Abstract of "Parallel ENO Schemes Applied to Shock/Cylinder Interactions & Numerical Methods by Radon Transform," by Carl B. Quillen, Ph.D., Brown University, May 1995

The construction of an implementation of the high-order ENO finite difference method for the computation of compressible reactive supersonic flow with real gas equations of state on the CM5 parallel supercomputer is described. The equation of state is given by $p = \rho RT$, and a polynomial fit is used to describe $C_p$. An empirical model is used for the viscosity parameter $\mu$. The thermal and species diffusion rates are derived from this by use of Prandtl and Schmidt numbers respectively. A single step, non-stiff reaction provides a crude model for hydrogen/oxygen combustion.

The numerical method is applied to the two-dimensional calculation of planar shock waves interacting with cylindrical light gas domains contained in air. The interaction with a combusting Hydrogen gas inclusion is also computed. The results are compared with existing experiments, previous computational results, and with a shock-capturing spectral method.

The results suggest that numerical diffusion may have seriously contaminated the results of previous researchers, and that mixing efficiencies are considerably less than those computed previously. On the other hand, high wave number perturbations of a sharp cylindrical interface are observed to cause intense mixing. This contradicts previous observations from lower resolution simulations.

The high order ENO scheme compares surprisingly well with the spectral method. The comparison is increasingly close as the formal order of accuracy of the scheme and/or spatial resolution is increased. The results confirm that high order methods are essential for accurate simulation of time-varying flow fields.

In the second part of the thesis a basis derived from a singular value decomposition of the Radon transform is used to construct spectral methods for a 2D circular disk. A numerical scheme is also created which consists of a finite difference method operating in Radon space. Both methods are applied to the computation of linear hyperbolic flows. The spectral method has no singularities at the origin, and is seen to perform well when tau boundary conditions are applied.

# Parallel ENO Schemes Applied to Shock/Cylinder Interactions

# &

# Numerical Methods by Radon Transform

by

Carl B. Quillen

A.B., Harvard University, 1987
Sc.M., Brown University, 1995

Thesis

Submitted in partial fulfillment of the requirements for
the Degree of Doctor of Philosophy
in the Division of Applied Mathematics at Brown University

May 1995

This dissertation by Carl B. Quillen is accepted in its present form by
the Division of Applied Mathematics as satisfying the
dissertation requirement for the degree of
Doctor of Philosophy

Date............................ ..........................................................
Chi-Wang Shu

Recommended to the Graduate Council

Date............................ ..........................................................
Wai Sun Don

Date............................ ..........................................................
George Em Karniadakis

Approved by the Graduate Council

Date............................ ..........................................................

The Vita of Carl B. Quillen

Carl Quillen was born March 22, 1965 in Boston Massachusetts.

He received an A.B. degree in June 1987 from Harvard University in Applied Mathematics.

In May 1991, he married Dr. Karen Kong.

In May 1993, he and his wife were delighted by the arrival of their daughter, Deirdre Elaine.

'D.D.' Quillen

Publications:

(with Chi-Wang Shu) *Parallel Implementation and Application of High Order ENO Schemes* in *Modeling and Simulation Volume 23 Part 5*, 1992, William G. Vogt and Marlin H. Mickle, eds.

(with Wai Sun Don) *Numerical simulation of supersonic reactive flow, Part I: Resolution requirement* (submitted to J. Comput. Physics.)

## Acknowledgments

I would like to acknowledge the sincere and determined efforts of my thesis advisor, Professor Chi-Wang Shu, who over years did his best to help me to graduate, often with modest success, but always with good humor. I am grateful for the assistance of Professor Wai Sun Don, who provided help with all aspects of the computational work and physical modeling in the first part of this thesis, and led me daily through this research. I would especially like to thank Jan S. Hesthaven for his advice, which I found extremely useful when I tried to create a spectral method in the second part of this thesis. The extent to which I manage to overcome my profound ignorance of this subject area reflects his assistance. Linus Torvalds deserves thanks for creating the operating system I used for much of my work. Above all I would like to thank my wife for remaining patient with me over the many years I have been a graduate student, and my daughter for managing to understand and forgive all the craziness involved with writing this dissertation.

I would also like to acknowledge the support of DARPA via a graduate fellowship, and support under ARO grants DAAH04-93-G-0329 and DAAH04-94-G-0205, AFOSR grant 93-1-0090, and DARPA grant N00014-91-J-4016. Computational support was provided to me by the Army High Performance Computing Research Center in Minnesota, and the Northeast Parallel Architectures Center in Syracuse.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The construction of numerical methods for non-linear hyperbolic systems of equations is a difficult challenge. Theoretical understanding of even basic properties like existence and uniqueness of solutions is lacking in many cases of interest. It is difficult to use a numerical scheme, when the underlying equations may not have physically meaningful solutions.

Yet in many cases, it is possible to have confidence that the equations have solutions, especially if one is interested in a problem of engineering significance. For example, the equations of gas dynamics are used for engineering in many situations involving fluid flows. They are applied for designing airplanes and turbines, and generally in any area where compressible flow with low viscosities is encountered. It is possible to dream up situations where these equations break down. For example consider the one-dimensional problem in figure 1.1 where initially a gas with uniform density $\rho$, pressure and temperature traveling with velocity $-u$ is at the left of the origin, and the same gas is present at the right traveling at velocity $+u$. A rarefaction will develop at the origin. Certainly if $|u|$ is greater than the sound speed, a vacuum will form there as well. Theoretically, this is not an easy situation to analyze—the continuum hypothesis underlying the derivation of the Euler equations of gas dynamics will break down in the vacuum, and a complete understanding may require use of the Boltzmann equation. The theoretical description of gas dynamics is complicated by this difficulty—quite simple flow conditions may produce situations with zero density.[1]

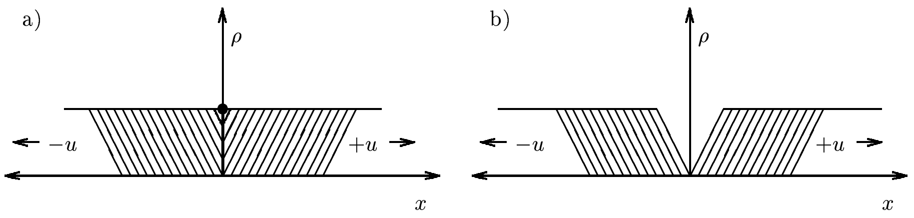


Figure 1.1: A uniform initial condition with velocity discontinuity at time $t = 0$ (a) and $t = \epsilon$ (b)

Quite obviously, the example cited above is not a flow that one can expect to see often in nature. In fact, it wouldn't be easy to make this situation occur if one tried. In cases arising from practical engineering problems, the physical existence of flow solutions is usually not a surprise: these flows occur in existing machinery. Where theoretical understanding is currently useful is in providing intuition for the construction of viable computational methods.

Let me begin by summarizing briefly the situations where the problem is theoretically well understood. Certainly linear hyperbolic systems of equations with constant coefficients are completely understood. Nonlinear scalar conservation laws

$$u_t + \sum_i f_i(u)_{x_i} = 0,$$

are well understood, even for non-convex fluxes. Solutions to this equation exist, provided one is willing to admit the existence of discontinuities (shocks). Unique solutions exist provided one makes use of an 'entropy' criterion. This entropy is physically relevant, and implies that the result is stable to small perturbations.

For nonlinear systems of equations the situation is much more bleak. Systems of two equations in one space variable

$$u_t + f(u, v)_x \quad = \quad 0$$

[1]On the other hand, the Boltzmann equation itself, which describes the evolution of the probability distribution function for the velocities of gas molecules, is better understood theoretically (see Diperna and Lions[12]).

$$v_t + g(u,v)_x = 0,$$

are the one exception. Examples of this would be adiabatic or isentropic flow in one dimension. Again an entropy criterion is available which permits the selection of a unique solution.

For systems of more than two equations and for multi-dimensional equations, much less is known in general. In the case with one spatial dimension the Glimm scheme provides a constructive proof for the existence of solutions for genuinely nonlinear systems of hyperbolic equations. However, it does not provide any hint as to the uniqueness of solutions, and it does not suggest an entropy to provide this.

Because the main area of interest in this thesis is the two-dimensional system of equations for gas dynamics, there currently is very little way of constructing a numerical scheme that has complete theoretical justification. The only way of doing this currently is a direct numerical simulation of the Boltzmann equation. Even in one dimension this is computationally demanding. In more than one dimension it is completely impractical.

The approach that most researchers have taken is to first construct schemes for the problems that are understood—scalar conservation laws. The theoretical analysis developed to understand these P.D.E's then assists in proving that numerical methods converge to the correct, unique solution in these simple cases. The intuition gained from this analysis is then extremely useful when the numerical methods are extended to systems in more than one dimension. Certainly a numerical method that performs badly in the scalar conservation law case will be a poor choice for calculations in more complicated problems. Verifying that the resulting numerical schemes actually work correctly is then an exercise of generating confidence as they are applied to more and more complicated physical situations and compared with other numerical schemes and experimental results.

This is the approach that has been taken for the ENO schemes that I use in the first part of this thesis. They are designed for one-dimensional scalar laws, then extended to more complicated problems. The extension to systems of conservation laws occurs here by using a linearization to approximately diagonalize the system locally. The extension to more than one spatial dimension then can happen in several different ways.

One way to split the spatial operator into $n$ partial time steps, where $n$ is the number of spatial dimensions. For example $u_t + f(u)_x + g(u)_y = 0$ can be advanced in time by solving for one partial time step

$$u_t + f(u)_x = 0 \tag{1.1}$$

and for the next partial step

$$u_t + g(u)_y = 0. \tag{1.2}$$

If this is done correctly the result is a scheme with at most second order accuracy in time. An additional advantage is that if the scheme can be proven to work for (1.1) and (1.2), and certain commutativity requirements are met, then the split operator scheme above converges for the multi-dimensional problem.

Other ways for extending into more than one dimension depend more on the details of the specific scheme. The particular ENO schemes used here work essentially by calculating the derivatives $\frac{\partial}{\partial x} f(u)$ and $\frac{\partial}{\partial y} f(u)$ in fancy way that works well for hyperbolic equations. These derivatives are computed from conservative fluxes

$$\frac{\partial}{\partial x} f(u(x_j, y_k)) \approx \frac{\hat{f}_{j+1/2,k} - \hat{f}_{j-1/2,k}}{\Delta x}$$

$$\frac{\partial}{\partial y} g(u(x_j, y_k)) \approx \frac{\hat{g}_{j,k+1/2} - \hat{g}_{j,k-1/2}}{\Delta y}$$

A special version of the standard Runge-Kutta ODE solver algorithms may then be used to advance the semi-discrete equation

$$\frac{\partial}{\partial t} u(x_j, y_k) = -\frac{\hat{f}_{j+1/2,k} - \hat{f}_{j-1/2,k}}{\Delta x} - \frac{\hat{g}_{j,k+1/2} - \hat{g}_{j,k-1/2}}{\Delta y}$$

in time. The advantage of doing this over splitting the spatial operator is that higher order time-stepping methods are available. Because the concern here will be a time-varying problem where the method will be advanced in time tens of thousands of time steps, high order time-stepping accuracy will be quite useful. Unfortunately, this cannot assist a proof of convergence like the split operator method above.

This methodology of extending schemes for scalar conservation laws to more complicated systems has produced useful schemes that work reasonably well in practice, something that is demonstrated in considerable detail here. But for most problems of engineering interest, the numerical methods in use are only partially justified by theory. The existing numerical analysis does provide qualitative insights. A reasonable question to ask is whether one can get away with constructing numerical methods that bypass the one-dimensional problems that are understood theoretically. This is a question that I try to address one aspect of in the second part of this thesis. In particular, I take as a starting point a perspective that is inherently multi-dimensional—there really no way using the same viewpoint in one spatial dimension. In several ways, the approach taken modifies the traditional theoretical framework. While the attempts I describe here are only a beginning, I hope that these efforts may inspire others to examine the traditional theory, which has changed little since the seminal paper of Crandall and Majda on monotone schemes[10]. Their approach has been very successful, and has been pushed to dizzying heights by a number of talented researchers. It may be time to examine the foundations of the subject, to see what possibilities have been overlooked.

## 1.1 The existing theoretical framework

Let us begin by briefly summarizing the theory suggested by Crandall and Majda[10], which has been applied to monotone schemes, then extended to TVD[20] and TVB[44] methods. This can be summarized in several steps:

1. Use a weak formulation to the problem $u_t + f(u)_x = 0$. This is the first approach to finding a solution of the P.D.E. which may contain discontinuities. The problem is that a smooth initial condition may become discontinuous within finite time. The P.D.E. is an equation containing

derivatives, and as such ceases to have meaning in this situation. The way to resolve this problem is to convert the P.D.E. to an integral equation

$$\frac{\partial}{\partial t}\int_a^b u\,dx = f(a) - f(b) \tag{1.3}$$

for an interval $(a, b)$. If this integral equation holds for all intervals, then it implies the P.D.E. for smooth solutions $u$. The integral equation also has meaningful solutions with discontinuities when no continuous solution exists, and the quantity $u$ will be conserved. This is the right formulation to use when $u$ is a conserved quantity like mass or energy in the physical system being modeled. The disadvantage of this formulation is that it may admit more than one solution. An entropy condition must be found to select out the unique physically correct one.

2. The numerical scheme is constructed in conservative form. For example

$$\frac{\partial}{\partial t}u(x_j) = -\frac{\hat{f}_{j+1/2} - \hat{f}_{j-1/2}}{\Delta x}. \tag{1.4}$$

The flux $\hat{f}_{j+1/2}$ is calculated from the values of $u$ at grid points $x_k$, for a set of values k close to j. $\hat{f}$ is called consistent if when applied to a uniform state $u(x_j) = C \quad \forall j$, $\hat{f} = f(C)$. This is a nice property, because a theorem by Lax and Wendroff shows that if a computation in conservative form based on a consistent continuous numerical flux $\hat{f}$ converges, it converges to the weak solution to the P.D.E. The proof is easy, and is accomplished by performing a discrete sum of (1.4) over an interval. The interior values of $\hat{f}$ cancel, leaving just the boundary terms, which converge to the physical flux $f$ by consistency and continuity as the grid is refined.

3. A discrete form of the entropy condition in item 1 above is then used to demonstrate that if the scheme converges, it converges to the physically correct solution.

4. The above is everything that we need, except a guarantee that the numerical scheme will converge to something as the grid is refined. This is often referred to as a stability requirement, although it is somewhat stronger than what is meant by classical numerical stability. The

approach taken is to observe that for an initial condition of bounded total variation (TV), the analytical solution always has less total variation at later times. If the numerical fluxes are Lipschitz continuous, then the set of numerical solutions at different grid resolutions is pre-compact and has a convergent subsequence. The fact that the numerical scheme is continuous implies that all subsequences converge to the same thing—the correct answer.

The compactness implied by bounded TV is exploited in the Monotone, TVD and TVB methods to prove convergence. In each case, less severe restrictions are imposed upon the numerical method to achieve bounded TV of the solution. With less severe restrictions comes increased accuracy. Monotone schemes, for example, are only first order accurate. In one dimension, TVD schemes degenerate to first order accuracy at critical points, but are of high-order accuracy elsewhere. In two dimensions, they are only first order accurate.[17] TVB schemes are of uniform high order accuracy.

ENO schemes themselves don't fit into the theoretical framework described above. The numerical fluxes are not even continuous! If it were known that the solution that they converge to is piecewise continuous, then it can be shown that they are TVB. But it has not been proven that ENO schemes do in fact converge. What can be said is that in many numerical experiments, ENO schemes do seem to converge, and they converge to the right thing.

## 1.2 Conservation laws via Radon transform

Armed with the example of ENO schemes, which are motivated by the classical theoretical framework, but don't fit in it, it begins to seem more reasonable to try something radically different. The approach tried here is borrow an idea from Tomography and to make use of the Radon transform, which is defined in the following way:

$$\mathcal{R}: L_1(\mathbb{R}^n) \to L_1(\mathbb{R}\times S^{n-1})$$

where

$$\mathcal{R}: f(x) \mapsto \int_{\mathbb{R}^n} f(x)\delta(p - x\cdot\xi)\, dx.$$

The Radon transform $\mathcal{R}f(p,\xi)$ consists of the set of the integrals of $f$ over lines $l_{p,\xi}$ at distance $p$ from the origin and perpendicular to the direction vector $\xi$ (see figure 1.2). Here I describe it as a mapping from $L_1$ to $L_1$, but it may be extended to $L_2$, and even Sobolev spaces if the norms are defined in the right way in Radon Space.

The key point about the Radon transform is that it is invertible. In other words, if all the line integrals of a function are known, then the pointwise values of the original function may be found. This is useful in medical imaging for example because line integrals of certain densities in a human body may be calculated by passing radiation through the body and measuring how much is transmitted through to the other side. The pointwise values of these densities may be computed, and an image of the interior of the body printed out. This is a much better way of doing things than carving up somebody to see what is going on inside, which was, unfortunately, the only alternative in some cases before MRI and CAT scans were invented.

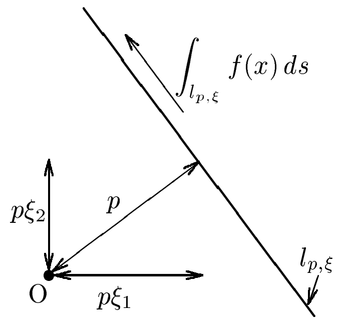

Figure 1.2

What interests us here about the Radon transform is that it has some interesting consequences for the weak formulation of a conservation law. Take for example the multi-dimensional counterpart of the weak formulation in equation (1.3):

$$\frac{\partial}{\partial t}\int_X u(x)dx = -\int_{\partial X} \vec{f}(u)\cdot\vec{n}ds, \tag{1.5}$$

which is a weak formulation of the multi-dimensional scalar conservation law

$$u_t + \nabla\cdot\vec{f}(u) = 0.$$

Clearly, if the integral (1.5) holds for a smooth function $u$ for enough sets $X$, then it implies that $u$ satisfies the P.D.E. Now one might ask what collection of sets $X$ is actually necessary. One might guess that a collection that can generate the measurable sets would be required. The Radon transform implies that a smaller collection will do, if the solution $u$ and $\vec{f}(u)$ satisfy certain integrability conditions. For example, the sets $S_\epsilon = \{x : p - \epsilon \leq x{\cdot}\xi \leq p + \epsilon\}$ will work. This is clear, because using $S_\epsilon$ for $X$ in (1.5) results in

$$\frac{1}{2\epsilon}\frac{\partial}{\partial t}\int_{S_\epsilon} u(x)\,dx = -\frac{1}{2\epsilon}\int_{\partial S_\epsilon} \vec{f}(u){\cdot}\vec{n}\,ds = -\frac{1}{2\epsilon}\left(\xi{\cdot}(\mathcal{R}\vec{f})(p+\epsilon,\xi) - \xi{\cdot}(\mathcal{R}\vec{f})(p-\epsilon,\xi)\right)$$

Taking the limit of list as $\epsilon \downarrow 0$ and by using the linearity of the Radon transform we arrive at the equation

$$\mathcal{R}(u_t) = \frac{\partial}{\partial t}\mathcal{R}u = -\xi{\cdot}\frac{\partial}{\partial p}(\mathcal{R}\vec{f})(p,\xi). \tag{1.6}$$

It can be seen without much difficulty that if the inverse of the Radon transform is applied to both sides of this equation, then the result is the original P.D.E. For this to work, obviously, $\mathcal{R}(u_t)$ and $\mathcal{R}f_i(u)_{x_i}$ will have to be defined, and thus the integrability conditions on $u$ and $\vec{f}(u)$.

The interesting thing about equation (1.6) is that it can be applied to construct a numerical scheme. In particular, it suggests a new, multi-dimensional 'conservation form' for a numerical scheme operating in Radon space:

$$\frac{\partial}{\partial t}(\mathcal{R}u)(p_j,\xi) = -\frac{\hat{f}_{j+1/2}(\xi) - \hat{f}_{j-1/2}(\xi)}{\Delta p}. \tag{1.7}$$

This looks very much like the one-dimensional formulation in (1.4). The difference will be that the numerical fluxes $\hat{f}_{j+1/2}(\xi)$ necessarily depend in a global fashion on the coordinate direction $\xi$. They will also need to satisfy a rather different consistency requirement than before, as they are computed from $\mathcal{R}(\xi \cdot \vec{f})$ not $f$.

The second part of this thesis addresses the construction of a numerical scheme using the formulation in equation (1.7). The approach taken is to use the underlying ideas behind the ENO method to construct a multi-dimensional scheme. Along the way, I also encounter some good bases for implementing spectral schemes on disks in two and higher dimensions. While I am not able to answer all the questions arising from finite difference implementations of (1.7), these spectral schemes do work quite well for the simple problems I apply them to.

A question I do not address is how close one may come within this formulation to finding a parallel to the standard theoretical framework listed above. One might ask if it were possible to construct numerical methods with a TVB property in Radon space. For example, one might measure the total variations along each coordinate direction $\xi$ in Radon space, and then take the maximum of these variations. If this is a bounded quantity, then Tychonoff's theorem implies compactness in Radon space. One might ask whether this criterion is incompatible with higher order schemes. But this seems to be an unimportant detail, because it would be difficult to check this condition in a numerical scheme. The total variation in one coordinate direction depends on the total variation in all the other directions, and this will be reflected in any numerical scheme. To see that this is true, consider a rotating flow in two dimensions

$$u_t - yu_x + xu_y = 0.$$

The solution rotates in time. Total variation that is seen in one coordinate direction, will be seen (hopefully) unchanged in other coordinate directions at later times.

# Chapter 2

# A Parallel ENO code

Current parallel machines provide a computational ability one to two orders of magnitude greater than conventional vector machines. Increasingly, they are based upon standard microprocessor chips, which seem to be doubling in capability every one to to two years. Conventional supercomputers have a generation life-cycle much longer than this, and there is little about this picture that can be expected to change in the next decade. The future is clear. In order to attack the largest, most complex problems possible, computational techniques and codes have to be converted to these parallel machines.

As vastly increased computational ability becomes available, more complex physical problems become approachable. For example, much past work with CFD has concentrated on solving stationary flow problems. These problems are typically solved using a guess for the solution as an initial condition for a flow solver which marches in time until the computation converges to an equilibrium. A low-order accurate flow solver and low order time stepping provide reasonable accuracy for these situations. The error in the end solution will reflect only the spatial approximation error in the numerical scheme, and second order accuracy often provides completely adequate results at low computational cost.

With improved computational ability, one is tempted to try and attack more complicated flow fields that are inherently evolving with time. Higher order methods become attractive if the method is to be advanced in time a substantial number of time steps. This is because the nature of truncation error has important consequences for an evolving flow field. Low order schemes tend to have large artificial numerical viscosities, which can smooth away small scale features. Dispersive errors can cause substantial phase errors in the flow as well. For multi-dimensional flows, the anisotropic nature of the substantial truncation error can also introduce odd artifacts in the flow, and activate fluid instabilities.

Thus the motivation for using high-order schemes is actually rather subtle. A comparison of the magnitude of truncation errors may not necessarily show them to be at an obvious advantage. One might argue that a substantial part of the flow-field may contain discontinuities and will be resolved to first order only. Lower order schemes are much less expensive to implement for a given resolution, so if memory is available one may compute with finer grids for a given amount of computational resources. But it appears in practice that the high order methods when implemented just produce a model with inherently more realistic physics. They are able to resolve features in the flow that low order methods cannot at any practical resolution and their results are easier to interpret and compare better with experiment.

The fact that the ENO finite difference method is a family of schemes with varying numerical accuracy is particularly valuable in this regard. It is easy to perform a comparison of results computed from different ENO schemes with different formal accuracy.

One of the main disadvantages that ENO has on standard vector computers is that it does not vectorize well at all. The basic element in the ENO method is a finite difference stencil that can vary from grid point to grid point. At each partial time step in the method, the stencil used can change at each grid point. The result is an irregular access of data interspersed with logical calculations at the core of the algorithm.

The CM5 parallel computer provides hardware support for two very important elements needed by the ENO algorithm. The CM5 is able to vectorize logical operations. It also has hardware support for vectorized indexing of arrays. The memory architecture of the existing CM5 machines is not staggered, and so very little penalty is paid for localized irregular memory access. Thus the core operations in ENO can be vectorized trivially in an efficient way on this machine.

The result is that the ENO algorithm easily achieves a quarter of the peak speed of the CM5, when measuring elapsed time for the entire program. In contrast, on a Cray YMP performance is seen to be less than a tenth of peak.

In the following I describe the implementation of the ENO algorithm to solve the two-dimensional compressible Navier Stokes equations with multiple reacting chemical species and simple, non-stiff chemical terms. The problems of interest are high speed gas dynamic flows with low viscosity and real equations of state. These flows are predominantly characterized at large scales by convective phenomena, and so an ENO method may be used to discretize the Euler fluxes. Centered differences may then be used to compute the viscous terms.

## 2.1 The Physical Model

The physics of the problem are described by the two dimensional Navier-Stokes equations in conservative form, which can be written

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = \mathbf{V}(\mathbf{U})_x + \mathbf{W}(\mathbf{U})_y + \mathbf{C}(\mathbf{U}). \tag{2.1}$$

The terms $\mathbf{V}$ and $\mathbf{W}$ on the right side of this equation contain all the diffusion in the problem and $\mathbf{C}$ is the chemical source term. Thus the inviscid Euler equations are obtained by dropping these terms. The state vector $\mathbf{U}$ is written as

$$\mathbf{U} = (\rho\,\mathbf{f}, \rho, \rho u, \rho v, \rho E)^t\,.$$

The fluid density is $\rho$, $\mathbf{u} = (u, v)^t$ is the fluid velocity, $E$ is the specific total energy, and $\mathbf{f}$ is the vector of mass fractions $(f_1 \cdots f_{N_{s-1}})$ for $N_{s-1}$ chemical species. There are $N_s$ species in the problem, but the mass fraction of the last species $f_{N_s}$ may be derived from that of the first $N_{s-1}$ by applying the principle of conservation of mass:

$$\sum_{i=1}^{N_s} f_i = 1.$$

Each species $i$ drifts at a velocity $\tilde{\mathbf{u}} = (\tilde{u}_i, \tilde{v}_i)^t$ relative to the mean mixture velocity $\mathbf{u}$.

In terms of these variables and the pressure $p$, the Euler fluxes are given by

$$\mathbf{F} = \begin{pmatrix} (u+\tilde{u}_1)\rho f_1 \\ \vdots \\ (u+\tilde{u}_{N_{s-1}})\rho f_{N_{s-1}} \\ u\rho \\ u\rho u + p \\ u\rho v \\ u(\rho E + p) \end{pmatrix} \qquad \mathbf{G} = \begin{pmatrix} (v+\tilde{v}_1)\rho f_1 \\ \vdots \\ (v+\tilde{v}_{N_{s-1}})\rho f_{N_{s-1}} \\ v\rho \\ v\rho u \\ v\rho v + p \\ v(\rho E + p) \end{pmatrix}$$

The Stokes assumption implies that the elements of the viscous stress tensor take the form

$$\tau_{xx} = \frac{4}{3}u_x - \frac{2}{3}v_y$$

$$\tau_{yy} = \frac{4}{3}v_y - \frac{2}{3}u_x$$

$$\tau_{xy} = \tau_{yx} = u_y + v_x,$$

and in terms of these the viscous terms are

$$\mathbf{V} = \mu \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \tau_{xx} \\ \tau_{yx} \\ \sigma_1 \end{pmatrix} \qquad \mathbf{W} = \mu \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \tau_{xy} \\ \tau_{yy} \\ \sigma_2 \end{pmatrix}$$

where $\mu$ is the mixture viscosity. A Prandtl number $P_r$ is used to relate fluid viscosity and thermal diffusion, and a Schmidt number $S_c$ relates viscosity and molecular diffusion, so in terms of the temperature $T$ and the mixture specific heat at constant pressure $\bar{C}_p$ and the species specific enthalpy $h_i$ we have

$$\sigma_1 = u\tau_{xx} + v\tau_{xy} + \frac{\bar{C}_p T_x}{P_r} - \rho \sum_{i=1}^{N_s} h_i \tilde{u}_i f_i / \mu$$

$$\sigma_2 = v\tau_{yy} + u\tau_{yx} + \frac{\bar{C}_p T_y}{P_r} - \rho \sum_{i=1}^{N_s} h_i \tilde{v}_i f_i / \mu.$$

$$\tilde{\mathbf{u}} = \frac{\mu}{\rho S_c} \nabla \mathbf{f}.$$

### 2.1.1 Thermodynamic Model

The high temperatures produced by combustion imply that the perfect gas equation of state is a reasonable approximation:

$$p = \rho \bar{R} T.$$

$\bar{R}$ is a mixture gas constant computed by weighting the perfect gas constant $R$ by the appropriate average of the species molecular weights $M_i$:

$$\bar{R} = R \sum_{i=1}^{N_s} f_i / M_i.$$

The specific total internal energy is given by

$$E = \int_0^T \bar{C}_p(t)\, dt - \frac{p}{\rho} + \frac{u^2}{2} + \frac{v^2}{2} + \sum_{i=1}^{N_s} f_i h_i^0$$

where $h_i^0$ is the reference enthalpy of species $i$ and $\bar{C}_p$ is a mixture specific heat at constant pressure. It is computed from the individual species specific heats $C_{p_i}$ by

$$\bar{C}_p = \sum_{i=1}^{N_s} C_{p_i} f_i / M_i$$

$C_{p_i}$ is approximated by a fourth order polynomial in temperature, and thus its integral may be obtained trivially. The coefficients for this approximation were obtained from[33]. The specific species enthalpy $h_i$ above may be obtained similarly:

$$h_i = h_i^0 + \int_0^T C_{p_i}(t)\,dt.$$

### 2.1.2 Diffusion Models

Sutherland's Law provides an individual species viscosity

$$\frac{\mu_i}{\mu_0} = \left(\frac{T}{T_0}\right)^{3/2}\left(\frac{T_0+S}{T+S}\right)$$

$\mu_0, T_0$ and $S$ are species dependent constants, and are tabulated by Svehla[50]. The mixture viscosity may then be obtained by Wilke's Law[55], which is an empirical model:

$$\mu = \sum_{i=1}^{N_s} \frac{\mu_i X_i}{\sum_{j=1}^{N_s} X_j \phi_{ij}}$$

$$\phi_{ij} = \frac{\left\{1 + [(\mu_i/\mu_j)(f_j/f_i)]^{\frac{1}{2}}\,(M_i/M_j)^{\frac{1}{4}}\right\}^2}{[8(1+(M_i/M_j))]^{\frac{1}{2}}}$$

where $X_i = f_i/M_i$ is the mole fraction of species $i$.

### 2.1.3 Chemical Model

The chemical reaction is modeled by a one step reaction and four chemical species ($N_s = 4$):

$$2\mathrm{H}_2 + \mathrm{O}_2 \Leftrightarrow 2\mathrm{H}_2\mathrm{O}.$$

The fourth species is $\mathrm{N}_2$, and is considered inert. A Modified Arrhenius Law gives the reaction rate constant $k_f$

$$k_f = 5.5410{\times}10^{14}\,\exp\left(\frac{-3625}{T}\right)$$

The reverse reaction rate $k_b$ is derived from an equilibrium constant $k_{eq}$

$$k_{eq} = 82.06\,T \exp(4.606(\frac{12925}{T} - 2.915))$$

$$k_b = k_f / k_{eq}.$$

The species production rate $w$ may then be written

$$\dot{w} = k_f [\mathrm{H}_2]^2 [\mathrm{O}_2] - k_b [\mathrm{H}_2\mathrm{O}]^2.$$

If we identify species 1 with $H_2$ species 2 with $O_2$ and species 3 with $H_2O$ then we may write the chemical source term **C** as

$$C = (-2\dot{w}M_1, -\dot{w}M_2, 2\dot{w}M_3, 0, 0, 0, 0)^t\,.$$

## 2.2 A quick summary of the ENO scheme

The ENO schemes of Osher and Shu are a class of high order finite difference schemes in conservative form. We summarize briefly here the ideas behind the discretization. Consider the model equation

$$u_t = -g(u(x))_x\,, \tag{2.2}$$

which is a one-dimensional scalar conservation law. We may discretize the spatial derivative in the following way:

$$u_t(x_j) = \frac{\hat{g}(x_{j-1/2}) - \hat{g}(x_{j+1/2})}{\Delta x}. \tag{2.3}$$

Here we assume a regular grid of points $x_j$. The grid spacing is $\Delta x = x_{j+1} - x_j$. $\hat{g}(x_{j+1/2})$ is a consistent low order approximation to the flux $g(u(x))$ evaluated at a point midway between $x_j$ and $x_{j+1}$. In flux based ENO schemes, this approximation is cleverly arranged so that $\frac{1}{\Delta x}[\hat{g}(x_{j-1/2}) - \hat{g}(x_{j+1/2})]$ is actually a high order approximation to the derivative $-g(u)_x$. This may be accomplished in the following way. Let

$$P(x_{j+1/2}) = \sum_{k \le j} g(u(x_k)). \tag{2.4}$$

This defines $P(x)$ on the points $x_{j+1/2}$. Given a desired order $s_0$ of approximation of the spatial discretization, a stencil of points is chosen $\{x_{s+1/2}, x_{s+3/2}, \ldots, x_{s+s_0+1/2}\}$ consisting of $s_0 + 1$ grid points. We may then compute a polynomial interpolant of $P$. The obvious way is to use a Lagrange polynomial interpolant

$$P(x) = \sum_{k=s}^{s+s_0} \prod_{\substack{l=s \\ l \ne k}}^{s+s_0} \left( \frac{x - x_{l+1/2}}{x_{k+1/2} - x_{l+1/2}} \right) P(x_{k+1/2}). \tag{2.5}$$

Then $\hat{g}(x_{j+1/2}) = \Delta x \frac{\partial P(x)}{\partial x}|_{x_{j+1/2}}$ will yield a high order approximation to the derivative in equation (2.2):

$$\frac{\partial g(u)}{\partial x}|_{x=x_j} = \frac{\hat{g}(x_{j-1/2}) - \hat{g}(x_{j+1/2})}{\Delta x} + O(\Delta x^{s_0}). \tag{2.6}$$

This can be explained in the following way: suppose a function $h(x)$ exists which is a deconvolution of $g$

$$g(x) = \frac{1}{\Delta x} \int_{x-\Delta x/2}^{x+\Delta x/2} h(\zeta)\, d\zeta. \tag{2.7}$$

Then $g'(x_j) = \frac{1}{\Delta x}[h(x_{j+1/2}) - h(x_{j-1/2})]$ exactly.

