Pseudospectral Techniques For Non-Smooth Evolutionary Problems

By

Chris Guenther

A Dissertation

Submitted to Eberly College of Arts and Science

at

West Virginia University

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Mathematics

Department of Mathematics

Morgantown, West Virginia

1998

Abstract

Pseudospectral Methods for Non-Smooth Evolutionary Problems

By Chris Guenther

A pseudospectral approach is used to solve non-smooth evolutionary problems using Fourier collocation and Chebyshev collocation. It is well known that pseudospectral methods for smooth problems can offer superior accuracy over finite difference and finite element methods.

This paper explores the use of pseudospectral methods for non-smooth evolutionary problems in the area of hyperbolic heat transfer. Boundary and initial conditions are considered which cause instantaneous jumps, in the temperature and flux, prior to the propagation of a thermal wave into the medium. There is a considerable amount of literature that has investigated hyperbolic heat transfer under similar conditions, the common problems throughout theses investigations is the presence of numerical oscillation at the wave front. Finite difference and finite element methods have both been used, and both methods exhibit severe numerical oscillation at the wave front. In an attempt to reduce this oscillation extremely fine grids and severe timestep restrictions had to be introduced, but even these attempts still

exhibited some oscillation.

This paper will demonstrate that pseudospectral methods, when used correctly, can eliminate the numerical oscillation at the wave front and accurately resolve the instantaneous jump at the boundary. Furthermore, pseudospectral methods can be used successfully with coarser grids and larger timesteps and still provide superior results.

This paper will also investigate hyperbolic heat transfer with boundary conditions that contain a continuous periodic flux with surface radiation. These boundary conditions have never before been investigated in the literature on hyperbolic heat transfer. Previous research has only considered boundary conditions that contain a constant flux with radiation r a periodic on-off pulse with radiation. In either case, extremely fine grids were needed to prevent severe numerical oscillation. This paper will compare the hyperbolic and parabolic thermal response due to the periodic flux, under a wide range of frequencies, as well as show how pseudospectral methods can be used successfully in the case of periodic flux with surface radiation without the need to introduce fine spatial grids and prohibitively small timesteps.

Pseudospectral Techniques For

Non-Smooth Evolutionary Problems

By

Chris Guenther

A Dissertation

Submitted to

West Virginia University

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy of Mathematics

APPROVAL OF EXAMINING COMMITTEE

_____

I. Christie, Ph.D.
(committee chair)

_____

G. Ganser, Ph.D.

_____

C. Zhang, Ph.D.

_____

J. Wilder, Ph.D.

_____

M. Gautam, Ph.D.

Acknowledgement

I would like to take this opportunity to thank the members of my committee for all of their help during my four years at West Virginia University. I would especially like to thank Dr. Ian Christie for all of his support, suggestions, and friendly conversation. I also would like to thank the mathematics department for the opportunity to pursue my goal. Special thanks go out to Dr. Margaret Lyell for all the wonderful courses in fluids. These courses where not only educational and enjoyable, but have been a tremendous help in finding employment. Finally, I would like to thank my family, Coleman, Nelli, and Sidney for their continuous support. They were there from the beginning when things were so hard and my confidence was so low and have continued to give me love and support throughout. Had it not been form Coleman's encouragement and long talks I would have never gotten so far. To her I am eternally grateful.

List of Symbols

c speed of thermal wave

c_v specific heat at constant volume

D pseudospectal differentiation matrix

D^p p-th derivative

DFT discrete Fourier transform

D_{Ψ_j} Chebyshev collocation derivative

f_r reference heat flux

F_N NxN Fourier transform matrix

$\bar{F}_N$ complex conjugate of Fourier transform matrix

F_N^{-1} inverse of Fourier transform matrix

FFT fast Fourier transform

$g(\xi)$ coordinate transformation mapping

I_N interpolant operator

$IDFT$ inverse discrete Fourier transform

k thermal conductivity

L_∞ maximum error norm

N number of collocation polynomials

P_N permutation matrix

q heat flux

$q^*(x^*, t^*) = \frac{q(x,t)}{f_r}$ non-dimensional heat flux

t^n time at the n-th time level

$t^* = \frac{c^2 t}{2\alpha}$ non-dimensional time

T temperature

$T^*(x^*, t^*) = \frac{T(x,t) - T_0}{\alpha f_r / kc}$ non-dimensional temperature

