgers' equation at $t = 2$.

	DONOR		MINMOD		VANLEER	
Cells	L^1	Order	L^1	Order	L^1	Order
40.	1.283E-01	–	1.183E-02	–	9.146E-03	–
80.	8.111E-02	0.661	2.568E-02	-1.118	2.338E-02	-1.354
160.	3.505E-02	1.210	9.371E-03	1.454	8.193E-03	1.513
320.	1.700E-02	1.044	2.390E-03	1.971	1.963E-03	2.061
640.	1.021E-02	0.737	1.650E-03	0.535	1.489E-03	0.398
1280.	4.525E-03	1.173	3.060E-04	2.431	2.305E-04	2.692
2560.	2.322E-03	0.963	6.185E-04	-1.015	5.412E-04	-1.231

	SUPERBEE		LT3		KP3	
Cells	L^1	Order	L^1	Order	L^1	Order
40.	1.752E-02	–	3.504E-02	–	3.328E-02	–
80.	2.336E-02	-0.415	1.756E-02	0.997	1.889E-02	0.817
160.	8.098E-03	1.529	4.787E-03	1.875	5.458E-03	1.791
320.	1.963E-03	2.044	6.891E-04	2.796	8.208E-04	2.733
640.	1.164E-03	0.754	1.087E-03	-0.658	1.083E-03	-0.400
1280.	2.212E-04	2.396	1.279E-04	3.088	1.440E-04	2.911
2560.	4.705E-04	-1.089	3.079E-04	-1.267	3.411E-04	-1.244

Table 3.12: Errors and estimated order of convergence for solutions to Burgers' equation at $t = 3$.

3.2.3 Euler Equations

In this section, we evaluate the accuracy of the ALEGRA/ECL algorithms using a system of conservation laws, i.e., the Euler equations, with an ideal gas equation of state. Three problems are considered, one with smooth initial conditions, one with non-smooth initial data, and a shock tube.

Euler Equations: Gaussian Pulse Advection

The Gaussian pulse advection problem is defined using a periodic domain with $0 \leq x \leq 1$. The initial conditions consist of a one-dimensional density pulse in the x-direction surrounded by a region of uniform pressure $p_0 = 10^{-6}$, density $\rho_0 = 10^{-3}$ and velocity $u_0 = 1$. The pulse is initially centered at $x = 0.25$, and the density distribution is computed using

$$\rho(x) = \rho_{pulse} e^{-(x/w)^2} + \rho_0[1 - e^{-(x/w)^2}] \tag{3.19}$$

where $w = 0.1$ is the characteristic half-width of the pulse.

For these initial conditions, the Euler equations reduce to a single wave equation with wave speed u_0. Thus, this problem tests the ability of the solution algorithm to preserve the primary features of the initial signal, e.g., half-width, peak amplitude and phase. Ideally, the solution algorithms should be capable of advecting a smooth signal and preserve the features of the initial Gaussian distribution with minimal errors.

Figure 3.7(a) shows snapshots of the density profile on a 512-cell grid ($\Delta x = 1/512$) at $t = 0.4$. Three solutions are plotted corresponding to the three different limiters used with second-order reconstruction: MINMOD, VANLEER and SUPERBEE. At this point in the calculation, the MINMOD limiter has reduced the peak amplitude by about 30%, VANLEER by nearly 15%, and SUPERBEE by about 5%. This suggests that VANLEER and MINMOD are much more dissipative than SUPERBEE. In contrast, the artificial steepening effects commonly associated with SUPERBEE are revealed by the narrowing at the base of the Gaussian pulse. While all three limiters yield formally second-order methods, there is a significant difference between the results for the SUPERBEE, VANLEER and MINMOD limiters on this grid.

While conducting these calculations, it was observed that the dissipative effects are heavily dependent on mesh resolution – a result that can be anticipated from the spatial form of the limiters. Table 3.2.3 shows the L^1 errors and approximate convergence rates for a series of calculations with grid resolution ranging from 64 cells to 4096 cells in the x-direction. On the coarse grids, i.e., less than 512 cells, the MINMOD and VANLEER limiters perform poorly relative to SUPERBEE as indicated by the convergence rates. A mesh with 512 cells corresponds to resolving the Gaussian half-width by

about sixteen cells, i.e., about sixteen cells per wavelength. For simulations that require the resolution of smooth features, the SUPERBEE limiter will provide better results when less than sixteen cells per wavelength are used.

	MINMOD		VANLEER		SUPERBEE	
Cells	L^1	Order	L^1	Order	L^1	Order
64.	3.282E-02	–	2.865E-02	–	2.289E-02	–
128.	2.535E-02	0.380	2.036E-02	0.502	1.243E-02	0.898
256.	1.622E-02	0.650	1.022E-02	1.003	4.924E-03	1.349
512.	7.720E-03	1.066	4.233E-03	1.266	4.586E-03	0.102
1024.	2.638E-03	1.545	1.489E-03	1.504	2.497E-03	0.875
2048.	9.078E-04	1.541	4.482E-04	1.734	1.174E-03	1.091
4096.	2.848E-04	1.673	1.262E-04	1.830	4.226E-04	1.474

Table 3.13: Errors and convergence rates for the Gaussian pulse advection problem using various reconstruction methods.

Euler Equations: Square Pulse Advection

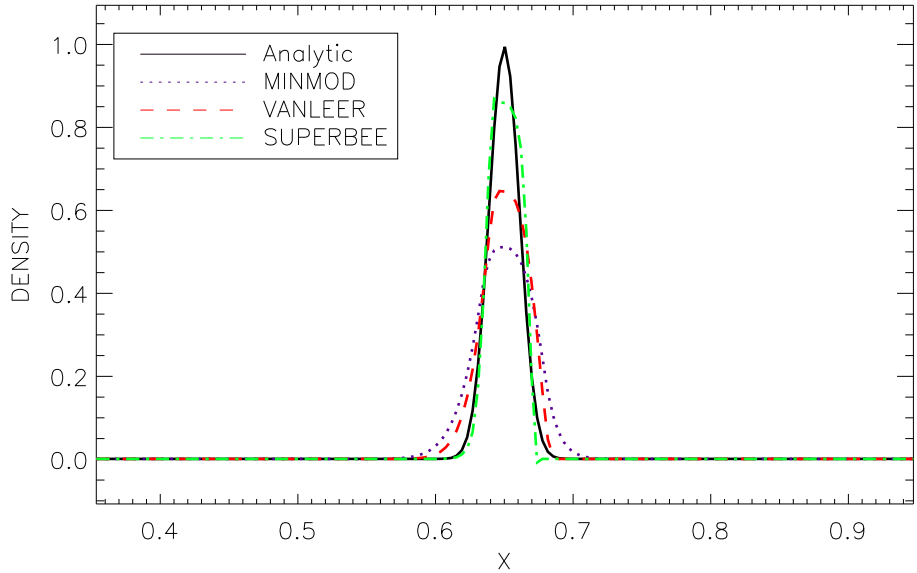
This problem is similar to the Gaussian pulse advection problem, but uses non-smooth initial density data. Like the Gaussian pulse problem, the computational domain is periodic with $0 \leq x \leq 1$.

The initial conditions for the square pulse advection problem are defined using a one-dimensional density pulse in the x-direction that is surrounded by a region of uniform density $\rho_0 = 10^{-3}$ and velocity $u_0 = 1$. The square pulse is initially centered at $x_0 = 0.25$, and has a width $w = 0.2$. The initial density distribution is defined as

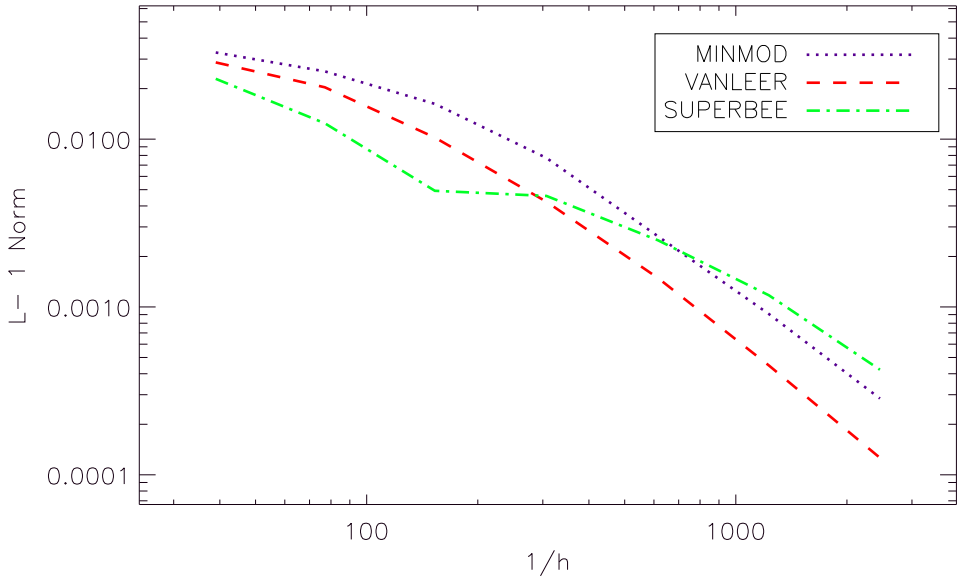
$$\rho(x) = \begin{cases} 1 & \text{if } |x - 0.25| \leq 0.1 \\ \rho_0 & \text{otherwise.} \end{cases} \tag{3.20}$$

The pressure is uniform throughout the domain, $p_0 = 10^{-6}$. For these initial conditions, the solution is simply a moving contact discontinuity with speed u_0. This problem tests the ability of the solution methods to preserve contact discontinuities, i.e., the ability to treat linearly degenerate eigenvalues.

Figure 3.8(a) shows a snapshot of the density at $t = 0.4$ for each of the second-order limiters in ALEGRA/ECL for a mesh with 128 cells. Figure 3.8(b) shows the L^1 error norm as a function of $1/h$ for each of the three limiters. The corresponding errors and convergence rates are shown in Table 3.14. As was observed for the linear advection problem with non-smooth initial conditions, SUPERBEE attempts to preserve the sharp features of the square pulse. The MINMOD and VANLEER limiters appear to smooth



(a) Solutions at $t = 1.0$.



(b) Global L^1 error vs. mesh refinement.

Figure 3.7: Comparison of density solutions for the Gaussian pulse advection problem using second-order reconstruction and various limiters.

the sharp corners both at the bottom and top of the square pulse. Again, only SUPERBEE achieves a linear convergence rate, $O(\Delta x)$, while the MINMOD and VANLEER limiters yield sub-linear rates.

	MINMOD		VANLEER		SUPERBEE	
Cells	L^1	Order	L^1	Order	L^1	Order
64.	5.945E-02	–	4.471E-02	–	2.441E-02	–
128.	3.699E-02	0.685	2.671E-02	0.743	1.156E-02	1.078
256.	2.440E-02	0.600	1.720E-02	0.635	6.348E-03	0.865
512.	1.519E-02	0.684	1.035E-02	0.733	2.754E-03	1.205

Table 3.14: Errors and convergence rates for the square pulse advection problem using various reconstruction methods.

Euler Equations: Sod Shock Tube

In order to compare the first, second and third-order methods in ALEGRA/ECL in a more realistic setting, we considered the well-known Sod shock tube problem. This problem exercises the in-line ideal gas Euler EOS in ALEGRA/ECL. The problem is one-dimensional, with initial conditions

$$\begin{aligned} \rho_l &= 1.0, \\ p_l &= 1.0, \end{aligned} \tag{3.21}$$

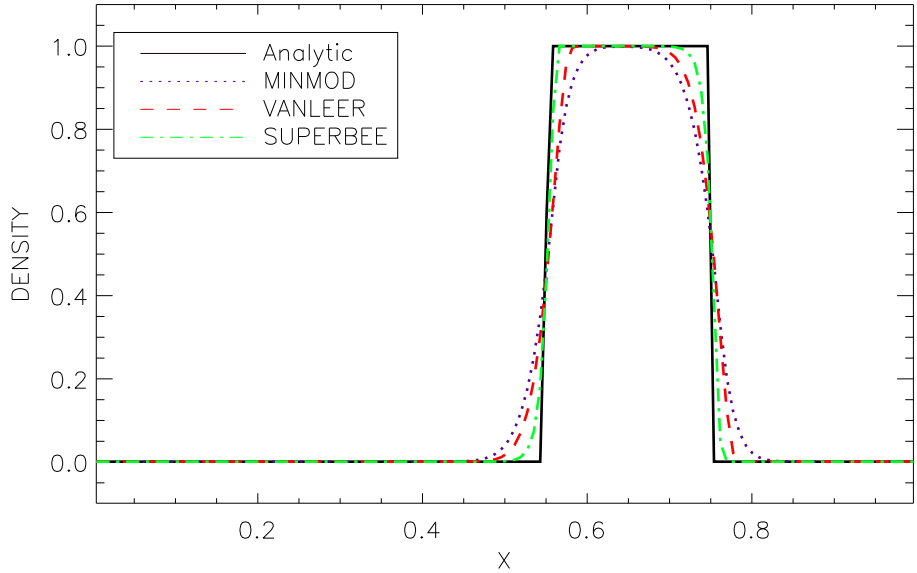
for the left-state (l), and

$$\begin{aligned} \rho_r &= 0.125, \\ p_r &= 0.1, \end{aligned} \tag{3.22}$$

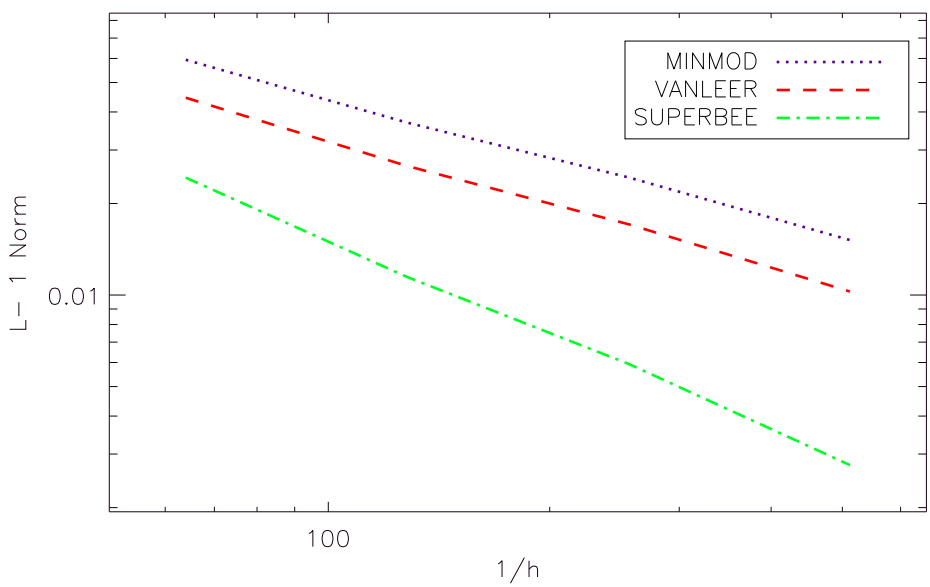
for the right state (r), with the interface located at $x = 1.0$. The computational domain is periodic with $0 \leq x \leq 2$. For the purposes of comparison, we consider the solution on the interval $0.5 < x < 1.5$ at time $t = 0.1644$ to avoid any possible side-effects from the periodic boundary conditions in the x-direction.

Snapshots of the density profile at $t = 0.1644$ are shown in Figure 3.9(a) along with the error distribution in Figure 3.9(b) for a grid with $\Delta x = 1/128$. The corresponding pressure distribution and error in the pressure are shown in Figure 3.10, and the velocity profile and error are shown in Figure 3.11. As expected, the DONOR reconstruction yields solutions that exhibit excessive smearing with the concomitant errors indicated in Figures 3.9 – 3.11.