So we need only find a high order approximation to $h$, and this will serve for $\hat{g}$. Now

$$\frac{1}{\Delta x} \int_{-\infty}^{x_{j+1/2}} h(\zeta)\, d\zeta = \sum_{k \le j} \frac{1}{\Delta x} \int_{x_{k-1/2}}^{x_{k+1/2}} h(\zeta)\, d\zeta = \sum_{k \le j} g(x_j). \tag{2.8}$$

so by (2.4) $\int_{-\infty}^{x} h(\zeta)\, d\zeta = \Delta x P(x)$, and $\Delta x P'(x) = h(x)$. Therefore $\Delta x$ times a high order approximation to the derivative of $P(x)$ will serve for $\hat{g}$. The solution is to interpolate $P$ with a polynomial of order $s_0$. Then $\hat{g}$ will be approximate $h$ to order $s_0 + 1$ and the difference in (2.3) will be accurate to order $s_0$.

### 2.2.1 Choosing a stencil

Missing in the above discussion is a method for choosing a stencil of points $\{x_{s+1/2}, \ldots, x_{s+s_0+1/2}\}$ to be applied. In fact, the stability of the numerical scheme is determined by this choice. For example, it is known that a consistent bad choice of the stencil will yield a linearly unstable scheme. A method that has been seems to work well in practice is to construct the stencil recursively by comparing magnitudes of undivided difference coefficients.

The undivided difference coefficients $D[x_{s-1/2}, \ldots, x_{s+n+1/2}]$ are defined inductively in the following way:

$$\begin{aligned} D[x_{s-1/2}, x_{s+1/2}] &= g(x_s) \\ D[x_{s-1/2}, \ldots, x_{s+n+1/2}] &= D[x_{s+1/2}, \ldots, x_{s+n+1/2}] - D[x_{s-1/2}, \ldots, x_{s+n-1/2}]. \end{aligned} \tag{2.9}$$

Stencil construction in an ENO scheme works in the following way: If a stencil

$$\{x_{s+1/2}, \ldots, x_{s+n+1/2}\}$$

is chosen for the scheme of order $n$, then either the stencil $\{x_{s-1/2}, \ldots, x_{s+n+1/2}\}$ or

$$\{x_{s+1/2}, \ldots, x_{s+n+3/2}\}$$

are possible choices for the stencil of the scheme of order $n + 1$. The choice between stencils is made depending upon the magnitude of the difference coefficients. The first stencil is chosen if its associated difference coefficient is less in magnitude that the difference coefficient associated with the second stencil:

$$\left|D[x_{s-1/2}, \ldots, x_{s+n+1/2}]\right| < \left|D[x_{s+1/2}, \ldots, x_{s+n+3/2}]\right|. \tag{2.10}$$

A minor modification of this algorithm has proven to work better in practice[40]. The idea is to preferentially choose stencils that are closer to being centered differences by biasing the comparison in (2.10) by a multiplied constant. That is the approach used in this paper. The choice of the initial stencil for the zeroth order scheme is determined by upwinding considerations and is described below.

### 2.2.2 Upwinding

In order to achieve stability the numerical scheme must take into account the natural propagation of waves in the problem. Doing this well in practice is a difficult exercise for a nonlinear system of equations. We begin by considering the scalar equation 2.2 again. In this paper we apply a version of the Lax-Friedrich flux splitting:

$$g(u) = g^{+}(u) + g^{-}(u) \tag{2.11}$$

where $g^{\pm}(u) = \frac{1}{2}\left(g(u) \pm \alpha u\right)$, and $\alpha$ is a wave speed, taken as $\max|g'(u)|$ for a global Lax-Friedrich splitting, for a less diffusive flux $\alpha(x_{j+1/2}) = \max_{u_j}^{u_{j+1/2}} |g'(u)|$. Compromises between these two extremes are also possible, where a maximum is computed locally over some number of grid points.

$\hat{g}^{+}(x_{j+1/2})$ and $\hat{g}^{-}(x_{j+1/2})$ are calculated separately in the same manner that $\hat{g}(x_j)$ was above. The initial starting stencil is taken as $\{x_{j-1/2}, x_{j+1/2}\}$ for $\hat{g}^{+}$ and $\{x_{j+1/2}, x_{j+3/2}\}$ for $\hat{g}^{-}$. For $\hat{g}^{\pm}$, $P^{\pm}(x_j) = \sum_{k \leq j} g^{\pm}(x_k)$ is interpolated, instead of $P(x_j)$ as before. Having calculated $\hat{g}^{\pm}$, we revise our definition of $\hat{g}$,

$$\hat{g} = \hat{g}^{+} + \hat{g}^{-}.$$

It is this quantity that is actually used in equation (2.3) to discretize the equation (2.2).

### 2.2.3 Systems

A linear system of equations

$$\mathbf{U}_t = -A\mathbf{U}_x \tag{2.12}$$

may be discretized by diagonalizing the system, and then solving for each component as before. For example, if $A$ can be written $A = L^{-1}\Lambda L$ Where $\Lambda$ is diagonal, then $L\mathbf{U}_t = -\Lambda L\mathbf{U}_x$. Define a new state vector $\mathbf{Y} = L\mathbf{U}$. then in terms of this new parameterization

$$\mathbf{Y}_t = -\Lambda \mathbf{Y}_x.$$

The system is completely decoupled, and each component may be handled independently as before.

This approach may be applied locally in nonlinear systems to approximately decouple them as well. Suppose we desire a spatial discretization of the system

$$\mathbf{U}_t = -\mathbf{F}(\mathbf{U})_x = -A(\mathbf{U})\mathbf{U}_x$$

of the form

$$\mathbf{U}_t(x_j) = \frac{1}{\Delta x}(\hat{\mathbf{F}}(x_{j-1/2}) - \hat{\mathbf{F}}(x_{j+1/2})).$$

In order to compute $\hat{\mathbf{F}}$ at each half grid point $x_{j+1/2}$ we construct a different flux splitting

$$\mathbf{F}(x) = L^{-1}(x_{j+1/2})(\mathbf{F}^+(x) + \mathbf{F}^-(x))$$

where

$$\mathbf{F}^{\pm}(x) = L(x_{j+1/2})\left(\mathbf{F}(\mathbf{U}(x)) \pm \bar{\Lambda}(x_{j+1/2})\mathbf{U}(x)\right).$$

$L^{-1}(x)$, $\Lambda(x)$ and $L(x)$ are derived from the diagonalization of a matrix $A(x)$ as with the linear case. The question is, what matrix $A$ to use at the half grid points $x_{j+1/2}$. Now $A(x_j) = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}}|_{U(x_j)}$. We need some average of $A(x_j)$ and $A(x_{j+1})$ to use for $A(x_{j+1/2})$. There is more than one possible way of taking this average, but for the purposes of this paper, adequate performance is obtained by a simple average:

$$A(x_{j+1/2}) = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}}|_{\frac{U(x_j)+U(x_{j+1})}{2}}.$$

$\bar{\Lambda}$ Is a diagonal matrix of maximum wave speeds in the problem. It may be taken as the global maximum of $|\Lambda_{ii}|$ where $\Lambda$ is the matrix of eigenvalues of $A$. Or some localized maximum may be used.

### 2.2.4 Boundary Conditions

Two types of boundary are implemented in the current code. In both cases ghost points outside the computational domain are employed. For example, consider the boundary $x = 0$. The computational domain consists of grid points $(x_j, y_k)$ for $j \geq 0$ and less than some maximum, and $y$ in some bounded range as well. The ghost grid points $(x_{-1}, y_k), \cdots, (x_{-s_0-1}, y_k)$ are computed from the known data.

For reflecting boundary conditions, set $\mathbf{U}(x_{-l}, y_k) = \mathbf{U}(x_{l-1}, y_k)$ for $l = 1, \ldots, s_0$, for all elements of the state vector $\mathbf{U}$ except the $y$ momentum $\rho v$. The sign of this element is inverted.

For non-adsorbing boundary conditions a characteristic projection may be used. Given $\mathbf{U}(x_0)$ and a free stream boundary condition $\mathbf{U}_\infty$ and the Jacobian $A = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} = L^{-1}\Lambda L$ evaluated at some intermediate state $\mathbf{U}$, we project to characteristic variables

$$\mathbf{Y}^0 = L\mathbf{U}(x_0)$$

and

$$\mathbf{Y}^\infty = L\mathbf{U}_\infty.$$

Then if $\Lambda_{ii} < 0$ set $Y_i \rightarrow \mathbf{Y}_i^0$ otherwise set $Y_i \rightarrow \mathbf{Y}_i^\infty$. Then set

$$\mathbf{U}(x_{-1}) \rightarrow L^{-1}\mathbf{Y}.$$

$\mathbf{U}(x_{-s_0-1}), \ldots, \mathbf{U}(x_{-2})$ may be set equal to $\mathbf{U}(x_{-1})$.

### 2.2.5 ENO Applied to a Real Gas in 2-D

In this section, we derive the Jacobian for the 2-D ENO fluxes. In order to discretize (2.1) the approach taken in this paper is to discretize the inviscid left side of the equation using the ENO methodology in each space dimension separately. For example, to discretize

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = 0. \tag{2.13}$$

We calculate $\mathbf{F}(\mathbf{U})_x$ with $y$ held fixed as if solving the system $\mathbf{U}_t+\mathbf{F}(\mathbf{U})_x$, and then calculate $\mathbf{G}(\mathbf{U})_y$ with $x$ held fixed.

The alternative to doing this would be to use Strang time-splitting, which effectively involves solving the equations $\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0$ and $\mathbf{U}_t + \mathbf{G}(\mathbf{U})_y = 0$ in two partial time steps. This time-splitting has better theoretical justification than just applying ENO derivatives to each coordinate direction separately. But the numerical order of the resulting scheme is at most second order in time. Here the major concern is constructing a high-order method for the purpose of tracking a time-varying flow over several tens of thousands of time steps. A high-order in time seems essential.

The right hand side of (2.1) may be computed in a straightforward way using central differences.

The remaining obstacle is obtaining diagonalization of the Jacobians $\frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}}$ and $\frac{\partial \mathbf{G}(\mathbf{U})}{\partial \mathbf{U}}$. The Diagonalization for $\frac{\partial \mathbf{G}(\mathbf{U})}{\partial \mathbf{U}}$ may be obtained from that of $\frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}}$ by a suitable relabeling of variables and row and column exchanges in the equations, due to the isotropic nature of the fluxes in the problem.

With some difficultly we may obtain

$$\frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} = L^{-1}\Lambda L$$

with

$$L^{-1} = \begin{pmatrix} I - \mathbf{f}\otimes {\mathbf{b}_3}^t & \mathbf{f} & \mathbf{f} & 0 & \mathbf{f} \\ -{\mathbf{b}_3}^t & 1 & 1 & 0 & 1 \\ -u\,{\mathbf{b}_3}^t & u-c & u & 0 & u+c \\ -v\,{\mathbf{b}_3}^t & v & v & 1 & v \\ -H\,{\mathbf{b}_3}^t & H-cu & H-1/b_1 & v & H+cu \end{pmatrix}$$

$$L = \begin{pmatrix} I & -\mathbf{f} & 0 & 0 & 0 \\ \frac{1}{2}{\mathbf{b}_3}^t & \frac{b_2c+u}{2c} & -\frac{1+b_1uc}{2c} & -\frac{b_1v}{2} & \frac{b_1}{2} \\ 0\cdots 0 & -\mathbf{b}_3\cdot\mathbf{f}^t - b_2 + 1 & b_1u & b_1v & -b_1 \\ 0\cdots 0 & -v & 0 & 1 & 0 \\ \frac{1}{2}{\mathbf{b}_3}^t & \frac{b_2c-u}{2c} & \frac{1-b_1uc}{2c} & -\frac{b_1v}{2} & \frac{b_1}{2} \end{pmatrix}$$

and

$$
\Lambda = \begin{pmatrix} u & & & & & & \\ & \ddots & & & & & \\ & & u & & & & \\ & & & u-c & & & \\ & & & & u & & \\ & & & & & u & \\ & & & & & & u+c \end{pmatrix}.
$$

Here $\otimes$ is the usual matrix product, and $c$ is the sound speed, given by

$$
c^2 = \frac{p}{\rho^2}\frac{\partial p}{\partial e} + \frac{\partial p}{\partial \rho}
$$

The total specific enthalpy $H = \frac{p}{\rho} + E$, and

$$
b_1 = \frac{1}{\rho c^2}\frac{\partial p}{\partial e}
$$

$$
b_{3i} = \frac{1}{\rho c^2}\frac{\partial p}{\partial f_i}
$$

$$
b_2 = \frac{1}{\rho c^2}\left(\frac{\partial p}{\partial e}(\frac{u^2+v^2}{2} - e) + \rho\frac{\partial p}{\partial \rho} - \sum_{i=1}^{N_s-1} f_i \frac{\partial p}{\partial f_i}\right)
$$

Using the thermodynamic model presented in section 2.1.1, computing the partial derivatives of $p$ is straightforward:

$$
p = \left(\sum_{i=1}^{N_s-1}\frac{f_i}{M_i} + \frac{1-\sum_{i=1}^{N_s-1} f_i}{M_{N_s}}\right)\rho R T(e, \mathbf{f})
$$

so

$$
\frac{\partial p}{\partial f_i} = \left(\frac{1}{M_i} - \frac{1}{M_{N_s}}\right)\rho R T + \frac{p}{T}\frac{\partial T}{\partial f_i}.
$$

Now

$$
e(T, f_i) = \sum_{i=1}^{N_s-1} f_i \left(\int^T \frac{1}{M_i}(C_{p\,i}(T) - R)dT + h_i^0\right) + (1 - \sum_{i=1}^{N_s-1} f_i)\int^T \frac{1}{M_{N_s}}(C_{p\,N_s}(T) - R)dT.
$$

and $de = \frac{\partial e}{\partial T} dT + \frac{\partial e}{\partial f_i} df_i$, so

$$\frac{\partial T}{\partial f_i}|_e = -\frac{\partial e/\partial f_i}{\partial e/\partial T}$$

$$\frac{\partial e}{\partial f_i} = \int^T \left( \frac{C_{p\,i}}{M_i} - \frac{C_{p\,N_s}}{M_{N_s}} \right) dT + h_i^0$$

$$\frac{\partial e}{\partial T} = \bar{C}_p - \bar{R}.$$

Thus

$$\frac{\partial p}{\partial f_i} = \left( \frac{1}{M_i} - \frac{1}{M_{N_s}} \right) \rho R T - \frac{p}{T(\bar{C}_p - \bar{R})} \frac{\partial e}{\partial f_i}.$$

Calculating $\frac{\partial p}{\partial e}$ is somewhat easier.

$$\frac{\partial p}{\partial e} = \rho \bar{R} \frac{\partial T}{\partial e} = \rho \bar{R} / \frac{\partial e}{\partial T} = \frac{\rho \bar{R}}{\bar{C}_p - \bar{R}}.$$

And simplest of all

$$\frac{\partial p}{\partial \rho} = \frac{p}{\rho}.$$

Due to the fact that $u$ is a repeated eigenvalue in $\Lambda$ the above diagonalization is not unique. The above choice is designed to produce a matrix $L$ that is as sparse as possible. This has the effect of reducing the number of operations carried out in the inner loop of the computational kernel, and is the most obvious way of reducing computational cost of the procedure.

## 2.3 Time stepping

The previous discussion has concentrated on finding a discretization of the spatial derivatives in (2.1). The result is a system of ODE's that must be solved to advance in time. The third order TVD Runge-Kutta ODE solver suggested by Shu and Osher[45] is used for this. It is a simple ODE solver that preserves the TVD property in a a scheme that is TVD if advanced by the Euler method.

$$\mathbf{U}^* = \mathbf{U}^n + \Delta t L(\mathbf{U}^n)$$

$$
\begin{aligned}
\mathbf{U}^{**} &= \frac{1}{4}(3\mathbf{U}^n + \mathbf{U}^* + \Delta t L(\mathbf{U}^*)) \\
\mathbf{U}^{n+1} &= \frac{1}{3}(\mathbf{U}^n + 2\mathbf{U}^{**} + 2\Delta t L(\mathbf{U}^{**}))
\end{aligned} \tag{2.14}
$$

Here $\mathbf{U}^n$ at time step $n$ is advanced one time step to $\mathbf{U}^{n+1}$, $t \leftarrow t + \Delta t$, and the two intermediate time steps are denoted by $\mathbf{U}^*$ and $\mathbf{U}^{**}$. $L$ is the spatial derivative operator. This method is easy to apply and requires only two additional levels of storage.

### 2.3.1 Numerical difficulties with the approach

Several criticisms could be made of the methods chosen to simulate the physical model. There are two different problems that are known to occur with the straight-forward approach used. One pertains to the fact that I use a conservative discretization for the individual chemical species mass fractions. Another problem has been observed to occur with chemical source terms. Both of these problems seem to not significantly affect the flows studied below, but this remains a concern for using the code, especially in other settings.

There are two fundamental difficulties with using a conservative formulation for each species. One is that density $\rho_i$ of each species $i$ is limited by $0 \leq \rho_i \leq \rho$, and situations where the density is at one of the extremes of its range are quite common. If a situation occurs in a flow where there is an interface across which a transition is made to either one of these extremes, then for a method that is not TVD, a (small) overshoot and undershoot in $\rho_i$ will arise at the transition as it propagates in the flow. At certain grid points $\rho_i$ may become negative. At others, it could rise above the total fluid density, but this is less likely in practice.

Undershoots also occur with the total density $\rho$, but these are usually less serious, as $\rho$ is usually sufficiently positive to prevent negative densities from occurring.

Negative densities can cause bad things to happen if they appear. The most unpleasant events are produced when they influence the O.D.E's responsible for computing the effects of chemistry ($L_c$ above). These O.D.E's are guaranteed to work for positive concentrations of reactants but typically will blow up when they see a negative reactant concentration. One can avoid this by lying to the chemistry and never reporting negative concentrations to it. This works fairly well, but does not seem to solve all problems, especially for high order schemes at high spatial resolutions.

There are a couple of different ways to prevent negative densities, both of which involve abandoning the conservative formulation to some extent. This is not as serious as it might seem, because the total density $\rho$ will remain conserved. Effectively what happens is that the flow solver introduces some chemistry of its own to the problem in areas where there is a significant conservation error. If conservation error can be controlled, and the kind of chemistry that occurs isn't too unpleasant, then the situation may be quite reasonable. It is quite possible to control the kind of extra chemistry that occurs through various tricks. Wilson and Sussman[56] describe some techniques in this regard that they use to avoid having elements transmuted by conservation error.

One approach to avoiding negative densities is to clip them when they are outside the permissible range. This produces a loss of conservation at undershoots. This is a simple approach, and it has the advantage that the location of the conservation error is easy to see. This is what is done here.

Another approach that has much to recommend it is a suggestion of Sussman and Wilson.[49] The idea is to replace the flow equation for the species $\rho_i$ to an equivalent equation in terms of a non-conserved quantity. This proceeds in the following way. In 1-D the element of the system 2.1

$$\frac{\partial}{\partial t}\rho_i + \frac{\partial}{\partial x}(u\rho_i)$$

can be reduced to a equation in $f_i$

$$\frac{\partial}{\partial t}f_i + u\frac{\partial}{\partial x}f_i$$

by using $\rho_i = \rho f_i$ and subtracting $\rho_t + (\rho u)_x$. Now this is equivalent to the equation

$$\frac{\partial}{\partial t}\ln f_i + u\frac{\partial}{\partial x}\ln f_i. \tag{2.15}$$

The species mass fraction $f_i$ is limited in range by $0 \leq f_i \leq 1$, but the mapped variable $\ln f_i$ has no lower bound. A different mapping could be used to eliminate the upper range restriction, but as this problem is encountered less often it may not be necessary. Sussman and Wilson[49] suggest the logarithmic mapping because species concentrations tend to fall off exponentially behind reaction fronts. This kind of decay is difficult to resolve well in a scheme based on polynomial interpolation, and the logarithmic mapping eliminates the problem. The disadvantage in using equation (2.15) is that it is not easy to discretize this in a way that conserves $\rho_i$.

Karni[24] has suggested another reason for abandoning the conservative formulation in flows with multiple species. One problem that a conservative formulation may have is that the pressure $p$ must be derived from the conserved quantities. At a contact discontinuity the pressure is continuous, but flow species concentrations are not. This results in discontinuities in computed pressures, especially pressures used to calculate numerical fluxes at half grid points. These perturbations in the effective pressure field produced by the numerics may be expected to force the flow locally, introducing noise into the solution. But as Ton[52] observed, for the particular ENO schemes considered here, the problem isn't anywhere as severe as suggested by Karni. Why is not clear, but the characteristic projection is probably helping the situation. Ton observes that the initial condition chosen by Karni to demonstrate the method implicitly assumes a difference in temperature across the contact discontinuity. With a continuous temperature, the perturbations driven by the discontinuities are barely visible to the eye in the same test problem. With the same discontinuity in temperature, the pictures are somewhat worse, but the errors are probably an order of magnitude less than demonstrated by in the conservative method used by Karni.

The initial conditions are (from Ton[52]): on the left

$$\rho = 1, p = 1, u = 0, \gamma = 1.4, R = 287, f_0 = 1, f_1 = 0,$$

on the right

$$\rho = 0.125, p = 0.1, u = 0, \gamma = 1.2, R = 299.6, f_0 = 0, f_1 = 1.$$

With these conditions the temperature $T = p/(\rho RT)$ is the same on both sides. In figure 2.1 the results of this problem are shown. Also shown are the results with $R = 287(1.2 - 1)/(1.4 - 1)$ on

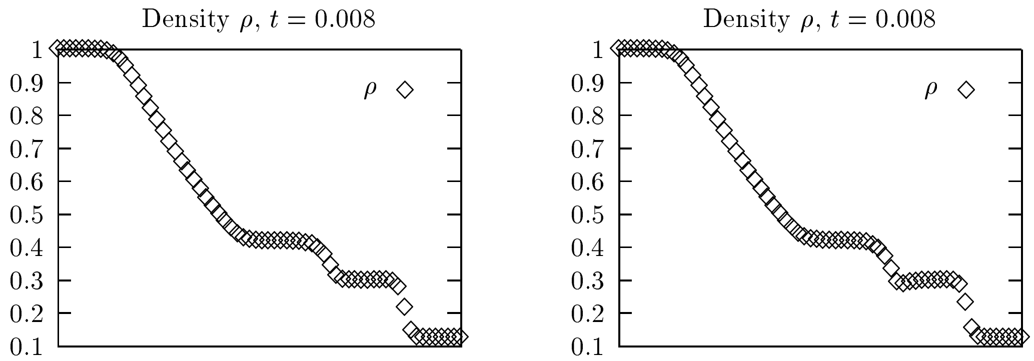


Figure 2.1: Multi-species shock tube problem. Initial condition had (left) uniform $T$, (right) uniform $C_v$.

the right. Because $C_v = R/(\gamma - 1)$, this results in the same $C_v$ on the left and right, which is closer to Karni's initial condition.

When the numerical method described here is applied to a real flow field, the results also seem to indicate that a conservative method does not unduly perturb contact discontinuities. As will be described below, irregularities in the magnitude of the effective numerical diffusion can easily produce much larger perturbations in the flow field than the pressure errors described above seem to introduce. One should note however, that the current method does not apply any method to sharpen contact discontinuities, and the contact discontinuities do not remain particularly discontinuous for long. If sub-cell resolution or artificial compression were applied to avoid this smoothing pressure errors would be much more important, and Karni's suggestions might prove quite valuable.

The other known difficulty that the current code can encounter is related to the chemical source term $\mathbf{C}(\mathbf{U})$ in 2.1. In some situations where the chemical O.D.E's are stiff, this source term can cause a spurious numerical wave to propagate out from a detonation wave at a speed proportional to the spacing between grid points. Leveque and Yee[27] attribute this to the fact that the discontinuity at the detonation wave is resolved in a shock capturing scheme across two or more grid points. The chemical reaction can initiate early at these grid points, and if it does, a numerical detonation wave of reduced amplitude will propagate out ahead of the physical detonation wave at a speed close to one grid point per time step. This problem is more severe for time-split schemes where chemistry and fluid flow are solved in separate partial time steps, but it has been observed in schemes without time splitting.

An analysis of when this phenomena can be expected to occur is provided by Colella *et al.*[9], and Engquist and Sjorgreen[15] provide a simple suggestion to modify a numerical scheme to render it resistant to this failing. The idea is to detect the detonation wave transition, and then modify the temperature at the grid points located within the transition so as to suppress any reaction there.

Because the current numerical scheme is being applied to a situation where reactants are not pre-mixed, combustion rates are controlled by diffusion, and this potential failing does not manifest itself. No detonation waves are to be found anywhere in the flow, and the model chemistry is simple and non-stiff. As the problem becomes apparent in a quite dramatic way when it does occur, producing detonation waves of unexpected speed and low amplitude, it seemed unnecessary to modify the scheme to prevent the possibility.

## 2.4 Implementation and optimization

First let us consider the case of the ENO method implemented on a single rectangular computational domain. For this explicit scheme, parallelization may be accomplished by implementing essentially the same method on each processing node of a parallel machine for a rectangular sub-domain of the global computational mesh. The issues related to this are discussed in §2.5 below.

### 2.4.1 Computing $L$

The core of the implementation is contained in the routine to calculate the spatial derivative $L(\mathbf{U})$ from the given values of the state vector $\mathbf{U}$. $L$ is decomposed into four parts

$$L(\mathbf{U}) = L_f(\mathbf{U}) + L_g(\mathbf{U}) + L_v(\mathbf{U}) + L_c(\mathbf{U}).$$

The first two, $L_f(\mathbf{U})$ and $L_g(\mathbf{U})$ are due to Euler fluxes in the $x$ and $y$ direction respectively. The remaining two are represent the contribution of viscosity and chemistry to the Euler fluxes. $L_v$ and

$L_c$ are calculated in separate routines, but $L_f$ and $L_g$ are actually calculated by the same subroutine. This is possible because the physical flux in the problem is isotropic: $\mathbf{F}(\mathbf{U}) = \mathbf{G}(\tilde{\mathbf{U}})$ where $\tilde{\mathbf{U}}$ is obtained from $\mathbf{U}$ by exchanging the $x$ and $y$ momentum elements:

$$\begin{aligned} \mathbf{U} &= (f_0 \ldots f_{N_s}, \rho, \rho u, \rho v, \rho E)^t , \\ \tilde{\mathbf{U}} &= (f_0 \ldots f_{N_s}, \rho, \rho v, \rho u, \rho E)^t . \end{aligned}$$

Thus $L_g(\mathbf{U}) = \frac{\partial}{\partial y}\mathbf{G}(\mathbf{U})$ may be calculated by transposing the array data $\mathbf{U}$, and then passing $\tilde{\mathbf{U}}$ instead of $\mathbf{U}$ to the ENO derivative routine, which calculates a derivative in the $x$ axis direction. This is equivalent to calculating a $y$ derivative in an untransposed data array.

There are advantages to carrying out this transpose besides the obvious one that it simplifies the coding requirements. Depending on the memory architecture of the machine in question it may lead to more favorable locality of reference. For example, if a $y$ derivative is calculated at $y$ coordinate $j$, the computational elements will require elements at location $j \pm k$, and these will be separated in distance by one entire row in the matrix. For the CM5, this is not an important issue. It is not a cached machine, and the minimum size of importance for locality of reference is the DRAM page size of 8K bytes, which corresponds to very large data arrays. On the other hand, the vectorized indexing instructions used to implement the irregular gather operation implied by ENO are apparently limited to 128 byte offsets. This more or less implies that the transpose will be necessary. The transpose operation may be carried out locally on each computational node and does not imply any communication.

The computation of $L$ may be summarized in the following way:

1. Compute pressure from the conservative variables and the previous value of temperature at each grid point. Update the temperature.
2. Compute $L_v$ and $L_c$, the viscosity and chemistry contributions to $L$.
3. Compute the $x$ derivative $L_f$.

4. Transpose $\mathbf{U}$ and the pressure. Exchange the $x$ and $y$ momentum elements in $\mathbf{U}$.

5. Compute $L_g$ in transposed coordinates.

6. Transpose back.

7. Advance to the next partial time step in the Runge-Kutta o.d.e. solver using the computed value for $L$.

8. Update the boundary values for $U$ using boundary conditions (and communication for a parallel implementation).

### 2.4.2 $\Delta t$

It should be noted that the time step $\Delta t$ that the computation uses is derived from a CFL condition during the first partial time step in the Runge Kutta routine. Each element of the computation of $L$ adds a separate contribution to the CFL calculation. For example, during the calculation of $L_f$, the maximum wave speed is $\max_j(|u_{j,k}| + c_{j,k})$. For $L_g$, it is $\max_k(|v_{j,k}| + c_{j,k})$. For viscosity, it is $\mu/\Delta x$. $\Delta t$ is then calculated from

$$\left(\max_{j,k}(|u_{j,k}| + c_{j,k}) + \max_{j,k}(|v_{j,k}| + c_{j,k}) + \max_{j,k} \mu/\Delta x\right) \frac{\Delta t}{\Delta x} < \text{CFL}.$$

The CFL number necessary depends on the spatial order of the scheme. For a third order scheme, CFL=0.8 appears to be sufficient. CFL=0.5 works for the fourth order scheme, and CFL=0.3 seems to be required for the fifth order scheme.

The CFL condition calculated from the second partial time step is also computed. If it exceeds by a certain tolerance the the result of the previous partial time step, the entire Runge-Kutta routine is restarted with the new smaller time step. This occurs only very rarely in practice.

### 2.4.3 Computation of $L_f$ and $L_g$

$L_f = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x}$ is computed by use of an ENO derivative. This is the most expensive operation in the entire code, and deserves some careful attention.

Let us return for a moment to §2.2 where the derivative $\frac{\partial g(u(x))}{\partial x}$ of the scalar flux $g(u)$ was calculated by a conservative difference of the numerical flux $\hat{g}$ (equation (2.3)). $\hat{g}(x_{j+1/2})$ was calculated by interpolating a function $H(x)$ by a polynomial, and then taking its derivative. The method suggested was to use a Lagrange polynomial interpolant (equation (2.5)), and then to take the derivative of this polynomial, which can be carried out from analytical formula by a matrix multiply. The method actually used in this implementation is to make use of the Newton polynomial interpolant for $H$. Assume that $H(x)$ is to be interpolated on the stencil $\{x_{j-1/2}, \ldots, x_{j+s_o+1/2}\}$. Then

$$H(x) = H(x_{j-1/2}) + \sum_{l=0}^{s_o} D[x_{j-1/2}, \ldots, x_{j+l+1/2}] \prod_{k=0}^{l} \frac{x - x_{j+k-1/2}}{x_{j+l+1/2} - x_{j+k-1/2}}.$$

where the difference coefficients are defined above in (2.9). It turns out that it is possible to take the derivative of the polynomial in this form at the same time that the stencil of points $\{x_{j-1/2}, \ldots, x_{j+s_o+1/2}\}$ is chosen. Given a starting stencil $\{x_{l-1/2}, x_{l+1/2}\}$ where $l = j$ or $l = j+1$, the algorithm shown in figure 2.2 enlarges the stencil depending on the magnitude of the difference coefficients $D$ at the same time that if computes the derivative $H'(x_{j+1/2})$.

This algorithm does have somewhat inferior roundoff error when compared with the Lagrange polynomial method, due to the fact that floating point precision is lost in the repeated subtractions used to calculate the difference tables. But this does not appear to be a severe problem for the cases considered here.

This is essentially the algorithm used to calculate $L_f$ and $L_g$. There are some additional complications—the algorithm above is applied twice to each element of the approximately diagonalized system of equations in a manner described in §2.2.2 and §2.2.3. Instead of just one difference table where $D[x_{j-1/2}, x_{j+1/2}] = g(x_j)$, there are $2(N_s + 4)$ difference tables constructed from

$$D_i^{\pm}[x_{j-1/2}, x_{j+1/2}] = F_i^{\pm}(x_j) = L_i \left( \mathbf{F}(\mathbf{U}(x_j)) + \bar{\Lambda} \mathbf{U}(x_j) \right)$$

using the recurrence (2.9). Here $L_i$ is the $i$−th row of the projection matrix $L$. The projection matrix $L$ varies throughout the computational domain—it is calculated anew for each half grid

```
if (l = j) then
    p ← 1
else
    p ← −1
H' ← D[x_{l−1/2}, x_{l+1/2}]
k ← 1

do
{
    if (|D[j − 1/2, ..., j + k + 1/2]| > |D[j − 3/2, ..., j + k − 1/2]|) then
    {
        H' ← pD[l − 3/2, ..., l + k − 1/2]/(k + 1)!
        l ← l − 1
        p ← (j − l + 1)p
    }
    else
    {
        H' ← pD[l − 1/2, ..., l + k + 1/2]/(k + 1)!
        p ← (j − l − k)p
    }
    k ← k + 1
}
while k < s_o
```

Figure 2.2: The ENO derivative algorithm

point $x_{j+1/2}$. $\Lambda$ may also be chosen to vary, or may be a constant. In any even, the difference table $D_i^{\pm}$ will have to be calculated anew for each half grid point.

Because the difference calculation in (2.9) is a linear operation, the difference table $D_i^{\pm}$ above may by constructed in a variety of different ways. Individual difference tables for each flux and element of the state vector may be constructed ahead of time

$$D_{\mathbf{F}_i}[x_{j-1/2}, x_{j+1/2}] = \mathbf{F}_i(\mathbf{U}(x_j)),$$

$$D_{\mathbf{U}_i}[x_{j-1/2}, x_{j+1/2}] = \mathbf{U}_i(x_j),$$

and then obviously

$$D_i^{\pm} = L_i\left(D_{\mathbf{F}_i} + \bar{\Lambda} D_{\mathbf{U}_i}\right). \tag{2.16}$$

The difference tables $D_{\mathbf{F}_i}$ and $D_{\mathbf{U}_i}$ may be shared between grid points, and (2.16) applied only within the derivative calculation. This reduces the storage required for difference tables, and it turns out to require the same number of floating point operations as the previous method.

If a constant $\Lambda$ is available, then the same trick results in a considerably faster scheme. $N_s + 4$ difference tables may be shared between grid points

$$D'^{\pm}_i[x_{j-1/2}, x_{j+1/2}] = \mathbf{F}_i(\mathbf{U}(x_j)) + \Lambda\mathbf{U}(x_j)$$

and these may be multiplied by $L$ to construct the difference $D_i^{\pm}$. This is faster because it dramatically reduces the number of memory accesses within the derivative routine.