$T_k(x)$ Chebyshev polynomial of degree k

$T_k^*(x)$ modified Chebyshev polynomial of degree k

$\hat{u}_k(t)$ collocation coefficient

$\hat{u}_k^{(1)}(t)$ first derivative collocation coefficient

$\hat{u}_k^{(p)}(t)$ p-th derivative collocation coefficient

$v(x)$ steady state solution

$\hat{V}$ quanity defined on midpoint mesh

$\widetilde{V}$ conservatively smoothed quanity

$w(x,t)$ transient solution

x_j grid point location

$x^* = \frac{cx}{2\alpha}$ non-dimensional spatial coordinate

Greek Symbols

α thermal diffusive coefficients

δ_{kl} Kronecker Delta function

ϵ surface absorptivity constant

λ_k Fourier collocation derivative coefficient

λ_k^p Fourier collocation p-th derivative coefficient

Λ diagonal matrix in pseudospectral differentiation matrix D

Λ^p diagonal matrix in pseudospectral differentiation matrix D^p

ξ transformed spatial coordinate

ρ density

$\tau = \frac{\alpha}{c^2}$ relaxation term for the modified Fourier's law

$\phi_k(x_j)$ trial function used in the interpolating polynomial

ψ_j Lagrange polynomial

ω boundary condition frequency of the flux

Tables

Table 2.1 Maximum error using Chebyshev collocation.

Table 2.2 Maximum error comparison of Chebyshev collocation and second order finite difference.

Table 2.3 Comparison between Chebyshev collocation and exact solution at the Gauss-Lobatto grid points.

List of Figures

Figure 1 Chebyshev polynomials.

Figure 2 Modified Chebyshev polynomials.

Figure 3 Second order polynomial approximation (Chebyshev polynomials)

Figure 4 Second order polynomial approximation (Modified Chebyshev polynomials).

Figure 5 Numerical vs. Exact solutions of KdV equation at t= .52.

Figure 6 Numerical solution of Burgers equation N=32.

Figure 7 Numerical solution of Burgers equation N=64.

Figure 8 Numerical solution of Burgers equation N=128.

Figure 9 Burgers equation with coordinate transformation N=32.

Figure 10 Burgers equation with coordinate transformation N=64.

Figure 11 Burgers equation with coordinate transformation N=128.

Figure 12 Hyperbolic temperature profile at t=.5, N=16.

Figure 13 Hyperbolic temperature profile at t=.75, N=16.

Figure 14 Hyperbolic temperature profile at t=1.0, N=16.

Figure 15 Exact solutions of hyperbolic vs. parabolic profiles at t=.5, omega=0.

Figure 16 Exact solutions of hyperbolic vs. parabolic profiles at t=.5, omega=.5.

Figure 17 Exact solutions of hyperbolic vs. parabolic profiles at t=.5, omega=1.

Figure 18 Exact solutions of hyperbolic vs. parabolic profiles at t=.5, omega=10.

Figure 19 Exact solutions of hyperbolic vs. parabolic profiles at t=.5, omega=100.

Figure 20 Surface response to a periodic flux omega=1.

Figure 21 Surface response to a periodic flux omega=10.

Figure 22 Surface response to a periodic flux omega=100.

Figure 23 Hyperbolic system numerical profile N=16.

Figure 24 Hyperbolic system numerical profile N=35.

Figure 25 Perturbing the initial conditions for the hyperbolic system.

Figure 26 Conservative smoothing for the hyperbolic system.

Figure 27 Hyperbolic system with surface radiation at t=.5.