The VANLEER reconstruction dramatically sharpens the contact discontinuity relative to the constant DONOR method and improves the rarefaction. Surprisingly, the VANLEER and LT3 results appear to be quite



(a) Density distributions at $t = 0.4$ with $\Delta x = 1/128$.



(b) Global L^1 error vs. mesh refinement.

Figure 3.8: Snapshots of the density solutions, and L^1 errors at $t = 0.4$ for the square pulse advection problem using second-order reconstruction with various limiters.

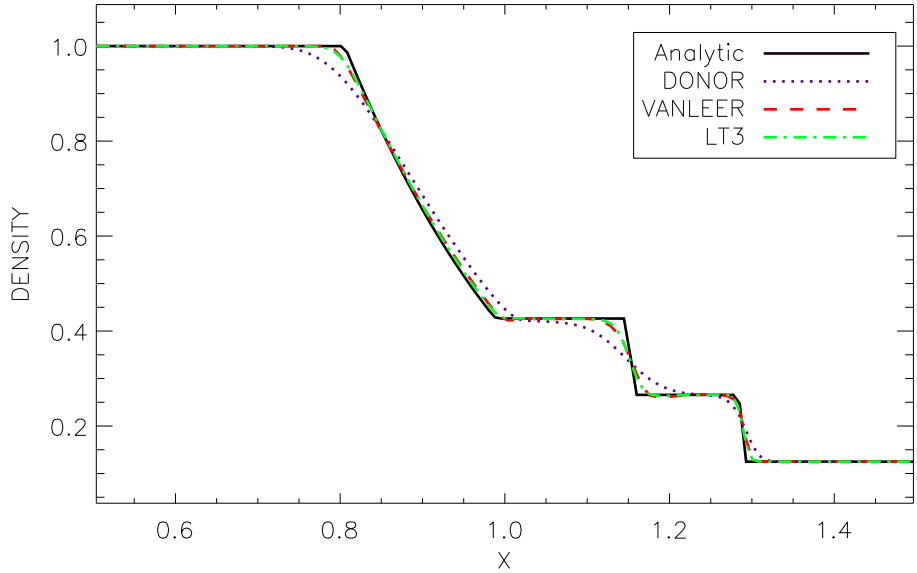
comparable, although LT3 produces slightly smaller errors. Thus, quadratic methods do not give a dramatic improvement over linear methods, partly because the reconstruction is necessarily reduced to first-order in regions where the solution is non-smooth. Again, this suggests that the higher-order methods may not be of value for this type of problem as the dominant errors are concentrated at the shock and contact discontinuity. However, it is not clear whether this is true of higher-order methods in general, or is only a consequence of the fact that the LT3 reconstruction is not provably third-order accurate.

Spatial convergence rates for the density solution for each method are presented in Table 3.15. As expected, the linear and quadratic methods yield convergence rates of approximately 1, while DONOR yields a rate of about $\frac{2}{3}$.

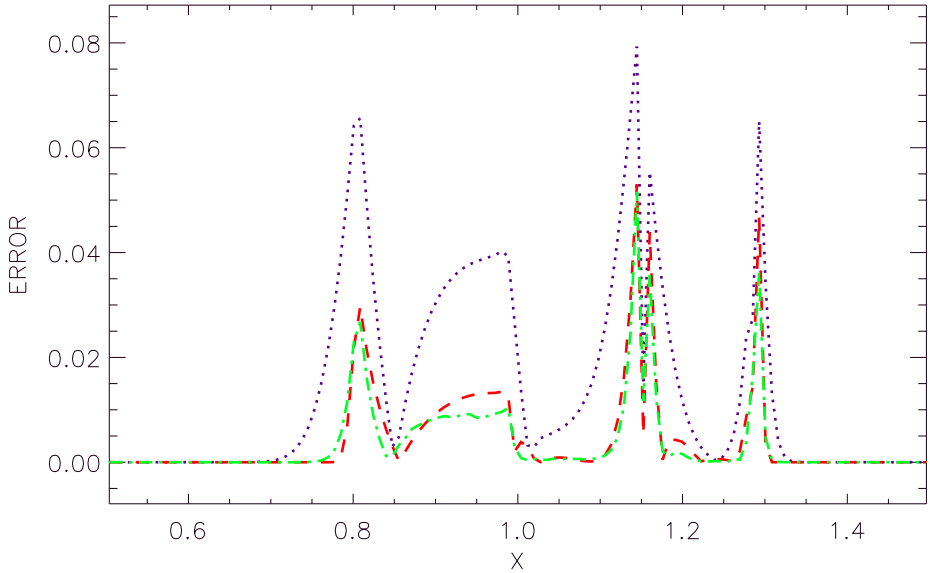
	DONOR		VANLEER		LT3	
Cells	L^1	Order	L^1	Order	L^1	Order
64.	2.135E-02	–	8.912E-03	–	7.133E-03	–
128.	1.434E-02	0.574	4.507E-03	0.983	3.788E-03	0.913
256.	9.367E-03	0.615	2.550E-03	0.822	2.184E-03	0.794
512.	5.918E-03	0.663	1.378E-03	0.888	1.174E-03	0.895
1024.	3.770E-03	0.650	7.762E-04	0.828	6.700E-04	0.809

Table 3.15: Global convergence rates for first, second, and third-order methods on the Sod shocktube problem.

We now turn our attention to the question of the computational cost vs. error. Figure 3.12 shows L^1 error vs. computational cost measured in terms of CPU Time. The linear and quadratic methods seem to be fairly similar in terms of the trend in solution quality vs. computational cost for the Sod problem. However, the cost to obtain a solution with an error level of 0.001 with the quadratic LT3 method is slightly more than with the linear VANLEER method. Using the constant DONOR method to reach this error-level would be significantly more computationally intensive than with either of the linear methods. This result seems to suggest that linear methods, with their reduced memory requirements and algorithmic simplicity are nearly optimal for this class of problem. However, other third-order accurate reconstruction methods, such as ENO and WENO schemes, may provide a worthwhile improvement. We do not consider such schemes in this work.

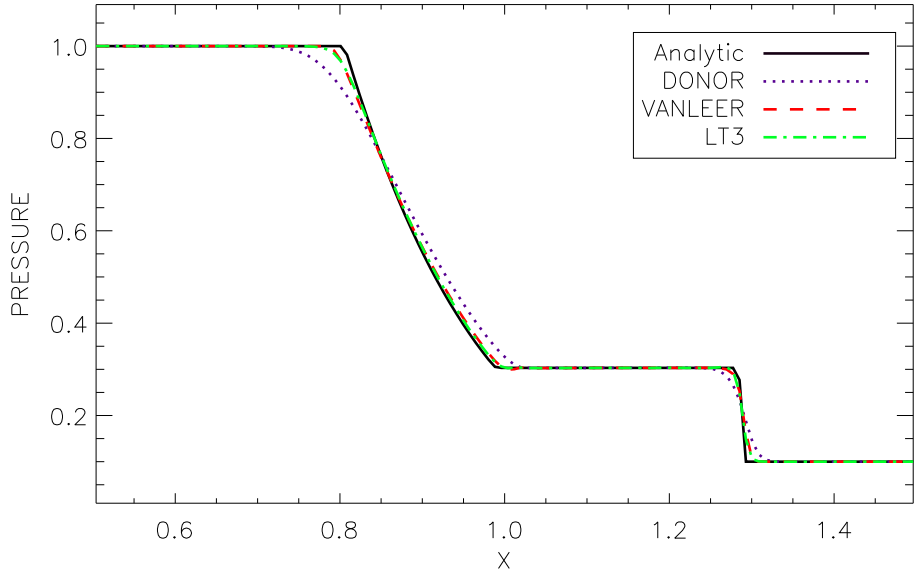


(a) Density profile.

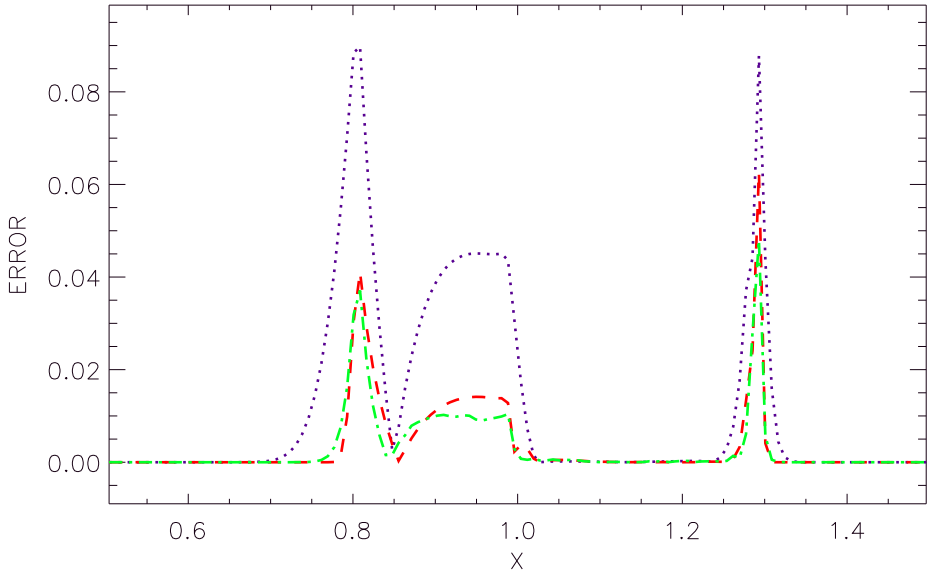


(b) Error distribution in density.

Figure 3.9: Snapshot of the density and density error distribution at $t = 0.1644$ for constant, linear, and quadratic reconstruction methods on the Sod shock tube problem.

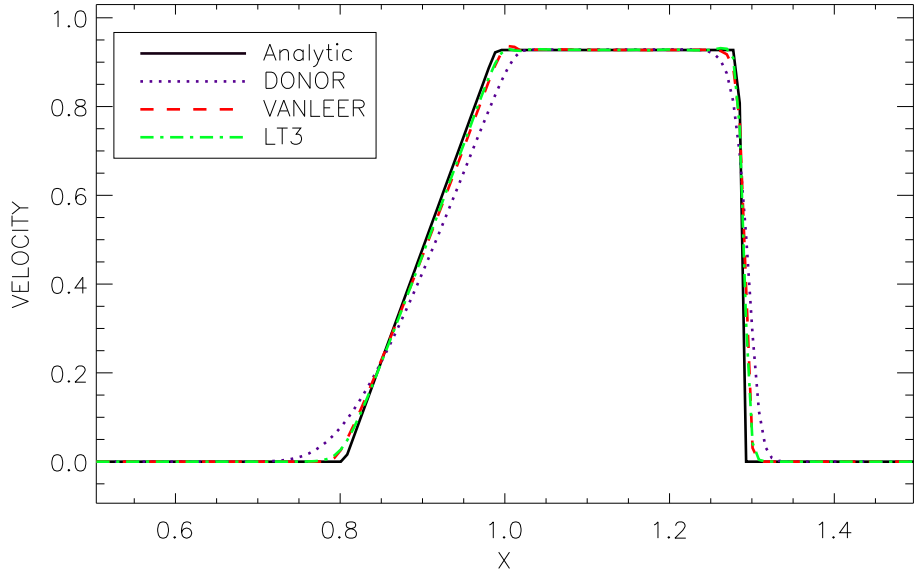


(a) Pressure solution.

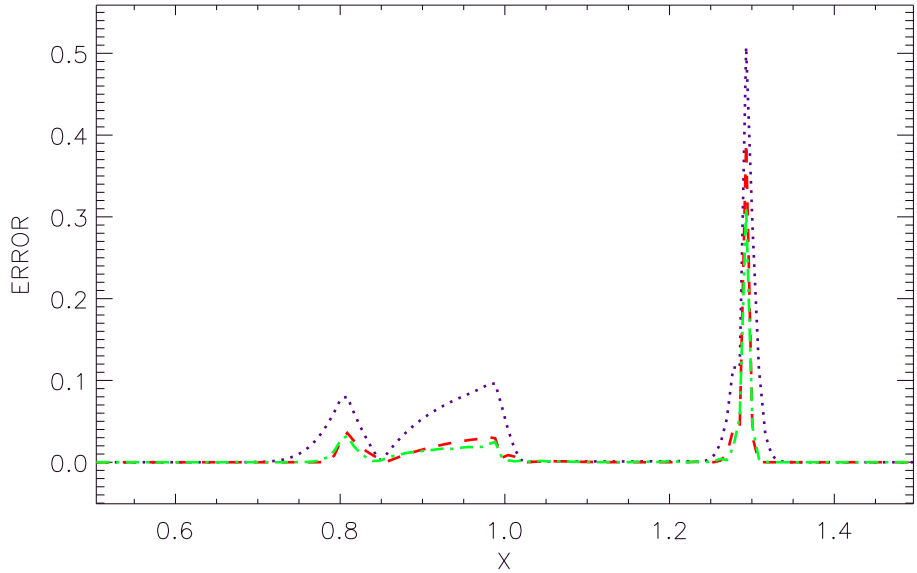


(b) Error distribution in pressure.

Figure 3.10: Snapshot of the pressure and pressure error distribution at $t = 0.1644$ for constant, linear, and quadratic reconstruction methods on the Sod shock tube problem.



(a) Velocity profile.



(b) Velocity error distribution.

Figure 3.11: Snapshot of the velocity and velocity error distribution at $t = 0.1644$ for constant, linear, and quadratic reconstruction methods on the Sod shock tube problem.

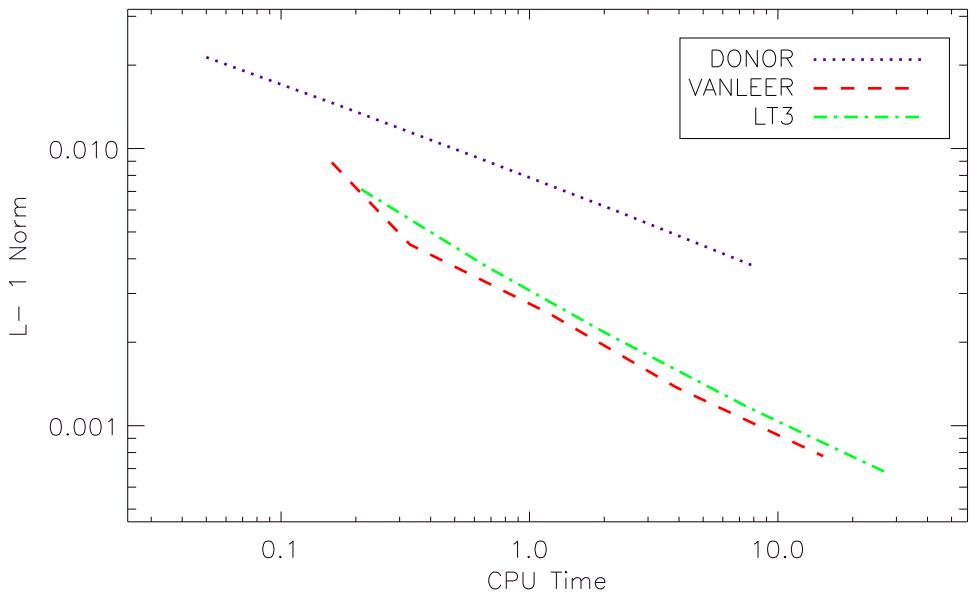


Figure 3.12: Error vs. cpu time for constant, linear, and quadratic reconstruction methods on the Sod shock tube problem.

3.3 Methods Comparison

In this section, we compare multiple codes and solution methods on a series of shock problems using a "figure-of-merit" based on both accuracy and computational cost. Two primary codes are used for the comparison, ALEGRA and FLASH [4]. ALEGRA is built upon on the NEVADA framework and uses both structured and unstructured grid methodologies. The ALEGRA code has three variations: a) ALEGRA/UNS, b) ALEGRA/STR and c) ALEGRA/ECL.