In the actual program, an additional complication occurs that modifies the algorithm of figure 2.2. The comparison between difference coefficients

$$\texttt{if}\ (|D[j-1/2,\ldots,j+k+1/2]| > |D[j-3/2,\ldots,j+k-1/2]|)$$

is modified by allowing a bias towards the choice of a centered stencil

$$\texttt{if}\ (\texttt{BIAS}\ |D[j-1/2,\ldots,j+k+1/2]| > |D[j-3/2,\ldots,j+k-1/2]|)$$

where BIAS is calculated from a constant factor $C$ by

```
if 2(l − j) > 2 − k then
   BIAS = C
else
   if 2(l − j) < 2 − k then
      BIAS = 1/C
   else
      BIAS = 1.0
```

## 2.5 Parallelization of ENO

The ENO implementation was carried out in two different ways on the CM5, with different consequences for the cost of communication. In one implementation, the ENO scheme was coded in the CM5 version of Fortran 90, CM Fortran. Communication was implemented by the Fortran CSHIFT intrinsic function, which can move an array an arbitrary number of elements along one axis element. Here it was used iteratively to construct stencils by shifting arrays one element left, right, up or down. Because the CSHIFT operation moves data locally on each node and communicates a fairly small amount of information off-node for each operation, it is a fairly expensive operation. Some effort was made to reduce the number of CSHIFTS by interlacing the computation of the difference tables with the stencil construction in the algorithm in figure 2.2, and by applying the algorithm to all elements of the state vector at the same time. These CSHIFTs were applied to a three-dimensional array containing all the two dimensional arrays of the state vector to maximize the amount of communication occurring for each call. The result of these optimizations was a improvement in speed of a factor of three or more for small problem sizes. Communication time went from dominating the time for computation to amounting to 30-40% of computation time. But a side affect was that code responsible for the algorithm in figure 2.2 ballooned in size to several hundred lines. When optimization errors in the compiler started to occur, it seemed reasonable to try another approach.

An implementation was then carried out using message passing. Unlike CM Fortran, which effectively hides the fact that the computation is occurring on a parallel machine, each node is programmed as if it were a separate machine. Because the ENO method is an explicit finite difference method that can be implemented by straight-forward domain decomposition, the logical structure for the message passing program running on each node separately is quite similar to that of the CM Fortran implementation. The major difference is the fact that no array transposes are performed in order to compute $L_g$ in the CM Fortran implementation. An array transpose would imply a large amount of communication across nodes. Separate versions of the ENO derivative routines are used to compute $L_f$ and $L_g$ by performing the derivatives in the $x$ and $y$ directions separately.

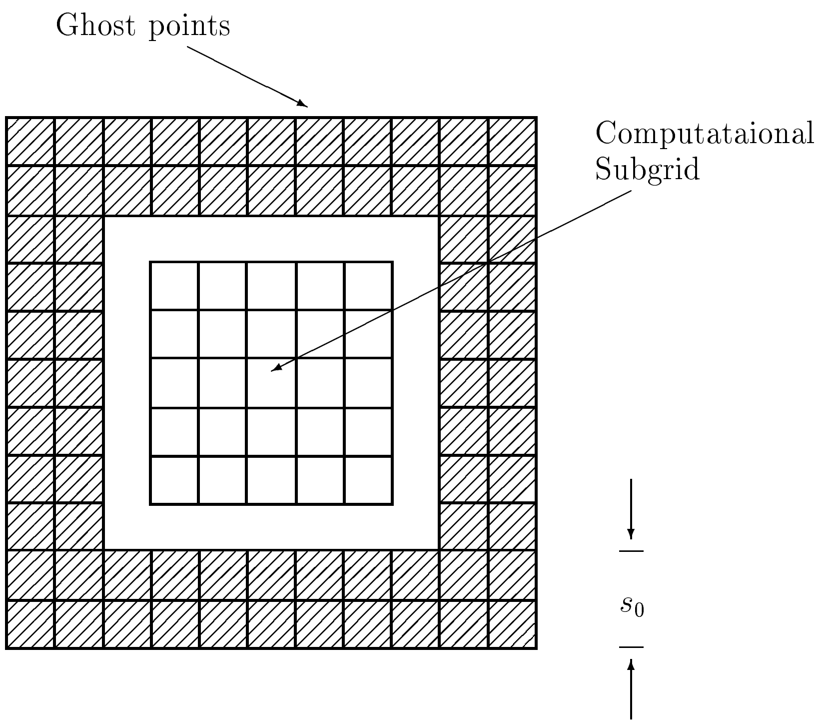


Figure 2.3: Ghost points and the computational mesh

### 2.5.1 Message passing implementation

A restriction for the flux based ENO schemes considered here is that the computational domain be a Cartesian mesh. Here only a rectangular grid is considered. In a two dimensional scheme, for each grid point $(j, k)$, the scheme with formal accuracy $s_0$ may make use of the $s_0$ grid points to the right $(j - s_0, k) \ldots (j - 1, k)$ and left $(j + 1, k) \ldots (j + s_0, k)$, as well as the points above $(j, k + 1) \ldots (j, k + s_0)$ and below $(j, k - s_0) \ldots (j, k - 1)$. If viscous terms are implemented using centered differences, then in order to compute the partial derivatives $\frac{\partial^2}{\partial x \partial y}$ the entire square of grid points $(j + a, k + b) : -s_0 \leq a, b \leq s_0$ must be used. For a rectangular mesh, this implies that a border of 'ghost points' $s_0$ elements wide is required at the borders of the mesh. (See figure 2.3.) Numerical values need to be assigned to the ghost points before the computational scheme can march in time. Boundary conditions for the underlying PDE that is being solved may be discretized to initialize these data points.

If a parallel machine is available with $N$ nodes, to implement a message passing scheme the computational mesh may be split into $N$ separate sub-domains. Ghost points will surround each sub-domain. These ghost points are initialized either from the boundary conditions as before, or if they correspond to points in the interior of the global computational domain, they are initialized from data points from the adjacent sub-domain.

Because a Runge-Kutta ODE solve is used to advance the numerical method to go from one time step to the next, ghost points must be initialized at each partial time step. For the third order TVD Runge-Kutta routine used[45], there are three of these partial time steps. If the sub-domains are of dimension $M \times M$, then $4s_0(M + s_0)$ double words of ghost points per data array must be sent. On the CM5 this is complicated by the fact that each node is logically divided into four vector units, and each vector unit is a basic processing element that maintains a sub-array. The consequence is that $4s_0(M + 2s_0)$ grid points are communicated. The cost will be approximately proportional to $s_0 M$, plus a latency time for each communication operation. Computational cost is proportional to $M^2$, so it can be seen that communication cost will be relatively most important for small problem sizes.

If there are $N_s$ chemical species then there are $N_s + 4$ data arrays. Four `send_and_receive()` subroutine calls are used to perform communication. If we assume a communication bandwidth $\beta$ bytes/second, and a latency of $\alpha$ seconds per subroutine call, then the communication time is

$$(N_s + 4)4s_0(M + 2s_0)8/\beta + (N_s + 4)4\alpha.$$

Figure 2.4 shows the measured communication time per partial Runge-Kutta time step. Fitting a line to this curve results in an estimate of $\beta = 1.4$Megabytes/s, $\alpha = 130\mu$s. Both these figures are considerably less than it is possible to achieve on the machine—$\beta = 10$Mb/s and $\alpha = 8\mu s$[29]. The communication subroutines copy data more than necessary and also perform the operation of communicating data between vector units, perhaps halving the effective measured bandwidth. Neverless, even this kind of performance is acceptable for this application. In figure 2.5, the fraction of total time spent in communication is presented.

A word here about manner in which the computational code here was implemented may be useful.

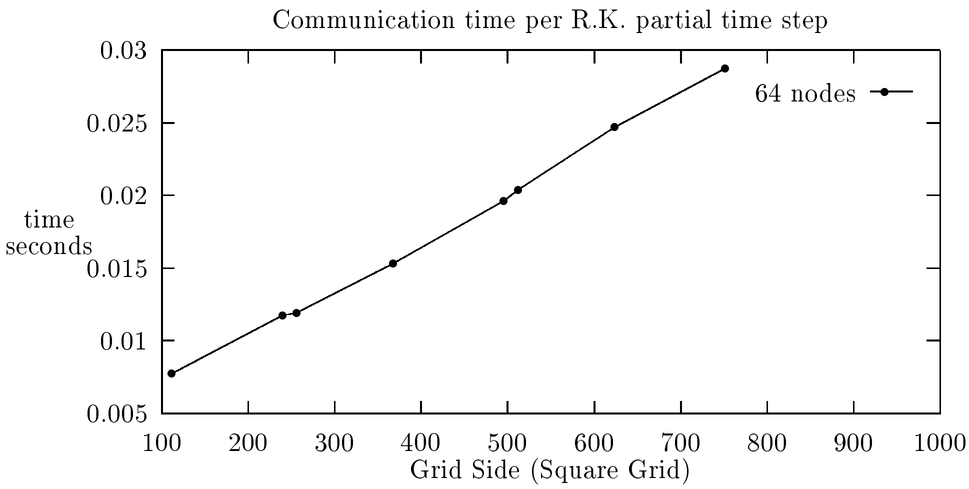


Figure 2.4: Communication time for a 3rd order ENO simulation of the Euler equations with periodic boundaries.

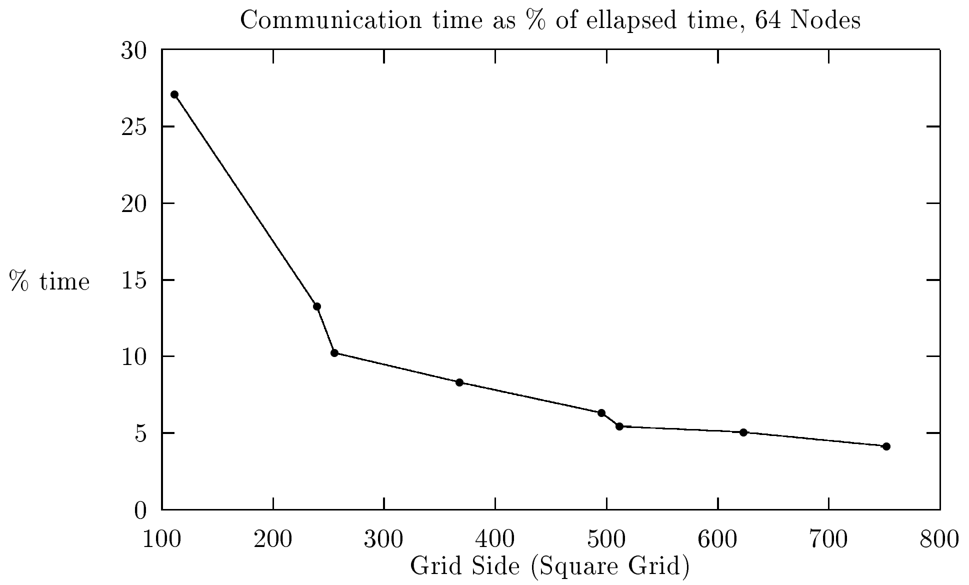


Figure 2.5: Communication as a percentage of ellapsed time for a 3rd order ENO simulation of the Euler equations with periodic boundaries.

The individual CM-5 nodes were programmed using a combination of a low-level 'C' compiler for the vector units known as AC[8], and the vector unit assembly language CDPEAC. It is possible to program individual nodes in CM Fortran, and in retrospect this might have been a better approach, as the AC compiler is quite unreliable.

## 2.6 Code validation

For the most part, the parallel implementation of ENO is close enough to the standard ENO algorithms so that the correctness of the code could be checked by comparison with existing well-tested ENO programs. A comparison was made with 1-D codes for one-dimensional problems, and the parallel code was found to calculate identical results up to round-off error.

The most serious potential remaining source of bugs is the computation of viscous terms in the physical model 2.1. Happily, a physical test problem exists that allows verification of these terms. Following Ton[52], a comparison was made between a computed compressible viscous flow over a thermally non-conductive plate and the known similarity solution for a compressible boundary layer.

The compressible boundary layer theory suggests that if the $y$ coordinate perpendicular to the plate is transformed by the Dorodnitzyn transformation

$$\bar{y} = \int_0^y \rho \, dy$$

and $\rho\mu$ is a constant throughout the flow, then velocity profiles can be related to the Blasius similarity solution of incompressible flow.

The Blasius equation is

$$ff'' + 2f''' = 0.$$

This is solved as a two point boundary problem, with $f(0) = 0$, $f'(0) = 0$ and $f'(1) = 1$. For the

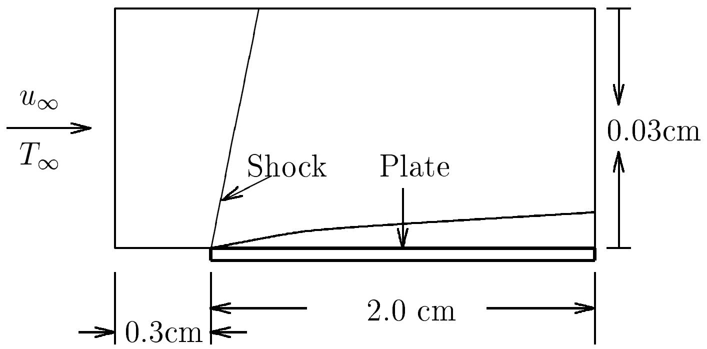


Figure 2.6: Geometry for a compressible boundary layer simulation.

compressible boundary layer, the similarity variable $\bar{\eta}$ is defined by

$$\bar{\eta} = \bar{y}\sqrt{\frac{u_\infty}{\rho\mu x}}$$

The streamwise component of velocity $u$ is then given in terms of the free stream value $u_\infty$ by

$$\frac{u}{u_\infty} = f'.$$

The transverse component $v$ is given by

$$\frac{v}{\frac{1}{2}\sqrt{\frac{u_\infty \mu}{\rho x}}} = \bar{\eta} f' - f.$$

If the Prandtl number $P_r = 1$, the temperature profile is given by

$$\frac{T - T_\infty}{T_\infty}\frac{2}{(\gamma - 1)M_\infty^2} = 1 - (f')^2.$$

where $M_\infty$ and $T_\infty$ are the free stream Mach number and temperature.

The numerical simulation was carried out using Ton's geometry, which can be seen in figure 2.6. The viscosity was given by

$$\mu = 3{\times}10^{-7}T$$

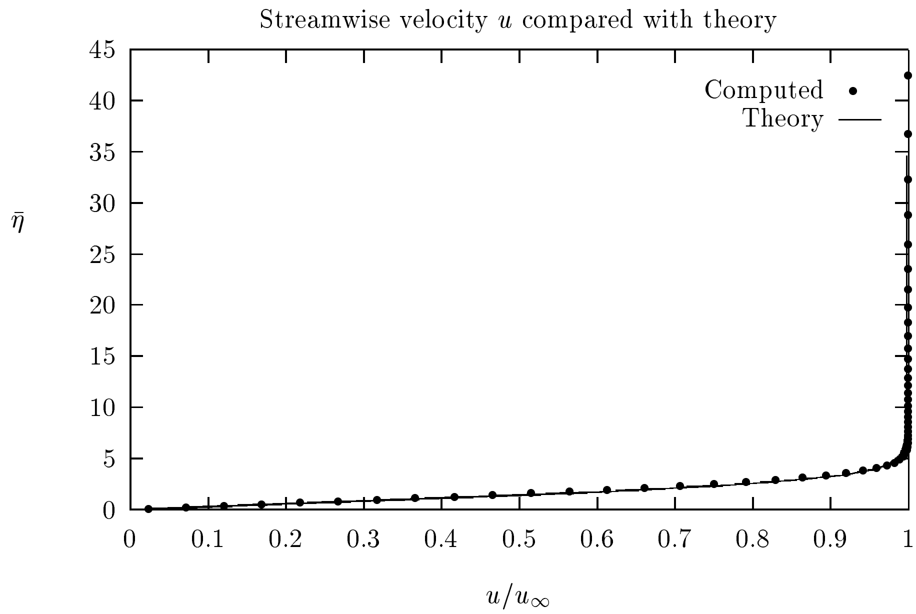


Figure 2.7: $u$ compared with theory for a Mach 1.2 compressible boundary layer.

and the computational region was 112 uniformly spaced points in the horizontal direction, 56 points vertically. A mapping was applied vertically to concentrate points within the boundary layer. Characteristic boundary conditions were applied to the top left (inflow) and right (outflow) sides. A reflection boundary condition was applied at the bottom boundary to the left of the plate, and a no-slip condition to the right. The flow simulated was a Mach 1.2 flow in air at 1 atm at $298°K$. The flow solver was initialized with a uniform flow condition with all variables at the free stream values, and was advanced in time until a steady state was reached.

Because the outflow boundary affected the flow somewhat more that the zero-gradient condition applied by Ton, a comparison of the flow solution with theory was made further away from the outflow boundary, 1 cm from the beginning of the plate. This Comparison may be viewed in figures 2.7, 2.8, 2.9. the discrepancy seen in $v$ is quite similar to that seen by Ton, and probably reflects the characteristic boundary condition at the top boundary.

### 2.6.1 Mflop rates

As it is necessary to count floating point operations by hand, a rough estimate of performance is provided here. For the simple case where $s_0 = 3, N_s = 0$ with periodic boundary conditions and a

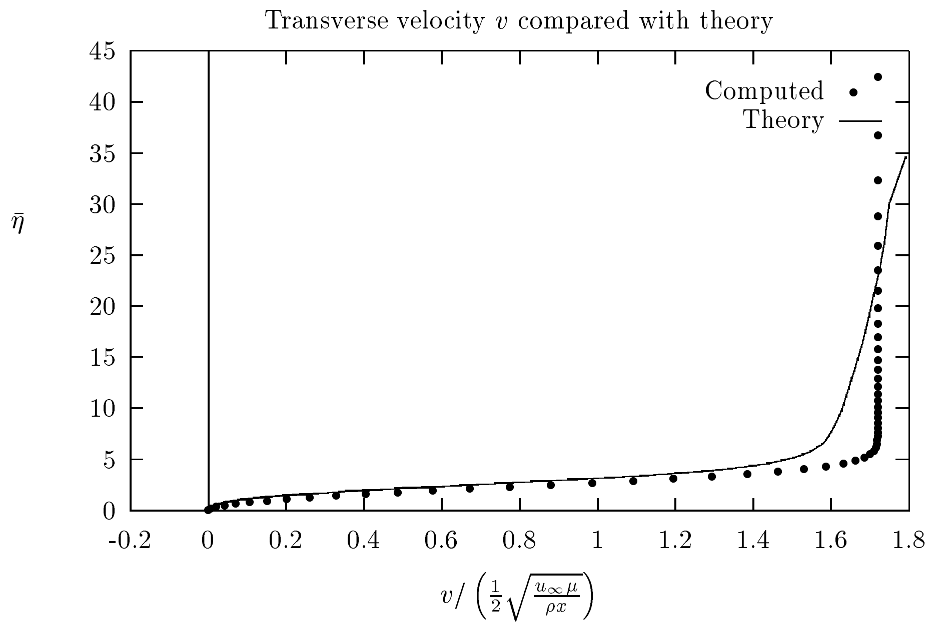


Figure 2.8: $v$ compared with theory for a Mach 1.2 compressible boundary layer.

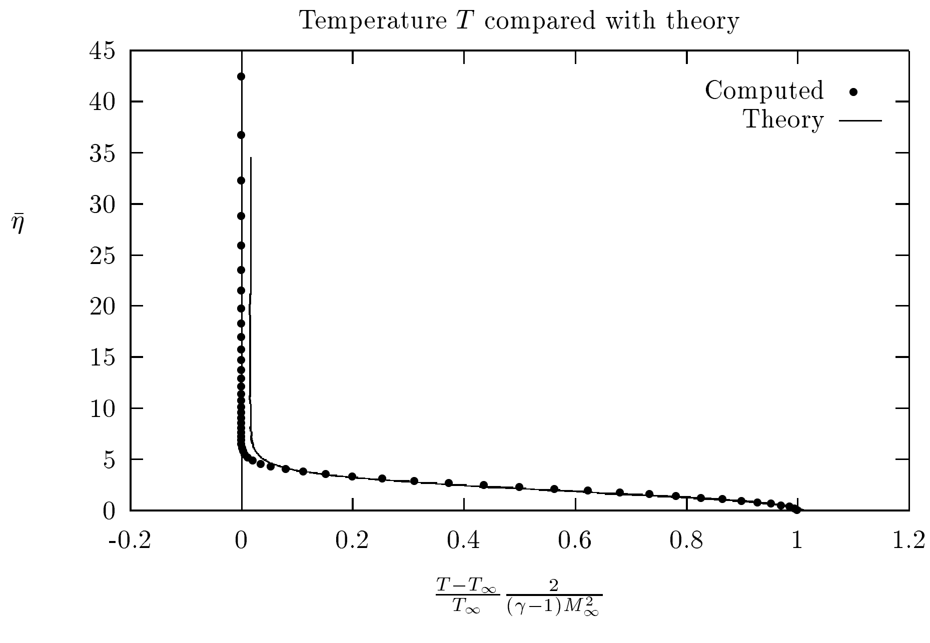


Figure 2.9: $T$ compared with theory for a Mach 1.2 compressible boundary layer.

real equation of state, a quick estimate yields 600 floating point operations per grid point for the ENO derivative routine. The divided difference tables require 32 operations, and the remainder of the computation of $L_f$ requires about 100 operations. This is repeated 6 times per time step for a third order Runge-Kutta routine, and so the requirement is about 4400 operations per grid point, if the calculation of pressure and temperature from the real equation of state is omitted. For the case $s_0 = 5, N_s = 0$, a similar estimate yields about 7550 floating point operations per grid point. Applying these numbers to the measured elapsed times yields an Mflop rating that can be seen in figure 2.10. A word about the diagram may be in order. The computation timed was a third order periodic ENO with no chemical species and a perfect gas equation of state. For the 64 node case the grid was square, and the $x$ axis gives the grid width in grid points. For the 512 node case the grid is twice as wide as high. The 'PE side' is a measure of the vector length seen by each vector unit. The flops rate is optimized when the 'PE side' is one less than a multiple of eight, and this accounts for the irregular form of some of the graphs. On others, only the most efficient grid widths were sampled, and the resulting graphs are regular. The much better performance for the $s_0 = 5$ case can be explained as the result of the fact that relatively more computation occurs within the ENO derivative routines, which are optimized kernels and perform considerably better than the rest of the program. Figure 2.11 presents a graph of performance serveral versions of the parallel code in comparison with a vector code on the Cray YMP.

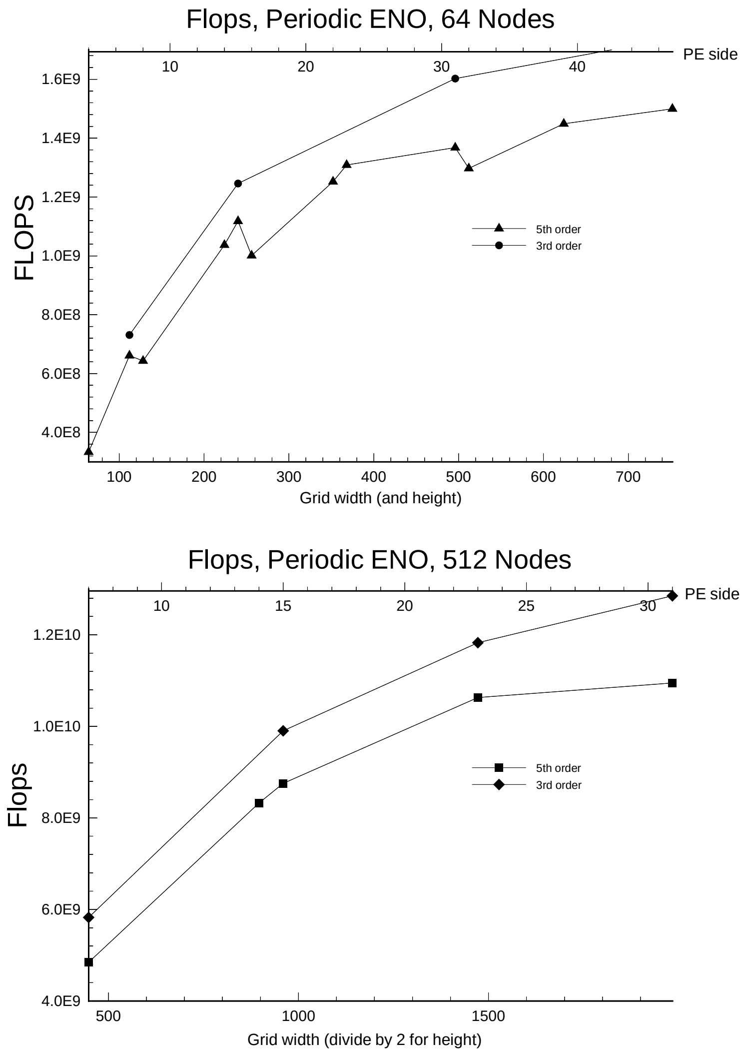


Figure 2.10: Flop rates on 64 and 512 node machines for periodic Euler simulations. Note change in scale. Maximum possible performance is 128 Mflops/node.

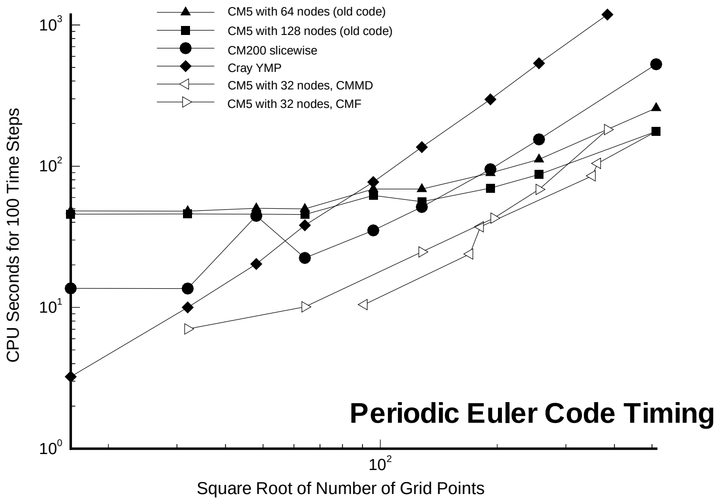


Figure 2.11: 3rd order ENO periodic Euler times on a variety of machines. The 'CMF code' is the CM Fortran 90 program. The 'CMMD code' is the message passing program. 'Old code' refers to the initial CM Fortran implementation.

# Chapter 3

# 2-D Shock/Cylinder Interactions by ENO

Air-breathing SCRAM-jet engines present a difficult challenge to the designer. Extremely rapid combustion is a necessity in these engines, where supersonic flow through the reaction chamber sweeps reactants out of the engine within a few milliseconds. In addition, the reaction must take place well away from the walls of the combustion chamber to prevent excessive heat transfer to the engine itself.

In a combustion process with reactants that are not pre-mixed, the combustion rate is limited by the diffusion rate of reactants across the air-fuel interface. The rate can be increased by stretching out the interface by the motion induced by a point vortex [31]. This not only increases the total length of the interface, but at the same time it steepens the concentration gradient and thus the diffusion rate of reactants by bringing them in proximity to the interface. Marble[32] suggests a way to induce this vorticity—to have a shock pass through a jet of fuel nearly perpendicular to its axis (see figure 3.1). While passing through the region of inhomogeneous density the shock will produce vorticity via the $\frac{\nabla\rho\times\nabla p}{\rho^2}$ term in the vorticity equation:

$$\frac{D\omega}{Dt} = (\omega\cdot\nabla)V - \omega\nabla\cdot V + \frac{\nabla\rho\times\nabla p}{\rho^2}. \tag{3.1}$$

If the fuel (usually hydrogen) and air have significantly different densities, then an intense production of vorticity results from the gradient in pressure across the shock in conjunction with the gradient in fluid density across the interface. The vorticity induced by the normal shock will be concentrated along the fuel-air interface and have the largest magnitude where $\nabla\rho$ is perpendicular to $\nabla p$, e.g. at the top and bottom edges of the circular fuel jet.[22]. One might expect that the induced vorticity surface would roll up to form a vortex propagating in the same direction as the shock (for a light fuel). The fuel-air interface should roll up along with the vorticity in this flow, and barring unforeseen interactions between the chemical reaction and the generated flow field, mixing and the resulting combustion should be efficient.

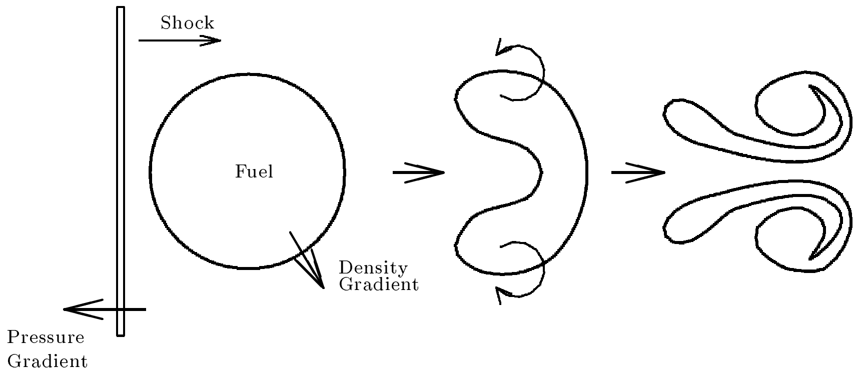


Figure 3.1: Roll-up in a shock-cylinder interaction due to induced vorticity.

Another approach to understanding the initial evolution of the flow has been suggested by Haas and Sturtevant[19] The initial evolution of the flow induced by the passage of the shock can be interpreted in terms of the Richtmyer-Meshkov instability[39]. This instability is related to the Rayleigh-Taylor instability, which describes denser fluid lying above a lighter one in a gravitational field. The heavier fluid forms fingers descending down into the lighter fluid below. A similar fingering can result from the impulsive acceleration due to a propagating shock. It differs from the Rayleigh-

Taylor instability in that the perturbation grows linearly rather than exponentially, and that there is no stable fluid layering—fingering will occur irrespective of in what direction the shock propagates between denser and lighter fluids. Given the amplitude of a disturbance of a planar interface between two fluids, the Richtmyer-Meshkov instability predicts a growth rate

$$\frac{d\eta(t)}{dt} = k[u]A\eta(0).$$

Here $A$ is the Atwood ratio $(\rho_2 - \rho_1)/(\rho_1 + \rho_2)$, k is the wave-number of the disturbance, and $[u]$ is the flow speed jump across the shock. $\rho_1$ and $\rho_2$ are the upstream and downstream densities respectively. Haas and Sturtevant apply this linear theory by comparing a hydrogen bubble with a sinusoidal interface discontinuity with period $2\pi R$ where $R$ is the cylinder radius. Picone and Boris[36] observe in their numerical experiments that $4R/3$ is a better fit for the period of the interface, casting some doubt on the direct applicability of this linear theory to a cylinder. However, one might expect that the evolution of high wave-number perturbations on a cylindrical interface might be successfully modeled by this mechanism. Recent experimental results of Brouillette and Sturtevant[5] suggest that the growth rate of instabilities may be substantially reduced from the predicted values if the fluid interface is diffuse rather than discontinuous. This affect is most noticeable for wave numbers of the perturbation on the order of the interface thickness. The flow is also stabilized by viscosity. For interfaces that are much thicker than the wave number considered, the growth rate is characterized entirely by the interface thickness. Despite these weaknesses, this theoretical approach does describe the observed initial flow features, and suggests what the influence of high wave number perturbations in the fuel-fluid interface might be.

### 3.0.2 Previous Work

The interaction between shock waves and inhomogeneous regions of density has been studied previously experimentally, analytically and numerically. The starting point for many has been studying the Richtmyer-Meshkov instability of a shock passing through a perturbed planar interface, beginning with the experiments of Meshkov[34]. Experimental work in this area continued with Zaitsev *et* *al.*[61] and Brouillette and Sturtevant[5]. Oblique plain interfaces attracted the attention of Bonazza *et al.*[4] and Yang, Zabusky and Chen[60]. Spherical interfaces were first studied by Rudiger and Sommers[42]. Haas and Sturtevant[19] studied both cylindrical and spherical interfaces experimentally, and analyzed the results in terms of linear geometric optics as well as the Richtmyer-Meshkov instability. These experimental results inspired the numerical work of Picone and Boris[36] and further experimental work of Jacobs[22] concentrating on the shock-cylinder interaction.

More recently a number of researchers have continued these initial attempts at numerical modeling. Among these, Yang, Kubota and Zukoski[58] applied the FCT computation methodology to the conservation equations in order to analyze a shock-cylinder interaction, and derive scaling laws for the amount of circulation produced by the interaction, the time scale of the interaction, and vortex spacing. Ton[52] used a detailed physical real gas model including detailed models of viscosity and chemistry and examined the contribution of these to the production of circulation and the degree of mixing and combustion. Samtaney and Zabusky[43] studied oblique, sinusoidal and cylindrical slow-fast interfaces using the shock polar analysis technique of Henderson[21] to model the interaction and make comparisons with a second order numerical scheme. Recently Quirk and Karni[37] have updated the work of Picone and Boris by carrying out an adaptive high resolution numerical simulation of Haas and Sturtevant's experiment using adaptive mesh refinement (AMR) and a finite difference scheme based on primitive instead of conservative variables. Numerical calculations have begun in three dimensions as well. Waitz[54] has simulated a realistic combustor design using a 3-D code at fairly coarse resolution, and has made a comparison with experimental data. Bell[3] has carried out a fully three-dimensional shock sphere calculation using an adaptive code.

### 3.0.3 High order schemes

The motivation for the current research has been the recent development of high order shock capturing numerical schemes for the computation of flows with shock waves[13, 6, 45] etc. The fact that these schemes have less artificial dissipation than lower order schemes makes them attractive for computation of an unsteady transient flow, especially one where a solution is required over several tens of thousands of time steps. The presence of large amounts of artificial dissipation in lower order methods suppresses fluid instabilities and results in artificially smooth simulations. Lagging phase errors are another affliction of these numerical methods. Some computational schemes applied previously also suffer from the anisotropic nature of the resulting truncation error, producing odd artifacts that are clearly visible in the computed results[58].