Contents

1 Pseudospectral Methods **1**

1.1 Introduction 1
1.2 Fourier Collocation 4
1.3 Chebyshev Collocation 18
1.4 Boundary Conditions 38

2 Evolutionary Problems **42**

2.1 Introduction 42
2.2 Linear Problems 43
2.3 Non-Linear Problems 51
2.4 Coordinate Transformations 58

3 Hyperbolic Heat Transfer **63**

3.1 Introduction 63
3.2 Hyperbolic Heat Equation 67
3.3 Perturbing the Initial Conditions 81
3.4 Conservative Smoothing 83

Summary and Recommendations **92**

Bibliography **94**

A Fourier Collocation **99**

B Chebyshev Collocation **101**

C Modified Chebyshev Collocation **102**

D Figures **103**

Chapter 1

Pseudospectral Methods

1.1 Introduction

The solutions of most partial differential equations can not be found in closed form. Even on the rare occasion an exact solution can be found, it is usually hard to work with. Alternatively, numerical solutions can be found to most well posed partial differential equations. The more popular methods include finite difference and finite element. These methods are similar in that they both are applied locally along the domain of interest to produce a global approximation. Spectral methods, on the other hand, are numerical techniques applied globally across the entire domain of interest to produce

an approximation. Further discussion on the comparison of finite difference methods and spectral methods can be found in Fornberg (1987,1996).

Given a mixed initial boundary value problem, the idea behind spectral methods is to approximate the solution $u(x,t)$ by $\sum_k \hat{u}_k(t)\phi_k(x)$. The immediate questions that arise are how to select the trial functions $\phi_k(x)$ and how to determine the coefficients $\hat{u}_k(t)$.

For periodic problems the appropriate trial functions are trigonometric polynomials. For non-periodic problems, orthogonal polynomials of Jacobi type are the correct choice. Chebyshev and Legendre polynomials are the most common, and this work considers only Chebyshev polynomials. The expansion coefficients $\hat{u}_k(t)$ will be determined by the particular spectral method employed. The two types of spectral methods covered in this paper are Fourier collocation and Chebyshev collocation. These methods are also known as pseudospectral or collocation methods.

The idea of approximating a solution by a truncated series is certainly not a recent development. In the past the problem was how to numerically compute these series with a sufficiently large number of terms, and how to handle non-linear terms. For certain applications the problems mentioned above were virtually insurmountable which led to the use of finite difference or

finite element methods to numerically solve the problem. For many problems, especially in the area of fluid mechanics, the relatively low accuracy of these methods will not produce accurate representations of the flow.

In the 1970's, the need for accurate representation of complex flows led to a revival of Fourier series methods. Some of the earliest applications to partial differential equations were done by Kreiss and Oliger (1972) and Orszag (1972). The first comprehensive report on the theory of spectral methods was written by Gottlieb and Orszag (1977). The reason for the success of these methods during that decade is due to two facts. First, the rapid development of faster and more powerful computers, and second, efficient algorithms (Fast Fourier Transforms) to handle large sums. These improvements allowed one to efficiently transform the problem from physical space to the space of the trial functions. Spatial derivatives are calculated in this space and the problem is then inverted back to physical space. Once the problem has been inverted back to physical space, non-linear terms are evaluated as they appear in the problem and derivatives are replaced by their spectral approximations. Orszag (1972) called this treatment of non-linear terms pseudospectral. The literature also refers to this procedure as collocation. This paper will begin with this type of spectral method.

1.2 Pseudospectral (Fourier Collocation)

Define I_N to be an interpolant operator. If $u(x,t)$ is a periodic function of x then $I_N(u)$ is the trigonometric interpolation polynomial of u on some chosen set of grid points x_j. That is $I_N(u(x_j,t)) = u(x,t)_{x=x_j}$. The following set of grid points is common to Fourier collocation methods: $x_j = \frac{2\pi j}{N}, j = 0,1,2,\ldots,N-1$. The reason for the popularity of these grid points is the availability of Fast Fourier Transforms (FFT) which speed up the calculations for large N. For simplicity, this section considers N to be an integer power of 2, but the theory that follows is not restricted to this choice of N.