ALEGRA/UNS is an arbitrary Lagrangian-Eulerian (ALE) code that is based on a blend of finite element and finite volume technologies. The ALEGRA/UNS ALE solution algorithm consists of a Lagrangian step followed by a remap (advection) step. The Lagrangian step uses a modified finite element method that relies on mass lumping, constant-gradient element technology and hourglass stabilization. The remap algorithm uses a finite volume formulation that provides both cell-centered and node-centered advection algorithms. The overall computational cost of the ALE algorithm per unit time step includes both the cost of the Lagrangian step plus the cost of the remap step.

The ALE algorithm in ALEGRA has recently has been extended to treat structured grids and is referred to as ALEGRA/STR. Although algorith-

mically similar to ALEGRA/UNS, ALEGRA/STR takes advantage of the multi-block structured grid layout to reduce memory requirements and memory addressing overhead. However, like its unstructured counterpart, ALEGRA/STR uses a Lagrangian step followed by a remap step for Eulerian calculations.

The ALEGRA/ECL code is also based on the structured grid capability in the NEVADA framework, and provides the primary application interface for the conservation law framework. Although ALEGRA/ECL shares the same structured grid infrastructure as ALEGRA/STR, the algorithms in ALEGRA/ECL are separate and distinct.

The FLASH code is the astrophysics code developed at the University of Chicago in its DOE ASCI Alliance center. FLASH was designed around a block adaptive mesh refinement strategy, and is used here as a basis of comparison with the multi-block structured solvers in ALEGRA. The FLASH code provides a directionally split piecewise parabolic method solver (FLASH/PPM) descended from the PROMETHEUS code [10]. The basic PPM algorithm is described in detail by Woodward and Colella [44]. FLASH implements the direct-Eulerian version of PPM. Also available in FLASH are central methods (FLASH/KNP) comparable to those implemented in ALEGRA/ECL. All FLASH results presented herein are from FLASH version 2.1.

3.3.1 Comparison Metrics

Simulation codes can be evaluated by several metrics. These include:

- Computational Time

  This is the time required to solve a problem with a given grid resolution. This can be misleading since there is no consideration of solution accuracy in this metric.

- Memory Usage

  This metric is the total memory required to run the calculation at a given grid resolution. Memory usage requirements arise from the basic algorithm but are strongly influenced by the software design and the algorithm implementation. Hidden memory costs may arise from parallel I/O requirements and material model interfaces which can significantly impact optimal memory usage.

- Accuracy per unit computational cost

  This type of metric is most meaningful since it gives some measure of goodness of solution as a function of cost. In this case, cost can be

total computer time, total memory, or a composite cost based on total computational time and memory.

For our purposes, we have chosen to consider the accuracy per unit computational cost as our figure-of-merit. In the ensuing discussion, we consider a variety of compressible flow problems that include the Sod shock tube problem, an isentropic vortex problem, and the Sedov blast wave problem.

3.3.2 The Sod Shock Tube

In this section, we compare the ALEGRA/ECL solution algorithms to those of ALEGRA/UNS, FLASH/PPM and FLASH/KNP on the Sod shock tube problem. A series of two-dimensional meshes with periodic boundary conditions and of varying resolution are used for the comparison. All computations performed with ALEGRA/ECL use the Kurganov-Noelle-Petrova (KNP) numerical flux. Both ALEGRA/ECL and FLASH/KNP employ MINMOD, $\omega = 1.3$ reconstruction for this test.

Before proceeding, several comments regarding the FLASH code are required. The FLASH code is designed for block-based adaptive mesh refinement (AMR). In our comparisons we essentially turned off this feature and ran with a uniform structured grid. Each block in FLASH has a 4-cell overlap with surrounding blocks, so we increased the block size to 32x32 cells in an attempt to minimize the extra cost associated with the overlap. The overhead may still account for a significant portion of the disparity in computation time that we observed between the FLASH and ALEGRA/ECL implementation of the Godunov-type central schemes. In addition, FLASH uses a third-order Runge-Kutta time integrator, as compared to the second-order SSP Runge-Kutta integrator used for most of the ALEGRA/ECL computations. A more precise and fair comparison will be made in the future when a full multi-block implementation in ALEGRA/ECL using inter-block/parallel communications as well as local boundary conditions is completed.

To begin the comparison, Figure 3.13(a) shows snapshots of the density solution and density error distribution for ALEGRA/ECL, ALEGRA/ECL-W, ALEGRA/UNS, FLASH/KNP and FLASH/PPM. ALEGRA/ECL-W uses Williamson's low-storage third-order Runge-Kutta time integrator which is the same time integrator used for the FLASH/KNP calculations. The results computed with ALEGRA/ECL and FLASH/KNP are nearly identical (as expected). The reason for the slightly reduced accuracy in the FLASH/KNP solution is unclear.

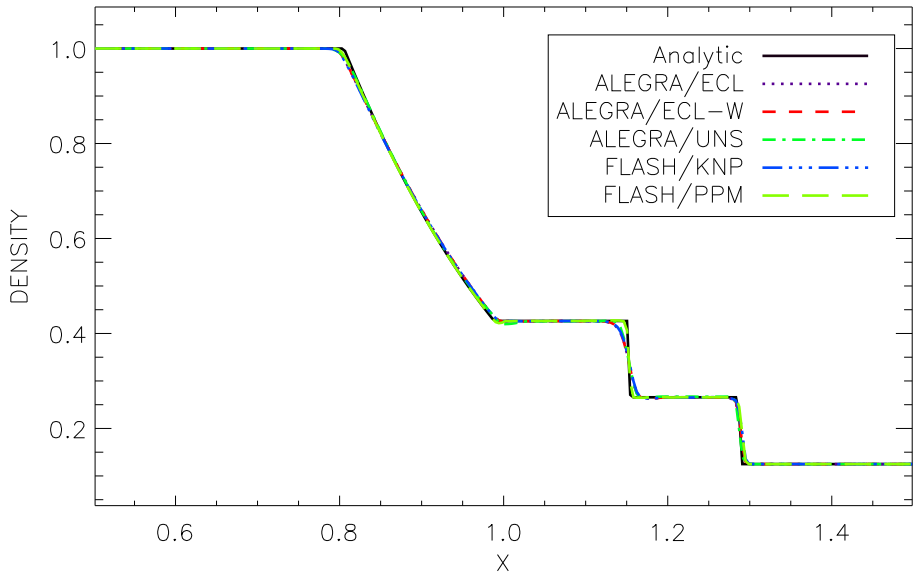
The smearing of the contact discontinuity, relative to ALEGRA and PPM, is evident in both. Overall, ALEGRA/UNS yields results similar to ALEGRA/ECL on a given mesh (but at a much higher computational cost), while FLASH's PPM method yields a significantly more accurate solution.

The increased computational cost of PPM results in an accuracy vs. cost metric similar to ALEGRA/ECL, as indicated by the fact that the curves lie close to each other in Figure 3.15(b). We note, however, that PPM may be preferable on a memory-constrained system, since it attains similar accuracy with a smaller grid, as compared to ALEGRA/ECL. The fact that the FLASH/KNP integration results are slow compared to the ALEGRA/ECL, even when employing identical time integrators, appears indicative of large overhead costs in the FLASH multi-block adaptive framework. If this is the case then it would be expected that a high resolution Riemann solver based numerical flux function approach like PPM would be extremely advantageous inside of the ALEGRA/ECL framework.

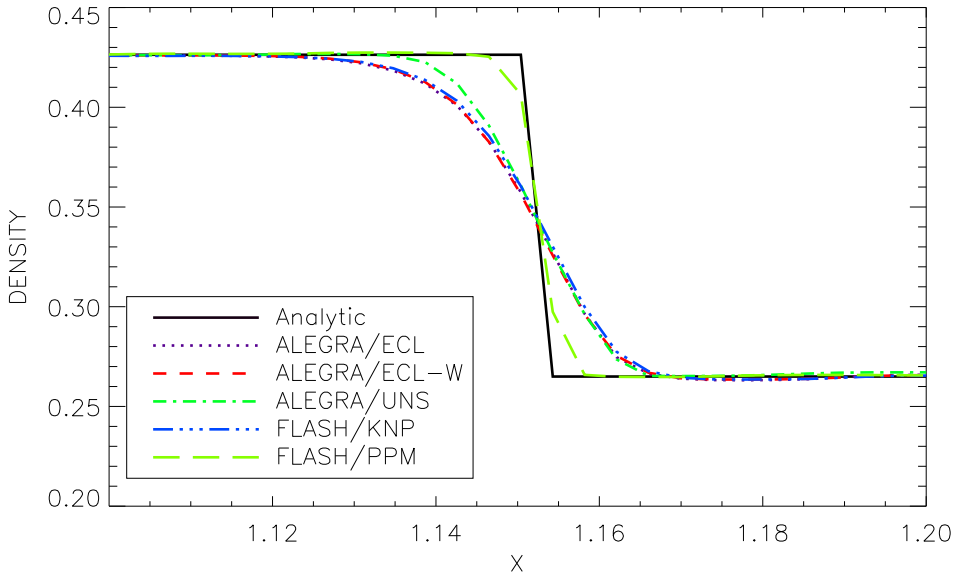
Each solution method attains approximately first-order spatial convergence as shown in Figure 3.15(a) and as expected for a non-smooth solution. Figure 3.15(b) shows the results from each code in terms of figure-of-merit, i.e., error vs. CPU time. Table 3.3.2 shows a tabular comparison of this data. Each method yields approximately the same error levels with the exception of FLASH/PPM which yields a relative error 40% smaller than ALEGRA/ECL. Surprisingly, the FLASH algorithms are 5 to 12 times more expensive than the ALEGRA/ECL methods, while ALEGRA/UNS is 60 to 137 times more expensive than the ALEGRA/ECL algorithms.

Code	Run time (s)	Relative time	Error	Rel. Error
ALEGRA/ECL	348.0	1.0	1.921E-03	1.000
ALEGRA/ECL-W	790.2	2.3	1.800E-03	0.937
ALEGRA/UNS	47801.1	137.4	1.553E-03	0.809
FLASH/KNP	4267.4	12.3	2.151E-03	1.120
FLASH/PPM	4676.1	13.4	7.753E-04	0.404

Table 3.16: Performance of ALEGRA/ECL vs. other codes running the shocktube on a 256x256 mesh to time $t = 0.1644$ s.



(a) Density snapshots.



(b) Closeup of contact discontinuity.

Figure 3.13: Snapshots of the density solutions for the Sod shock tube problem at $t = 0.1644$ on a 128×128 grid for four solution algorithms.

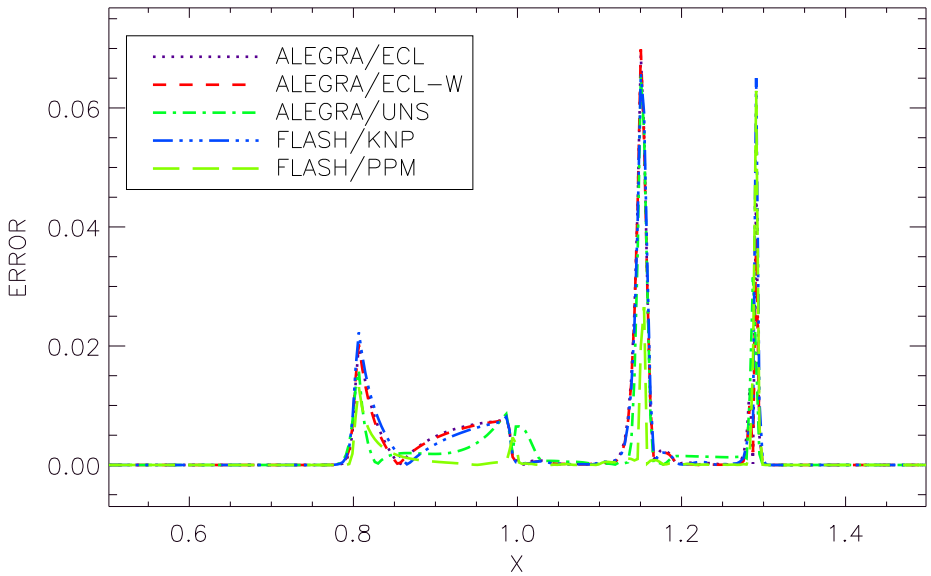
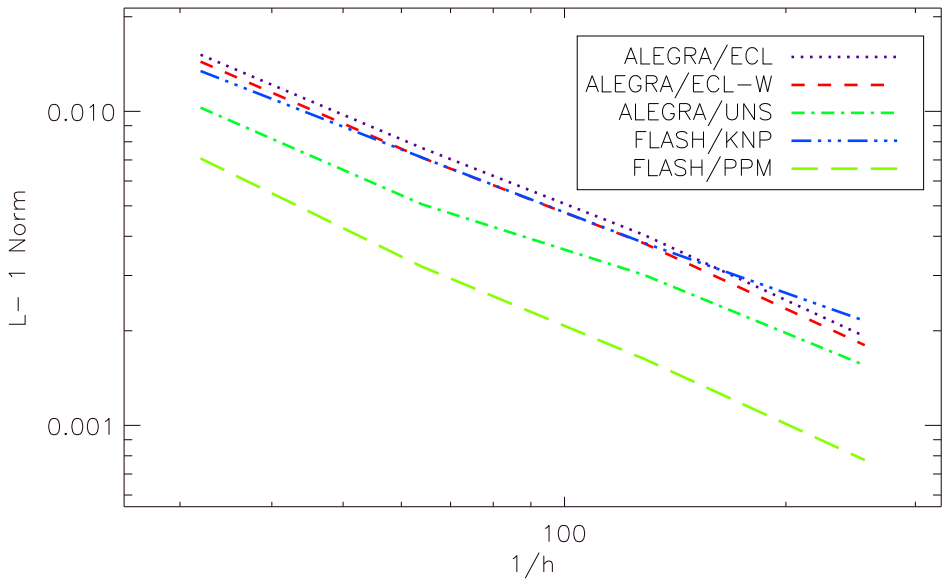
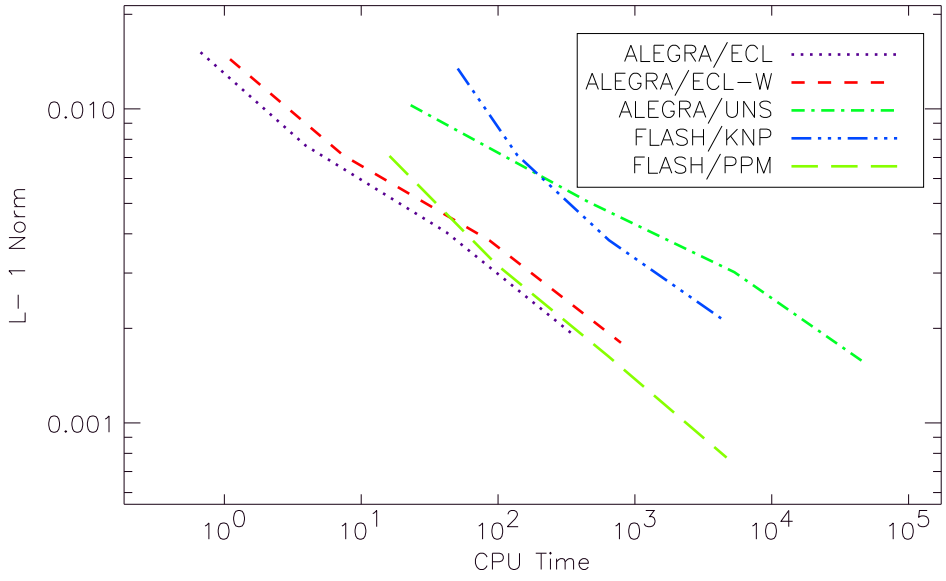


Figure 3.14: Errors in the density solutions for the Sod shock tube problem at $t = 0.1644$ on a 128×128 grid for four solution algorithms.