In this investigation I apply a parallel 2-dimensional ENO code to study this problem. A comparison was also made in many cases with results generated by Wai Sun Don using a pseudospectral code devised by him based on Chebyshev collocation[7, 18]. In this code a weak exponential smoothing of the fluxes and the solution is used to stabilize the discretized PDE [13]. If this stabilization is sufficiently weak oscillations related to the Gibbs phenomena are generated by shocks and other discontinuous points in the flow field. Nevertheless, it appears to be possible to reconstruct an accurate solution from these oscillatory computations. Recent developments of sophisticated reconstruction techniques for the removal of Gibbs phenomena [1], enable the computed raw data to be post-processed to produce a non-oscillatory solution away from discontinuities, and the result compares quite well with ENO code calculations.

Because the ENO code and spectral codes are so different, the fact that a comparison can be made and that the results are close does suggest strongly that both codes are calculating results that are close to the correct answer. The fact that ENO is a family of schemes with different numerical accuracy allows for a determination of the exact advantage that high order methods have for this problem, and also provides a convincing demonstration that the spectral method, despite the use of an array of tricks to maintain accuracy, is computing a high-order solution.

## 3.1 ENO Comparison with Experiment

Two experimental results are available that permit a comparison of numerical experiments with reality. Haas and Sturtevant[19] studied in one set of experimental runs shock interactions with a helium cylinder. The cylinder is 5cm in diameter, placed in a experimental section 8.9cm wide. All gases are at atmospheric pressure and room temperature, taken as 298°K. In the experiment, the cylinder was contained by a nitrocellulose microfilm, which permitted some gaseous diffusion. The helium was estimated to be contaminated 28% by mass with air. The air surrounding the cylinder was continuously refreshed, and helium contamination was observed to be small enough to ignore. The Mach number of the incident shock was either 1.085 or 1.22. Visualization in the experiment is by spark shadowgraph, which provides a reasonably precise determination of the interface between helium and the surrounding air. The well-defined nature of the initial condition in this experiment, and the fact that the researchers took great care to provide accurate quantitative measurements of shock speeds enables a comparison with numerical simulations. The results of an interaction with a Mach 1.22 shock simulated via the ENO code is presented in next section. This provides a good benchmark for the code.

Jacobs[22] presents an experiment with a helium cylinder interacting with shocks of Mach number 1.093 or 1.15. Unfortunately, the cylinder was created by introducing a helium jet at the bottom of an experimental section of a shock tube. The jet is then vented at the top of the section. The requirement that the jet retain a stable cross section limited the flow rate of the helium to $65.1\text{cm}^3/s$, emanating from a 0.9525cm diameter nozzle. This low flow rate permits a substantial diffusion of helium into the surrounding air. Buoyancy also substantially affects the jet by thinning it. Flow visualization was obtained by seeding the helium by a fluorescent gas (biacetyl) and illuminating by a planar laser beam. The visualization thus avoids being contaminated by three dimensional effects due to the finite height of the helium cylinder, and this is an advantage of the experimental setup. Unfortunately the disadvantages of this apparatus seem to outweigh the advantages. The heavy biacetyl gas diffuses much more slowly into air than helium, and the extent of the diffuse helium air interface is not apparent. This presents two difficulties for numerical reproduction of the experiment. The first is that one can only guess at what initial conditions to use. The other is the fact that the very diffuse helium air interface is known to have the effect of suppressing perturbation growth generated by the Richtmyer-Meshkov instability. The resulting images, taken from the center of a diffuse jet, are very smooth. They obviously cannot resolve the issue of how three dimensional effects affected the helium air interface in Haas's experiment, as the interface is more stable and in any event, perhaps not visualized at all.

Both of these experiments applied weak shocks to the helium cylinder. The low Mach number of these shocks prohibit simulation by use of the Spectral Code applied here. Spectral codes place more demands on the applied boundary conditions in order to achieve stability. A consequence of the limited abilities of the current boundary conditions is an inability to handle the subsonic inflow conditions present in this type of problem.

### 3.1.1 Numerical comparison

The numerical simulation of the from Haas and Sturtevant's experiment was carried out in in a computational domain which is depicted in figure 3.2. The sharp boundary between helium and air was smoothed slightly to extend over two to three grid points to enable it to be resolved better by the numerical method. All elements of the conservative variables $\mathbf{U}$ and temperature $T$ were modified by a factor

$$s(r) = \exp(-\alpha(r/r_0)^\delta) \tag{3.2}$$

where $r$ is the radial distance from the center of the helium jet and $\alpha = -\ln\epsilon$ with $\epsilon$ being the machine zero. $\delta$ was chosen to be 200.

In order to keep the cylinder from moving out of the small computational grid, a transformation is applied within the calculation so that the computational grid travels to the right at 125m/s.

The result of the ENO numerical simulation of this Mach 1.22 shock-helium cylinder interaction may be compared with the experiment in figure 3.3. The experimental shadowgraphs provide a visualization of the magnitude of the second derivative in density[28]. A reasonable comparison may be obtained with a numerical Schlieren photograph, which provides a picture of the norm of density gradient. We follow the methodology of Quirk and Karni[37] and plot a grayscale image of

$$\exp\left(-k|\nabla\rho|/\max(|\nabla\rho|)\right).$$

Esthetically pleasing results are obtained with $k = 120$.

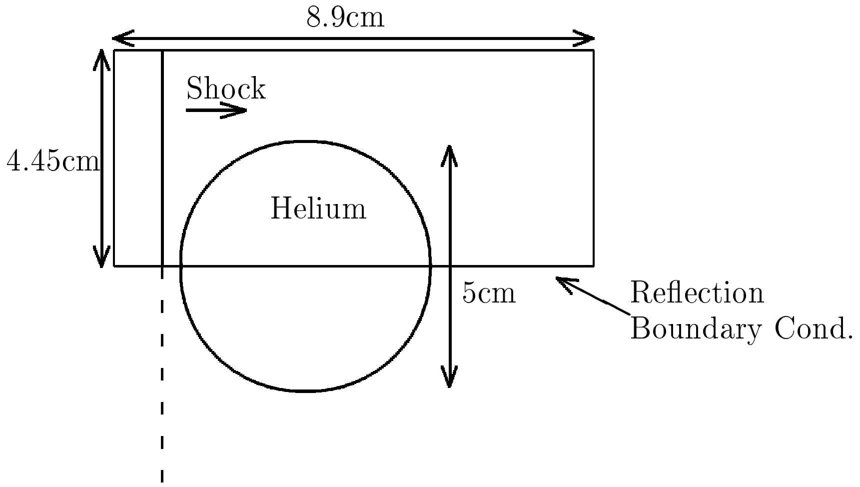


Figure 3.2: Computational geometry used to emulate Sturtevant and Haas's experiment.

Figure 3.3(A–H) provide a picture of the early time behavior of the flow as the shock initially compresses the helium cylinder and refracts through it. We compare them with Haas and Sturtevant's[19] results (frames a–h). Because the times measured experimentally did not have high absolute precision, some discrepancies may be seen in the relative position of shocks, especially in the earlier pictures.

Due to the low Mach number one might expect linear geometrical optics to provide a qualitative explanation of the results. At time $32\mu s$ and $52\mu s$ the shock reacts to the higher sound speed in helium by refracting and advancing ahead of the surrounding shock front in air. Meanwhile, the shock undergoes a complicated form of Mach reflection as it diffracts around the exterior of the cylinder, forming a four shock intersection. At $62\mu s$, the refracted shock emerges from the cylinder, leaving behind an internal reflection which forms two cusps inside the cylinder. At $82\mu s$, the shocks derived from the first internal reflection inside the cylinder emerge and cross just in front of it.

The later pictures present the long time evolution of the cylinder as induced vorticity causes it to first assume a kidney shape and then split into two vortices propagating in the streamwise direction. Reflections returning from the top boundary, provide a complex picture in frame (o).

Initial perturbations in the surface of the cylinder continue to grow causing mixing at the interface.

It is worth making an additional remark regarding the last picture and time $983\mu s$. The vortex is well developed by this stage in the flow evolution, and it is causing fluid to be entrained inside the cylinder, effectively splitting it in two. The entrained fluid appears to be forming a 'mushroom' shape as it penetrates into the cylinder from the top. This 'mushroom' seems to be a robust feature of the flow. We have seen it in a variety of different simulations with different flow conditions. If the numerical scheme has enough resolution, it almost always seems to be there. But observing this phenomenon does require a high resolution numerical simulation, and it this is used below to provide an indication of when flow features are being properly resolved.

In figure 3.4 and table 3.1 we present a comparison of shock speeds and the speed of various elements of the flow field and compare them with Haas and Sturtevant's results. The results agree about as closely as might be expected. The most significant errors are obtained with the measurement of the vortex propagation speed $V_v$ and the up-stream interface space $V_{ui}$. The error in these results may mostly be due to misinterpreting the definitions used by Haas and Sturtevant. The position of the upstream interface initially does not possess a constant velocity—it is accelerated over some distance by the flow. The velocity derived by a line fit thus varies depending on how many points are used. Values of velocity may be derived that vary from $160m/s$ to $200m/s$. We chose a high value within the 10% experimental error as a compromise between honesty and a desire to indicate that values measured by the experiment might be lower than in a calculation which ignores the contribution of the microfilm present in the experimental apparatus. The definition of the vortex head is not precisely clear. It was taken as the asymptotic slope of the downstream interface. We omit the measurement of $V_{df}$ present in Haas & Sturtevant, as it is unclear where to measure it in the flow field. Another obvious comment to make about the measured speeds is that it is unfair to criticize Quirk & Karni's method for being only approximately conservative. The conservative ENO method essentially duplicates the shock speeds they reported.

These numerical results are obtained by applying a fifth order ENO method for the spatial discretization, and a third order TVD Runge-Kutta solver in time. The computation is performed on

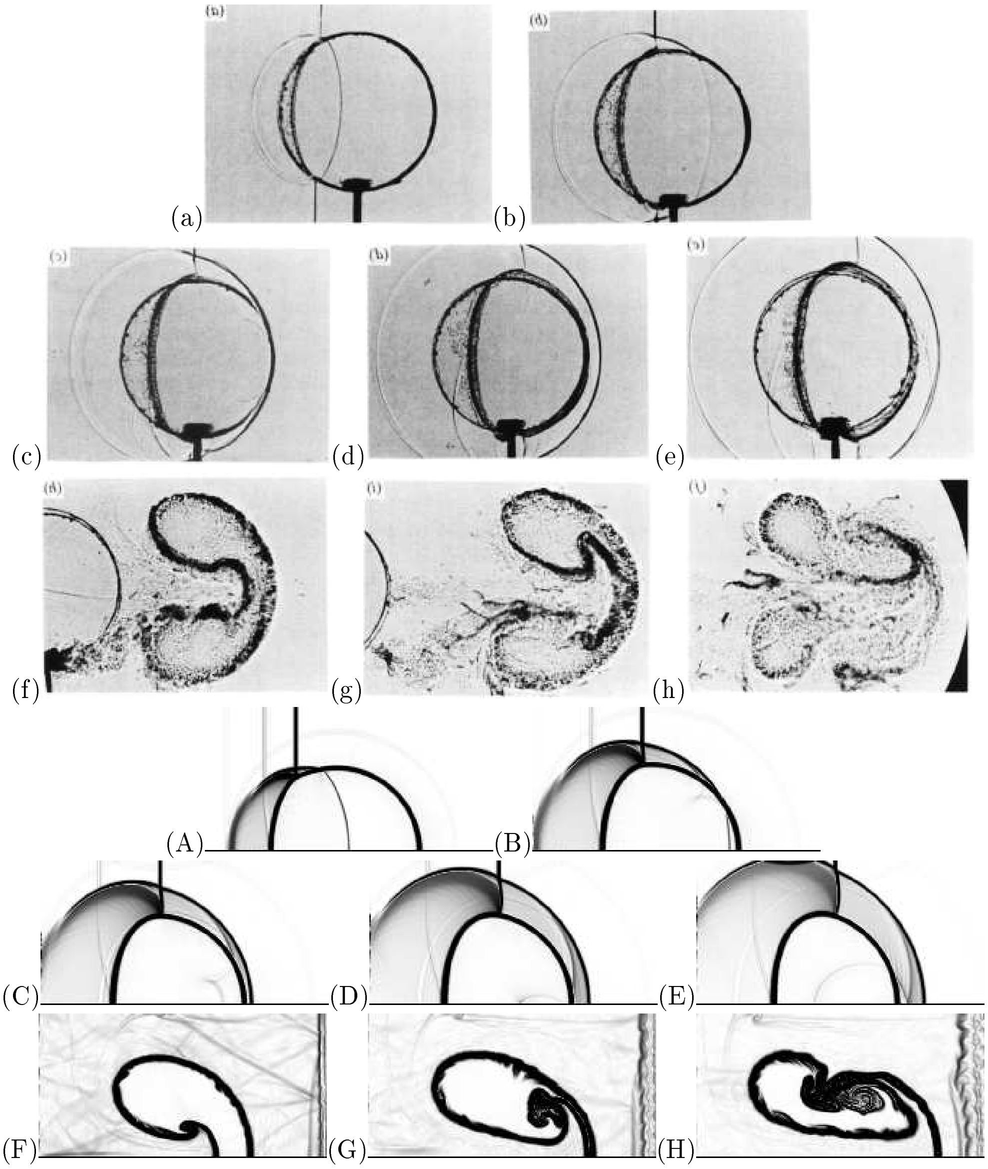

Figure 3.3: Comparison of the experiment (a–h) and ENO calculation (A–H) for the Mach 1.22 shock helium cylinder interaction. The corresponding times are (a,A) $32\mu s$, (b,B) $52\mu s$, (c,C) $62\mu s$, (d,D) $72\mu s$, (e,E) $82\mu s$, (f,F) $427\mu s$, (g,G) $674\mu s$ and (h,H) $983\mu s$. Images (a–h) are copied from Haas and Sturtevant[19].

| Velocity | $V_s$ | $V_R$ | $V_T$ | $V_{ui}$ | $V_{uf}$ | $V_{di}$ | $V_j$ | $V_v$ |
|---|---|---|---|---|---|---|---|---|
| Computation | 424 | 936 | 377 | 186 | 115 | 144 | 226 | 146 |
| Experiment | 410 | 900 | 393 | 170 | 113 | 145 | 230 | 128 |
| Quirk & Karni | 422 | 943 | 377 | 178 | – | 146 | 227 | – |
| ENO Error, % | +3.4% | +4.0% | –4.1% | +9.4% | +1.8% | –0.7% | –1.7% | +14% |

Table 3.1: Velocities measured from the ENO calculation compared with Haas & Sturtevant's experiment and Quirk & Karni's calculation.

a 496×248 point uniform grid, and it uses a moving coordinate system that translates with approximately the same mean velocity as the cylinder after the interaction. The characteristic boundary conditions seem to be sufficient to permit shocks to leave the computational region. Reflecting waves from these boundaries appear to be entropy waves which have a strength of approximately 1% of the pressure of the out-going shock. For the times measured here, these entropy waves do not succeed in propagating back close enough to the cylinder to cause interactions.

## 3.2 Comparison with a spectral code

As can be seen in the previous section, Haas and Sturtevant's shock-helium cylinder interaction is well simulated by an Euler code, even with no attempt made to model viscosity. But this experimental problem is not an ideal setting for studying shock-cylinder interactions.

The first objection is that the flow geometry itself complicates the phenomena under study. The cylinder is more than one half the width of the experimental channel. Blockage effects in the flow field should thus be quite important. From the point of view of studying how fluid instabilities are driven by the shocks present in the flow field, the picture is considerably complicated by the fact that reflections from the top and bottom walls of the flow section return quickly to influence the cylinder. The nearly discontinuous initial condition ensures that small wavelength perturbations will be amplified by the fluid instabilities[5]. A less complicated evolution may be obtained by starting with a more diffuse interface. Finally, once vortices form and begin propagating, they will be influenced by their images due to reflection and the boundaries, and travel faster than otherwise would be expected.

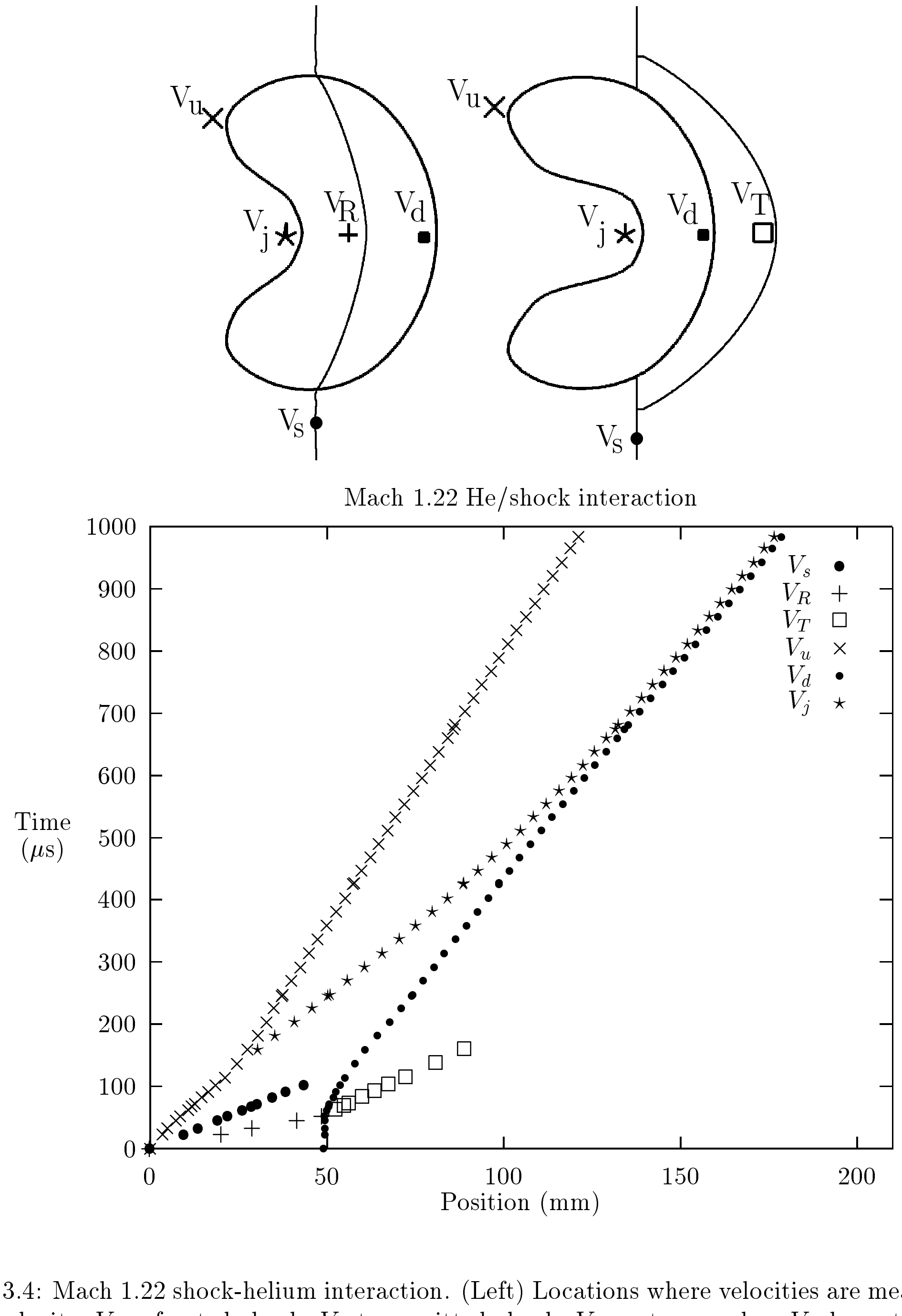


Figure 3.4: Mach 1.22 shock-helium interaction. (Left) Locations where velocities are measured. $V_s$: shock velocity, $V_R$:refracted shock, $V_T$:transmitted shock, $V_u$:upstream edge, $V_d$:downstream edge, $V_j$:air jet head. (Right) $x - t$ diagram.

Another objection arises purely for numerical reasons. The central concern with these numerical simulations is the fact that in long time integrations of the flows, errors (dissipative, dispersive, roundoff, to name a few) arising from the spatial discretization are cumulative. Eventually the solution can become unreliable. The low Mach number of the interaction, denser gas within the cylinder, and denser gases in general due to a lower initial temperature imply that the characteristic time scale for the flow evolution will be quite slow (see Yang[58] for a derivation of flow time scales). A faster Mach number and hydrogen within the cylinder imply a much faster time scale, and this substantially reduces the number of time steps required by the numerical scheme, even taking into account the reduced time step implied by the CFL restrictions of an explicit code. The resulting computation is far less computationally demanding, and it relies much less on the accuracy of the time integration method. The result is a much better setting in which to determine how schemes with different spatial accuracies affect the outcome of the simulation.

### 3.2.1 Evolution of the shock-hydrogen interaction

In this case, a Mach 2 planar shock located at $x = 0.5$cm is moves downstream and interacts with a circular cylinder of hydrogen with center at $(x_c, y_c) = (2.75\text{cm}, 0\text{cm})$ and radius $r = 2$cm (see figure 3.5). A smooth interface similar to that used in the previous section is applied, taking $\delta = 16$ in equation (3.2).

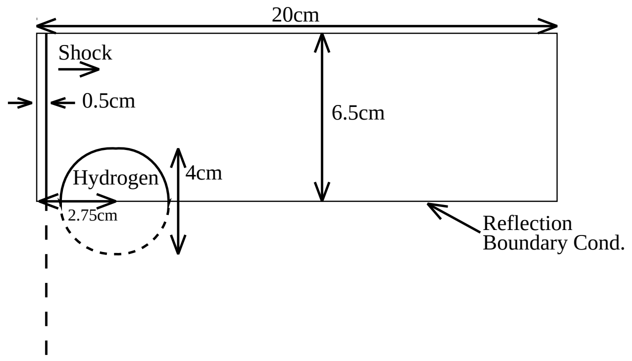


Figure 3.5: Computational geometry used for the Mach 2 shock-hydrogen cylinder interaction.

In figure 3.6 the density of the spectral simulation and the two best complete ENO runs for the

Mach 2 shock interaction with the hydrogen cylinder at various time $t = 20, 40, 60, 80, 120\mu s$ are shown. All density contour plots have 40 contour lines between 0.05 and 1.2.

The ENO solutions are based on the third order version with 1472×736 grid points (left) and the fifth order version with 752×376 grid points (middle). The rightmost figure is the raw data of the spectral simulations with 512×512 Chebyshev collocation points. The post-processed data of the spectral solution is given in the left most column of figure 3.7. The size of the domain used is $0 \leq x \leq 20\text{cm}, -6.5\text{cm} \leq y \leq 6.5\text{cm}$. In figure 3.7, the same Spectral Code is run with the same parameters as the previous case with the exception that the domain size in $x$ is reduced to $0 \leq x \leq 12.5\text{cm}$. With the same number of Chebyshev collocation points, this increases the effective resolution of the solution for the smaller domain, or equivalently increases the number of collocation points to $\approx$ 800×512 in the larger domain. Both the raw and post-processed density solution are given in the middle and the rightmost column respectively. The time shown are $t = 20, 40, 60, 80\mu s$.

The evolution of the flow in this case is very similar to that presented by the Mach 1.22 shock-helium cylinder interaction. The shock passes through the cylinder compressing it. The same four-shock intersection may be seen as the shock diffracts around the circular fuel-jet. A vortex forms, and it entrains fluid inside the cylinder. The difference in this case is the much higher speed of the interaction—the vortex forms while the shock is still diffracting around the cylinder.

It worth noting that the vortex forming at time $40\mu s$ in the spectral simulation (see middle column of figure 3.7) was not seen for low resolution/low order schemes. The vortex manifests itself as a growing instability in a kink at time $20\mu s$ in the upstream the air-hydrogen interface. This kink was dismissed as a numerical artifact in the beginning of this research. It seems to be a manifestation of the numerical 'ringing' seen when using spectral methods for discontinuous solutions (Gibbs phenomena). It was a surprise when its existence was confirmed by running higher resolution and higher order ENO calculations. This points out the fact that the spectral methods can still retain high order information when applied to discontinuous problems, even when the computed solution is very oscillatory.

Numerically, despite the oscillatory pictures of the raw density data, the post-processed density

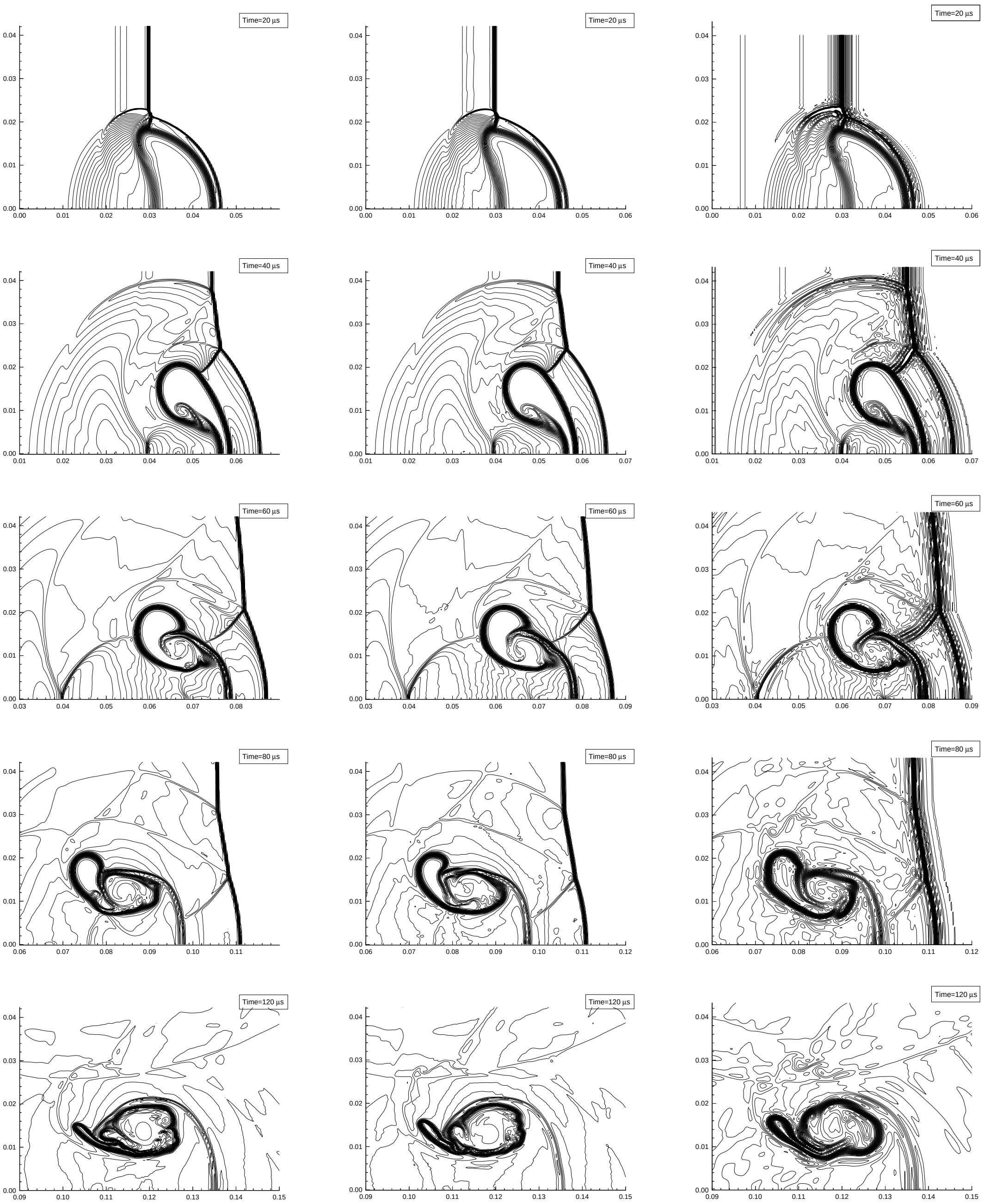


Figure 3.6: Density contour plot of the ENO code versus Spectral Code for the Mach 2 Shock-Hydrogen cylinder interaction at $t = 20, 40, 60, 80, 120\mu s$. ENO 3rd order with 1472×736 grid (left), ENO 5th order with 752×376 grid (middle) and Spectral raw data with 512×512 Chebyshev collocation points (right) with $0 \leq x \leq 20$cm, $-6.5\text{cm} \leq y \leq 6.5$cm.

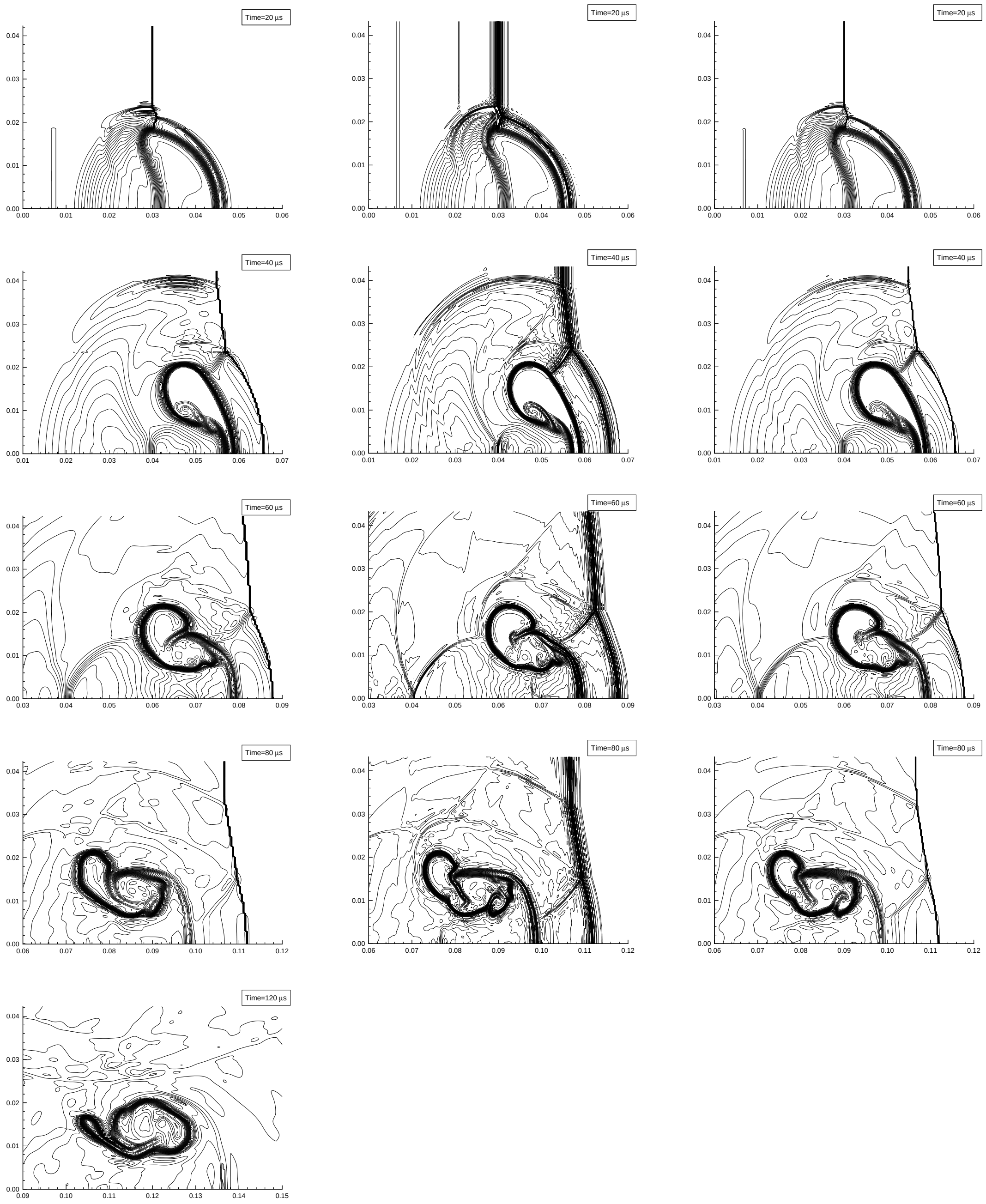


Figure 3.7: Continuous from the previous figure. Post-processed spectral data (left) at $t = 20, 40, 60, 80, 120\mu s$. Spectral raw data (middle) and post-processed data (right) with 512×512 grid for $t = 20, 40, 60, 80\mu s$ with $0 \leq x \leq 12.5$cm, $-6.5\text{cm} \leq y \leq 6.5$cm.

data agree quite well with the high resolution/high order ENO schemes. The large scale features are basically consistent between the spectral and the high resolution ENO results. The most variation is seen in the internal structure of the hydrogen cylinder, and this is best seen by plotting the mass fraction $f_{\mathrm{H}_2}$ of hydrogen. All hydrogen mass fractions are plotted with 15 contour lines with the range of $0.01 - 1.01$.

**Time=** $40\mu s$

A consistent picture can be seen as early as time $40\mu s$. Figure 3.8 shows the results of the ENO third order solution for grid size $496\times248, 752\times376$ and $1472\times736$. Figure 3.9 shows the results of the ENO fifth order solution for grid size $752\times376$ and $1472\times736$. It also shows the solution obtained by the Spectral Code with $512\times512$ Chebyshev collocation points.

At time $t = 40\mu s$, a small narrow jet of fluid begins penetrating the cylinder from the interior of the air jet and roll up to form a vortex. It is clearly seen in the figures as soon as enough grid resolution is attained. In a few relatively low grid resolution cases, the formation of vortex is not as well defined as the higher grid resolution or spectral solution due to the numerical dissipation. In general, these figures seen to be consistent and similar to each other. However, we might expect that the lower order ENO scheme would have increasing difficulties as later times are considered.

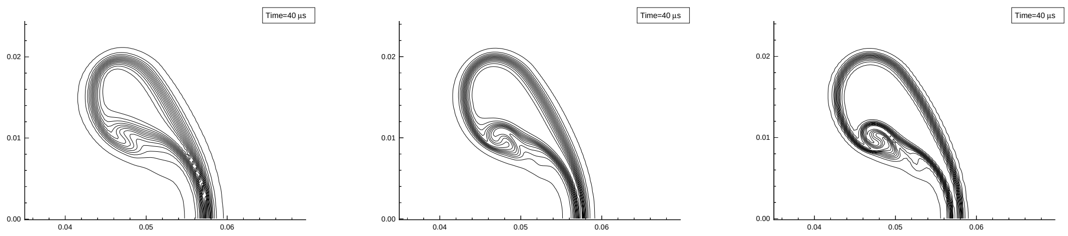


Figure 3.8: Hydrogen mass fraction for the Mach 2 Shock-Hydrogen cylinder interaction at $t = 40\mu s$. ENO 3rd order with grid size $496\times248$ (left), $752\times376$ (middle) and $1472\times736$ (right). The domain size is $0 \le x \le 20\text{cm}, -6.5\text{cm} \le y \le 6.5\text{cm}$.

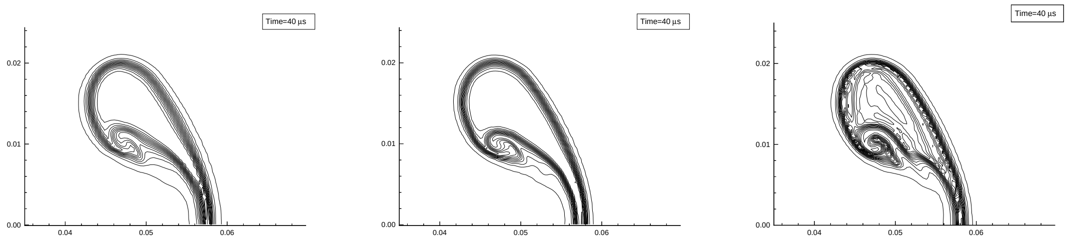


Figure 3.9: Hydrogen mass fraction for the Mach 2 Shock-Hydrogen cylinder interaction at $t = 40\mu s$. ENO 5th order with grid size 496×248 (left), 752×376 (middle) and spectral raw data with 512×512 Chebyshev collocation points (right). The domain size is $0 \leq x \leq 20\text{cm}, -6.5\text{cm} \leq y \leq 6.5\text{cm}$.

**Time=** $60\mu s$

At time $t = 60\mu s$, a jet of air penetrates the back top side of the cylinder for both the third order (figure 3.10), fifth order ENO scheme and Spectral Code (figures 3.10, 3.12) with various grid resolutions.

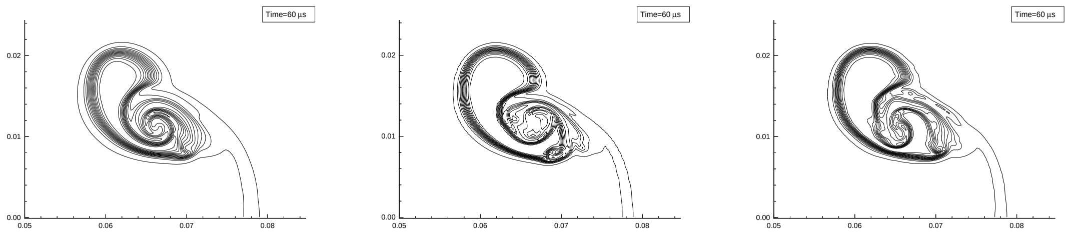


Figure 3.10: Hydrogen mass fraction for the Mach 2 Shock-Hydrogen cylinder interaction at $t = 60\mu s$. ENO 3rd order with grid size 752×376 (left) and 1472×736 (middle). ENO 5th order with grid size 752×376 (right). The domain size is $0 \leq x \leq 20\text{cm}, -6.5\text{cm} \leq y \leq 6.5\text{cm}$.

A 'mushroom' shape rollup of the air can be seen clearly in figure 3.12. It is related to the Rayleigh-Taylor instability which describes denser fluid lying above a lighter one in a gravitational field. The heavier fluid forms fingers descending down into the lighter fluid below. A similar fingering can result from the impulsive acceleration due to a propagating shock.

In the lower resolution cases this feature is captured only as a single jet that rolls up into a vortex. Higher resolution images provide a complicated picture that is rapidly becoming turbulent. Surprisingly the third order calculation on the 1472×736 grid produces a simple jet without a

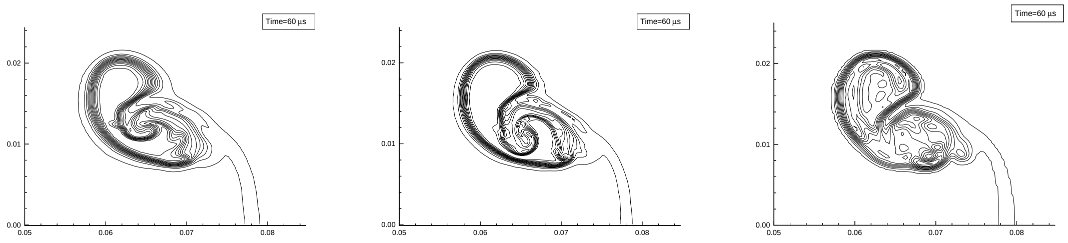

Figure 3.11: Hydrogen mass fraction for the Mach 2 Shock-Hydrogen cylinder interaction at $t = 60\mu s$. ENO 5th order with grid size 496×248 (left), 752×376 (middle) and post-processed spectral data with 512×512 Chebyshev collocation points (right). The domain size is $0 \leq x \leq 20\text{cm}, -6.5\text{cm} \leq y \leq 6.5\text{cm}$.

mushroom. But it is visible for the fifth order scheme on the 752×376 grid, which still compares fairly closely with the spectral scheme. At later times, however, features increasingly disappear due to numerical diffusion, and the ENO scheme presents a misleadingly simple picture of the internal structure (see figures 3.6 and 3.7).

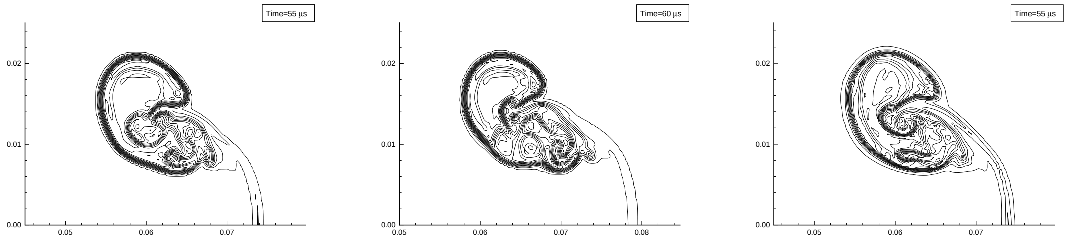

Figure 3.12: Hydrogen mass fraction for the Mach 2 Shock-Hydrogen cylinder interaction. Grid size is 512×512 Chebyshev collocation points. The domain size is $0 \leq x \leq 12.5\text{cm}, -6.5\text{cm} \leq y \leq 6.5\text{cm}$. Post-processed spectral data at $t = 55\mu s$ (left) and $t = 60\mu s$ (middle). Spectral raw data with viscosity and combustion at $t = 55\mu s$ with $0 \leq x \leq 20\text{cm}, -6.5\text{cm} \leq y \leq 6.5\text{cm}$ (right).

### 3.2.2 Discussion

The truly surprising result of the calculations described above is the amount of resolution that is required to properly resolve features within the circular fuel-jet. Pictures consistent with the expected physics are visible at quite low resolutions. For example, a beautiful roll-up in the cylinder due to vortex formation is visible in the third order ENO calculation on the 752×376 grid (figure 3.10). This picture is essentially unchanged for somewhat lower resolutions. Since this is precisely the physics that one might expect, it is natural to believe that the flow has been resolved.

But the third order 1472×736 calculation is beginning to show that the picture is not as simple as one might have hoped, and higher order calculations demonstrate clearly that more complicated phenomena are afoot. The spectral scheme provides especially valuable hints in this regard. It demonstrates these phenomena at lower grid resolutions and computational cost.

To demonstrate that this general picture remains valid even when the physical model is substantially changed, we present one plot derived from a real gas Navier-Stokes simulation which includes chemistry modeled by a one step reversible reaction $2H_2 + O_2 \leftrightarrow 2H_2O$. We show the hydrogen mass fraction raw data at time $55\mu s$ in the right-most picture of figure 3.12. Comparing with the left-most picture, the shape of the mushroom and the internal structures are slightly modified by the combustion and viscous process, but the general features are quite similar for both the Euler and the reactive Navier-Stokes simulations.

The results presented here hint very strongly at the necessity for high order schemes for calculating long-time evolution of flow fields. The major disadvantage these schemes have is the difficulty in applying these methods to irregular adaptively refined grids. The particular ENO scheme chosen is based on calculating derivatives of fluxes. It requires four times less computational work to implement this scheme than a finite volume ENO scheme (based on cell averages) in two dimensions. But unlike the finite volume schemes, there is currently no variant of the method with higher than second order accuracy that remains conservative when the grid is locally refined.

The computational burden of this particular flow problem may be greatly reduced by taking advantage of an adaptive grid. Quirk and Karni[37] reported that the cost may be decreased by up to a factor of 50 using the adaptive mesh refinement, especially for earlier times in the problem. The fact that they used a second order method could affect the reliability of their results for late times.

An additional complication of adaptive grids may be worth mentioning—the effective numerical viscosity present in their scheme depends on the local grid resolution, and this varies considerably over the computational region. It is possible that this introduces local perturbations in the flow which are amplified by the fluid instabilities. The results depicted in figure 3.3 use a regular grid,

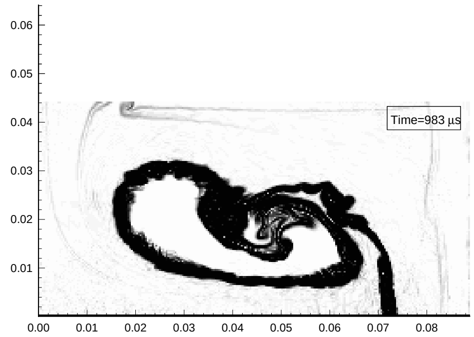


Figure 3.13: Mach 1.22 Shock-Helium cylinder interaction at time $983\mu s$, using a local Lax-Friedrich flux.

and the method chosen to do upwinding is a Lax-Friedrich flux using a wave speed chosen as the maximum over an entire line in the computational region, as described in section (2.2.1). Thus the effective numerical viscosity is quite uniform. If we change the flux to a local Lax-Friedrich flux, the numerical viscosity becomes more irregular. In figure 3.13 we present the results of a calculation using a local Lax-Friedrich flux were the maximum wave speed over the nearest 6 grid points is used. The result is very different from the previous case (figure 3.3H). The vortex has broken up, and the 'mushroom' formed by the air penetrating the downstream side of the cylinder is not recognizable. The picture presented is much less realistic and correlates much less well with the Mach 2 case.

### 3.2.3 Computational Cost

As the previous discussion makes clear, high order schemes provide more accurate solutions and resolve features better for a given grid resolution. They are computationally much more demanding. But for a given amount of CPU resources, they seem to provide better resolved results. The fifth order ENO schemes require roughly 60% more CPU time per time step, at a CFL number 1/3 to 1/2 that of the third order schemes, and so are up to five times as expensive. But for these explicit schemes, computational cost rises inversely as the third power of the grid point spacing (see table 3.2). The experience of the previous section seems to indicate that the fifth order schemes were providing superior results to third order schemes at twice the resolution by time $60\mu s$.

| Grid ($N \times M$) | Order | # Time Steps | Elapsed Time (seconds) | nodes | machine |
|---|---|---|---|---|---|
| 240×120 | 3 | 1908 | 1072 | 32 | CM5 |
| 496×248 | 3 | 4492 | 8470 | 32 | CM5 |
| 752×376 | 3 | 7534 | 31486 | 32 | CM5 |
| 240×120 | 5 | 5858 | 4930 | 32 | CM5 |
| 496×248 | 5 | 12962 | 37824 | 32 | CM5 |
| 752×376 | 5 | 20448 | 136143 | 32 | CM5 |
| 1472×736 | 3 | 15508 | 18396 | **512** | CM5 |
| 1472×736 | 5 | 42183* | 77760* | **512** | CM5 |
| 512×512 | Spectral | 3650 | 4562 | 1 | C90 |
| 800×512 | Spectral | 6900 | 13800 | 1 | C90 |

Table 3.2: CPU time requirements for the Mach 2 shock-hydrogen cylinder problem calculated to $135\mu s$. *By extrapolation from a partial computation.

The spectral method resolves better both the large and small scale structures for long time integration, and does this at lower computational cost, if measured from the point of view of the amount of time it takes a user to generate results. Figure 3.14 shows a graphic summary of the CPU time usage of all calculations run in this section. It can be seen that the spectral method is highly competitive with high order/high resolution ENO schemes when the two are compared on the same grid. Boundary conditions are more difficult for the spectral methods however, and it may be possible to carry out ENO calculations on a small enough domain so that they are competitive. In any event, finite difference methods like ENO are easier and more efficient to parallelize, and may be expected to eventually have an advantage as larger parallel machines become available.

## 3.3 Mixing and the effects of combustion and viscosity

In order to study mixing I will consider a somewhat smaller computational domain than the cases considered above. It extends over the region $-6.5\text{cm} \leq x \leq 6.5\text{cm}, 0\text{cm} \leq y \leq 6.5\text{cm}$, the initial shock is at $-2.3$cm and a 4cm diameter cylinder is located at the origin. A transformation is applied within the computation so that the grid is move along with the flow at 600m/s towards the right. As before reflection boundary conditions are applied at the top and bottom, and characteristic boundary conditions at the left and right boundaries. A sharp interface between the hydrogen and air with

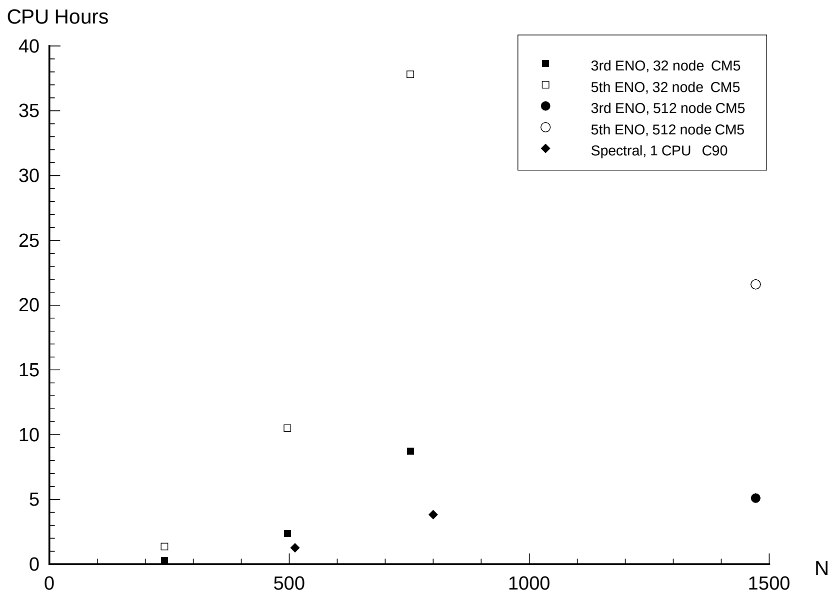


Figure 3.14: CPU hour usage of each schemes for different problem size. $N$ is the number of grid points in $x$.

$\delta = 100$ in equation (3.2) is used.

### 3.3.1 Mixing and circulation

From the point of view of building an actual engine, the most important information to be extracted from the flow field is the amount of mixing that is produced from the interaction of the shock with the cylinder. As noted above, the proposed mechanism for mixing is the generation of a vortex pair which causes air to be entrained within the cylinder and mixed with it.

Previous researchers addressed this aspect of mixing by computing the circulation $\Gamma$ about a contour $C$ in the flow field, where

$$\Gamma = \int_C u \cdot ds.$$

and by modeling the dependence of $\Gamma$ upon the Mach number, density, and equation of state of the interacting gases. This is easier to do for the cases that they considered, which were low mach number interactions. The slower time scale in these situations imply that at late times in the simulation,

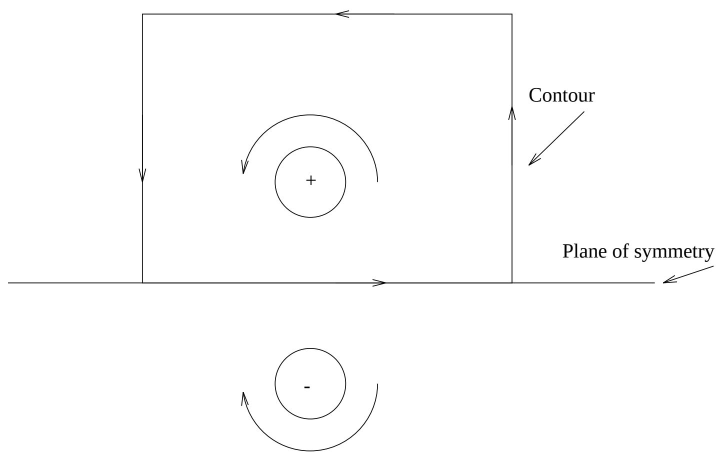


Figure 3.15: A vortex pair and a contour to find its strength

all shocks leave the computational domain. A contour $C$ may be chosen that essentially computes vorticity within and adjacent to the interacting cylinder and isn't affected by the surrounding flow. For the Mach 2 cases considered here, the complicated assortment of shocks and Mach reflections generated by the interaction never quite leave the region, (see figure 3.16), and it is difficult to place a contour to avoid them. Figure 3.17 presents the circulation computed on a contour surrounding the entire computational domain for the same two calculations depicted in figure 3.17. The measured circulation oscillates wildly as main shock passe through the hydrogen, and then drops after it emerges from the cylinder and progressively straightens out. If it were not contaminated by the influence of the shock, the circulation would be expected to reach a steady state and slowly decay due to viscosity.

The circulation is seen to be essentially the same with or without chemistry, which is quite surprising. One would expect the circulation to be less in the case with combustion: the gases are heated locally and expand, and by conservation of angular momentum, vorticity is destroyed—this is described by the $\omega\nabla\cdot v$ term in the vorticity equation (3.1). Moving the contour $C$ closer to the hydrogen cylinder results in a measurement contaminated by shocks and other waves close to the cylinder, and the measured circulation jumps by a factor of two or three. A direct comparison of circulation in this flow with previous work seems difficult.

(a)

(b)

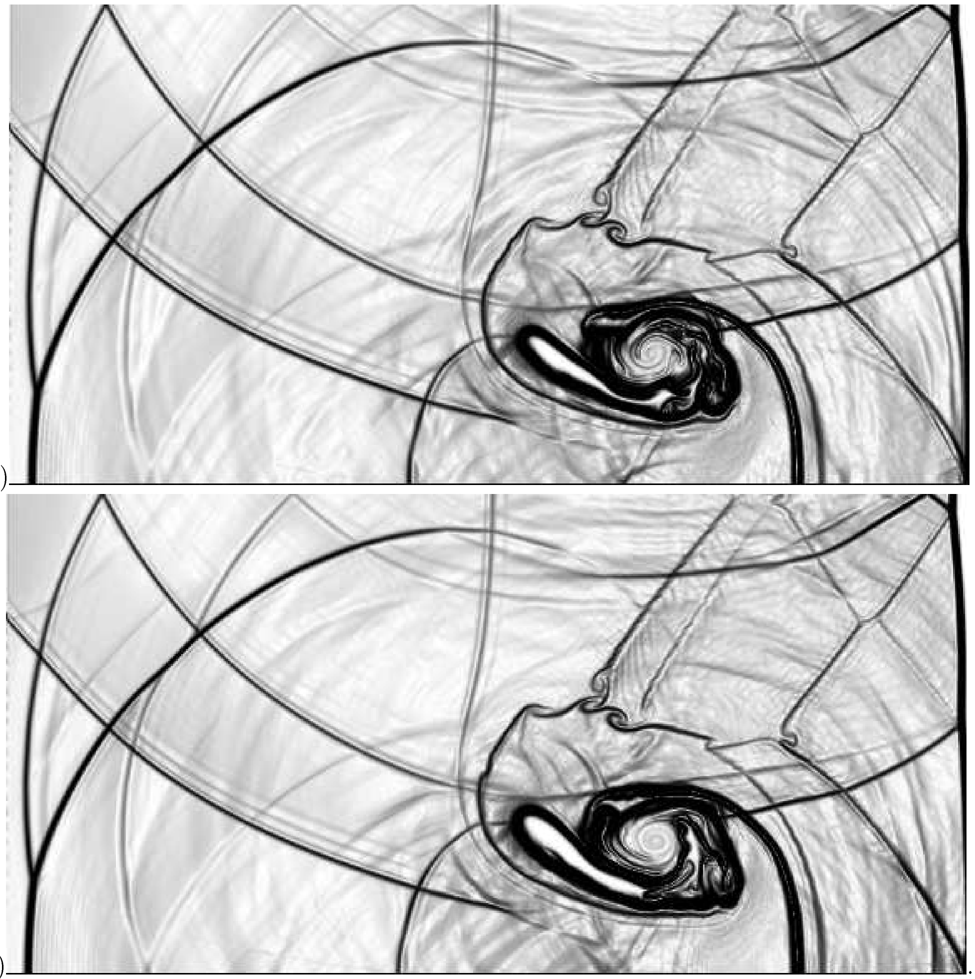

Figure 3.16: Numerical Schlieren images of a Mach 2.0 shock/hydrogen cylinder interaction at $130\mu s$ (a) with viscous modeling but no chemistry, (b) with chemistry.

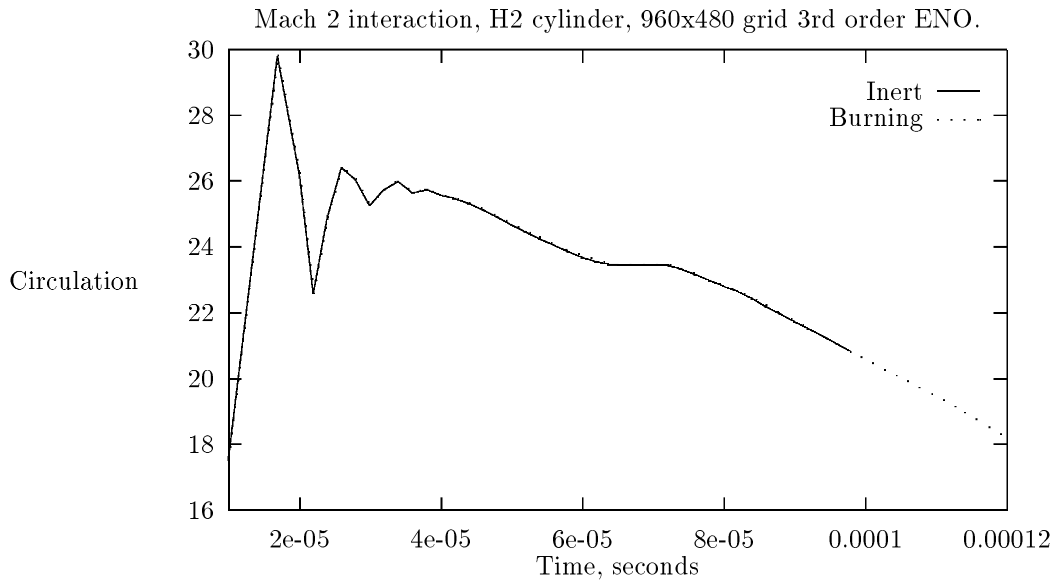


Figure 3.17: Numerical Schlieren images of a Mach 2.0 shock/hydrogen cylinder interaction at $130\mu s$ (a) with viscous modeling but no chemistry, (b) with chemistry.

### 3.3.2 Mixing measured by contour length

As a direct measurement of vorticity seems difficult, other measurements of the mixing occurring in the flow will be necessary. A variety of more direct measurements are possible.

The broadest numerical study of mixing so far has been that pursued by Yang *et al.*[59]. The approach taken there is to measure the contour length $C_l$ of the mass fraction for hydrogen at the 0.05 level, the argument being that this is the level at which stoichiometric combustion occurs in an engine, and that the combustion rate will be roughly proportional to the contour length. Because exponential stretching is expected, Yang *et al.* compute the rate of change of the logarithm of the contour length

$$\frac{d}{dt} \ln C_l(0.05)$$

and use this quantity to predict mixing.

There are a number of criticisms that may be made about this approach. First of all, it only makes sense for an inviscid non-reacting simulation, as the contour length is affected by diffusion. The effect of diffusion can enlarge contours as mass above a contour level is distributed over a larger area. Contour length also will drop suddenly as elements drop in height below a contour level. Surprisingly, a low contour level like 0.05 can have the effect of decreasing contour length if a flow consists mostly of concave objects. At lower contour levels, mass begins to fill in the concavities reducing surface area.

Despite the difficulties involved with using contour length as a measure of mixing rates, it is a strategy that I will pursue here among others, for a variety of reasons. Looking at contour lengths provides a surprising amount of information. If the lengths reported at different contour levels are very different, and this difference is increasing in time as mass moves from higher contour levels to lower ones, then this is an effective way of seeing how numerical diffusion is affecting the results. We shall use this below to provide some qualitative information about how much numerical diffusion is present in the inviscid cases. Another piece of information is also provided by the case when an element in the flow field drops below a contour level causing an abrupt jump in the trend of contour growth. When this sort of jump is seen in many different contour levels at once, the obvious conclusion is that some details in the flow are being diffused away by the numerical method. The measured contour lengths confirm the estimates of the previous section on the times to which reliable flow solutions have been calculated.

In figure 3.18 contours for the hydrogen mass fraction have been plotted for a third order inviscid computation. In the beginning contour lengths decrease as the bubble is compressed by the shock. Different contour levels remain closely packed together until about time 54$\mu$s indicating very little numerical diffusion. At this time a tendril of air has penetrated inside the hydrogen cylinder, and the situation is rapidly beginning to look turbulent. The lines begin to separate rapidly as diffusion moves hydrogen mass fraction below the contour levels. It is unfortunately the case that even with computation at very high resolution, some loss of resolution occurs about this time. The flow solution quite probably is becoming turbulent, and very little can be done about it. This case is similar to the previous runs compared with the spectral method, except a smaller numerical domain was used which permitted a higher resolution simulation. The picture is essentially unchanged with the addition of hydrogen chemistry. The results of a reacting viscous simulation with species diffusion

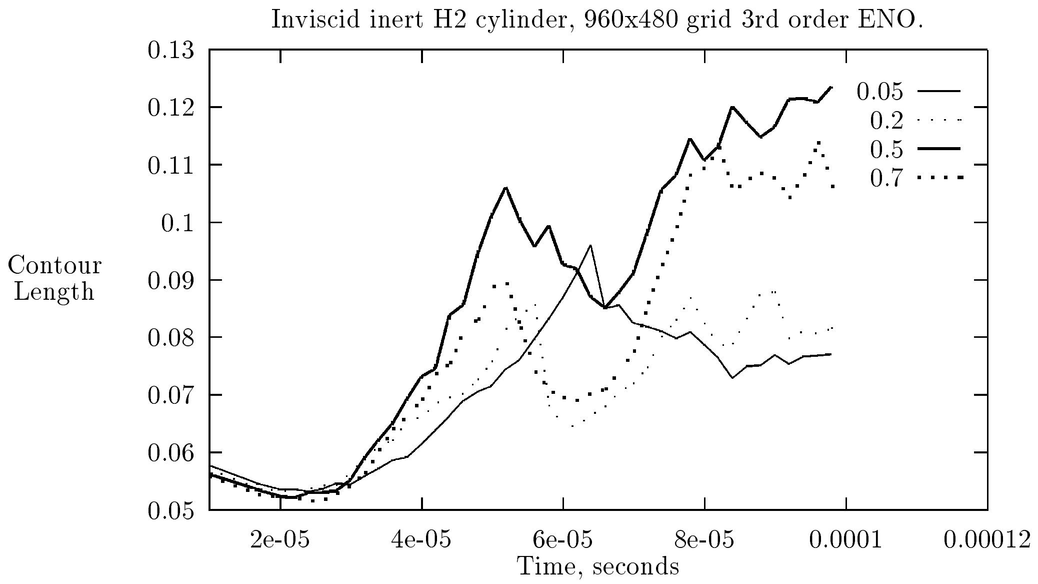


Figure 3.18: Mach 2.0 inviscid flow: contour lengths for hydrogen mass fraction of a cylinder interaction.

can be seen in figure 3.19. The fastest mixing rates measured in these graphics occur just before time $t = 54\mu$s, and are about 1 for the contour at level 0.5 when written using non-dimensionalized time $\bar{t} = tc/R_0$, $c = 620$ m/s is the sound speed in front of the shock, and $R_0 = 0.02$m is the radius of the cylinder. This is considerably faster that the rate of 0.08 measured by Yang *et al.*[59] for Helium and heavier gases at Mach 2.0.

A more direct comparison with the results of Yang *et al.* may be obtained by looking at the contour lengths derived from the simulation of Haas and Sturtevant's experiment [19] carried out in §3.1. Figure 3.20 is a graph of contour length in this case. The sound speed is 347m/s and the jet radius is 2.5cm, so the contour stretching rate is about 0.2 for the contour at level 0.5, which is about eight times the rate reported by Yang *et al.*[59] for a Mach 1.2 interaction. Part of the difference could be the much sharper interface of the initial condition. This was explored by Yang *et al.* and can considerably affect the measured rate—by a factor of three for the cases they considered. The lower resolution of their simulations and the 0.05 contour chosen probably also exaggerates this affect.

Another way of quantifying the amount of mixing that is occurring is to measure the total amount

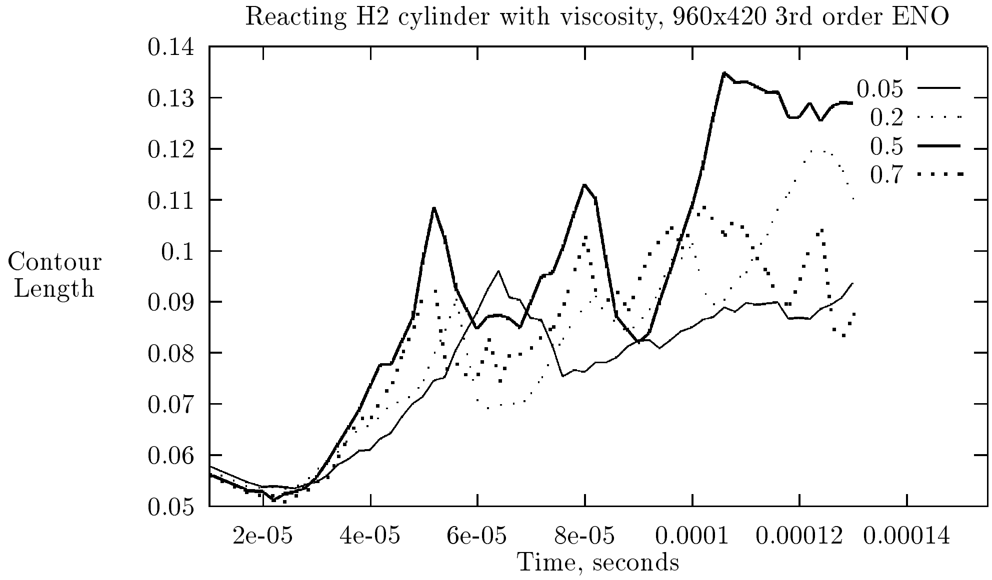


Figure 3.19: Mach 2.0 viscous combusting flow: flow contour lengths for hydrogen mass fraction of a cylinder interaction.

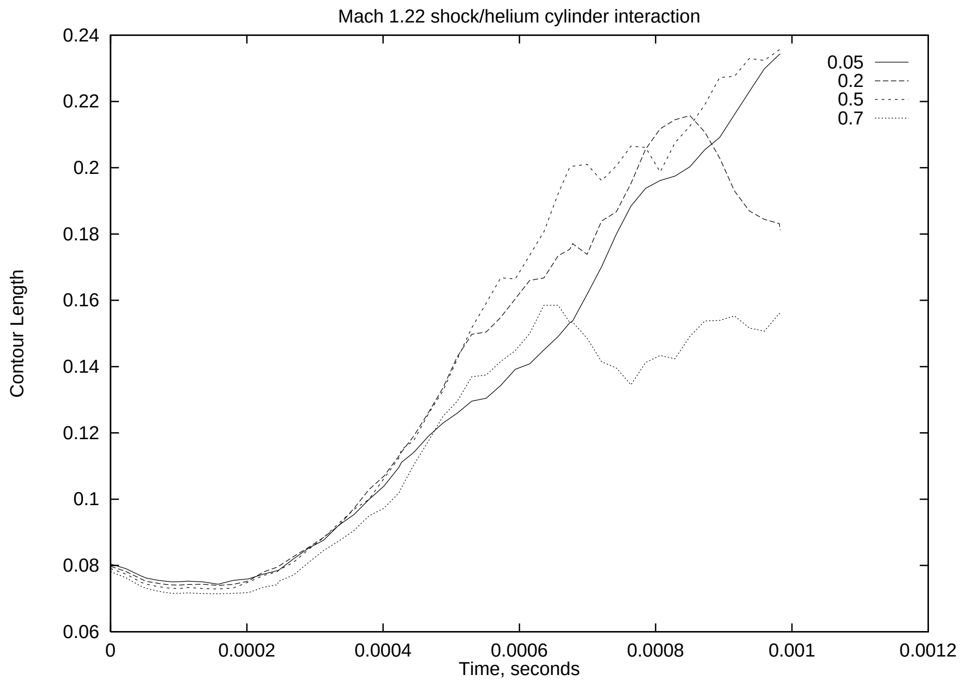


Figure 3.20: Mach 1.22 inviscid flow: flow contour lengths for Helium mass fraction of a cylinder interaction.