(a) Error vs. mesh refinement



(b) Error vs. CPU time

Figure 3.15: Comparison of Sod shock tube problem solution convergence and cost for four different solution methods.

3.3.3 The Isentropic Vortex

In order to compare the multi-dimensional performance of the central schemes implemented in ALEGRA/ECL with a method that uses a Riemann solver, we selected the isentropic vortex problem studied by Yee et. al. [45]. The initial conditions correspond to a vortex in dynamic equilibrium. The calculation is performed using a domain with $-5 < x < 5$, $-5 < y < 5$ and the vortex centered at $(0, 0)$. Far away from the vortex, the pressure and density are $p_\infty = 1.0$, and $\rho_\infty = 1.0$. The initial conditions are given in terms of x, y, and $r = \sqrt{x^2 + y^2}$ as

$$u = u_\infty - y\frac{\beta}{2\pi}e^{\frac{1-r^2}{2}}, \tag{3.23}$$

$$v = v_\infty + x\frac{\beta}{2\pi}e^{\frac{1-r^2}{2}}, \tag{3.24}$$

$$T = T_\infty[1 - \frac{(\gamma-1)\beta^2}{8\gamma\pi^2}e^{1-r^2}], \tag{3.25}$$

$$\rho = \rho_\infty(\frac{T}{T_\infty})^{\frac{1}{\gamma-1}}, \tag{3.26}$$

where $\beta = 5.0$ is the vortex strength and u_∞ and v_∞ are the far-field x and y-velocities, respectively, and T_∞ is the ambient temperature. The domain is initialized using cell averages.

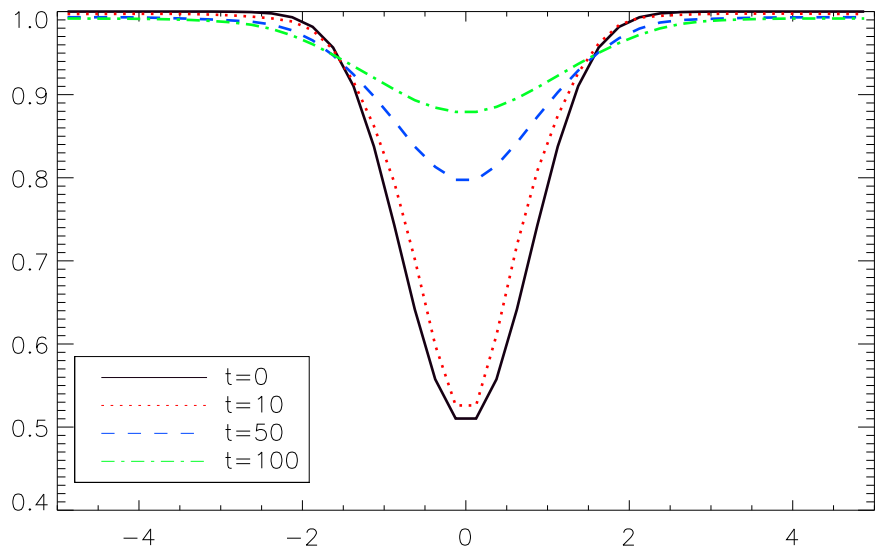
Two versions of the problem were used – a stationary vortex, and an advecting vortex. Both versions were run with ALEGRA/ECL and with FLASH/PPM. The options used for the ALEGRA/ECL runs are KNP flux, SSPRK42 time integration, and harmonic VANLEER reconstruction. Once again, we turned off the AMR features in FLASH and ran on fixed grids.

In the stationary rotating vortex problem, $u_\infty = v_\infty = 0$, so that the problem has no background flow. Density results using a 40×40 grid near the line $y = 0$ for times $t = 0, 10, 50$, and 100 are plotted in Figures 3.16(a) and 3.16(b). While the FLASH/PPM solution is relatively good even after 100 time units, the ALEGRA/ECL solution is very poor, reflecting the more dissipative nature of central schemes relative to schemes using Riemann solvers.

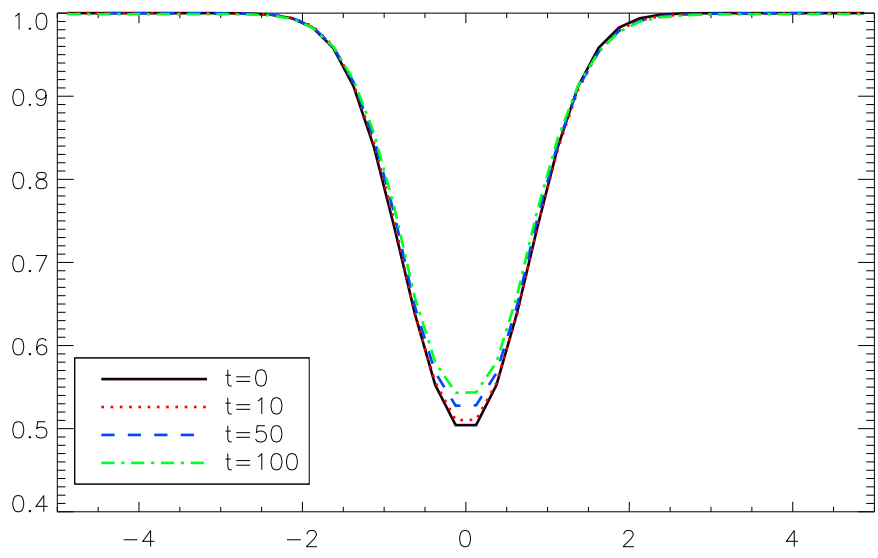
A more difficult problem involves the same initial conditions, but with a background flow field, $u_\infty = v_\infty = 1.0$. Thus the vortex is advected at a 45° angle relative to the coordinate axes. This problem has been used to test the FLASH code [4]. Again the calculation was performed using 40×40 grids. The results are shown in Figures 3.17(a) and 3.17(b) where the density profiles along the x-axis are plotted for times $t = 0$, 10, 50, and 100. The ALEGRA/ECL solution degrades even more quickly and is reduced to noise well before $t = 100$ in contrast to the results obtained using PPM (FLASH) on same grid.

As we discuss later, the computational cost of ALEGRA/ECL is much less than FLASH/PPM for equally refined grids. To obtain a more fair (in terms of computational cost) comparison between the two methods, we ran the stationary and advecting vortex problems on an 80×80 grid in ALEGRA/ECL. Results are shown in Figures 3.18(a) and 3.18(b). For the stationary problem, the results to $t = 100$ are reasonably good. For the advecting problem, however, the solution still deteriorates greatly over this time interval.

The errors in this case appear to be primarily dissipative, but phase errors, induced by the advection velocities, also contribute significantly, distorting the vortex and moving it off-center. Kurganov and Tadmor [22] present results of two-dimensional Riemann problems using genuinely multidimensional central schemes; their results show that such schemes give solutions similar to those obtained by schemes with Riemann solvers. It is possible that the poor performance of Alegra/ECL on this two-dimensional problem is at least partially due to the dimension-by-dimension implementation. It remains to be seen whether a fully multi-dimensional approach could significantly improve the performance of the central scheme approach on this problem. It is possible that good treatment of contact waves is the more critical feature to require in a shock capturing numerical method for this test problem. These questions remain open.

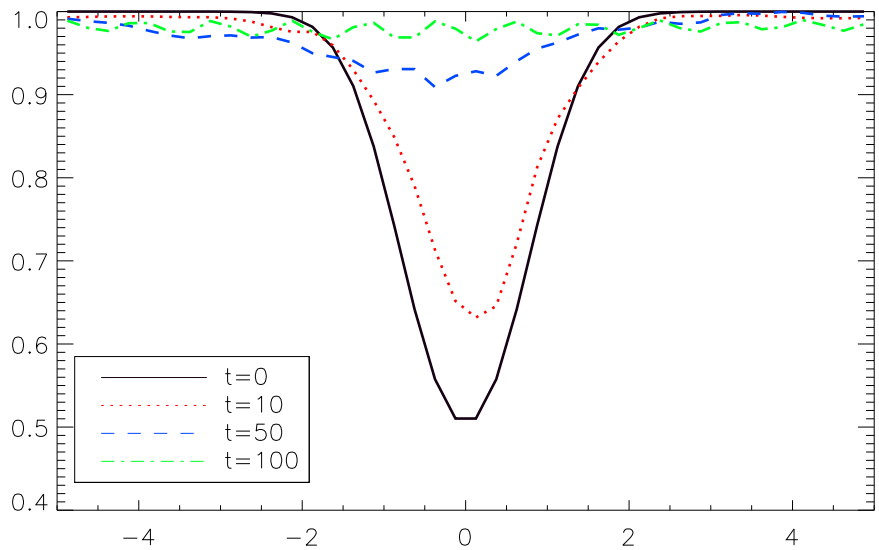


(a) ALEGRA/ECL

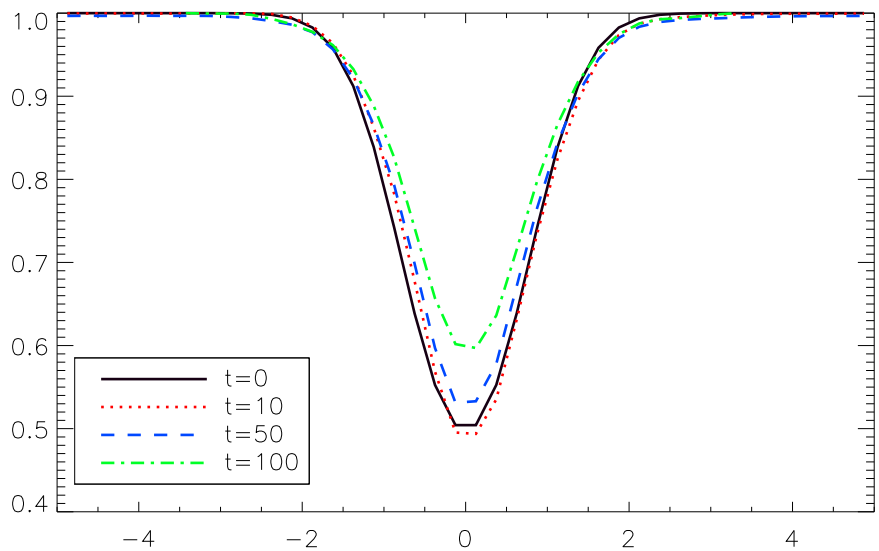


(b) FLASH/PPM

Figure 3.16: Deterioration of the density solution for the stationary isentropic vortex problem on a 40×40 grid using ALEGRA/ECL and FLASH/PPM. The density profiles along the x-axis are plotted for $t = 0$, 10, 50, and 100.

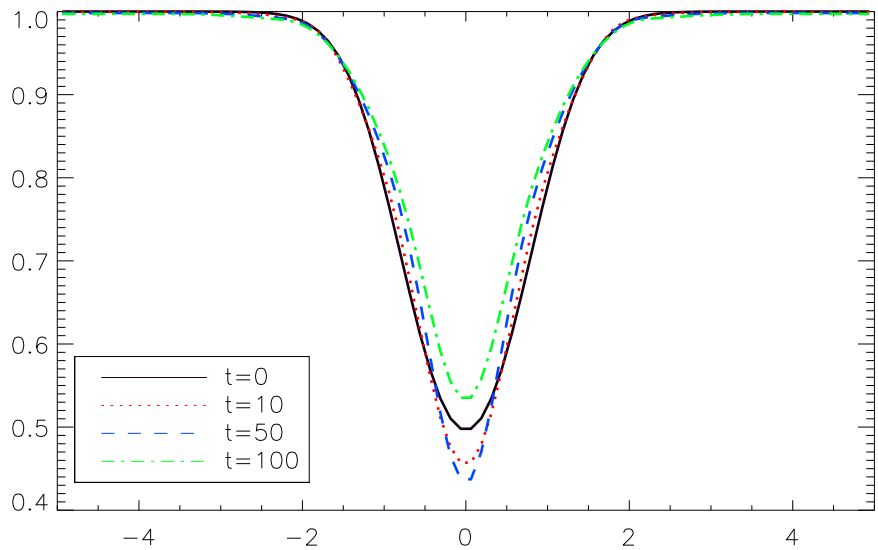


(a) ALEGRA/ECL

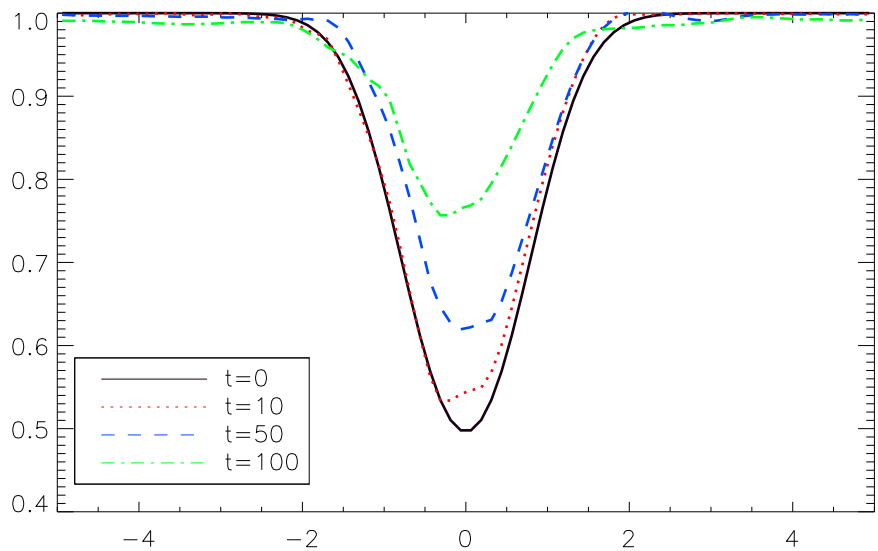


(b) FLASH/PPM

Figure 3.17: Deterioration of the density solution for the advecting isentropic vortex problem on a 40×40 grid using ALEGRA/ECL and FLASH/PPM. The density profiles along the x-axis are plotted for $t = 0$, 10, 50, and 100.



(a) Stationary vortex on 80×80 grid



(b) Advecting vortex on 80×80 grid

Figure 3.18: Deterioration of the ALEGRA/ECL solution for the stationary (top) and advecting (bottom) isentropic vortex problem on 80×80 grid. The density profiles along the x-axis are plotted for $t = 0$, 10, 50, and 100.