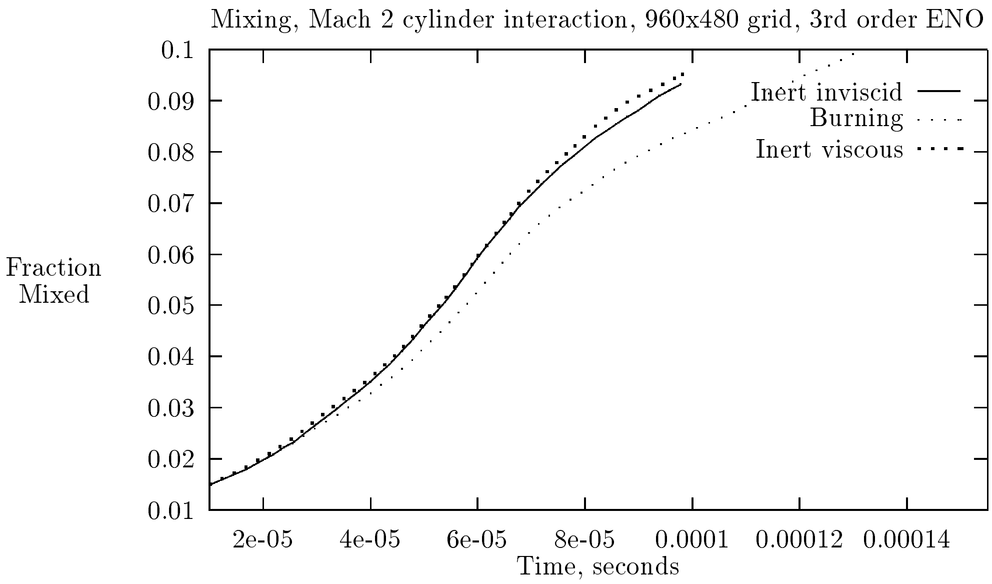


Figure 3.21: Mach 2.0 Fraction of Hydrogen that is mixed and could react.

of water that would be produced at each grid point if the reactants available there all reacted. The number of moles of water measured this way may be compared with the total number of moles of hydrogen initially present in the flow. For a completely combusting flow, the entire volume of hydrogen is consumed, and an equal number of moles of water is produced. The fraction of the hydrogen that is being consumed is plotted in figure 3.21, for both the third order non-reacting inviscid solution and the combusting case. Numerical diffusion in the scheme still is much larger than the physical diffusion, as the addition of viscosity to the modeling only slightly increases the diffusion rate at earlier times. At later times, the formation of water creates a barrier between the hydrogen and oxygen which slows diffusion. As can be seen, about 10% of the hydrogen has mixed with air and is in a position to combust in the length of time plotted here. This is a surprisingly low amount of combustion, given that the vortex towards the front of the cylinder is indeed rotating rapidly and efficiently mixing oxygen that has been entrained inside the cylinder. The problem is that very little oxygen has actually been entrained—the vortex, which initially forms at the boundary between the hydrogen and oxygen, has moved to the interior of the hydrogen, and is not creating as much mixing as might be hoped. It is quite possible to increase the amount of oxygen entrained by perturbing the initial cylinder interface, as will be seen below.

Ton[52] observed a 30% rate of combustion for a Mach 1.22 interaction computed to 1 millisecond. This seems to correspond to roughly time $100\mu s$ when compared to the Mach 2 flow here. The difference between his result and that which might be extrapolated from the calculation here could well be due to diffusion, and more specifically the low resolution of his calculation.

### 3.3.3 High wave number perturbations

Yang *et al.* use a low-order numerical method at quite low resolution, and so their solutions suffer from considerable numerical diffusion. But more important is the fact that they chose an extremely diffuse initial interface, in order to make comparisons with Jacobs'[22] experiment. The result is that the numbers they report for mixing are at best misleading, and probably underestimate the amount of mixing that is produced. As a particularly dramatic example, they observe that an initial conditions where the cylinder interface is perturbed by a sinusoidal wave $r = 1 + 0.1\cos(16\,\theta)$ where $r$ is the radius, does not enhance mixing rates over the unperturbed case. This is quite surprising, as these perturbations are affected by the Rayleigh-Taylor-Meshkov instability. While it is true that the linear theory suggests that initially perturbations at the front of the bubble should decrease in magnitude, cancelling growth at the back side of the bubble, the theory predicts that the perturbations will change sign and start to grow quite soon. Actually, this linear picture is a little misleading for a curved interface. Vorticity is deposited by the passing of the shock along the entire interface, which is much longer than in the unperturbed case. This vorticity interacts in a nonlinear fashion, causing tremendous rates of mixing at the interface. The results are visible in figure 3.22, which presents contour lengths of a fifth order simulation applied to a quite sharp initial condition, with $\delta = 100$ in equation (3.2). The interface length grows extremely rapidly compared to the previous cylindrical cases. The reason why can be seen from images of the simulation (figure 3.23). A vortex has formed within each wave of the perturbation.

As for why Yang *et al.*[59] did not observe this kind of mixing, two explanations are possible. It could be that the numerical scheme they employed diffused away the perturbation rapidly. Another

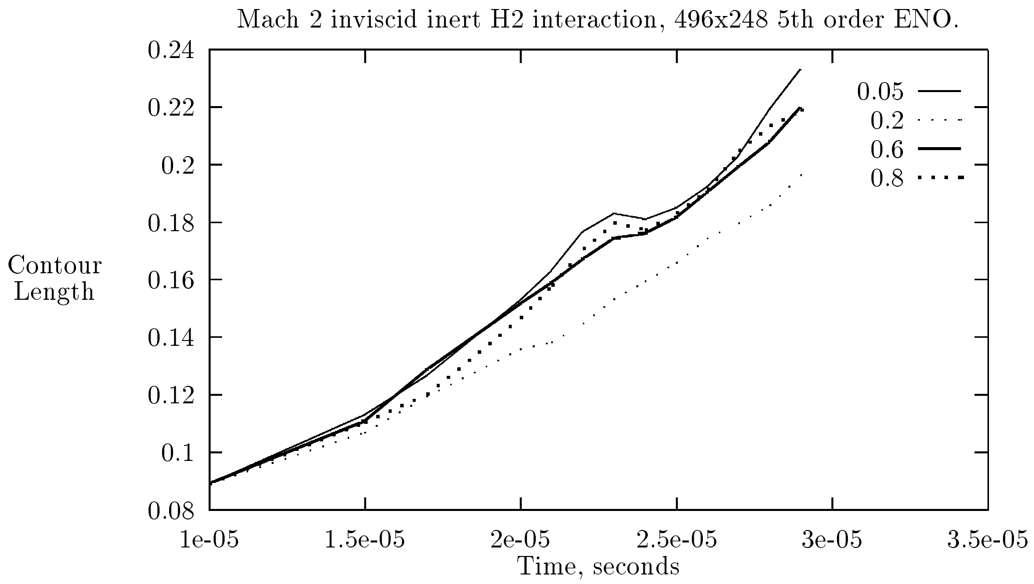


Figure 3.22: Contour lengths for hydrogen mass fraction of a Mach 2.0 shock interaction with a sinusoidally perturbed cylinder, modeled by inviscid flow without chemistry.

possibility is the fact that the initial condition they chose was quite diffuse, about as wide across as the width of the cosine waves of the perturbation. This kind of diffuse interface is known to suppress the growth rate of these instabilities.

### 3.3.4 Conclusions

Computation of this Mach 2.0 flow is seen to be a difficult problem. The numerical scheme loses a great deal of resolution even before a significant fraction of the reactants have a chance to mix. This is a problem which is seen in essentially any resolution that is practical to try, and so the results really can only be qualitative in nature. The results do seem to demonstrate that the mixing resulting from the interaction is substantially lower than previous researchers found, for a simple interaction with an unperturbed cylinder.

Nevertheless a tremendous amount of mixing can be accomplished very quickly if an initial condition can be produced with a fairly sharp interface that is perturbed by high wavenumber oscillations. High wave number perturbations would certainly be available in real life due to the relatively turbulent flow fields that would be present. A large amount of mixing can be produced

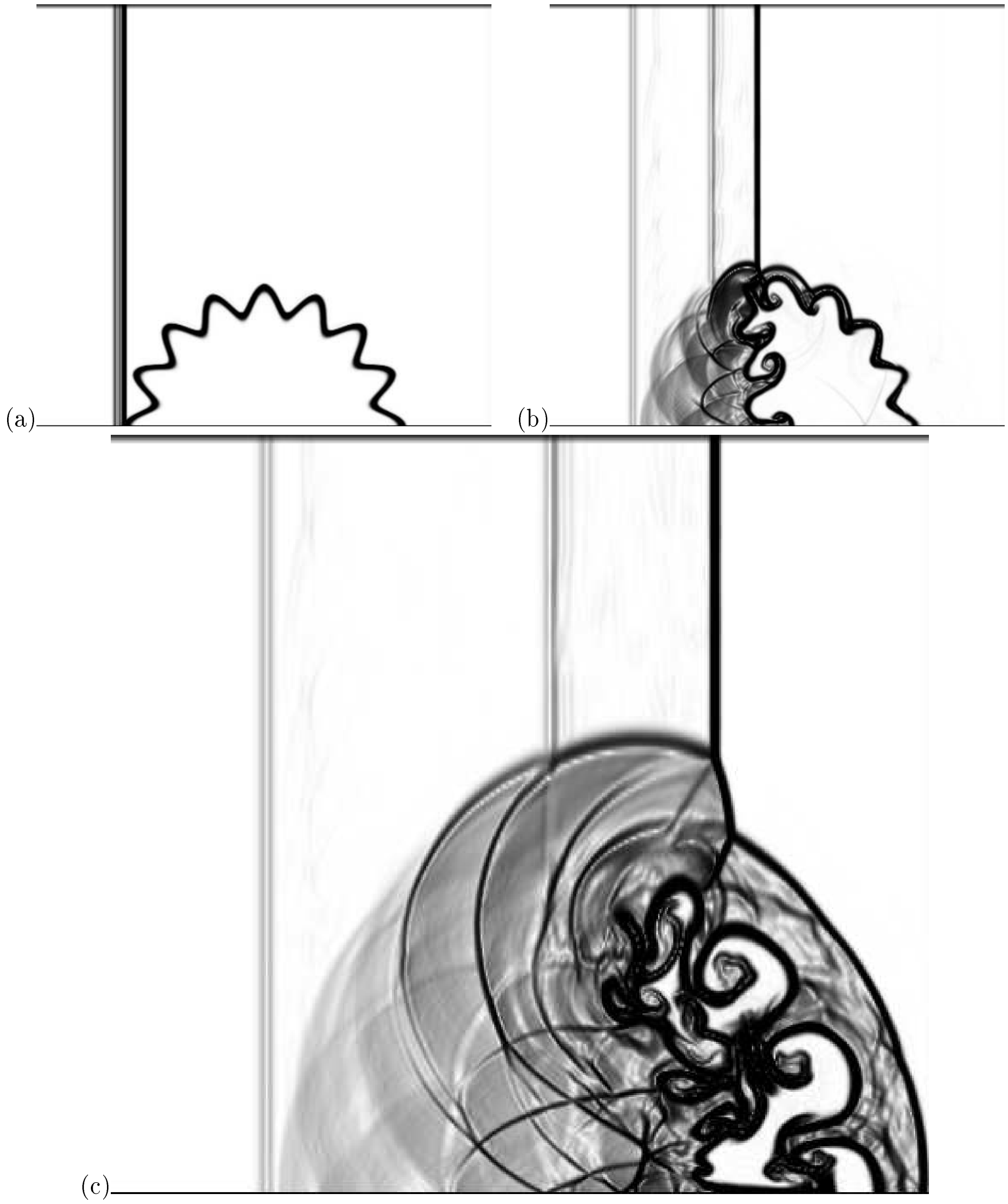


Figure 3.23: Numerical Schlieren images of a Mach 2.0 shock interaction with an inviscid sinusoidally perturbed cylinder at times (a) $1\mu$s, (b) $16\mu$s, and (c)$32\mu$s.

even before significant combustion can occur, so this mechanism is not diminished by combustion. The effect seems to be as strong as multi-jet mixture enhancement mechanisms.

# Chapter 4

# Numerical Methods and the Radon transform

## 4.1 Introduction

The Radon transform may be applied to the solution of linear hyperbolic systems of equations in much the same manner as that of Fourier transform. Fritz-John[23] developed this idea for the study of hyperbolic equations, creating a technique known as the "Method of Plain Waves." Just as the success of Fourier transform methods in solving linear PDE's suggests their use in non-linear numerical methods, the method of plain waves motivates the study of the Radon transform with a view towards its application in the creation of numerical schemes.

### 4.1.1 The Radon transform

$$\mathcal{R}: L_1(\mathbb{R}^n) \to L_1(\mathbb{R}\times S^{n-1})$$

where

$$\mathcal{R}: f(x) \mapsto \int_{\mathbb{R}^n} f(x)\delta(p - x\cdot\xi)\, dx. \tag{4.1}$$

Here $x \in \mathbb{R}^n, p \in \mathbb{R}$, and $\xi \in S^{n-1}$, the unit sphere in $\mathbb{R}^n$. I will employ the notation $\widehat{f}$ to denote the Radon transform of $f$, and use $\breve{f}$ for the inverse transform.

The Radon transform is just the set of integrals along hyperplanes perpendicular to $\xi$ at distance $p$ from the origin. It is obviously a bounded operator in $L_1$, as

$$\|\widehat{f}\|_{L_1} = \int_{S^{n-1}\cap\{\xi_0>0\}} dS_\xi \int_{-\infty}^{+\infty} |\widehat{f}(p,\xi)|\, dp \leq \int_{S^{n-1}\cap\{\xi_0>0\}} \|f\|_{L_1}\, dS_\xi = \frac{w_n}{2}\|f\|_{L_1},$$

where $w_n = \int_{S^{n-1}} dS_\xi$. The operator $\mathcal{R}$ may also be extended to $L_2$, based on the relation it has to the Fourier transform described below. While it is not surprising that the forward Radon transform is bounded, as it is an integration, we might expect that the inverse would be a form of derivative and would be a discontinuous operator.

### 4.1.2 Motivation for using the Radon transform

Consider the case of a conservation law with an isotropic flux

$$u_t + \nabla\cdot f(u) = 0. \tag{4.2}$$

Discretizing this in a way that is in fact isotropic is not easy. However, theoretically this may be accomplished in Radon space:

$$\begin{aligned} \int_{\mathbb{R}^n} (f(x+\Delta x) - f(x))\delta(p - x\cdot\xi)\, dx &= \int_{\mathbb{R}^n} f(x)\delta(p-(x-\Delta x)\cdot\xi)\, dx - \widehat{f}(p,\xi) \\ &= \widehat{f}(p+\Delta x\cdot\xi, \xi) - \widehat{f}(p,\xi). \end{aligned}$$

So by taking limits, in general the spatial derivative

$$\mathcal{R}\frac{\partial}{\partial x_j} f(x) = \xi_j \frac{\partial}{\partial p}\widehat{f}(p,\xi) \tag{4.3}$$

when transformed into Radon space. An immediate consequence is the fact

$$\widehat{\Delta} = \frac{\partial^2}{\partial p^2}. \tag{4.4}$$

If given $\widehat{u}$ one can obtain $\widehat{f}(u)$ then (4.2) may be solved by discretizing (4.3). Such a solution will conserve total mass and be rotationally invariant. In any event, this is the starting point taken here in a search for a numerical scheme.

In this chapter I will present two different methods for calculating the derivative in (4.3). We will see that there is an obvious orthogonal basis to use in Radon space, and in terms of that basis it is quite easy to create a derivative operator. A transformation of this basis to and from real space is then used to calculate the non-constant coefficient flux terms f(x). The result is a spectral scheme that I apply to solving simple convective flows. The use of Radon space for creating a spectral method is particularly advantageous, in the fact that Radon space is smoother than real space, so a spectral scheme operating there might expect higher convergence rates on a problem that is not $C^\infty$. This is related to an old idea for hyperbolic problems, which is to take the equation

$$u_t + f(u)_x = 0$$

and express it in integrated form, using

$$U = \int_\infty^x u(s)\,ds,$$

in terms of which

$$U_t + f(U_x) = 0.$$

This is known to be a good idea for non-linear $f$. It may also be advantageous for linear problems from the point of view of constructing boundary conditions. In any event, it is not easy to see how to extend this idea to more than one dimension. Radon space provides the extension. In two dimensions, and extra half-integral in smoothness is available, and in three, a full integral is available.

Another method for calculating derivatives is to use a finite difference operator. The spectral basis can be used to provide a transformation to and from real space, and the result will be an odd computation method which is really a mixed spectral-finite-difference method. The finite difference operator tried below is based on the same ENO derivative method used in the previous chapters of this thesis.

I begin below by surveying the available in inversion formulas in §4.2. The presentation is accomplished without mathematical rigor, and is intended as a quick introduction to the mathematics of tomography. A much more complete survey is available in Natterer's[35] excellent book.

In section §4.2.2 I introduce the singular value decomposition of Louis[30], which will provide the bases for the numerical schemes considered here. A spectral collocation method is proposed for two dimensions in §4.3, and its performance is evaluated for the wave equation in §4.4. In §4.5 the finite difference method in Radon space is used to implement numerical method for the wave equation.

## 4.2 Inversion formulas

The most common use of the Radon transform today is the area of tomography, especially for medical imaging. The key problem in this field is transforming data that has been acquired in Radon space to a real space picture in $\mathbb{R}^2$ (or $\mathbb{R}^3$ for MRI imaging). The focus in numerical algorithms is in constructing numerical inverses to the Radon transform, and to some extent the following presentation will suffer from this emphasis: computing the forward transform is considered really as an afterthought to finding an inversion algorithm with the desired properties.

The basic connection between the Radon and Fourier transforms is provided by the projection theorem:

$$\int_{-\infty}^{+\infty} \hat{f}(p,\xi) e^{-2\pi i p q}\, dp = \int_{-\infty}^{+\infty} \int_{R^n} f(x)\delta(p - x\cdot\xi) e^{-2\pi i p q}\, dx\, dp.$$

Interchanging the order of integration results in

$$= \int_{-\infty}^{+\infty} f(x) e^{-2\pi i x \cdot q\xi}\, dx = \hat{f}(q\xi).$$

The n-dimensional Fourier transform is just the 1-d Fourier transform of the Radon transform along the radial direction. This formula provides an immediate inversion theorem, and it is one that may be discretized along the same lines as the standard Discrete Fourier Transform (DFT). Let $N = q^n$ where $q$ is prime. Then the Discrete Radon Transform (DRT) may be defined on the $N \times N$ array $x(j,k), j = 0 \ldots N-1, k = 0 \ldots N-1$ as[16]

$$R_1(m,d) = \sum_{j=0}^{N-1} x(j, \langle d + mj \rangle_N)$$

for $d, m = 0, \ldots N-1$ and

$$R_2(s,d) = \sum_{j=0}^{N-1} x(\langle d + qsj \rangle_N, j)$$

where $d = 0, \ldots, N-1 \qquad s = 0, \ldots, N/q - 1$. Here $\langle j \rangle_N$ denotes $j$ mod $N$. In terms of these quantities the 2-d DFT may be written as

$$\hat{x}(\langle N - ml \rangle_N, l) = \sum_{d=0}^{N-1} R_1(m,d) e^{-2\pi i l d/N}$$

for $m = 0, \ldots, q^n - 1$, and

$$\hat{x}(k, \langle N - qsk \rangle_N) = \sum_{d=0}^{N-1} R_2(s,d) e^{-2\pi i k d/N}$$

for $s = 0, \ldots, q^{n-1} - 1$. What has happened here is that the DRT has been defined for integrals along lines of the slopes available in the lattice of points used by the standard DFT. Unfortunately as can be seen in figure 4.1 different lines are sampled at very different resolutions, and the picture depends strongly on the periodicity of the geometry in order to be interpreted as a Radon transform at all. The DRT has been exploited to derive algorithms faster than the standard 2-d FFT, especially in

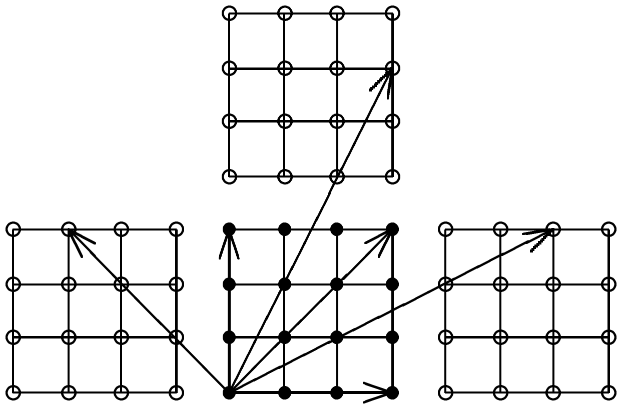

Figure 4.1: Directions available with a 4×4 DFT lattice.

situations where multiplication is much more expensive than addition[57]. But it is not clear that it is of any use for the type of numerical problems considered here.

This problem of unevenly sampled line integrals may be fixed if an interpolation is carried out in wave space. If the Radon transform $\hat{f}(p_j, \xi_k)$ is sampled at evenly spaced points $p_j$, then we may apply the projection theorem if $\mathcal{F}_p^{-1}\hat{f}$ is interpolated in the right way on a square lattice of points in wave space. Here $\mathcal{F}_p^{-1}$ is the inverse 1-d Fourier transform of the $p$ variable. Natterer[35] has established that band-limited interpolation suffices. The forward transform may also be performed via interpolation. While the algorithm may appear to have higher efficiency than the transforms considered below, the interpolation required for use of this discretization is probably an expensive calculation, especially for functions that are not very smooth. This algorithm is most advisable with functions constant outside a compact region, due to the fact that the radial transform will disagree with the transform in Cartesian coordinates on the domain on which periodicity is expected. An additional disadvantage is the fact that depending on how interpolation was accomplished this forward transform may not be the exact discrete inverse of the reverse transform. For the purposes considered here, this problem and the difficulties associated with periodicity suggest the use of other discrete transforms.

### 4.2.1 The classical inversion formula in 2-D

The standard algorithms for inverting the Radon transform are derived from the classical inversion formula, which is easily derived using Fourier analysis of distributions from the projection theorem (following the argument of Rowland[41]):

$$f(x) = \int_{\mathbb{R}^n} \hat{f}(\omega) e^{2\pi i\omega\cdot x}\, d\omega, \qquad x \in \mathbb{R}^n.$$

Expressing this in polar coordinates $q \in \mathbb{R}, \xi \in S^1$ in wave space:

$$f(x) = \int_0^\pi \int_{-\infty}^{+\infty} \hat{f}(q,\xi) e^{2\pi i x\cdot\xi q} |q|\, dq\, dS_\xi.$$

Now

$$\int_{-\infty}^{+\infty} \hat{f}(q,\xi) e^{2\pi i p q} |q|\, dq = \check{|q|} * \widehat{f}(p,\xi)$$

where the convolution is in the $p$ variable, and $\check{|q|} = \int_{-\infty}^{+\infty} |q| e^{2\pi i q p}\, dq$ in the distribution sense. So

$$f(x) = \int_0^\pi (\check{|q|} * \widehat{f}(p,\xi))_{|_{x\cdot\xi}}\, dS_\xi = \frac{1}{2}\mathcal{B}(\check{|q|} * \widehat{f}). \tag{4.5}$$

$\mathcal{B}$ is known as the "back projection" operator: it is the adjoint operator to the Radon transform, for the obvious inner products in Radon space and real space:

$$\mathcal{B}\colon S^{n-1} \times \mathbb{R} \to \mathbb{R}^n$$

$$\mathcal{B}\colon \widehat{f}(p,\xi) \mapsto \int_{S^{n-1}} \widehat{f}(x\cdot\xi,\xi) dS_\xi.$$

$$\begin{aligned}
(\mathcal{B}\colon \widehat{f}, g) &= \int_{\mathbb{R}^n} \int_{S^{n-1}} \widehat{f}(x\cdot\xi,\xi) g(x)\, dS_\xi\, dx \\
&= \int_{\mathbb{R}^n} \int_{S^{n-1}} \int_{\mathbb{R}} \widehat{f}(p,\xi) \delta(p - x\cdot\xi) g(x)\, dp\, dS_\xi\, dx \\
&= \int_{S^{n-1}} \int_{\mathbb{R}} \widehat{f}(p,\xi) \widehat{g}(p,\xi)\, dp\, dS_\xi
\end{aligned}$$

$$= \quad (\hat{f}, \hat{g}).$$

The classical inversion formula may now be easily obtained from equation (4.5). This is accomplished by writing $|q|$ as $\mathrm{sgn}(q)q$, which in real space may be interpreted as the differential operator

$$-\frac{1}{2\pi x} * \frac{\partial}{\partial x}.$$

This may be established by computing the Fourier transform of $1/(x \pm i\epsilon)$ via contour integration and then taking the limit as $\epsilon \downarrow 0$. Substituting this into (4.5) results in the the inversion formula

$$f = -\frac{1}{4\pi}\mathcal{B}\left(\frac{1}{p} * \frac{\partial}{\partial p}\hat{f}(p, \phi)\right)$$

This formula is discretized directly in the most popular inversion algorithm, the so-called 'filtered back projection' algorithm. Typically, the operator $\frac{1}{p}\frac{\partial}{\partial p}$ is implemented approximately by convolution with a kernel that is a smoothed version of $\check{|q|}$. Considerable lore exists on the best type of kernel to use.[35, 41] Using this formula to obtain a forward transform is somewhat harder. But it may be accomplished by realizing that the back-projection operator applied to $\hat{f}$ is proportional to the potential $\frac{1}{r^{n-1}} * f$:

$$\begin{aligned}
\mathcal{B}\hat{f}(y) &= \int_{S^{n-1}} \int_{-\infty}^{+\infty} \int_{\mathbb{R}^n} f(x)\delta(p - x\cdot\xi)\delta(p - y.\xi)\, dx\, dp\, dS_\xi. \\
&= \int_{S^{n-1}} \int_{\mathbb{R}^n} f(x)\delta((x - y)\cdot\xi)\, dx\, dS_\xi \\
&= \int_{\mathbb{R}^n} \frac{1}{|x - y|^{n-1}} f(x) \int_{S^{n-1}} \delta(\eta\cdot\xi)\, dS_\xi\, dx.
\end{aligned}$$

where $\eta = (x - y)/|x - y|$. Now $c_n = \int_{S^{n-1}} \delta(\eta\cdot\xi)\, dS_\xi\, dx$ is a constant. $c_2 = 1$, for example. So we have

$$\mathcal{B}\hat{f}(y) = c_n \int_{\mathbb{R}^n} \frac{1}{|y - x|^{n-1}} f(x)\, dx. \tag{4.6}$$

This suggests that the back-projection operator may be inverted via deconvolution. If this operation is carried out by FFT, the forward Radon transform may be implemented by directly inverting (4.5).

The deconvolution itself is not a straightforward operation—some information has been lost by the convolution in (4.6) which eliminates the 0 wavenumber. Another problem with this formula will be that the discrete versions of the forward and backward transforms will not be exact discrete inverses. As before, one might implement either the forward or inverse transform by employing the conjugate transform in an iterative solver. But this solution will be less accurate and less efficient than might be desirable.

Deconvolution of the back-projection of a function in Radon space may be used to compute an inverse Radon transform in the same way. This operation will suffer from all the same problems as the filtered back-projection, with the additional complications imposed by loss of the 0 wavenumber (and partial loss of other low wave numbers caused by smoothing in the discrete implementation).

### 4.2.2 Other inversion algorithms

The previous discussion considered algorithms based on quite complicated inversion formulas that are discretized only with the aid of assorted interpolations and filtering. Practical algorithms based on these techniques tend to rely on filters with characteristics chosen to optimize the esthetics of the resulting images. Quantifying the nature of this machinery and for the purpose of a numerical scheme is a daunting task. There are still some other transforms to consider. The most obvious is a forward transform based on discretizing the Radon integral (4.1). One might choose a basis for the region in $\mathbb{R}^n$ of interest, for example the piecewise constant functions on rectangles. One can then calculate the contribution of each rectangle to any given line integral. The Radon transform may then be expressed as a $M \times N$ sparse matrix operator, where M is the number of different line integrals desired, and N is the number of rectangles in real space. Arranging for an invertible matrix will then depend on the position of the line integrals computed. In any event, one would expect the matrix to be fairly ill conditioned. Iterative matrix solvers have been developed that perform adequately in this setting. But again, due to the large matrices involved, achieving good performance with these algorithms may be quite difficult. On the other hand, the choice of basis function in real space may be quite flexible, and may allow for accurate treatment of discontinuities in real space.

There is another disadvantage that this and all the previous inversion algorithms possess. The Radon transform does not have full range. To quote a precise statement (see Smith-Solmon-Wagner[48], thm. 13.4)

**Theorem 1** *Let $\alpha \geq (1-n)/2, g \in H^{\alpha+(n-1)/2}(X)$ is the Radon transform of $f \in H_0^\alpha$ if and only if*

1. *$g(p,\xi) = 0$ for hyperplanes missing $\Omega$.*

2. *$g$ is even on $X$.*

3. *for all $m \in \mathbb{N}, \int p^m g(p,\xi)dp$ is a homogeneous polynomial in $\xi$ of degree $\leq m$.*

Ensuring that this consistency requirement has been met with a numerical scheme running in Radon space is obviously a problem, one that the previous inversion techniques provide little assistance with. One might hope to find a basis for Radon space that satisfies these conditions, and express a numerical scheme in terms of this basis. Such bases exist, and may be exploited to provide inversion formulas in so-called series methods.

Series methods involve exploiting the linearity of the Radon transform by expanding Radon space in terms of a set of basis functions, and then computing the inverse of each basis function separately. In fact, series expansions exist for functions with support on the unit disk ($f: \mathbb{R}^n \to \mathbb{R}, \quad f \in L_2(D^n), \quad f(x) \mapsto 0$ for $|x| > 1$) with ideal properties. Incredibly, the right basis functions remain orthogonal with respect to appropriate weight functions as they are mapped between Radon space, real space, *and* Fourier space[30]. In Radon space the basis consists of products of Gegenbauer polynomials $C_s^\nu$ with spherical harmonic functions $Y_l$. These transform to products of Jacobi polynomials $P_l^{(\alpha,\beta)}$ with spherical harmonics in real space, and the products of Bessel functions of the first kind $J_l$ with spherical harmonics in Fourier space. Here I state the results of Louis[30]: given $\nu > n/2 - 1$ and,

$$f: \mathbb{R}^n \to \mathbb{R}, \quad x \in \mathbb{R}^n, \quad r = |x|, \quad \eta = x/r$$

$$f(r\eta) = \sum_{s>0,l>0} a_{s,l}\, c(n,s,\nu,l)(1-r^2)^{\nu-n/2} Q_{s,l}^{\nu,n}(r) Y_l(\eta)$$

with

$$Q_{s,l}^{\nu,n}(r) = r^l P_s^{(\nu-n/2,l+n/2-1)}(2r^2-1)$$

and

$$c(n,s,\nu,l) = 2^{1-2\nu}\pi^{1-n/2}\frac{\Gamma(2s+l+2\nu)\Gamma(s+1)}{\Gamma(2s+l+1)\Gamma(\nu)\Gamma(s+1+\nu-n/2)}.$$

$$\hat{f}(q\omega) = \sum_{s>0,l>0} a_{s,l}(-1)^{2s+l}\pi^{\frac{1}{2}}2^\nu\,\Gamma\left(\nu+\frac{1}{2}\right) C_{2s+l}^\nu(1)\left(\frac{q}{2\pi}\right)^{-\nu} J_{2l+s+\nu}\left(\frac{q}{2\pi}\right) Y_l(\omega).$$

In radon space,

$$\hat{f}(p,\xi) = \sum_{s>0,l>0} a_{s,l} w_\nu C_{2s+l}^\nu Y_l(\xi).$$

$Y_l$ is a spherical harmonic function of order $l$, which will depend in general on $f$. $Y_l$ can be decomposed further using a basis for the spherical harmonics to arrive at a basis independent of $f$. For example in $S^1$, $Y_l$ will be some linear combination of $e^{2\pi il}$ and $e^{-2\pi il}$. The function $w_\nu(p) = (1-p^2)^{(\nu-1/2)}$ is the weight for the Gegenbauer polynomials $C_l^\nu$. The Jacobi polynomials $P_l^{(\alpha,\beta)}(x)$ are orthogonal with respect to the weight function $|1-x|^\alpha|1+x|^\beta$, and the Bessel functions $J_l(r)$ are orthogonal with respect to the weight function $1/r$.

The bases suggested above have merit for application in spectral schemes in their own right. They are inherently multi-dimensional, and despite the fact that they are defined on n-dimensional disks, There is no lack of smoothness at the origin. In fact, there is some example in the published literature of their use. Leonard and Wray[26] make use of a basis in two dimensions which is similar to the $n = 2, \nu = 1$ case, and they derived a Galerkin method for pipe flow from this. They apparently arrived at this basis by insisting on bandedness for the resulting system of ODE's, which is considerably assisted by formulas like (4.4).

Recently Dubiner[14] proposed bases for the disk $D^2$ for use in spectral schemes on the disk and pipe flow. One basis proposed, modulo typographic errors, appears to be the same as the one used here. The quadrature points he suggests are the same as the ones used here.

It is worth mentioning that these bases may have advantages over Cartesian product bases even

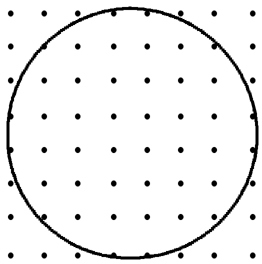

Figure 4.2: The Fourier modes compared with a circle.

when rotational symmetry isn't essential. If we look at the Fourier modes on the square $[0,1]\times[0,1]$, for example (figure 4.2) we see that real space hasn't been uniformly sampled. Modes with larger distance from the origin correspond to smaller length scales. In some sense, the modes outside the circle in the corners are wasted. Not only is extra storage required for them, but they affect the time step, as they have larger eigenvalues for derivative operators. One would expect that basis functions on rotationally invariant domains might have some advantage in this regard, and that there will be some room for algorithmic complexity and other inefficiencies when using the above bases of Louis.

**Connections with P.D.E. eigenfunctions**

Recently Batcho and Karniadakis[2] demonstrated that spectral bases with good approximation qualities may be derived from singular P.D.E operators, and were even able to apply constraints to the problem in order to ensure that the resulting eigenfunctions were solenoidal, a useful feature for simulating incompressible flow. It is interesting to note that there is a connection between the basis functions here and eigenfunctions of certain partial differential operators. Thus it may eventually be possible to derive eigenfunctions using the techniques of Batcho and Karniadakis for general geometries and at the same time exploit the enhanced smoothness of a related Radon space basis.

The connection arises because of the orthogonality properties of Louis's bases. They provide a singular value decomposition of the Radon operator, using the $L_2$ inner product on the disk in real space and the $L_2$ inner product in Radon space. This implies that the basis elements are respectively left and right eigenfunctions of the operator $R^*R$. Now we saw that $R^*$ is the back-projection operator $\mathcal{B}$ in §4.2.1, and (4.6) provides a formula for $R^*R$

$$R^*R = c_n \int_{\mathbb{R}^n} \frac{1}{|y-x|^{n-1}} f(x)\,dx.$$

The $\frac{1}{|y-x|^{n-1}}$ resembles the Green's function for Laplace's equation $\frac{1}{|y-x|^n}$, and probably is the Green's function for for a related partial differential operator. Eigenfunctions for this operator will be eigenfunctions of $R^*R$.

I will not explore this connection in detail here, principally because it is quite likely that the ideas are already present in the literature. A paper by Trimèche[53] seems to explore this subject area. It is unfortunately quite difficult to read.

## 4.3 Two dimensions and the Zernike polynomials

In the remainder of this chapter, in order to avoid experiencing the complications of higher dimensional spherical harmonics, I will consider the two dimensional case $n = 2, \nu = 1$. In this case one obtains Chebyshev polynomials of the second kind in Radon space, and Zernike polynomials in real space:

$$\hat{f}(p,\phi) = \sum_{s=0}^{\infty} \sum_{l=-\infty}^{\infty} a_{s,l} \sqrt{(1-p^2)}\, U_{2s+|l|}(p)\, e^{2\pi i l\phi} \tag{4.7}$$

$$f(r,\theta) = \sum_{s=0}^{\infty} \sum_{l=-\infty}^{\infty} a_{s,l} \frac{2s+|l|+1}{2} Z_{2s+|l|}^{|l|}(r)\, e^{2\pi i l\theta} \tag{4.8}$$

where

$$Z_{2s+l}^{l}(r) = r^l P_s^{(0,l)}(2r^2-1). \tag{4.9}$$

Table 4.1 (taken from Deans[11]) lists the first few basis functions in real space.

An orthogonal expansion in terms of Bessel functions is available for Fourier space, but it is probably worth mentioning that the picture in wave space presented by this basis is affected by the fact that the real space function is discontinuous at the surface of $S^1$, and is zero outside the unit disk $D^2$. The Fourier transpose of this will be affected everywhere by this discontinuity, and this

Table 4.1: The first few functions in the Zernike basis in real space.

| | $l+2s$ | | | | | |
|---|---|---|---|---|---|---|
| $l$ | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 | 1 | | $2r^2-1$ | | $6r^4-6r^2+1$ | |
| 1 | | $re^{2\pi i\theta}$ | | $(3r^2-2r)e^{2\pi i\theta}$ | | $(10r^5-12r^3+3r)e^{2\pi i\theta}$ |
| 2 | | | $r^2e^{4\pi i\theta}$ | | $(4r^4-3r^2)e^{4\pi i\theta}$ | |
| 3 | | | | $r^3e^{6\pi i\theta}$ | | $(5r^5-4r^3)e^{6\pi i\theta}$ |
| 4 | | | | | $r^4e^{8\pi i\theta}$ | |
| 5 | | | | | | $r^5e^{10\pi i\theta}$ |

will make calculating derivatives impractical in Fourier space.

Now that we have a basis to work in, we need to establish two things before using this basis in a spectral scheme. One is the approximation theory for that basis. It would be nice if smooth functions in real space were approximated with spectral accuracy in real space. The easiest way to do this is by comparison with expansions that are known to converge spectrally. It turns out we may compare with the Legendre polynomials $P_s = P_s^{(0,0)}$ because

$$\{Z_{2s+|l|}^{|l|}(r)e^{2\pi il\theta}\}$$

are orthogonal with respect to the constant weight function on the disk $D^2$ and for $2s+|l| \leq c$ span the same space as the polynomials $\{x^ay^b : a+b \leq c,\ (a,b,c \in \mathbb{N})\}$. Both these claims are easy to establish. In polar coordinates the orthogonality relationship is

$$\int_0^1 \int_0^1 Z_{2s+|l|}^{|l|}(r) Z_{2t+|m|}^{|m|}(r) e^{-2\pi il\theta} e^{2\pi im\theta} r\, dr\, d\theta$$

$$= \delta_{lm} \int_0^1 r^{2|l|+1} P_s^{(0,|l|)}(2r^2-1) P_t^{(0,|l|)}(2r^2-1)\, dr.$$

Substituting $r = \sqrt{\frac{x+1}{2}}$

$$= \delta_{lm} 2^{-l-2} \int_{-1}^{+1} (x+1)^{|l|} P_s^{(0,|l|)}(x) P_t^{(0,|l|)}(x)\, dx = \delta_{lm}\delta_{st} \frac{1}{2(|l|+2s+1)}. \tag{4.10}$$

Thus the Zernike polynomials are orthogonal polynomials in $r$. If $l$ is odd they have only odd powers of $r$, and if $r$ is even they have even powers. Now

$$x^a y^b = r^a \cos(2\pi\theta)^a r^b \sin(2\pi\theta)^b = \left(\frac{r}{2}\right)^{a+b} \left(e^{2\pi i\theta} + e^{-2\pi i\theta}\right)^a \left(ie^{-2\pi i\theta} - ie^{2\pi i\theta}\right)^b.$$

This consists of terms $r^{a+b}e^{2\pi i\theta(a+b-2j)}, j \in \mathbb{N}$ and these are in the span of $\{Z_{2s+|l|}^{|l|} e^{2\pi il\theta} : 2s + |l| \leq a + b\}$.

**Proposition 1** *If $f : [-1,1]\times[-1,1] \to \mathbb{R}, f \in C^\infty([-1,1]\times[-1,1])$ and $\Pi_z^N f$ is the $L_2$ projection of $f$ onto the subset Zernike polynomial basis*

$$\left(\Pi_z^N f\right)(r', \theta') = \sum_{2s+|l|<N} Z_{2s+|l|}^{|l|}(r')\, e^{2\pi il\theta'} \frac{1}{2(2s+|l|+1)} \int_0^1 \int_0^1 f(r,\theta) r Z_{2s+|l|}^{|l|}(r)\, e^{2\pi il\theta}\, dr\, d\theta$$

*and $\Pi_p^N$ is the projection of $f$ onto the Legendre polynomials*

$$\left(\Pi_p^N f\right)(x', y') = \sum_{l+m<N} P_l(x')P_m(y') \frac{1}{\|P_l\|_{L_2} \|P_m\|_{L_2}} \int_{-1}^1 \int_{-1}^1 f(x,y) P_l(x) P_m(y)\, dx\, dy$$

*Then $\|f - \Pi_z^N f\|_{L_2} \leq \|f - \Pi_p^N f\|_{L_2}$.*

Proof: Let

$$\|f(x)\|_{L_2(\Omega)} = \sqrt{\int_\Omega f^2(x)\, dx}.$$

$\Pi_p^N f(x,y)$ is in the span of $x^a y^b : a + b \leq N$, so $\Pi_p^N f(x,y)_{|D^2}$ is in the span of the Zernike basis $Z_{2s+|l|}^{|l|} e^{2\pi il\theta} : 2s + |l| \leq N$. So

$$\|f - \Pi_z^N f\|_{L_2(D^2)} \leq \|f - \Pi_p^N f\|_{L_2(D^2)} \leq \|f - \Pi_p^N f\|_{L_2([-1,1]\times[-1,1])}. \quad \blacksquare$$

This is enough to establish spectral convergence in the case that a function $f$ on $D^2$ can be extended to be $C^\infty$ on $[-1,1]\times[-1,1]$. But in general, $f$ may lack smoothness on a closed subset $A \subset \{[-1,1]\times[-1,1] \setminus D^2\}$. For a single isolated singularity, one could of course rotate the Legendre basis to avoid it. But in general one will have to patch up $f$ and make it smooth. This is easy enough to accomplish. Let $d > \epsilon$ be the distance between $D^2$ and $A$. $A$ is compact, so we may find a finite cover of $A$ consisting of disks $D^2_{\epsilon/2}(x_j) = \{y : |y - x_j| < \epsilon/2\}$. Let $g(x)$ be a smooth function such that $g : D^2_{\epsilon/2}(0) \to 0$, $g : \mathbb{R}^2 \setminus D^2_{\epsilon}(0) \to 1$. Then $\prod_i g(x - x_i) f(x)$ is smooth on $[-1,1] \times [-1,1]$ and identical to $f$ on $D^2$.

### 4.3.1 Collocation points for the Zernike basis

The fact that the Zernike polynomials are orthogonal with respect to different polynomial weight functions permits the use of Gaussian integration on the Legendre polynomial quadrature points in order to construct methods similar to classical collocation schemes. Suppose $f(r, \theta)$ is to be expanded in a subset of the Zernike basis. There are two indices, so one may choose different subsets to do this expansion in. For now, I will confine choice to a set that is generated by polynomials of order $\leq N/2$, where $N$ is a power of two. Let $S = \{(s, l) : 2s + |l| \leq N/2 \text{ and } l > -N/2\}$. then we may express a function in this subset of the Zernike basis as

$$f(r, \theta) = \sum_{(s,l) \in S} b_{s,l} Z^{|l|}_{2s+|l|}(r) e^{2\pi i l \theta} = \sum_{l=-N/2+1}^{N/2} g_l(r) e^{2\pi i l \theta} \tag{4.11}$$

If we discretize in the radial direction by sampling $\theta$ by the set $\{\theta_j = j/N : 0 \leq j \leq N-1\}$, then we may obtain the radial functions $g_l(r)$ by doing a DFT:

$$g_l(r) = \sum_{j=0}^{N-1} f(r, \theta_j) e^{-2\pi i l \theta_j}.$$

Now

$$b_{t,l} = 2(2t + |l| + 1) \int_0^1 g_l(r) Z^{|l|}_{2t+|l|}(r)\, r\, dr.$$

Now make the same substitution used to arrive at equation (4.10), $r = \sqrt{\frac{x+1}{2}}$. The integral becomes

$$b_{t,l} = 2(2t + |l| + 1) \sum_{2s \leq N-|l|} b_{s,l}\, 2^{-l-2} \int_{-1}^{+1} (x+1)^{|l|} P_s^{(0,|l|)}(x) P_t^{(0,|l|)}(x)\, dx$$

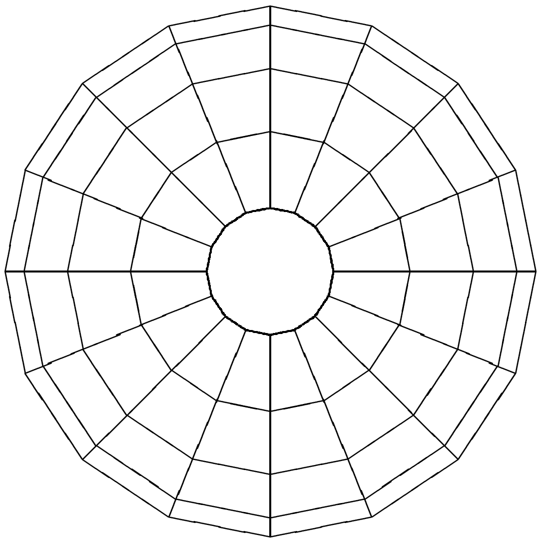

Figure 4.3: The real space computational grid for $N = 16$

which is the integral of a polynomial of order $\leq N/2$. This may be performed exactly by using Gaussian integration. Gaussian quadrature requires $N/4+1$ points to integrate polynomials up to order $N/2+1$ exactly, for $N/2$ even. Gauss-Radau quadrature is an obvious choice to avoid wasting the one degree of freedom. On $N/4+1$ points it is exact for polynomials of order $N/2$ and it allows the choice of one integration point, which in this geometry is most advantageously placed at the boundary $x = r = 1$. The constant weight function suggests use of the Legendre abscissas and weights.

It is worth pointing out that the actual coordinates used are mapped by $r = \sqrt{\frac{x+1}{2}}$, and this square root is exactly what is needed to avoid having the Legendre points cluster at their left boundary, which is mapped to the origin $r = 0$. A representative example of the computational grid used can be seen in figure 4.3.

The numerical quadrature suggested above is best understood when written in matrix form. If we have coefficients $b_{s,l}$ and want to evaluate the function $f$ in equation 4.11 on the grid points $r_j, \theta_k$, we apply the matrix

$$M_l(s,j) = P_s^{(0,|l|)}(q_j)\left(\frac{q_j+1}{2}\right)^{\frac{|l|}{2}},$$

so that

$$f(r_j,\theta_k) = \sum_{s,l} b_{s,l} M_l(s,j) e^{2\pi i l \theta_k}.$$

Gauss-Radau integration provides the pseudoinverse to this operation:

$$M_l^{\div}(s,j) = w_j P_s^{(0,|l|)}(q_j) \left( \frac{q_j+1}{2} \right)^{\frac{|l|}{2}},$$

where $q_j$ are the Legendre Gauss-Radau abscissas and $w_j$ are the weights.

The fact that a pseudo-inverses occurs instead of an inverse reflects the fact that we have more grid points than basis functions. Schemes based on this won't be exactly collocation methods, and aliasing phenomena will be somewhat of a worry. Nevertheless, it is possible to get fairly far by pretending that this is a collocation method.

$M_l(s,j)$ is a different 2-dimensional matrix for each $l$. One can precompute all of these matrices for small problems, but for fine grids this may not be practical. It may be necessary to calculate these matrices during the computation itself. The expensive part of evaluating the entries for $M_l$ is evaluating the Jacobi polynomials, and the obvious solution to this problem is to make use of the recursion relation (see Szegö[51], p. 72).

$$P_s^{(\alpha,\beta+1)} = \frac{2}{2s+\alpha+\beta+2} \frac{(s+\beta+1)P_s^{(\alpha,\beta)} + (s+1)P_{s+1}^{(\alpha,\beta)}}{(1+x)}.$$

One would apply this by saving at each stage the values of $P_l^{(0,|l|)}$, and then advance from $l$ to $l+1$ (or in the reverse direction, depending on the stability of the recursion).

### 4.3.2 Collocation points in Radon space

It is somewhat more straightforward to propose a discretization in Radon space. Again, there are two indices for the basis, $s$ and $l$, and a finite subset must be chosen. Let us simplify the presentation here by considering first the case $\{s,l \in \mathbb{N} : 2s+|l| \leq N/2, l > -N/2\}$ and then discuss the merits of $\{s,l \in \mathbb{N} : 2s+|l| \leq N/2-2\}$. Recalling (4.7) the Zernike basis transforms to the product of a radial function with the Fourier basis. So if $\hat{f}(p, \phi)$ is assumed to have an expansion

$$\hat{f}(p, \phi) = \sqrt{1-p^2} \sum_{\substack{2s+|l| \leq N/2 \\ l > N/2}} a_{s,l} U_{2s+|l|}(p) e^{2\pi i l \phi},$$

the coefficients $a_{s,l}$ may be obtained by performing the integral

$$a_{t,m} = 2 \int_{-1}^{+1} \int_{0}^{1} \hat{f}(p, \phi) U_{2t+|m|}(p) e^{-2\pi i m \phi} \, d\phi \, dp = 2 \int_{-1}^{+1} h_m(p) U_{2t+|m|}(p) \, dp \tag{4.12}$$

where the radial function $h_l(p)$ is defined by

$$h_l(p) = \int_{0}^{+1} \hat{f}(p, \phi) e^{-2\pi i l \phi} \, d\phi.$$

Equation (4.12) follows by the orthogonality relationship satisfied by the polynomials $U$:

$$\int_{-1}^{+1} \sqrt{1-p^2} U_b(p) U_c(p) \, dp = \frac{1}{2} \delta_{bc}$$

In order to perform collocation in Radon space, an exact quadrature formula for (4.12) is required. The integral in $d\phi$ may be discretized the same way the integral in $d\theta$ was in the previous section. A DFT using the discrete values $\phi_k = \frac{k}{N}, k = 0, \ldots, N-1$ computes this exactly. A sine transform may be used to compute the integral over $dp$ exactly due to the identity

$$\begin{aligned} \sqrt{1-p^2} U_s(p) &= \sqrt{1-p^2} \frac{\partial}{\partial p} T_{s+1}(p) \\ &= \sqrt{1-p^2} \frac{\sin((s+1)\cos^{-1}(p))}{\sin(\cos^{-1}(p))} \\ &= \sin((s+1)\cos^{-1}(p)). \end{aligned}$$

If apply the transformation $p = \cos(\psi)$ to the radial variable $p$, then we can perform an exact quadrature using a sine transform on the $N/2+1$ points $\psi_j = \frac{j\pi}{N/2+2}$, for $j = 1 \ldots N/2+1$:

$$\frac{1}{2}a_{t,m} = \mathcal{S}_\psi h_l(\cos(\psi_j)) = \sum_{j=1}^{N/2+1} \sin(\frac{\pi j(2s+|l|+1)}{N/2+2})h_l(\cos(\psi_j)). \tag{4.13}$$

It would be nice to be able to use fast transforms in both the radial and angular directions for these quadratures. To do this, $N$ will probably be chosen as a power of two so that an FFT may be used in the angular direction. In order to use a fast sine transform to perform the sum in 4.13, $N/2+2$ needs to be a power of two. The simplest way around this conflicting demand is to throw away enough radial modes so that using the $N/2-1$ points in the radial direction $\psi_j = \frac{\pi j}{N/2}$ is sufficient. The sine transform then becomes

$$\sum_{j=1}^{N/2-1} \sin(\frac{\pi j(2s+|l|+1)}{N/2})h_l(\cos(\psi_j)). \tag{4.14}$$

This corresponds to throwing away the modes $2s+|l| \geq N/2-1$. This sampling is advantageous even if fast transforms are not used—for $2s+|l|+1 < N/4$ one may omit every other data point and perform a sine transform of half the size:

$$\sum_{j=1}^{N/4-1} \sin(\frac{\pi j(2s+|l|+1)}{N/4})\hat{f}(\cos(\psi_{2j-1}), \phi_k).$$

This process may be continued for $2s+|l| < N/8$ and so on.

There is an additional benefit to dropping the basis elements with $2s+|l| \geq N/2-1$. The set of values $p_j = \cos(\psi_j)$, $\phi_k = 2\pi k/N$ correspond to sampling line integrals carried out along the lines depicted in figure 4.4. This is a version of the so-called "fan-beam" sampling geometry in tomography. It is the geometry that would occur if $N$ omni-directional radiation sources and $N$ detectors were arranged evenly around in a ring, one source and one detector per point. Sampling occurs in $N$ stages. At stage $j$, source $j$ is switched on, and every other detector besides $j$ measures the amount of radiation reaching it, computing the line integral between it and the source $j$. It is worth mentioning that the line integrals in this geometry intersect at the boundary points used in the real space discretization. This may have favorable consequences, depending on the way that boundary conditions are implemented.

If the sine transform is implemented by a simple sum, symmetry in the problem may be used to halve the number of operations in the radial transform. The key is the fact that

$$\hat{f}(-p,\phi) = \hat{f}(p,\phi+\pi).$$

This in turn implies that

$$h_l(-p)e^{2\pi il\phi} = h_l(p)e^{2\pi il(\phi+\pi)} = (-1)^l h_l(p)e^{2\pi il\phi}$$

So $h_l(-p) = (-1)^l h_l(p)$. Now

$$\sin((2s+|l|+1)\cos^{-1}(-p)) = \sin((2s+|l|+1)(\pi-\cos^{-1}(p))) = (-1)^{|l|}\sin((2s+|l|+1)\cos^{-1}(p)).$$

The signs here cancel in (4.13) and (4.14). The sum for positive values of $p = \cos^{-1}(\psi)$ is equal to the sum for negative values, and so the sum needs to be carried out over only half the domain.

Let us summarize the steps involved in using these transforms to go from real space to Radon space for the case $2s+|l| \le N/2, l > -N/2$:

1. Real DFT (or FFT) in the radial direction

$$\mathcal{F}_\theta f(r_j,\theta_k) \to g_l(r_j).$$

2. Legendre quadrature $M_l^{\div}(s,r_j)g_l(r_j) \to b_{s,l}$.

3. Scale the Zernike coefficients $b_{s,l}$ to get the coefficients in Radon space $a_{s,l}$

$$a_{s,l} = \frac{2}{|l|+2s+1}b_{s,l} = Wb_{s,l}.$$

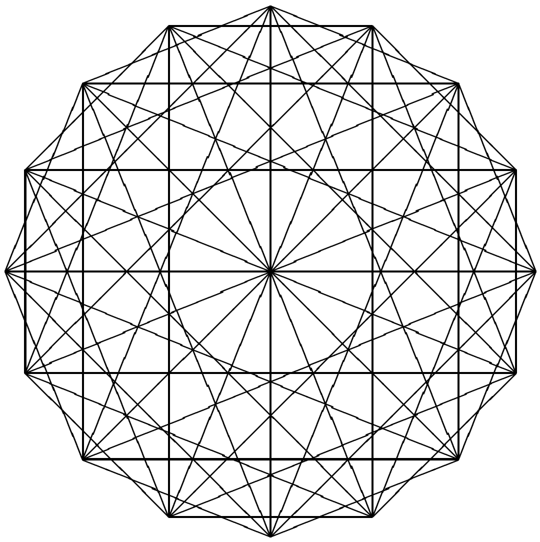

Figure 4.4: A fan-beam geometry for Tomography

4. Apply a sine transform for each $l$ to compute

$$h_l(p_j) = \sqrt{1-p^2}\sum_s a_{s,l} U_{2s+|l|}(p_j) = \sum_s a_{s,l}\sin(\frac{\pi j(2s+|l|+1)}{N/2+2}) = \mathcal{S} a_{s,l}.$$

5. Use the inverse DFT/FFT to compute

$$\hat{f}(p_j,\phi) = \mathcal{F}_l^{-1} h_l(p_j),$$

or more precisely

$$f(p_j,\phi_k) = \sum_l h_l(p_j) e^{2\pi i l \phi_k}.$$

In operator notation this is

$$\hat{f}(p,\phi) = \mathcal{F}_l^{-1}\,\mathcal{S}\,W\,M_l^{\div}\,\mathcal{F}_\theta\, f(r,\theta).$$

The (pseudo)inverse of these operations may be applied in the reverse direction to compute a transform from Radon to real space.

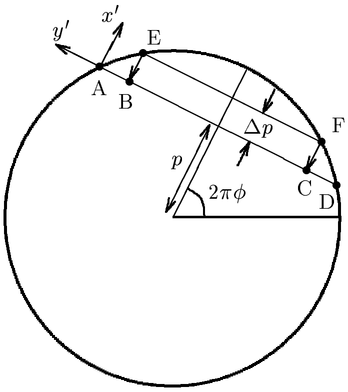


Figure 4.5:

## 4.4 Implementing collocation schemes

Now that a basis and a set of collocation points has been chosen, the only further requirement for creating a spectral scheme is to find a formula for computing derivatives. It is possible to do this in real space, using recurrence relations for the Jacobi polynomials. But this may not be a stable recurrence. The other approach is to exploit the Radon space basis set of the previous section. In this situation it is possible to derive an explicit formula for the derivative. The obvious approach is to make use of equation 4.3. But that equation is only valid for infinite domains, and our basis is supported only on the unit disk $D^2$. It will be necessary to modify (4.3) by adding boundary terms. This is easy to see, as naively applying (4.3) to our basis yields a singularity at $p = \pm 1$:

$$\frac{\partial}{\partial p}\hat{f}(p,\xi) = \frac{\partial}{\partial p}\left(\sqrt{1-p^2}\sum a_{s,l}U_{2s+|l|}(p)e^{2\pi i l\phi}\right) =$$

$$\frac{-p}{1-p^2}\hat{f}(p,\xi) + \text{ other terms.}$$

Consider a single basis function

$$w(p,\phi) = U_{2s+|l|}(p)e^{2\pi i l\phi} \tag{4.15}$$

in Radon space. We would like to take the derivatives $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ of

$$\breve{w}(x,y) = \frac{2s+|l|+1}{2} Z_{2s+|l|}^{|l|}(r) e^{2\pi i l \theta} \tag{4.16}$$

and transform them back into Radon space. It will be simpler to work for now in the transformed coordinate system $x'$ and $y'$ where $x' = r\cos(2\pi(\theta-\phi))$, and $y' = r\sin(2\pi(\theta-\phi))$ and write $\breve{w}(x',y')$. (see figure 4.5). Consider the integral of $\breve{w}(x',y')$ over the line segment $EF$ minus the integral over $AD$ in figure 4.5:

$$w(p+\Delta p,\phi) - w(p,\phi) = \int_{-\sqrt{1-(p+(\Delta p)^2}}^{\sqrt{1-(p+\Delta p)^2}} \breve{w}(p+\Delta p, y')\, dy' - \int_{-\sqrt{1-p^2}}^{\sqrt{1-p^2}} \breve{w}(p,y')\, dy'.$$

We would like to approximate this using $\frac{\partial}{\partial x'}\breve{w}(x',y')_{|x'=p}$ using the same approach as in §4.1.2. It can be done provided we subtract off the line segments $AB$ and $CD$ from the integral over $AD$:

$$\int_{-\sqrt{1-(p+\Delta p)^2}}^{\sqrt{1-(p+\Delta p)^2}} \frac{\partial}{\partial x'}\breve{w}(p,y')\Delta p\, dy'$$

$$= \int_{-\sqrt{1-(p+\Delta p)^2}}^{\sqrt{1-(p+\Delta p)^2}} \left(\breve{w}(p+\Delta p, y') - \breve{w}(p,y')\right)\, dy' + O\left((\Delta p)^2\right).$$

But

$$\begin{aligned}
\int_{AB} \breve{w}(p,y')\, dy' &= \int_{\sqrt{1-(p+\Delta p)^2}}^{\sqrt{1-p^2}} \breve{w}(p,y')\, dy' \\
&= -\Delta p \frac{\partial}{\partial p}(\sqrt{1-p^2})\, \breve{w}(p,\sqrt{1-p^2}) + O\left((\Delta p)^2\right).
\end{aligned}$$

and

$$\begin{aligned}
\int_{CD} \breve{w}(x',p)\, dy' &= \int_{-\sqrt{1-(p)^2}}^{-\sqrt{1-(p+\Delta p)^2}} \breve{w}(p,x')\, dy' \\
&= -\Delta p \frac{\partial}{\partial p}(\sqrt{1-p^2})\, \breve{w}(p,-\sqrt{1-p^2}) + O\left(\Delta p^2\right).
\end{aligned}$$

Substituting in,

$$\begin{aligned}
\frac{\partial}{\partial p} w(p,\phi) &= \frac{w(p+\Delta p,\phi) - w(p,\phi)}{\Delta p} + O(\Delta p) \\
&= \int_{-\sqrt{1-(p+\Delta p)^2}}^{\sqrt{1-(p+\Delta p)^2}} \frac{\partial}{\partial x'} \breve{w}(x',y')_{|_{x'=p}} \\
&- \frac{p}{\sqrt{1-p^2}} \left( \breve{w}(p,\sqrt{1-p^2}) + \breve{w}(p,-\sqrt{1-p^2}) \right) + O(\Delta p)
\end{aligned}$$

Taking limits, the result is

$$\begin{aligned}
\frac{\partial}{\partial p} w(p,\phi) &= \int_{-\sqrt{1-p^2}}^{\sqrt{1-p^2}} \frac{\partial}{\partial x'} \breve{w}(x',y')_{|_{x'=p}}\, dy' \\
&- \frac{p}{\sqrt{1-p^2}} \left( \breve{w}(p,\sqrt{1-p^2}) + \breve{w}(p,-\sqrt{1-p^2}) \right). \qquad (4.17)
\end{aligned}$$

An expression entirely in Radon space is desirable, and this is easy to achieve. Using the definitions of $Z$ (4.9) and $\breve{w}$ (4.16) and the fact that $P_s^{(0,l)}(1) = 1$

$$\breve{w}(p, \pm\sqrt{1-p^2}) = \frac{2s+|l|+1}{2} e^{il(2\pi\phi \pm \psi)}. \qquad (4.18)$$

where $\cos(\psi) = p$. Feeding this and (4.15) into (4.17) and using the transformation $p = \cos(\psi)$ results in

$$\begin{aligned}
&\int_{-\sqrt{1-p^2}}^{\sqrt{1-p^2}} \frac{\partial}{\partial x'} \breve{w}(x',y')_{|_{x'=p}}\, dy' = \\
&(2s+|l|+1) \frac{\cos(\psi)\cos(l\psi) - \cos((2s+|l|+1)\psi)}{\sin(\psi)} e^{2\pi i l \phi}. \qquad (4.19)
\end{aligned}$$

This looks like it might be singular at $p = \pm 1 \mapsto \psi = 0, \pi$, but trigonometric identities can be applied to show it isn't.

Finding an equation for $\frac{\partial}{\partial y'}$ is easier:

$$\int_{-\sqrt{1-p^2}}^{\sqrt{1-p^2}} \frac{\partial}{\partial y'} \breve{w}(p,y')\, dy' = \breve{w}(p,\sqrt{1-p^2}) - \breve{w}(p,-\sqrt{1-p^2}). \qquad (4.20)$$

writing this in Radon space yields

$$\int_{-\sqrt{1-p^2}}^{\sqrt{1-p^2}} \frac{\partial}{\partial y'} \breve{w}(p,y')\, dy' = (2s+|l|+1) i \sin(l\psi) e^{2\pi i l \phi}. \tag{4.21}$$

Now that $\frac{\partial}{\partial x'}$ and $\frac{\partial}{\partial y'}$ are available, a coordinate transformation on (4.19) and (4.21) produces what we are looking for:

$$\mathcal{R}\frac{\partial}{\partial x} Z_{2s+|l|}^{|l|}(r) e^{2\pi i l\theta} = \\ 2\left(\frac{\cos(\psi)\cos(l\psi) - \cos((2s+|l|+1)\psi)}{\sin(\psi)} \cos(2\pi\phi) - i\sin(l\psi)\sin(2\pi\phi)\right) e^{2\pi i l\phi}. \tag{4.22}$$

$$\mathcal{R}\frac{\partial}{\partial y} Z_{2s+|l|}^{|l|}(r) e^{2\pi i l\theta} = \\ 2\left(\frac{\cos(\psi)\cos(l\psi) - \cos((2s+|l|+1)\psi)}{\sin(\psi)} \sin(2\pi\phi) + i\sin(l\psi)\cos(2\pi\phi)\right) e^{2\pi i l\phi}. \tag{4.23}$$

In order to apply equations (4.22) and (4.23) to construct a derivative matrix, it is simpler to express the results in using terms of the form $h_{m,s,l}(p)\, e^{2\pi i l\phi}$, where $h_{m,s,l}(p)$ denotes the radial function at angular wave-number $m$ produced by taking the derivative of Radon space basis element $(s,l)$. For example consider the $x$ derivative case, if we expand $\sin(2\pi\phi)$ and $\cos(2\pi\phi)$ in terms of exponentials we find

$$\mathcal{R}\frac{\partial}{\partial x} Z_{2s+|l|}^{|l|}(r) e^{2\pi i l\theta} = h_{l+1,s,l}(p)\, e^{2\pi i(l+1)\phi} + h_{l-1,s,l}(p)\, e^{2\pi i(l-1)\phi} \tag{4.24}$$

where

$$h_{l+1,s,l}(p) = \frac{\cos(\psi)\cos(l\psi) - \cos((2s+|l|+1)\psi)}{\sin(\psi)} - \sin(l\psi).$$

and

$$h_{l-1,s,l}(p) = \frac{\cos(\psi)\cos(l\psi) - \cos((2s+|l|+1)\psi)}{\sin(\psi)} + \sin(l\psi).$$

Of course, if $m \neq l+1$ and $m \neq l-1$ then $h_{m,s,l} = 0$.

Now given a function

$$\hat{f}(p,\phi) = a_{s,l} \sum U_{2s+|l|}(p) e^{2\pi i l \phi}$$

A derivative may be found by

$$\mathcal{R}\frac{\partial}{\partial x} f(x,y) = \sum_{m,s,l} \frac{2s+|l|+1}{2} a_{s,l}\, h_{m,s,l}(p) e^{2\pi i m\phi} \tag{4.25}$$

In operator notation this will be written below in terms of the operator $D_x$, which is defined here as

$$D_x = \mathbf{F}_\phi \mathcal{R}\frac{\partial}{\partial x} f(x,y).$$

Only two values of $m$ are summed for each $l$ in (4.25), so this is reasonably efficient. The radial sum may be carried out first, and a real inverse FFT used to compute the angular sums, so this is not substantially more costly than the transform to and from Radon space. Given the smooth analytic formula for the matrix, it seems quite likely that matrix compression techniques could provide a fast transform method as well.

### 4.4.1 Numerical experiments with the wave equation

The computational grid proposed in §4.3 is not quite sufficient for implementing a collocation method. The set spanned by this basis does not contain 'delta' functions, zero on every grid point but one. Nevertheless, one can pretend that this is the case, and create schemes similar to collocation methods. For example, consider the wave equation

$$u_t - u_x = 0, \text{ where } u : \mathbb{R}^2 \to \mathbb{R}$$

with boundary condition

$$u(r, 2\pi\theta) = 0 \text{ for } r = 1, \quad -\frac{1}{2} \leq \theta \leq \frac{1}{2}. \tag{4.26}$$

Let $v$ be the Radon basis coefficients

$$v = W M_l^{\div} \mathcal{F}_\theta \, u.$$

Using the operator notation defined in previous sections. This is almost in Radon space, as $\widehat{u} = \mathcal{F}_p^{-1} \mathcal{S} v$. It turns out to be convenient to write a scheme in terms of $v$. In semi-discrete form the scheme may be written as

$$\dot{v} = B_v \, 2\mathcal{S} \, D_x B_v v \tag{4.27}$$

where $B_v$ is a boundary condition projection

$$B_v = W M_l^{\div} \, \mathcal{F}_\theta \, B \, \mathcal{F}_l^{-1} M_l \, W^{-1}$$

and $B$ is the boundary operator operating in real space which sets the collocation points on the inflow boundary to zero—as in equation (4.26). This scheme appears to work well when implemented. But a plot of the spectrum of the spatial derivative operator reveals that for $N = 8$ and $N = 16$ it is probably stable. For $N = 32$ and higher, the computed spectrum appears to have eigenvalues with positive real parts (see figures 4.6–4.10). It is not clear what contribution round-off error is making to these pictures. The matrices involved are large–for the 128 case the derivative operator from equation (4.27) is a 2144x2144 non-symmetric matrix.

The derivative matrix involved in these spectrum calculations was assembled by performing the operations in equation (4.27) in sequence upon each mode in turn, constructing the matrix one column at a time. This matrix itself may be used to construct a numerical scheme, but the result of doing this appears to be less stable than a scheme implemented by doing the transforms in (4.27).