3.3.4 The Sedov Blast Wave

The Sedov problem [38] involves the self-similar evolution of a spherical blast wave from a delta-function initial pressure perturbation in an otherwise homogeneous medium. The analytic solution at a given time t is a function only of distance from the initial perturbation. The material involved is an ideal gas with $\gamma = \frac{5}{3}$. This problem tests a code's ability to deal with strong shocks and maintain expected symmetries. In the following set of comparisons, we run the problem in a cube with $-1 \leq x, y, z \leq 1$. The initial conditions are

$$\rho_{amb} = 1 \tag{3.27}$$
$$T_{amb} = 10^{-5} \tag{3.28}$$

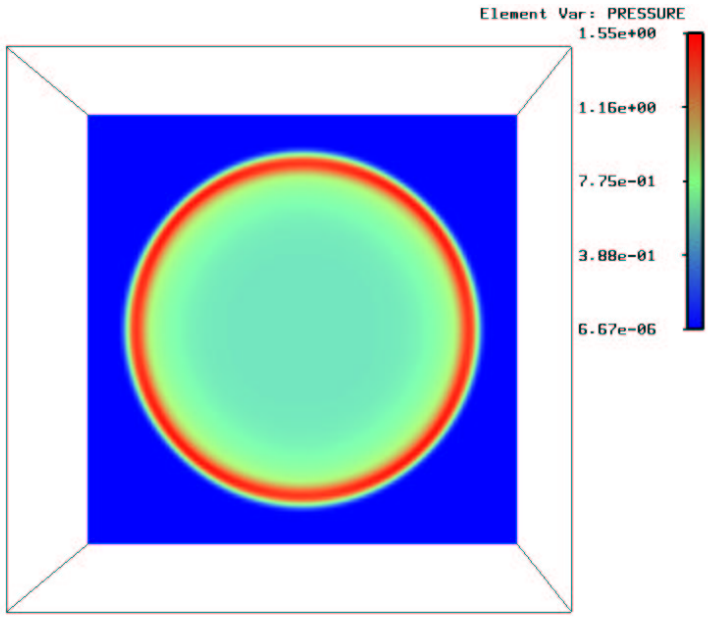
For this suite of computations, a slightly pre-expanded high-pressure region is specified as a sphere centered at $[0, 0, 0]$ with radius $r = 0.1$. This is done to avoid numerical difficulties presented by trying to approximate a delta-function for the initial pressure field. Inside this region the initial conditions are

$$\rho_{exp} = 1 \tag{3.29}$$
$$T_{exp} = 10^{3} \tag{3.30}$$

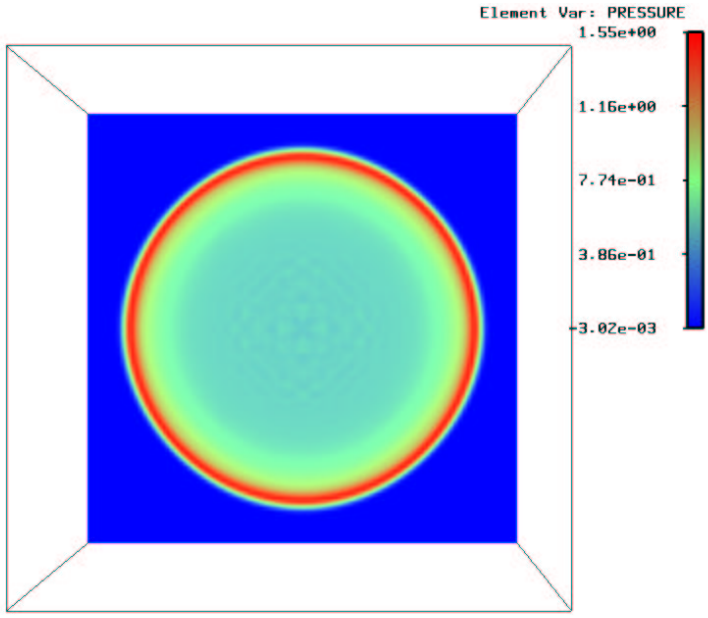
The computations are run to a simulated time of $t = 0.2$. The ALEGRA/ECL simulation is run with periodic boundary conditions, while the ALEGRA/STR simulation implements a free surface condition. In either case, no boundary effects interact with the explosion during the simulation. The ALEGRA/ECL simulation also employs KNP flux and MINMOD, $\omega = 1.3$ reconstruction. We note in passing that the FLASH code was not exercised on this problem because of the programming effort involved in duplicating the initial conditions. We ran the simulation in both ALEGRA/ECL and ALEGRA/STR on a 96^3 mesh.

Figure 3.19 shows snapshots of the pressure field at $t = 0.2$ for ALEGRA/ECL and ALEGRA/STR. The ALEGRA/STR result exhibits slightly negative pressures just outside the shock and noticeable ringing in the interior of the blast wave.

Figures 3.20 and 3.21 show radial density and pressure profiles of the analytic solution and the computed solutions at $t = 0.2$ on the 96^3 mesh for ALEGRA/ECL and ALEGRA/STR. The ALEGRA/ECL solution is accurate behind the shock and calculates the correct shock speed; however, it exhibits significant smearing of the shock peak. In the ALEGRA/STR solution, the shock speed is incorrect, although the peak is slightly sharper.



(a) ALEGRA/ECL solution.



(b) ALEGRA/STR solution.

Figure 3.19: Cutplanes showing snapshots of the pressure field for the Sedov problem at $t = 0.2$.

In order to achieve the correct shock speed, the ALEGRA/STR simulation was run with a modification that conserves total energy (referred to herein as ALEGRA/STR_TE). The results are displayed in Figure 3.22. In this case, the shock speed is correct, but the solution is inaccurate behind the shock. Errors for all three tests are plotted in 3.23.

Tables 3.17 and 3.18 show a comparison of the run times and L^1 error norms for the three codes. ALEGRA/ECL runs $2.6 - 2.7$ times faster and obtains a significantly better pressure solution with a computed error that is smaller by a factor of two.

Code	Run time (s)	Relative time	L^1 Error	Rel. Error
ALEGRA/ECL	3149.7	1.0	4.227E-01	1.000
ALEGRA/STR	8034.3	2.6	4.662E-01	1.103
ALEGRA/STR_TE	8456.9	2.7	7.797E-01	1.844

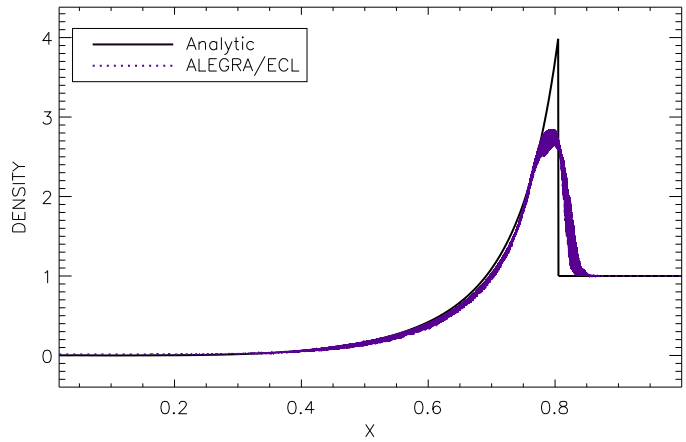
Table 3.17: Comparison of ALEGRA/ECL vs. ALEGRA/STR density solution and speed for Sedov problem.

Code	L^1 Error	Rel. Error
ALEGRA/ECL	2.317E-01	1.000
ALEGRA/STR	4.960E-01	2.141
ALEGRA/STR_TE	4.270E-01	1.843

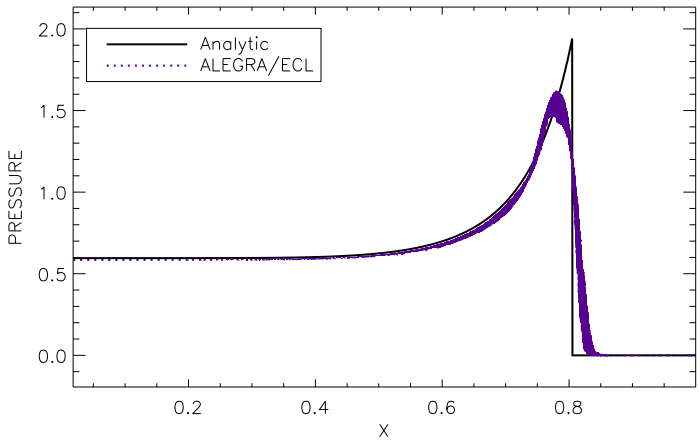
Table 3.18: Comparison of ALEGRA/ECL vs. ALEGRA/STR pressure solution for Sedov problem.

We also used the Sedov problem to investigate the symmetry-preserving abilities of the two codes. Ideally the solution should be perfectly spherically symmetric, but this is clearly not possible on a Cartesian grid – due in part to dispersive errors. However, it is possible and extremely desirable to preserve left-right symmetry, as well as symmetry through 90° rotations. Again using the results computed on the 96^3 meshes, we calculated the difference between the positive- and negative-x halves of the density solution. Similar checks were performed in the y and z directions (180° rotations). For both ALEGRA/ECL and ALEGRA/STR, all of the results were identically zero. This is an important verification result, demonstrating that there are no coding errors that result in left-right asymmetries.

We then examined asymmetries between the x-, y-, and z-axes by performing 90° degree rotations of the solution and subtracting the rotated solution from the actual solution. For ALEGRA/ECL, the results were, again,

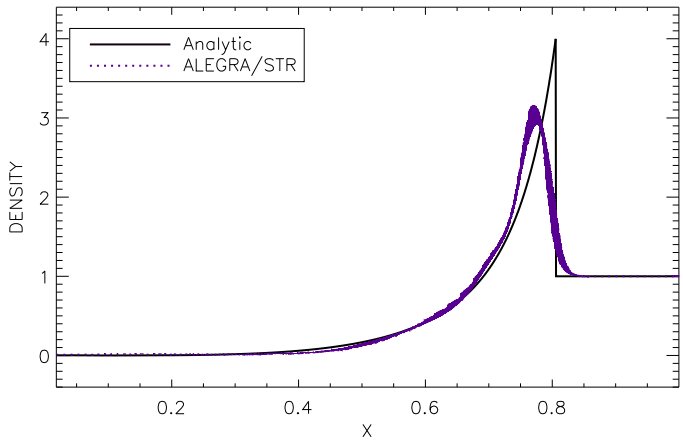


(a) ALEGRA/ECL density.

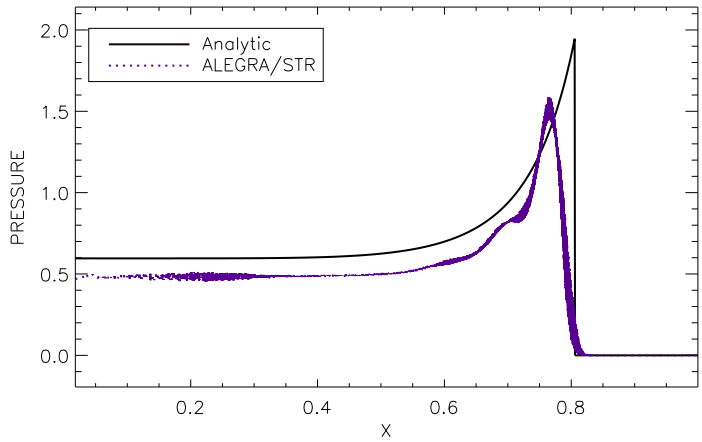


(b) ALEGRA/ECL pressure.

Figure 3.20: Density and pressure solutions as a function of distance from center for Sedov problem at $t = 0.2$.

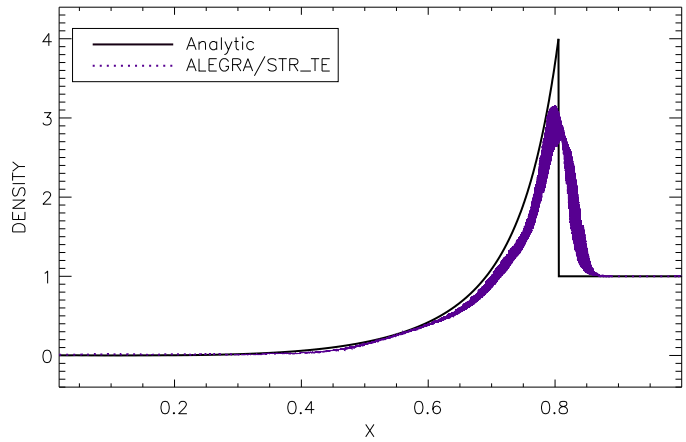


(a) ALEGRA/STR density.

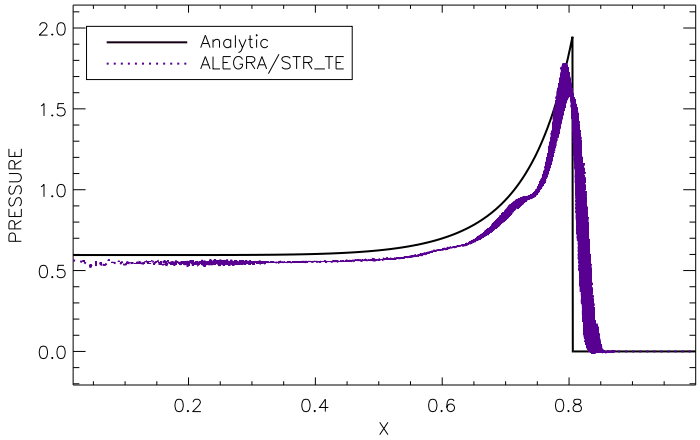


(b) ALEGRA/STR pressure.

Figure 3.21: Density and pressure solutions as a function of distance from center for Sedov problem at $t = 0.2$.

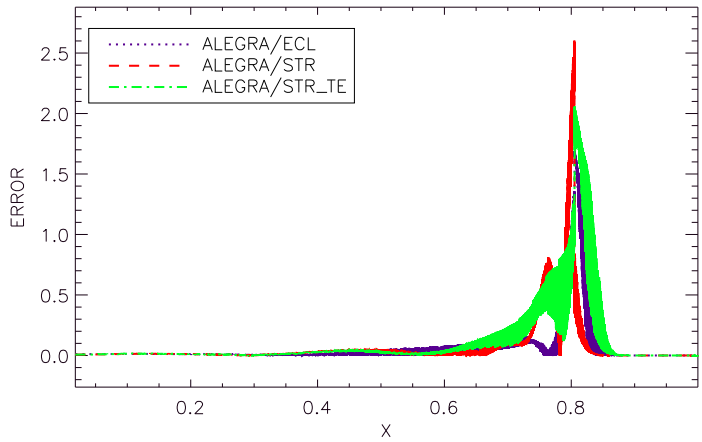


(a) ALEGRA/STR_TE density.

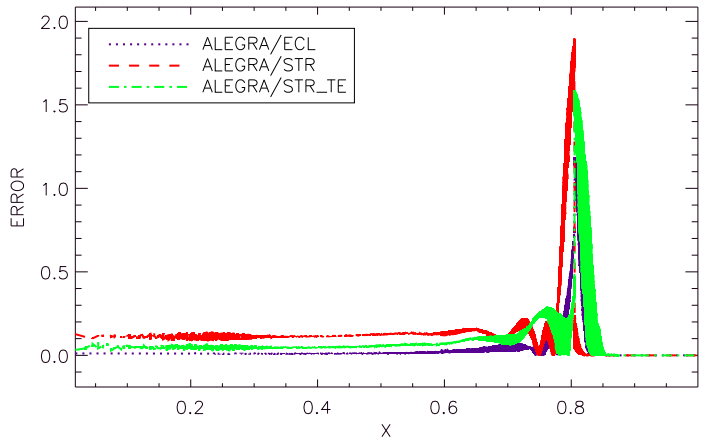


(b) ALEGRA/STR_TE pressure.

Figure 3.22: Density and pressure solutions as a function of distance from center for Sedov problem at $t = 0.2$.



(a) Errors in density.



(b) Errors in pressure.

Figure 3.23: Density and pressure solutions as a function of distance from center for Sedov problem at $t = 0.2$.

Line of Symmetry	Mesh Resolution $\Delta x = 1/32$	$\Delta x = 1/64$	$\Delta x = 1/96$
x-y	1.334e-3	1.212e-3	6.603e-4
x-z	1.946e-3	7.584e-4	9.051e-4
y-z	2.437e-3	1.401e-3	9.328e-4

Table 3.19: Symmetry metrics for ALEGRA/STR based on the Sedov problem at $t = 0.2$.

all identically zero indicating that sphericity is retained in the numerical solution. This shows that, although the methods we have implemented are formally dimension-by-dimension, they avoid directional asymmetries. The same tests gave non-zero results for the ALEGRA/STR solution as shown in Table 3.19. These errors are due to the use of a directionally-split remap method in which the order of the directional passes is important.

3.3.5 Numerical Flux Comparison

Figures 3.24(a) and 3.24(b) show results of a comparison using KNP vs. KT flux methods on the Sod shock tube problem. As expected, KNP flux appears to achieve somewhat better results with no significant increase in calculation time.