A stable method may be obtained for the $N = 32$ case by projecting the solution using the boundary operator $B_v$ at every partial time step in the ODE solver. In figure 4.11 the $L_\infty$ norm of the solution for a random initial condition is plotted against time. The ordinary differential equation solver used is the third order TVD Runge-Kutta method of Shu and Osher[45], and the projection

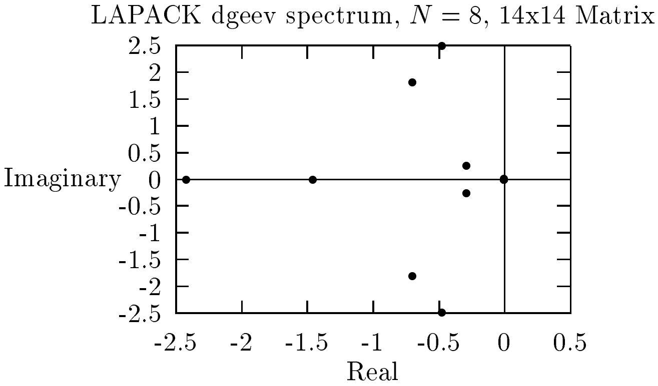


Figure 4.6: Spectrum for the spatial derivative operator for $u_t = u_x$, $N = 8$, collocation boundaries.

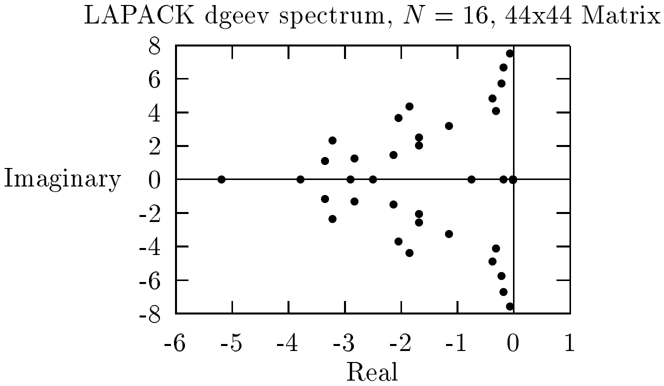


Figure 4.7: Spectrum for the spatial derivative operator for $u_t = u_x$, $N = 16$, collocation boundaries.

$v \leftarrow B_v v$ is applied at every partial time step. The boundary conditions force the problem, resulting in a nonzero norm for all time, but it doesn't grow. On the other hand, the $N = 64$ and $N = 128$ cases are unstable. To what extent this is purely the result of roundoff error is unclear.

The instability present in the higher resolution schemes, is associated with higher modes and grows slowly. Spectral accuracy can be checked quite easily for a smooth initial condition which does not excite these modes. Table 4.2 presents the $L_\infty$ error of this scheme, when applied to an initial condition

$$C \exp\left(-\frac{1}{a-0.15} - \frac{1}{1-a-0.15} - \frac{1}{b-0.1} - \frac{1}{1-b-0.1}\right),$$

if $0.15 < a = (x+0.8)/2 < 0.85, \quad 0.1 < b = (y+1)/2 < 0.9$, otherwise the initial condition is 0. $C$ is chosen so that the initial condition is 1 at its maximum. Again the ODE solver used is the

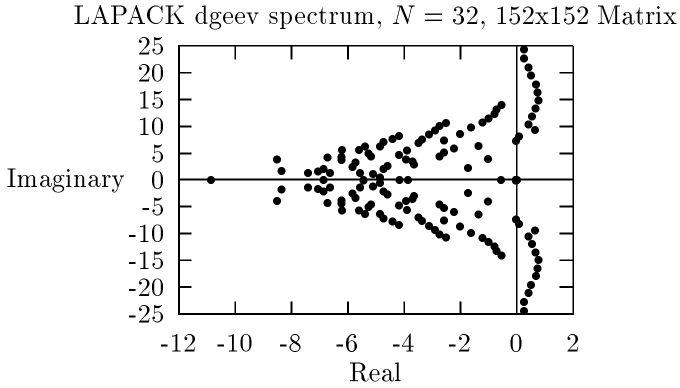


Figure 4.8: Spectrum for the spatial derivative operator for $u_t = u_x$, $N = 32$, collocation boundaries.

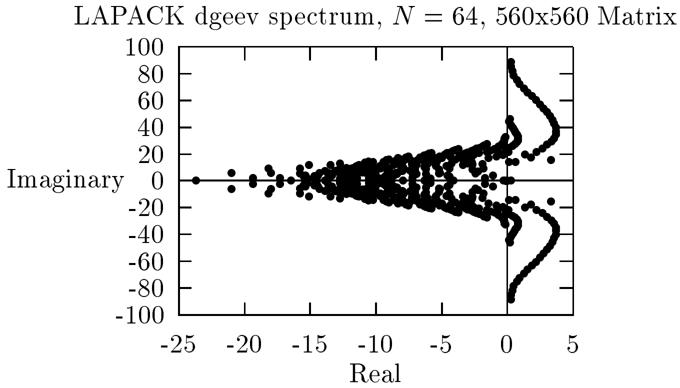


Figure 4.9: Spectrum for the spatial derivative operator for $u_t = u_x$, $N = 64$, collocation boundaries.

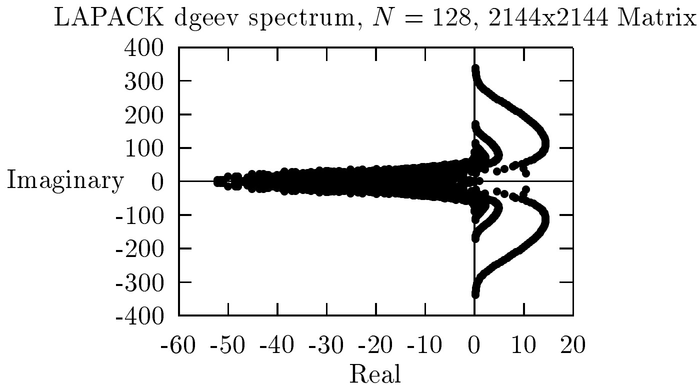


Figure 4.10: Spectrum for the spatial derivative operator for $u_t = u_x$, $N = 128$, collocation boundaries.

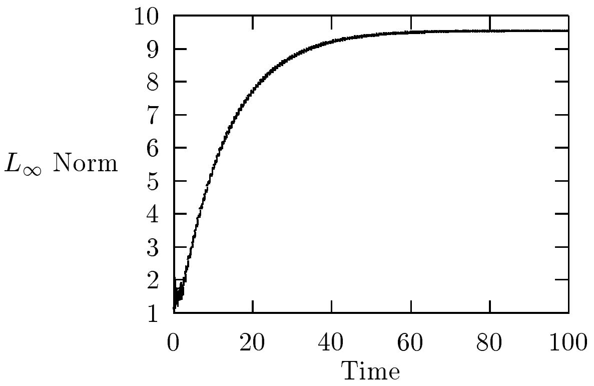


Figure 4.11: $L_\infty$ norm for $N = 32$, $\Delta t = 1/36$, random initial condition.

| N | Modes | $L_\infty$ error | $\Delta t$ |
|---|---|---|---|
| 8 | 14 | 0.473119 | 1/2 |
| 16 | 44 | 0.363711 | 1/6 |
| 32 | 152 | 0.0134922 | 1/36 |
| 64 | 560 | 0.000380967 | 1/200 |
| 128 | 2144 | 3.03962E-06 | 1/1000 |

Table 4.2: $L_\infty$ Errors at time 1.20 for $u_t = u_x$ with collocation boundary conditions. $N = 64$ and $N = 128$ are slightly unstable.

third-order TVD Runge-Kutta method[45]. Time differencing errors are negligible here.

### Tau method boundary conditions

Equation (4.18) provides a relation between Zernike modes and boundary values. It can be exploited in an efficient way to provide a boundary condition similar to tau method boundary conditions, and the resulting schemes seem to be stable.

Given Zernike coefficients $b_{s,l}$

$$u(r,\theta) = \sum_{s,l} b_{s,l} Z_{2s+|l|}^{|l|}(r) e^{2\pi i l \theta}$$

Then, using (4.18) $u$ takes on boundary values

$$u(1,\theta) = \sum_{s,l} b_{s,l} e^{2\pi i l \theta}.$$

If the expansion of $u$ is limited to a finite set of modes, say $\{s, l : 2s + |l| \leq N/2, l > -N/2\}$, then a set of the modes with high wave numbers may be manipulated to enforce the boundary conditions on $u$. For each angular mode $l$ the mode with the largest value of $s$, $s = s_l$ where

$$s_l = \max\{s : 2s + |l| \leq N/2\},$$

is changed. For the case of the wave equation $u_t = u_x$ we want an expansion that is zero at the right boundary points $\{(r, \theta) : r = 1, -\pi/2 < \theta < \pi/2\}$. The high modes $b_{s_t,t}$ may be set to

$$b_{s_t,t} = -\frac{1}{N} \sum_{k=0}^{N-1} e^{-2\pi i t k/N} B(k) \sum_{\substack{-N/2 < l \leq N/2 \\ s < s_l}} b_{s,l} e^{2\pi i l k/N} \tag{4.28}$$

where

$$B(k) = \begin{cases} 0, & \text{if } \pi/2 \leq 2\pi k/N \leq 3\pi/2 \\ 1, & \text{otherwise.} \end{cases}$$

The Fourier sum here may be evaluated by FFT.

This boundary condition leads to a stable scheme. Some spectra for this may be seen in figure 4.12. Observed accuracies for this method are very similar to the scheme with collocation boundaries, as can be seen in table 4.3. The eigenvalues in the spectrum have somewhat larger norm than the collocation method. If filtering were used to stabilize the collocation scheme, it might have an advantage in terms of time-stepping.

It should be noted that stable schemes are possible even if the right boundary points are not entirely taken to zero. For example,

$$B(k) = \begin{cases} 0, & \text{if } \pi/2 \leq 2\pi k/N \leq 3\pi/2 \\ C, & \text{otherwise.} \end{cases}$$

works for a constant C slightly less than one. This has the effect of decreasing the negative real part of the eigenvalues in the two 'tails', as can be seen in figure 4.13. Non-constant $C$ is also possible. $C(k) = \sqrt{\cos(2\pi k/N)}$ seems to improve the spectrum slightly upon the $C = 1$ case.

| N | Modes | $L_\infty$ error | $\Delta t$ used | Stable $\Delta t$ |
|---|---|---|---|---|
| 8 | 6 | 0.78933 | 1/5 | 1 |
| 16 | 28 | 0.336751 | 1/7 | 1/4 |
| 32 | 120 | 0.0153574 | 1/25 | 1/16 |
| 64 | 496 | 0.000395241 | 1/100 | 1/63 |
| 128 | 2016 | 3.62789E-06 | 1/400 | 1/257 |

Table 4.3: $L_\infty$ Errors at time 1.20 for $u_t = u_x$ with tau boundary conditions. Stability checked by running a random initial condition to $t = 100$.

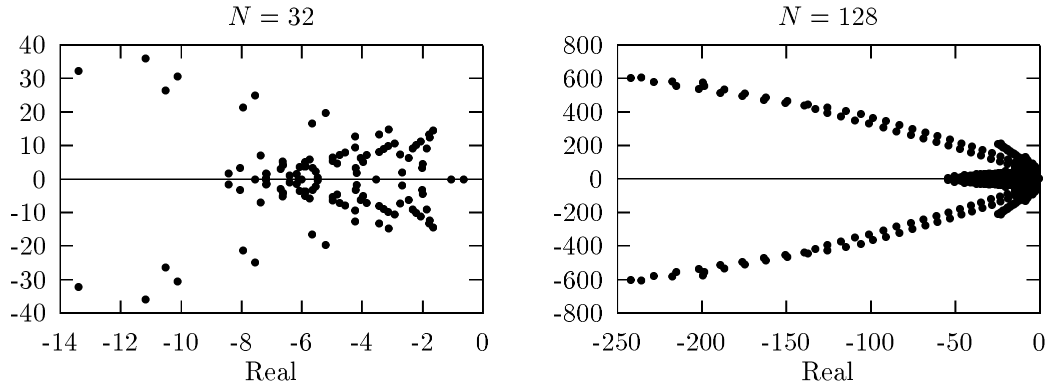


Figure 4.12: Spectrum for the spatial derivative operator for $u_t = u_x$, with $C = 1$ tau boundary conditions.

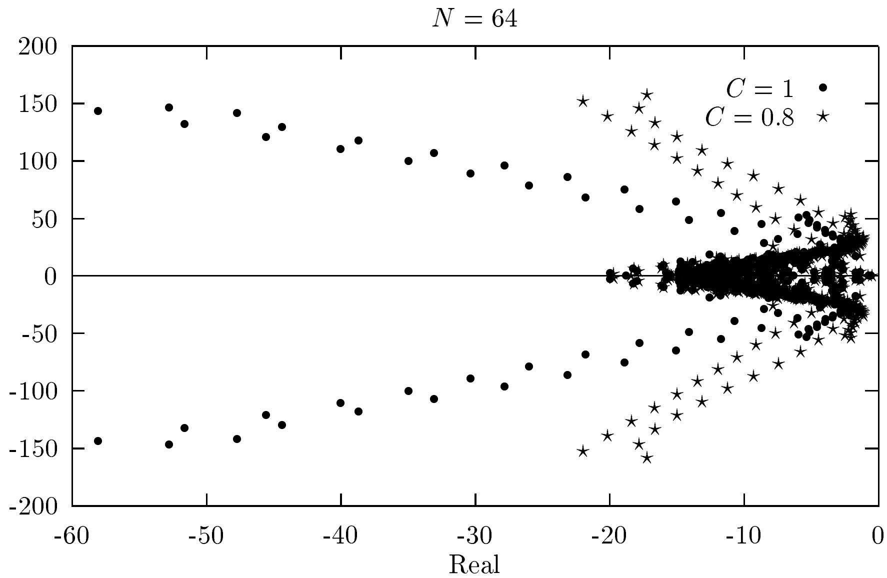


Figure 4.13: Spectrum for the spatial derivative operator for $u_t = u_x$, tau boundaries with $N = 64$ comparing $C = 0.8$ and $C = 1$.

| N | $L_\infty$ error | $\Delta t$ used | Stable $\Delta t$ |
|---|---|---|---|
| 8 | 0.21932 | 1/10 | 1/2 |
| 16 | 0.100476 | 1/20 | 1/4 |
| 32 | 0.0040099 | 1/40 | 1/9 |
| 64 | 0.00023783 | 1/80 | 1/18 |
| 128 | 3.95718E-06 | 1/320 | 1/36 |

Table 4.4: $L_\infty$ Errors at time $20\pi$ (10 rotations) for $u_t = -yu_x + xu_y$. Stability checked by running a random initial condition to $t = 100$.

**Rotating flow**

The rotational symmetry of the Zernike basis hints that it may be useful for rotating flow. This also avoids the problems of boundary conditions present in the previous section. Consider the model equation

$$u_t = -yu_x + xu_y.$$

This equation describes a flow that convects $u$ counter-clockwise. Again, it is simpler to implement a scheme in terms of $v = W M_l^{\div} \mathcal{F}_\theta\, u$. In semi-discrete form this becomes

$$\dot{v} = 2\mathcal{S}\left(-D_x\, W\, M_l^{\div}\, \mathcal{F}_\theta\, yI + D_y\, W\, M_l^{\div}\, \mathcal{F}_\theta\, xI\right) B\, \mathcal{F}_l^{-1}\, M_l\, W^{-1}\, v.$$

$B$ again is a boundary operator. In this case, it sets all boundary points to zero. The spectrum for this case may be seen in figure 4.14. For this particular problem, the eigenvalues appear to grow in magnitude proportionally to $N$, and not $N^2$ as in the previous case, and this is reflected in the permissible time step (see table 4.4). The same initial condition was used as in the previous section.

The boundary operator may be omitted altogether without affecting the stability of the scheme, but then boundary values may drift somewhat. The resulting spectra are quite similar to those depicted here.

The performance of the numerical scheme in this case is quite good. Rotating flows seem to be a natural application for this basis. This and the fact that gravitational potentials may be calculated by using the back-projection operator (4.6) suggest the application of this method to self-gravitating flows of astrophysical interest.

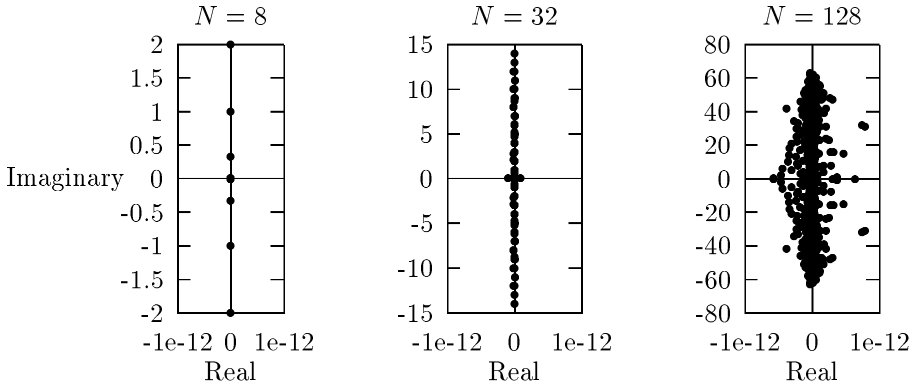


Figure 4.14: Spectra for $u_t = -yu_x + xu_y$.

## 4.5 Finite difference schemes in Radon Space

In the previous sections, the Radon space version of the Zernike basis was used in order to derive simple equations for the derivative operators. There is no real necessity for doing this, it just happens that the Radon space basis had a simpler analytic form than the real space basis. For higher dimensional versions of the Zernike basis derived by Louis, this advantage will be lost, and one will have to look for other excuses to work with the Radon space basis.

Equation (4.3) (and the versions with boundary conditions, (4.17) and (4.20)) provide an excuse. They imply that derivatives in real space may be carried out in the radial direction in Radon space. The Radon space derivative need not be calculated by a spectral derivative operator at all. A conservative finite difference derivative may do. A particularly interesting derivative to try would be an advanced high-order finite difference operator. Here I will explore this idea by employing the derivative used in the ENO schemes of Osher and Shu[45, 46].

Here I will solve the rotating convection problem

$$u_t = -(yu)_x + (xu)_y = -f(\hat{u})_x + g(\hat{u})_y, \quad u(r = 1, \theta) = 0.$$

where $f(\widehat{u}) = -y\mathcal{R}^{-1}\widehat{u}$, and $g(\widehat{u}) = x\mathcal{R}^{-1}\widehat{u}$. In Radon space this may be discretized

$$\widehat{u}(p,\phi)_t = \frac{\partial}{\partial p}(-\widehat{f}(\widehat{u})\cos(\phi) + \widehat{g}(\widehat{u})\sin(\phi)) = \frac{\partial}{\partial p}F(\widehat{u}(p,\phi)) \tag{4.29}$$

as we have zero boundary conditions and the boundary terms may be ignored. This equation appears to be a different one-dimensional conservation law for each value of the angular variable $\phi$. Coupling between these equations is accomplished through the flux $F$. Of course, this flux is going to have a very complex non-convex form, and it is not clear that standard techniques designed for convex fluxes will work.

The ENO method is a conservative discretization which takes the form

$$\widehat{u}(p,\phi)_t = \frac{\tilde{F}(p+\Delta p/2,\phi) - \tilde{F}(p-\Delta p/2,\phi)}{\Delta p}.$$

$\tilde{F}$ derived from $F$ in a tricky way that ensures right hand side of this equation is a high order approximation to the spatial derivative in (4.29) in smooth regions. In a practical ENO scheme, $\tilde{F}$ may be constructed in several different ways in order to provide upwinding. For a scheme based on the Lax-Friedrich flux, $F$ is split into two parts, $\tilde{F}^{+}$, and $\tilde{F}^{-}$, where

$$\frac{\tilde{F}^{\pm}(p+\Delta p/2,\phi) - \tilde{F}^{\pm}(p-\Delta p/2,\phi)}{\Delta p} = \frac{\partial\left(F(p,\phi) \pm \alpha\widehat{u}(p,\phi)\right)}{\partial p} + O(\Delta p^r).$$

where $r$ is the (high order) truncation error of the scheme. $\alpha$ is chosen to be approximation of the wave speed in the problem. For a non-convex flux this seems to be the area of most difficulty. The rotating flow problem is particularly bad—the effective wave speed depends on where mass is located with respect to the origin. Happily, practical ENO schemes do not appear particularly sensitive to the exact value of $\alpha$ supplied. It is not unrealistic to hope that an ad-hoc calculation of the wave speed would work. The obvious $\alpha$ to try is the following:

$$\alpha(p,\phi) = \frac{|\cos(\phi)|\,\mathcal{R}\left((|u|+\epsilon_1)|f(\widehat{u})|\right) + |\sin(\phi)|\,\mathcal{R}\left((|u|+\epsilon_1)|g(\widehat{u})|\right)}{\mathcal{R}(|u|+\epsilon_1)} + \epsilon_2$$

| N | $L_1$ error | $\Delta t$ |
|---|---|---|
| 32 | 0.0226785 | 1/100 |
| 64 | 0.0115647 | 1/200 |

Table 4.5: $L_1$ Errors at time 0.3 for $u_t = -yu_x + xu_y$ solved by ENO.

where $\epsilon_1$ and $\epsilon_2$ are small positive constants, say $\epsilon_1 = \epsilon_2 = 0.01$.

In an actual implementation of the above, if the scheme runs using the Chebyshev quadrature points, it is simpler to calculate the derivatives above using the mapping $p = \cos(\psi)$:

$$\widehat{u}(p,\phi)_t = \frac{\tilde{F}(\psi + \Delta\psi/2, \phi) - \tilde{F}(\psi - \Delta\psi/2, \phi)}{\Delta\psi} \frac{\partial\psi}{\partial p}.$$

The above scheme when implemented appears stable. Table 4.5 presents $L_1$ errors at time $t = 0.3$ for the third order ENO scheme, implemented using a third order TVD Runge-Kutta ODE solver to discretize in time. The initial condition is the same as before. The method of upwinding chosen seems to have some problems at the origin, and a spike in error occurs there. This may be reduced by playing with the form of $\alpha$, but possibly at the expense of stability in other areas of the flow.

It could be that finite difference schemes based on cell averages would perform somewhat better in this situation, as upwinding appears in a rather different way. This schemes would be implemented in essentially the same way as the derivative based scheme above. The fluxes would be computed by solving approximate (or not so approximate) Riemann solvers in real space at the computational grid points, and then transforming back to Radon space.

## 4.6 Conclusion

The singular value decomposition of the Radon transform derived by Louis[30] provides a natural basis for Spectral methods on the disk in $n$ dimensions. For the two dimensional case studied here, the Radon space interpretation proved quite useful to derive spectral derivative operators. Spectral schemes derived in this way appear to work quite well. Preliminary results suggest that it is also possible to create hybrid spectral/finite difference schemes operating in Radon space. Whether or not this is a useful idea remains to be seen.

A natural extension of this work would be to derive basis sets using mapped coordinate systems. For example, the circular computational domains applied here may be mapped by affine transformation trivially to elliptical domains. One might also hope to map the radial direction in order to remove the $1/N^2$ time-step restriction inherent in the schemes considered here, following the suggestion of Kosloff and Tal-Ezer[25].

One should also note that the Radon transform inversion techniques used here to create numerical schemes may have application to Tomography, especially when coupled with a method to locate discontinuities (in Radon space by taking advantage of the characterization of singularities by Ramm and Zaslavsky[38], or by using the Fourier space basis) and a method to remove the associated Gibbs phenomena in real space. Fast transforms based on matrix compression might be possible for large data sets. A combination of these techniques might prove quite useful for numerical methods for problems where discontinuities occur.

# Bibliography


[1] S. Abarbanel, D. Gottlieb, *Information Content in Spectral Calculations*, Progress and Supercomputing in Computational Fluid Dynamics, ed. E. Murman, et al., Proc. of U.S.-Israel Workshop, pp. 345–355 (1985)

[2] Paul F. Batcho, George Em Karniadakis *Generalized Stokes Eigenfunctions: A New Trial Basis for the Solution of Incompressible Navier-Stokes Equations* J. Comput. Physics **115** pp. 121–146 (1994)

[3] I. Bell, *3-D adaptive mesh refinement for conservation laws*, SIAM J. Sci. Comp. **15**, pp. 127 (1994)

[4] R. Bonazza, M. Brouillette, D. Goldstein, J.F. Hass, G. Winckelmans, B. Sturtevant, *Rayleigh-Taylor instability of oblique interfaces.* Bull. Am. Phys. Soc. **30**, pp. 1742 (1985)

[5] M. Brouillette, B. Sturtevant, *Experiments on the Richtmyer-Meshkov instability: single-scale perturbations on a continuous interface*, J. Fluid Mech. **263**, pp. 271–292 (1994)

[6] W. Cai, D. Gottlieb, C. W. Shu, *Essentially Non-Oscillatory Spectral Fourier Methods for Shock Wave Calculation*, Math. Comp. **52**, No. 186, pp. 389–410 (1989)

[7] C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, *Spectral methods in fluid dynamics*, Springer-Verlag (1987)

[8] William W. Carlson, Judith D. Schlesinger, *AC: A C Language and Compiler for the CM-5 Node Architecture*, Technical Report SRC-TR-93-094, Supercomputing Research Center Institute for Defense Analyses, Bowie MD. (1993)

[9] P. Collela, A. Majda, V. Roytburd, *Theoretical and numerical structure for reacting shock waves*, SIAM J. Sci. Statist. Comput., **7** pp 1059–1080 (1986)

[10] M. Crandall, A. Majda, *Monotone difference approximations for scalar conservation laws,* Math. Comp. **34** pp. 1–21 (1980)

[11] Stanley R. Deans, *The Radon transform and some of its applications,* New York, Wiley (1983)

[12] R. J. Diperna, J. L. Lions *Global Solutions of Boltzmann's Equation and the Entropy Inequality*, Archive for Rational Mechanics and Analysis, **114** p. 47 (1991)

[13] W. S. Don, *Numerical study of pseudospectral methods in shock wave applications*, J. Comp. Physics, **110,** pp. 103–111 (1994)

[14] M. Dubiner, *Spectral methods on triangles and other domains.* J. Sci. Comp., **6** p. 345 (1991)

[15] B. Engquist, B. Sjogreen, *Robust Difference Approximations of Stiff Inviscid Detonation Waves,* UCLA CAM Report 91-03 (1991)

[16] I. Gertner, *A new efficient algorithm to compute the two-dimensional discrete Fourier transform,* IEEE Trans., **ASSP-36**, pp. 1036–1050 (1988)

[17] J. Goodman, R. LeVeque, *On the accuracy of stable schemes for 2D scalar conservation laws,* Math. Comp. **45** pp. 15–21 (1985)

[18] D. Gottlieb, S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, CBMS conference Series in Applied Mathematics **26**, SIAM (1977)

[19] J. F. Haas, B. Sturtevant, *Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities*, J. Fluid Mech. **181**, pp. 41–76 (1987)

[20] A. Harten, *On a class of high-resolution total-variation-stable finite-difference schemes,* SIAM J. Numer. Anal. **21** pp. 1–23 (1984)

[21] L. F. Henderson, *The refraction of a plane shock wave at a gas interface*, J. Fluid Mech. **26**, pp. 607–637 (1966)

[22] J. W. Jacobs, *Shock-induced mixing of a light-gas cylinder*, J. Fluid Mech. **234**, pp. 629–649 (1992)

[23] Fritz John, *Plane Waves and Spherical Means Applied to Partial Differential Equations,* Interscience Publishers, 1955, Springer Verlag (1981)

[24] S. Karni, *Multi-component Flow Calculations by a Consistent Primitive Algorithm,* J. Comp. Physics, **112**, pp. 31–43 (1994)

[25] D. Kosloff, H. Tal-Ezer, *Modified Chebyshev Pseudospectral Methods With $O(N^{-1})$ Time Step Restriction*, J. Comp. Physics, **104**, pp. 457–469 (1993)

[26] A. Leonard, A. Wray, A New Numerical Method for the Simulation of Three-Dimensional Flow in a Pipe, in *Eighth International Conference on Numerical Methods in Fluid Dynamics: Proceedings of the Conference, Rheinisch-Westfalische Technische Hochschule, Aachen, Germany, June 28-July 2, 1982* Lecture Notes in Physics, **170**. Ed. E. Krause. (Springer Verlag) pp. 335–340 (1982)

[27] R. J. Leveque, H.C. Yee, *A Study of Numerical Methods for Hyperbolic Conservation Laws with Stiff Source Terms, J. Comp. Phys.* **86**, pp. 187–210 (1990)

[28] H. W. Liepmann, A. Roshko, *Elements of Gasdynamics*, John Wiley & Sons, New York (1957)

[29] Mengjou Lin, Rose P. Tsang, David H. C. Du, Alan E. Klietz, Stephen Saroff *Performance Characteristics of the Connection Machine Hypertree Network* Jour. Par. Dist. Comp. **19**, 245–254 (1993)

[30] Alfred K. Louis, *Orthogonal Function Series Expansions and the Null Space of the Radon Transform,* SIAM J. Math. Anal **15** pp. 621–633 (1984)

[31] F. E. Marble, *Growth of a Diffusion Flame in the Field of a Vortex* in *Recent Advances in the Aerospace Sciences* (ed. Corrado Casci), pp. 395–412 (1985)

[32] F. E. Marble, E. E. Zukoski, J. W. Jacobs, G. J. Hendricks, I. A. Waitz, *Shock Enhancement and Control of Hypersonic Mixing and Combustion*, AIAA Paper 90 (1981)

[33] B. J. McBride, S. Heimel, J. G. Ehlers, S. Gordon, *Thermodynamic Properties to 6000° K for 210 Substances Involving the First 18 Elements*, NASA SP-3001 (1963)

[34] E.E. Meshkov, *Instability of the interface of two gases accelerated by a shock wave*, Izv. Akad. Nauk. SSSR Mekh. Zhidk. Gaza **4**, pp. 101–104 (1969)

[35] Frank Natterer, *The Mathematics of Computerized Tomography,* Teubner, Stuttgart (1986)

[36] J. M. Picone, J. P. Boris, *Vorticity generation by shock propagation through bubbles in a gas*, J. Fluid Mech. **189**, pp. 23–51 (1988)

[37] J. J. Quirk, S. Karni, *On the dynamics of a shock-bubble interaction*, submitted to J. Fluid Mech.

[38] A. G. Ramm, A. I. Zaslavsky, *Singularities of the Radon Transform,* Bull. Amer. Math. Soc., **28** No. 1, pp. 109–115 (1993)

[39] R. D. Richtmyer, *Taylor instability in shock acceleration of compressible fluids*, Commun. Pure Appl. Maths, **23**, pp. 297–319 (1960)

[40] A. Rogerson, E. Meiburg, *A numerical study of the convergence properties of ENO schemes*, J. Sci. Comput. **5**, No. 2 (1990)

[41] S. W. Rowland, *Computer Implementation of Image Reconstruction Formulas,* In *Image Reconstruction from Projections: Implementation and Application* Topics Appl. Phys., No. **32**. Ed. G. T. Herman, Springer Verlag, Berlin (1979)

[42] G. Rudiger, L. M. Somers, *Behaviour of small regions of different gases carried in accelerated gas flows*, J. Fluid Mech. **7**, pp. 161–176 (1960)

[43] R. Samtaney, N. J. Zabusky, *Circulation deposition on shock-accelerated planar and curved density-stratified interfaces: models and scaling laws*, J. Fluid Mech. **269**, pp. 45–78 (1994)

[44] C.-W. Shu, *TVB uniformly high-order schemes for conservation laws,* Math. Comp. **49** pp. 123–134 (1987)

[45] C. W. Shu, S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comp. Physics, **77**, pp. 493–471 (1988)

[46] C. W. Shu, S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes, II*, J. Comp. Physics, **83** pp. 32–78 (1989)

[47] C. W. Shu, G. Erlebacher, T. Zang, D. Whitaker, S. Osher, *High Order ENO Schemes Applied to Two- and Three-Dimensional Compressible Flow*, Appl. Numer. Math. **9**, pp. 45–71 (1992)

[48] K. T. Smith, D. C. Solmon and S. L. Wagner, *Practical and mathematical aspects of the problem of reconstructing objects from radiographs,* Bull. Amer. Math. Soc. **83**, pp. 1227–1270 (1977)

[49] M. A. Sussman, G. J. Wilson, *Computation of Chemically Reacting Flow Using a Logarithmic Form of the Species Conservation Equations,* Proceedings of the 4th International Symposium on Computational Fluid Dynamics (Davis, CA), **2**, pp. 1113–1118 (1991)

[50] R. A. Svehla, *Estimated viscosities and thermal conductivities of gases at high temperatures*, NASA TR R-132 (1962)

[51] G. Szegö, *Orthogonal Polynomials,* American Mathematical Society Colloquium Publications, Vol. XXIII, rev. ed., Providence, R.I., Amer. Math. Soc. (1959)

[52] V. T. Ton, *A numerical Method for Mixing/Chemically Reacting Compressible Flow with Finite Rate Chemistry*, Ph.D. thesis, UCLA (1993)

[53] K. Trimeche *The Radon transform and its dual associated with partial differential operators and applications to polynomials on the unit disk.* J. Comput. Appl. Math. **49** pp. 271–280 (1993)

[54] Ian A. Waitz, *Investigation of a Contoured Wall Injector for Hypervelocity Mixing Augmentation* AIAA Journal **31** No. 6, pp. 1014–1021 (1993)

[55] C. R. Wilke, *A viscosity equation for gas mixtures*, Chem. Phys. **18** no. 4, pp. 517–519 (1950)

[56] G. J. Wilson, M. A. Sussman, *Computation of Unsteady Shock-Induced Combustion Using Logarithmic Species Conservation Equations* AIAA Journal **31** No. 2, pp. 295–301 (1993)

[57] D. Yang, *Fast discrete Radon transform and 2-D discrete Fourier transform,* Electronics Letters **26** pp. 550–551 (1990)

[58] Joseph. Yang, T. Kubota, E. E. Zukoski, *A model for characterization of a vortex pair formed by shock passage over a light gas inhomogeneity*, J. Fluid Mech. **258**, pp. 217–244 (1994)

[59] Joseph Yang, Toshi Kubota, Edward E. Zukoski, *Applications of Shock-Induced Mixing to Supersonic Combustion* AIAA Journal **31** pp. 854–862 (1993)

[60] X. Yang, N. J. Zabusky, I. L. Chen, *Breakthrough' via dipolar-vortex/jet formation in shock-accelerated density-stratified layers.* Phys. Fluids A, **2**, pp. 892–895 (1990)

[61] S. G. Zaitsev, E. V. Lazareva, V. V. Chernukha, V. M. Belyaev, *Experimental investigation of the hydrodynamic instability of the interface between media of different density in an acceleration field.* translated from *Teplofizika Vysokikh Temperatur* **23**, pp. 535–541 (1985)