3.3.6 EOS Interface Comparison

In order to evaluate the relative performance and accuracy of the general function evaluation EOS and the in-line ideal gas EOS, we ran the Sod shock tube problem on a series of identical meshes with the two modules. Figure 3.25 shows the results of these runs, demonstrating that the two give virtually identical results and run times. This is important since it indicates that the access procedures and data layouts will not significantly affect run times.

3.4 Non-Oscillatory Third-Order Reconstruction

As discussed previously we have implemented two non-oscillatory third-order reconstruction methods, referred to herein as LT3 and KP3 (see [30, 31] and [20], respectively). As shown in §3.2.2 we observed sub-optimal convergence rates using these methods for Burgers' equation. After investigating the foundations of the reconstruction algorithms we concluded that the limiting

procedures as documented are deficient and that the order of accuracy may actually be much less than claimed for smooth problems. We document below the basic flaw in the limited reconstruction procedure. The orders of accuracy discussed below are consistent with our numerical results for Burgers' equation. We have not developed modifications to either of the algorithms which will recover the desired third-order accuracy. A compact WENO methodology appears to be a useful and robust substitute at this time [27]. However, we have not implemented or tested alternative methods.

We comment that a realization that our results with LT3 and KP3 were actually due to the underlying formulation eluded us for a significant period of time. We can only emphasize the importance of careful and extensive convergence studies on a variety of problems for purposes of verifying proposed algorithms and implementations.

3.4.1 LT3 Reconstruction may be $O(\Delta x^2)$

The deterioriation of the LT3 scheme has already been discussed by Bianco, Puppo and Russo who show that if θ_j is not regular enough accuracy will deteriorate [6]. Our approach is to directly calculate the expected order of accuracy of θ_j.

The key failure in the proof that LT3 is third-order on smooth data is in the estimate for θ_j. It is required that the θ_j values used in each cell satisfy $\theta_j = 1 + O(\Delta x^3)$ as h approaches zero. If both the left and right sides of the cell satisfy this condition then the cell as a whole will satisfy it. It is stated that $M_{j+1/2} - M_j$ is $O(\Delta x^3)$. However, it is the θ_j values from the left or right-hand-side of each cell which are used in the reconstruction and these must be at most $O(\Delta x^3)$ away from 1. A typical formula for θ_j is

$$\theta_j = \frac{M_{j+1/2} - \overline{\mathbf{u}}_j}{M_j - \overline{\mathbf{u}}_j}. \tag{3.31}$$

Since we are using quadratic reconstructions, the values at the cells edges can be represented as

$$\theta_j \sim \frac{\mathbf{u}(x_{j+1/2}) + \alpha h^3 - \overline{\mathbf{u}}_j}{\mathbf{u}(x_{j+1/2}) + \beta h^3 - \overline{\mathbf{u}}_j} \tag{3.32}$$

as h approaches 0. Now $\alpha \neq \beta$ in general since the reconstructed third-order point values feeding into the formula come from different data sets. Then we obtain

$$\theta_j \sim 1 + \frac{\alpha - \beta}{\mathbf{u}(x_{j+1/2}) - \overline{\mathbf{u}}_j} h^3 \tag{3.33}$$

and estimating further

$$\theta_j \sim 1 + 2\frac{\alpha - \beta}{\frac{d\mathbf{u}}{dx}(x_j)} h^2 \tag{3.34}$$

In general θ_j may be $O(\Delta x^2)$ away from 1. Thus LT3 cannot be expected to be better than second-order accurate.

3.4.2 KP3 Reconstruction may be $O(\Delta x)$

Now let us examine a typical formula for θ_j used in the left/right estimate minimization procedure for KP3

$$\theta_j = \frac{M_{j+1/2} - L_{j+1/2}}{M_j - L_{j+1/2}} \tag{3.35}$$

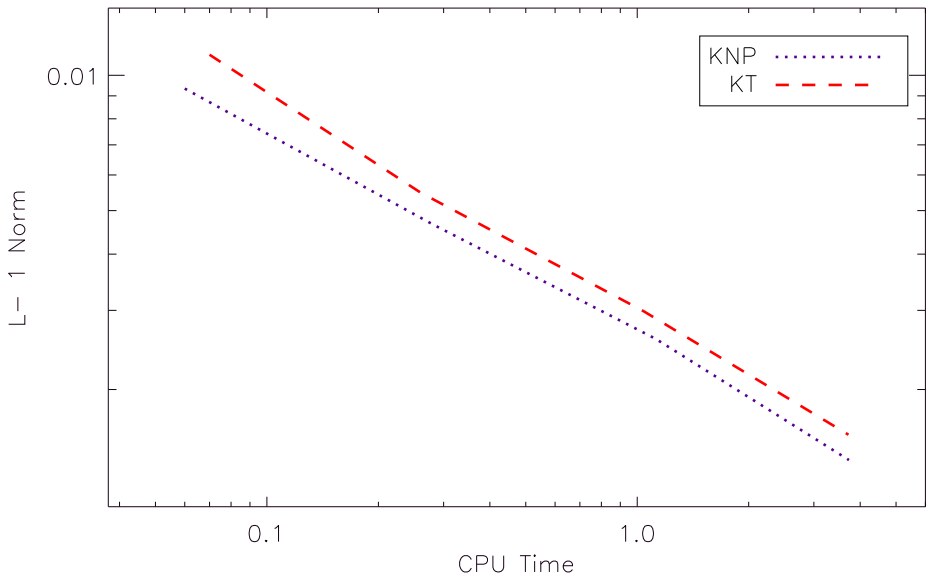
Since we are using quadratic reconstructions, the values at the cells edges can be represented as

$$\theta_j \sim \frac{\mathbf{u}(x_{j+1/2}) + \alpha h^3 - (\mathbf{u}(x_{j+1/2}) + \gamma h^2)}{\mathbf{u}(x_{j+1/2}) + \beta h^3 - (\mathbf{u}(x_{j+1/2}) + \gamma h^2)} \tag{3.36}$$

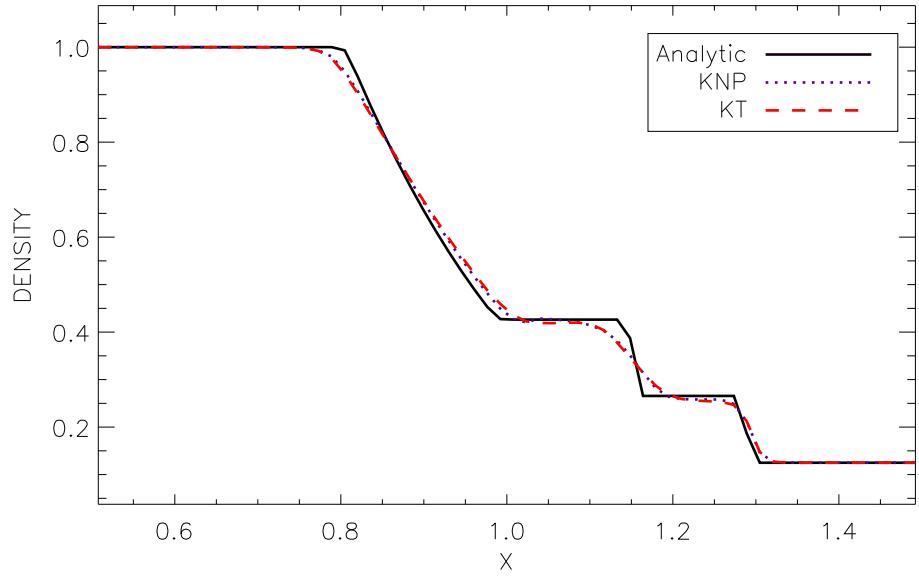
as h approaches 0. Now $\alpha \neq \beta$ in general since the reconstructed third-order point values feeding into the formula come from different data sets. Then we obtain

$$\theta_j \sim 1 + \frac{\alpha - \beta}{\gamma} h \tag{3.37}$$

In general θ_j may be $O(\Delta x)$ away from 1. Thus KP3 cannot be expected to be better than first-order accurate.



(a) Error vs. cost metric



(b) Density solutions at t=0.1644

Figure 3.24: Comparison of Sod shock tube problem solutions obtained with KT and KNP flux methods.

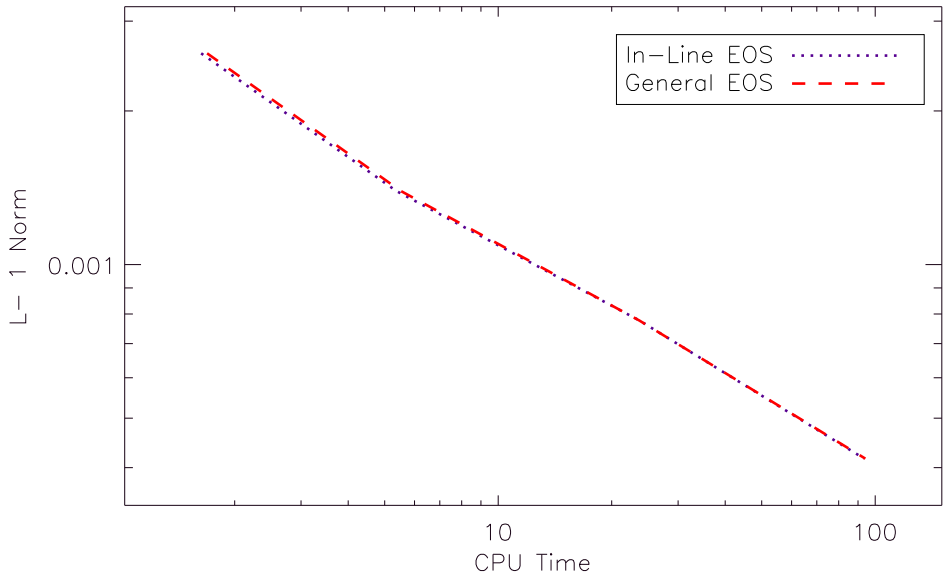


Figure 3.25: Error vs. cpu time comparison for the in-line ideal gas EOS and general EOS interface accessing ideal gas for solving Sod shock tube problem.

Chapter 4

SSP Runge-Kutta Time Integrators

A number of strong-stability preserving (SSP) time integration methods for the method of lines have been developed and documented in the literature. These methods were originally called TVD time discretization methods because they were associated with the total variation (TV) norm. They provide provable non-linear stability properties for time integration of non-linear hyperbolic equations [12, 13]. We give evidence below of the practical results of using (or not using) schemes with these provable non-linear properties. Our results indicate that standard Runge-Kutta methods that are optimized for other properties, e.g., truncation error, can yield very similar results to SSP methods. In this chapter, we investigate the extent to which the SSP property is essential or necessary.

As background, recall that Runge-Kutta (RK) methods of a given order of accuracy are left with free parameters to be chosen at will. These free parameters can be chosen to satisfy low-storage requirements, provide an SSP property, or deliver minimal truncation error. Of course, unfortunate choices can even be made that yield a very poor method which is still formally of the given order of accuracy. In comparison, Spiteri and Ruuth have provided s-stage, p-order SSP RK methods where $s > p$ and shown expanded SSP stability regions which are beneficial even after normalizing by cost [40]. A concise summary of important SSP results is given by Shu [39]. We have implemented several of these SSP methods, namely 2- to 4-stage second-order methods and 3- and 4-stage third-order methods. The optimal low-storage third-order SSP method was also implemented along with Williamson's low storage algorithm.

Godunov shock capturing schemes are designed to introduce the proper viscous dissipation near shocks, in part, to select the correct physical solution. The dissipation is introduced non-linearly into the scheme so it is dif-

ficult to determine exact stability limits when considering a time integration scheme. An experimental approach is one way to investigate the interaction between the dissipative mechanisms in the time integrator and the spatial discretization. We shall investigate the relationship between second-order and third-order time integrators when applied to a central scheme framework for hyperbolic conservation laws for a suite of test problems.

If Δt_{FE} is the strongly stable time step (associated with a particular norm) for a forward Euler integration then the SSP CFL coefficient c is defined as follows: A general s-stage Runge-Kutta scheme with $\alpha_{ij} \geq 0$ and $\beta_{ij} \geq 0$ can be written

$$u^{(0)} = u^n \tag{4.1}$$

$$u^{(i)} = \sum_{j=0}^{i-1}(\alpha_{ij}u^{(j)} + \Delta t\beta_{ij}\mathcal{L}(u^{(j)})),\ i = 1, ..., s \tag{4.2}$$

$$u^{n+1} = u^{(s)}. \tag{4.3}$$

The SSP CFL coefficient c^{ssp} associated with a particular set of α_{ij} and β_{ij} coefficients can be written

$$c^{ssp} = \min_{i,j} \frac{\alpha_{ij}}{\beta_{ij}} \tag{4.4}$$

The Runge-Kutta method is then said to be SSP under the restriction

$$\Delta t \leq c^{ssp}\Delta t_{FE}. \tag{4.5}$$

Generally there will be free parameters available in the Runge-Kutta coefficients so one attempts to find the largest such value c^{ssp} taken over all possible sets of available coefficients, α_{ij} and β_{ij}. Thus

$$c = \max_{\alpha_{ij},\beta_{ij}} c^{ssp} \tag{4.6}$$

defines the optimal SSP CFL coefficient under a given constraint set. The effective SSP CFL coefficient of a $p-$order, $s-$stage method with maximum SSP stable CFL coefficient c is defined as

$$c_{eff} = \frac{cp}{s} \tag{4.7}$$

The CFL number is defined for any RK method as

$$CFL = \frac{\Delta t}{\Delta x}\lambda, \tag{4.8}$$

where λ is the local wave speed. The effective CFL number is defined as

$$CFL_{eff} = \frac{\Delta t}{\Delta x}\frac{\lambda p}{s} \tag{4.9}$$

which is the CFL number normalized by the ratio of the number of stages p to obtain a given order of accuracy divided by the actual number of stages in the method.

4.1 Second-Order Time Integrators

In this section, we compare various second-order RK schemes with 2 stages. These schemes can be written as a one parameter family

$$q = u^n + \alpha \Delta t \mathcal{L}(u^n, t^n) \tag{4.10}$$

$$u^{n+1} = u^n + \beta \Delta t \mathcal{L}(u^n, t^n) + \gamma \Delta t \mathcal{L}(q, t^n + \eta \Delta t) \tag{4.11}$$

where, in order to obtain consistency through second-order, $\beta = 1 - \gamma$ and $\alpha = \eta = 1/2\gamma$. Given assumed constants M and L related to $\mathcal{L}$ and the time-derivatives of $\mathcal{L}$, the following error equation is obtained

$$|\epsilon| \leq \frac{h^3}{6} 2ML^2 C_\gamma \tag{4.12}$$

where

$$C_\gamma = 2\left|\frac{3}{4\gamma} - 1\right| + 1. \tag{4.13}$$

A minimum truncation error scheme is obtained for $\gamma = 3/4$. Details can be found in section 2.3 of Gear's text [11].

An SSP scheme can be written as convex combinations of Euler steps. To examine the SSP properties of a Runge-Kutta scheme we write the method as a non-unique convex combination of Euler steps and then optimize on the free coefficients to demonstrate the optimal stability result. Given an $\mathcal{L}$ dependent only on u we obtain

$$q = u^n + \frac{1}{2\gamma} \Delta t \mathcal{L}(u^n) \tag{4.14}$$

$$u^{n+1} = (1-\delta)(u^n + \frac{(1-\gamma) - \delta/(2\gamma)}{1-\delta} \Delta t \mathcal{L}(u^n)) + \delta(q + \frac{\gamma}{\delta} \Delta t \mathcal{L}(q)) \tag{4.15}$$

The SSP stability coefficient c is then

$$c = \min(2\gamma, \frac{1-\delta}{1-\gamma-\delta/(2\gamma)}, \delta/\gamma) \tag{4.16}$$

for $0 \leq \gamma, \delta \leq 1$. We wish to make c as large as possible by optimizing over δ while keeping non-negative coefficients. Then the optimal δ is

$$\delta = 2\gamma(1-\gamma) \tag{4.17}$$

and thus

$$c = \min(2\gamma, 2(1-\gamma)) \tag{4.18}$$

Name	γ	c	C_γ
GS non-TVD example (GSRK2)	-1/40	-	63
SSPRK22	1/2	1	2
minimal truncation error (MTRK2)	3/4	1/2	1
midpoint RK2 (RK2)	1	0	1.5

Table 4.1: SSP stability coefficients and truncation constants for various RK2 methods.

Name	c	c/2	cp/s	cp/(2s)
SSPRK22	1	1/2	1	1/2
SSPRK32	2	1	4/3	2/3
SSPRK42	3	3/2	3/2	3/4

Table 4.2: Limiting SSP CFL factors and effective CFL numbers.

We give several methods and their respective c coefficients. If $\gamma = 1/2$, then $c = 1$ and this method is the optimal SSP second-order method. It is also known as Huen's method or the trapezoidal rule RK2 method. Gottlieb and Shu (GS) give an example, to demonstrate the importance of SSP methods, with $\gamma = -1/40$. In terms of truncation error estimates it is clear that the GS non-TVD example is far out of normal practice for RK2 schemes. It has a truncation error coefficient 63 times larger than the optimal truncation error scheme. The optimal truncation error method is SSP stable at an SSP CFL factor of 1/2 but also has a truncation error estimate twice as small as the optimal SSP method.

We experimentally checked the stability characteristics of all implemented time integrators by running the Sod shock tube problem at varying CFL numbers. Results of this test are displayed in Figures 4.1 and 4.3(a). The SSP stability limits are also shown where applicable for each method. We find that, although the central schemes we employ have a theoretical CFL limit of $\frac{1}{2}$, the practical limit is not sharp. In fact, it is apparently possible to run near a CFL number of 1, although this results in slight degradation of the solution. This does not appear to be an effect of the RK time integrators, since Euler time integration shows the same effect. Taking this into account, the multistage SSP RK methods with extra steps seem to behave as expected, providing the additional stability characteristics. Namely, the methods with one extra step are stable to a CFL of two, and the method with two extra steps is stable up to a CFL of three. Figures 4.1 and 4.3 show the stability of the various time integrators vs. effective CFL number. It is clear that there is a significant benefit to using the $s > p$ integrators which improve the

effective CFL number significantly above 1/2.

To further evaluate the practical differences between SSP and non-SSP time integrators, we reproduced the numerical test originally used to demonstrate the value of SSP integrators in [12]. This scalar problem involves the solution to Burger's equation with initial data

$$u(x,0) = \begin{cases} u(x,0) = 1.0 & 0 < x < 0.5, \\ u(x,0) = -0.5 & 0.5 < x < 1. \end{cases} \tag{4.19}$$

The solution changes over time only in that the shock propagates to the right with velocity $v = 0.25$. Using GSRK2 time integration, we found that very severe oscillations, resulting in a diverging solution, occur at CFL numbers greater than about 0.3. Therefore, all results shown are run at CFL 0.3. Figures 4.2(a) and 4.2(b) show snapshots of the solution at time $t = 0.125$ for various second-order RK time integrators. As expected, we find a significant overshoot using the non-SSP integrator used in [12] and no such overshoot in solutions using various SSP integrators. Remarkably, neither of the other non-SSP integrators yield overshoots, even though these runs employed a CFL beyond the theoretical SSP limit of the midpoint and minimal truncation methods. We also calculated the TV norm

$$TV(u^n) = \sum_j |u_j^n - u_{j-1}^n| \tag{4.20}$$

at each time step for each method, and with the exception of GSRK2, all integrators gave strictly TVD results, i.e.

$$TV(\bar{u}^n) \leq TV(\bar{u}^0) \tag{4.21}$$

for all time steps n.

4.2 Third-Order Time Integrators

We have previously described several third-order Runge-Kutta time integrators. These include an optimal three stage SSP method, an optimal four stage SSP method, a minimal truncation error method, a 2 stage low storage SSP method and a classical non-SSP low storage method.

Table 4.3 gives the SSP stability factors for each of these methods. These numbers are computed by optimizing the SSP stability factors at each RK stage. Figures 4.3(a) and 4.3 show the experimental results. We note that even though the SSP stability factors are well below the practical limit, all the integrators are essentially equivalent up to a CFL of 1/2 which is the expected practical limit. The SSPRK43 scheme provides what appears to be

Name	c	c/2	cp/s	cp/(2s)
SSPRK33	1	1/2	1	.5
SSPRK43	2	1	3/2	3/4
RK3	0	0	0	0
LSSSP33	0.322349	0.322349/2	0.322349	0.322349/2
WLSRK33	-	-	-	-

Table 4.3: Limiting SSP CFL factors and effective CFL numbers for third-order schemes.

a reasonable solution up to the expected limit. There is no apparent reason to prefer the SSP LS scheme over the Williamson LS scheme.

Third-order LS schemes can be analyzed for SSP stability factors by comparing each RK step with a general convex combination of Euler steps. For the general two stage third-order LS scheme one obtains the relations

$$c_1 = \frac{1}{B_1} \tag{4.22}$$

$$c_2 = \min\left(\frac{1-\alpha_1}{B_2A_2+(1-\alpha_1)B_1}, \frac{\alpha_1}{B_2}\right) \tag{4.23}$$

$$c_3 = \min(\frac{1-\alpha_2-\alpha_3}{(1-\alpha_2-\alpha_3)B_1+A_2d}, \frac{\alpha_2}{d}, \frac{\alpha_3}{B_3}) \tag{4.24}$$

$$d = (1-\alpha_3)B_2 + B_3A_3) \tag{4.25}$$

$$c = \min(c1, c2, c3) \tag{4.26}$$

with the α_k as free parameters and all entries must be non-negative. For the SSP LS scheme $c_1 = 1.08158$ and maximizing with respect to α_1 on the c_2 equation yields

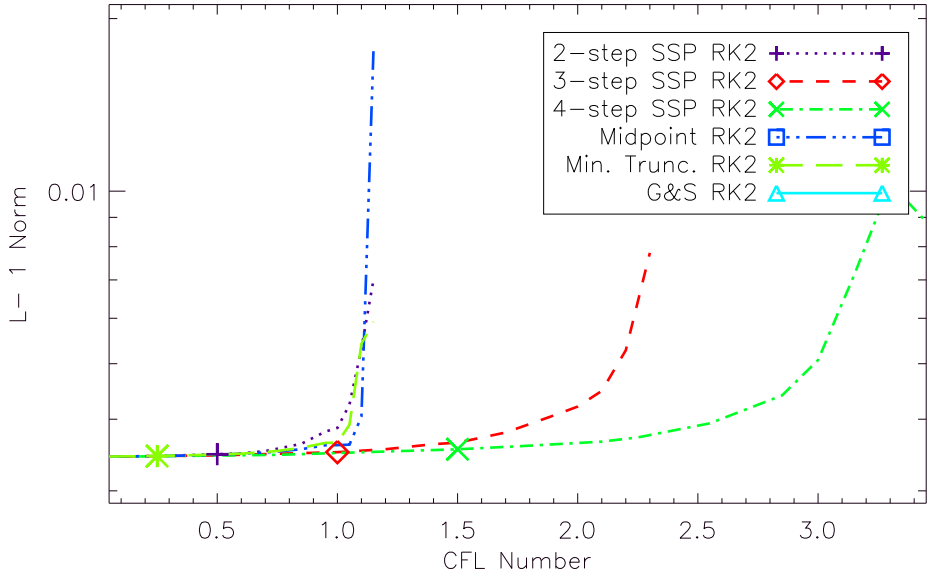
$$c_2 = \frac{B_1 + B_2A_2}{B_1B_2} = 0.322349. \tag{4.27}$$

Choosing α_2 and α_3 to match this c_2 in the second and third entries of the c_3 minimum equation gives a first entry 1.45986 which is larger than c_2. Thus $c_2 = .322349$ is the SSP CFL factor as claimed for the method. For Williamson's scheme it is impossible to chose an α_2 to give positive coefficients so it is a non-SSP scheme. A similar analysis for the third-order minimal truncation error scheme shows that it is SSP at $c = 0$.

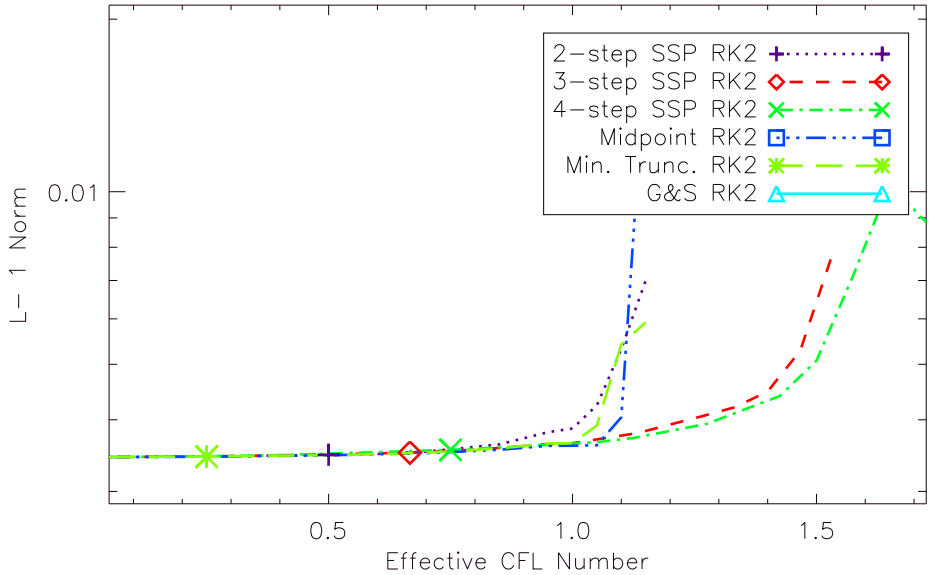
4.3 Summary of SSP Results

It appears that the second and third order SSP methods in the literature for $s > p$ clearly provide a useful enhanced stability region as predicted by the

analysis. However, non-SSP and non-optimal SSP schemes with SSP factors less than one can still be very useful if used in regions below the standard physical CFL limit. The well designed integration schemes we have examined appear to perform well in all instances where the time step is maintained below the standard physical CFL limit. These include low storage schemes.

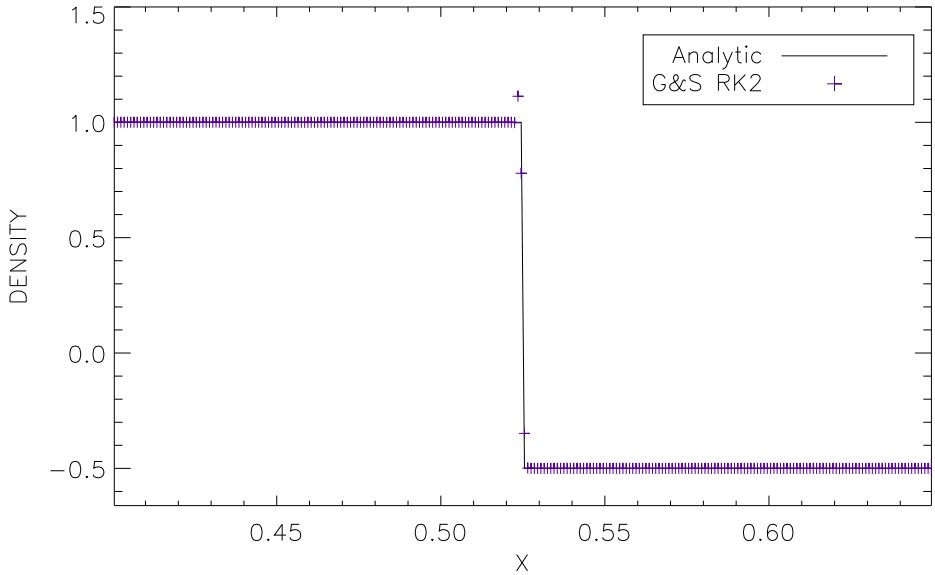


(a) L^1 error as a function of the CFL number.

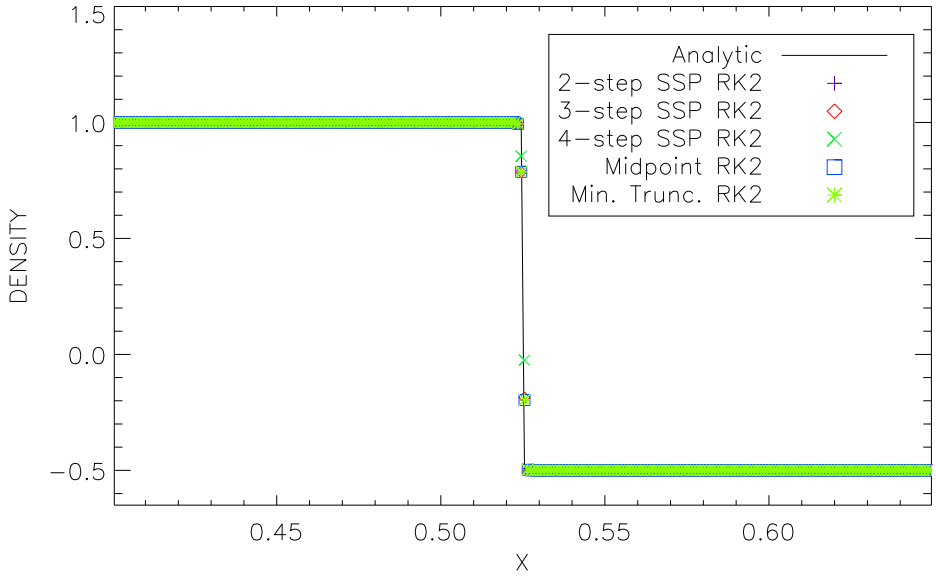


(b) L^1 error as a function of the effective CFL number.

Figure 4.1: Stability of various second-order time integration methods plotted versus CFL and effective CFL. Symbols indicate theoretical SSP limits for each method.

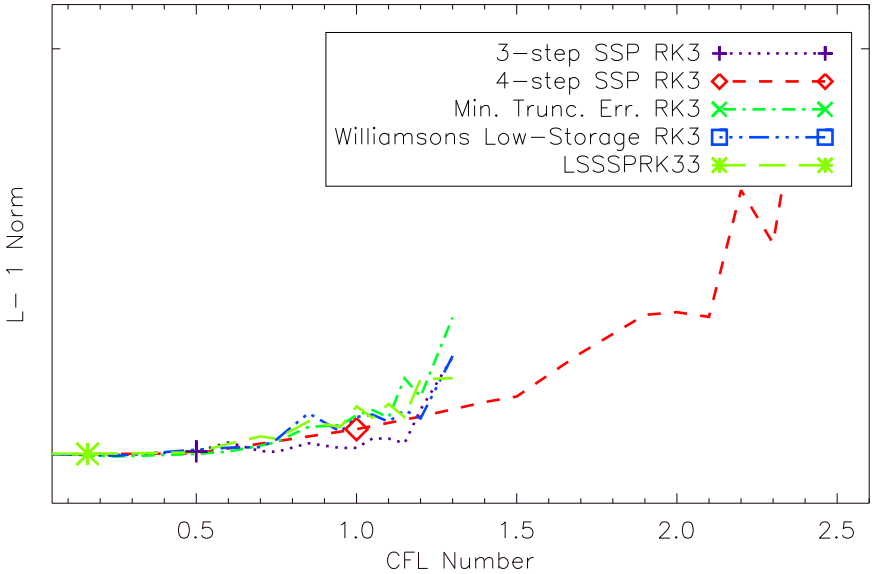


(a) Gottlieb & Shu's RK2.

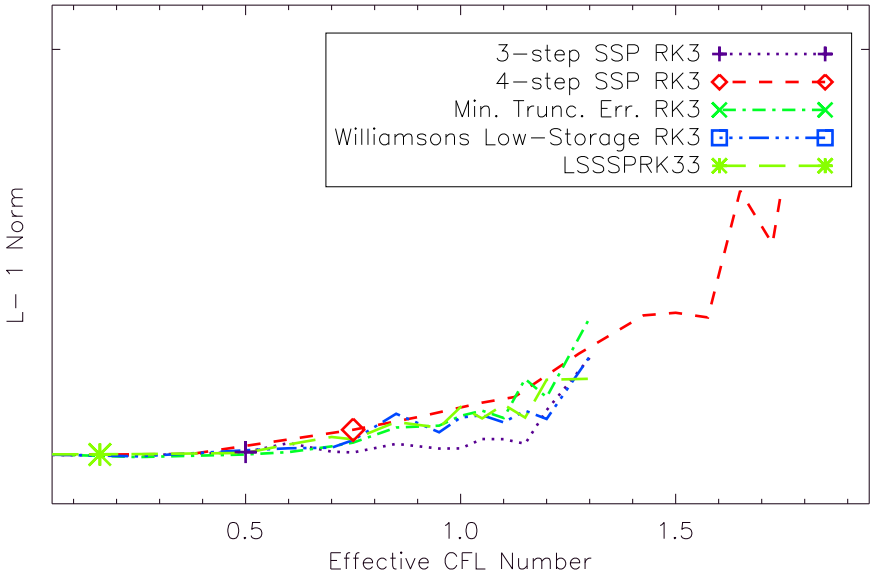


(b) Several other second-order RK methods.

Figure 4.2: Solutions using various SSP and non-SSP time integrators.



(a) L^1 error as a function of the CFL number.



(b) L^1 error as a function of the effective CFL number.

Figure 4.3: Stability of various 3rd-order time integration methods plotted versus CFL and effective CFL. Symbols indicate theoretical SSP limits for each method.

Chapter 5

Summary

In this work, we have constructed a generalized conservation law framework and the concomitant software framework based on Godunov-type central schemes for hyperbolic conservation laws. The generalized conservation law approach appears not only to be feasible, but also to result in significant advantages over existing arbitrary Lagrangian-Eulerian (ALE) technologies in current use at Sandia for shock hydrodynamics calculations. The primary advantages appear to be a relatively efficient design for good machine utilization and, more importantly, a significant separation of the solution algorithm from the physics. The advantages in terms of allowing for future options for implementing improved underlying algorithms are significant. We have been able to easily implement a variety of methods without difficulty while still maintaining a separation from specific physics descriptions.

In general, the convergence properties of the central schemes were shown to be consistent with the formal order of accuracy for these methods on smooth data. However, it was also demonstrated that the two third-order reconstruction procedures, LT3 and KP3, that were implemented as documented in the literature for use in association with central schemes, are deficient. In specific, due to the limiting procedures used in both third-order reconstruction methods, the spatial accuracy for smooth data for LT3 may be reduced to $O(\Delta x^2)$ and for KP3 may be reduced to $O(\Delta x)$.

While the algorithmic simplicity and computational efficiency of the central schemes is appealing, the numerical studies we have conducted indicate that they can not be considered sufficient for all problems. This is particularly true for problems that involve rotational flows, or where the details of contact discontinuities are important. Central schemes do provide a "black-box" framework that permits rapid implementation of new physics making them the "first" method to try on new problems. However, for many Sandia-centric applications, we anticipate that more refined algorithms will be desired.

When combined with central semi-discrete methods, the second and third

order SSP time integration methods as described in the literature for $s > p$ clearly provide a useful enhanced stability region as predicted by the analysis. However, well-designed non-SSP and non-optimal SSP schemes with SSP factors less than one appear completely acceptable if used in regions below the standard CFL limit. These include low storage schemes.

Future Directions

The work presented here constitutes a first step in the development of a generalized conservation law framework for problems that involve significant hyperbolic character. The strengths and weaknesses of the Godunov-type central schemes have made it clear that there are a number of key issues that need to be addressed in future efforts. Based on our most recent results, our future efforts will address the following issues.

1. Solution methods that make use of more sophisticated flux function evaluations, including exact Riemann solvers and more extensive approximate Riemann solvers in a general framework will be investigated to determine which solution strategies provide the best performance in terms of accuracy, ease of implementation, time-to-solution, and robustness.

2. The disappointing performance of the third-order reconstruction methods suggests that additional third-order schemes be investigated and/or developed in order to fully understand the utility of third order methods. One alternative here is the use of compact WENO (weighted essentially non-oscillatory) schemes.

3. Understanding exactly how best to accommodate more complicated physics models in the context of the conservation law framework will be addressed in terms of the tools required for generating characteristic decompositions of flux Jacobians and specific flux function technologies.

4. We will also consider the treatment of conservation laws with equations with source terms in order to deal with elastic-plastic materials and physics with multiple and disparate time scales, e.g., resistive magnetohydrodynamics.

5. In order to address problems with multiple materials and multiple material interfaces, the issues associated with interface reconstruction vs. multiphase formulations will be pursued.

Bibliography

[1] R. Abgrall and S. Karni, *Computation of compressible multifluids*, Journal of Computational Physics, 169 (2001), pp. 594–623.

[2] M. Arienti, P. Hung, E. Morano, and J. E. Shepard, *A level-set approach to Eulerian-Lagrangian coupling*, Tech. Rep. cit-asci-tr136, California Institute of Technology, 2001.

[3] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri, *Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations*, Applied Numerical Mathematics, 25 (1997), pp. 151–167.

[4] ASCI FLASH Center, *FLASH User's Guide – Version 2.1*, Chicago, IL, June 2002.

[5] T. D. Aslam, *A level-set algorithm for tracking discontinuties in hyperbolic conservation laws*, Journal of Computational Physics, 167 (2001), pp. 413–438.

[6] F. Bianco, G. Puppo, and G. Russo, *High-order central schemes for hyperbolic systems of conservation laws*, SIAM Journal of Scientific Computing, 21 (1999), pp. 294–322.

[7] E. A. Boucheron, K. H. Brown, K. G. Budge, D. E. Carroll, S. K. Carroll, M. A. Christon, R. R. Drake, C. G. Garasi, T. A. Haill, J. S. Peery, S. V. Petney, J. Robbins, A. C. Robinson, R. M. Summers, J. R. Weatherby, and M. K. Wong, *Alegra: User input and physics descriptions, version 4.0*, tech. rep., Sandia National Laboratories, Albuquerque, New Mexico, July 2001.

[8] D. E. Carroll, *The NEVADA framework*, tech. rep., Sandia National Laboratories, Albuquerque, New Mexico, USA, 2002.

[9] R. P. Fedkiw, T. Aslam, and S. Xu, *The ghost fluid method for deflagration and detonation discontinuities*, Journal of Computational Physics, 154 (1999), pp. 393–427.

[10] B. A. Fryxell, E. Muller, and D. Arnett, in Numerical Methods in Astrophysics, P. R. Woodward, ed., Academic, 1989.

[11] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, 1971.

[12] S. Gottlieb and C.-W. Shu, *Total variation diminishing Runge-Kutta schemes*, Mathematics of Computation, 67 (1998), pp. 73–85.

[13] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Review, 43 (2001), pp. 89–112.

[14] G.-S. Jiang, D. Levy, C.-T. Lin, S. Osher, and E. Tadmor, *High resolution nonoscillatory central schemes with nonstaggered grids for hyperbolic conservation laws*, SIAM Journal of Numerical Analysis, 35 (1998), pp. 2147–2168.

[15] C. A. Kennedy, M. H. Carpenter, and R. M. Lewis, *Low-storage, explicit Runge-Kutta schemes for compressible Navier-Stokes equations*, Applied Numerical Mathematics, 35 (2000), pp. 177–219.

[16] C. A. Kennnedy and M. H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations*, to appear in Applied Numerical Mathematics, (2002).

[17] A. Kurganov and D. Levy, *A third order semidiscrete central scheme for conservation laws and convection-diffusion equations*, SIAM Journal of Scientific Computing, 22 (2000), pp. 1461–1488.

[18] A. Kurganov, S. Noelle, and G. Petrova, *Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations*, SIAM Journal of Scientific Computing, 23 (2001), pp. 707–740.

[19] A. Kurganov and G. Petrova, *Central schemes and contact discontinuities*, Mathematical Modelling and Numerical Analysis, 34 (2000), pp. 1259–1275.

[20] ——, *A third-order semi-descrete genuinely multidimensional central scheme for hyperbolic conservation laws and related problems*, Numerische Mathematic, 88 (2001), pp. 683–729.

[21] A. Kurganov and E. Tadmor, *New high resolution central schemes for nonlinear conservation laws and convection-diffusion equations*, Journal of Computational Physics, 160 (2000), pp. 241–282.

[22] A. Kurganov and E. Tadmor, *Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers*, submitted to Numerical Methods for Partial Differential Equations, (2002).

[23] P. D. Lax, *Weak solutions of nonlinear hyperbolic equations and their numerical computation*, Communications on Pure and Applied Mathematics, 8 (1954), pp. 159–193.

[24] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhauser Verlag, 1992.

[25] R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambgridge University Press, Cambridge, UK, 2002.

[26] D. Levy, G. Puppo, and G. Russo, *Central WENO schemes for hyperbolic systems of conservation laws*, Mathematical Modelling and Numerical Analysis, 33 (1999), pp. 547–571.

[27] ——, *Compact central WENO schemes for multidimensional conservation laws*, SIAM Journal of Scientific Computing, 22 (2000), pp. 656–672.

[28] K.-A. Lie and S. Noelle, *On the artificial compression method for second-order nonoscillatory central difference schemes for systems of conservations laws*, submitted to SIAM Journal on Scientific Computing, (2001).

[29] T. Linde, *A practical, general-purpose, two-state HLL Riemann solver for hyperbolic conservation laws*, International Journal for Numerical Methods in Fluids, 40 (2002), pp. 291–402.

[30] X.-D. Liu and S. Osher, *Nonoscillatory high order accurate self similar maximum principle satisfying shock capturing schemes*, I. SINUM, 33 (1996), pp. 760–779.

[31] X.-D. Liu and E. Tadmor, *Third order nonoscillatory central scheme for hyperbolic conservation laws*, Numer. Math., 79 (1998), pp. 397–425.

[32] H. Nessyahu and E. Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, Journal of Computational Physics, 87 (1990), pp. 408–463.

[33] W. L. Oberkampf and T. G. Trucano, *Verification and validation in computational fluid dynamics*, Tech. Rep. SAND2002-0529, Sandia National Laboratories, Albuquerque, New Mexico, March 2002.

[34] D. Pathria, *The correct formulation of intermediate boundary conditions for Runga-Kutta time integration of initial boundary value problems*, SIAM Journal of Scientific Computing, 18 (1997), pp. 1255–1266.

[35] G. Puppo, *Numerical entropy production for central schemes*, preprint, (2002).

[36] V. V. Rusanov, *Calculation of interaction of non-steady shock waves with obstacles*, Journal of Computational Mathematical Physics, USSR, 1 (1961), pp. 267–279.

[37] R. Saurel and R. Abgrall, *A multiphase Godunov method for compressible multifluid and multiphase flows*, Journal of Computational Physics, 150 (1999), pp. 425–467.

[38] L.-I. Sedov, *Similarity and Dimensional Methods in Mechanics*, Academic, 1959.

[39] C.-W. Shu, *Chapter4: A survey of strong stabilty preserving high order time discretizations*, in Collected Lectures on the Preservation of Stability under Discretization, D. Estep and S. Tavener, eds., SIAM, 2002.

[40] R. J. Spiteri and S. J. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM J. Num. Anal., 40 (2002), pp. 469–491.

[41] E. Tadmor, *Convenient total variation diminishing conditions for nonlinear difference schemes*, SIAM J. Num. Anal., 25 (1988), pp. 1002–1014.

[42] M. Wesenberg, *A note on MHD Riemann solvers*, tech. rep., Albert-Ludwigs-Universitat Freiburg, 2000.

[43] J. H. Williamson, *Low-storage Runge-Kutta schemes*, Journal of Computational Physics, 35 (1980), pp. 48–56.

[44] P. Woodward and P. Collela, Journal of Computational Physics, 54 (1984), p. 115.

[45] H. C. Yee, M. Vinokur, and M. J. Djomehri, *Entropy splitting and numerical dissipation*, Journal of Computational Physics, 162 (2000), pp. 33–81.

Hardcopy Distribution:

1	MS 0841	T. C. Bickel, 09100
1	MS 0824	A. C. Ratzel, 09110
1	MS 0847	H. S. Morgan, 09120
1	MS 0824	J. L. Moya, 09130
1	MS 0835	J. M. McGlaun, 09140
1	MS 0834	J. E. Johannes, 09114
1	MS 0836	E. S. Hertel, 09116
1	MS 0836	R. O. Griffith, 09117
1	MS 1135	L. A. Gritzo, 09132
1	MS 0828	M. Pilch, 09133
1	MS 0835	S. N. Kempka, 09141
1	MS 0827	J. D. Zepper, 09143
1	MS 0321	W. J. Camp, 09200
1	MS 0310	P. Yarrington, 09230
1	MS 0847	S. Mitchell, 09211
1	MS 1110	D. Womble, 9214
1	MS 0819	R. Summers, 09231
1	MS 0820	P. Chavez, 09232
1	MS 0316	S. Dosanjh, 09233
1	MS 0316	J. Aidun, 9235
1	MS 1186	T. A. Mehlhorn, 01674
5	MS 0819	M. Christon, 09231
5	MS 0819	D. I. Ketcheson, 09231
5	MS 0819	A. Robinson, 09231
1	MS 9042	C. D. Moen, 08728
1	MS 0819	T. G. Trucano, 09231
1	MS 1111	B. Hendrickson, 09226
1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 09616

E-mail Distribution:

1	MS 1110	P. Bochev, 09214
1	MS 1110	M. Heroux, 09214
1	MS 1110	R. Lehoucq, 09214
1	MS 0819	K. Brown, 09231
1	MS 0819	S. Burns, 09231
1	MS 0819	D. Carroll, 09231
1	MS 0819	R. Drake, 09231
1	MS 0819	C. Garasi, 09231
1	MS 0819	D. Hensinger, 09231
1	MS 0819	J. Robbins, 09231
1	MS 0819	T. Voth, 09231
1	MS 0819	M. Wong, 09231
1	MS 0820	T. Bartel, 09232
1	MS 0820	A. Farnsworth, 09232
1	MS 1111	J. Shadid, 09233
1	MS 1111	T. M. Smith, 09233
1	MS 0316	H. Hjalmarson, 9235
1	MS 0836	M. R. Baer, 09100
1	MS 0826	D. K. Gartling, 09100
1	MS 0834	T. A. Baer, 09114
1	MS 0834	M. M. Hopkins, 09114
1	MS 0834	M. J. Martinez, 09114
1	MS 0834	H. K. Moffat, 09114
1	MS 0834	R. R. Rao, 09114
1	MS 0834	P. R. Schunk, 09114
1	MS 0825	F. G. Blottner, 09115
1	MS 0825	B. Hassan, 09115
1	MS 0825	C. J. Roy, 09115
1	MS 0847	S. W. Attaway, 09132
1	MS 0828	B. F. Blackwell, 09133
1	MS 0828	W. L. Oberkampf, 09133
1	MS 0835	S. W. Bova, 09141
1	MS 0847	J. W. Swegle, 09142
1	MS 0827	J. R. Stewart, 09143
1	MS 1186	T. A. Brunner, 01674
1	MS 1186	T. A. Haill, 01674