For numerical work, the basic issue is to replace $u(x,t)$ by $I_N(u)$ given in terms of a finite sum of trigonometric polynomials. Following the approach by Canuto et al.(1988) the interpolating function is

$$I_N(u) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_k(t)\phi_k(x_j). \tag{1.1}$$

For periodic problems the trial function is given by $\phi_k(x_j) = e^{ikx_j}$. Hence, the interpolant is

$$I_N(u) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_k(t)e^{ikx_j}. \tag{1.2}$$

Multiplying both sides of the interpolant by e^{-ilx_j}, $\frac{-N}{2} \leq l \leq \frac{N}{2} - 1$ and

summing over j, we get

$$\sum_{j=0}^{N-1} I_N(u)e^{-ilx_j} = \sum_{j=0}^{N-1} u(x_j,t)e^{-ilx_j} = \sum_{j=0}^{N-1} \sum_{k=\frac{-N}{2}}^{\frac{N}{2}-1} \hat{u}_k(t)e^{-ikx_j}e^{-ilx_j}$$

and, using the discrete orthogonality property,

$$\frac{1}{N}\sum_{j=0}^{N-1} e^{ipx_j} = \begin{cases} 1 & p = Nm, \quad m = 0, \pm 1, \pm 2, \ldots \\ 0 & otherwise \end{cases}$$

we get

$$\hat{u}_k(t) = \frac{1}{N}\sum_{j=0}^{N-1} u(x_j,t)e^{-ikx_j}. \tag{1.3}$$

The coefficient $\hat{u}_k(t)$ given above amounts to taking the discrete Fourier Transform (DFT) of $u(x,t)$ at the grid points x_j, while $I_N(u)$ supplies the inversion formula or inverse discrete Fourier transform (IDFT) i.e.,

$$u(x_j,t) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_k(t)e^{ikx_j}, \qquad j = 0, 1, \ldots N-1. \tag{1.4}$$

Problems that demonstrate the effectiveness of pseudospectral methods are differential equations with periodic boundary conditions. To calculate spatial derivatives required by these types of problems the DFT and IDFT can be used. For example, suppose the problem contains the term $\frac{\partial u}{\partial x}$. The pseudospectral approximation to the first derivative is

$$\frac{\partial u}{\partial x} \approx \frac{\partial(I_N(u))}{\partial x} = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_k(t)\frac{\partial(e^{ikx_j})}{\partial x_j} \tag{1.5}$$

where $\hat{u}_k(t)$ is given above. Hence, the first derivative can be approximated by first performing a DFT of $u(x,t)$ to find the discrete Fourier coefficients $\hat{u}_k(t)$. Multiplying these coefficients by the complex number $\frac{\partial(e^{ikx_j})}{\partial x_j}$ and summing over k, takes the result back to physical space and provides the approximation of the first derivative. The key to this type of differentiation is that derivatives of the interpolant $I_N(u)$ are taken exactly in Fourier space as opposed to finite difference methods where the derivative is approximated in physical space. Higher derivatives pose no real challenge since

$$\frac{\partial^p(I_N(u))}{\partial x^p} = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_k(t)\frac{\partial^p(e^{ikx_j})}{\partial x_j^p} \tag{1.6}$$

i.e., higher derivatives are still calculated in the same manner as the first derivative with just a change in the complex multiplication. For large problems ($N \approx 100$ or larger) the DFT and IDFT would be performed by an FFT algorithm which requires $O(N \log N)$ operations as opposed to $O(N^2)$ arithmetic operations without the FFT. For actual implementation of the above procedure it might be helpful to think of $u(x_j,t)$ not as an explicit function, but rather as a vector containing a discrete set of data points at some time level t. Since this approach for producing pseudospectral approximations of spatial derivatives is a linear process, an elegant alternative is to

use matrix-vector operations (Sanz-Serna 1994).

Sanz-Serna's approach defines the trial function as

$$\phi_k(x) = e^{\frac{2\pi k i}{L}x}, \qquad k = 0, \pm 1, \pm 2, \ldots. \tag{1.7}$$

where $\phi_k(x)$ forms a system of L-periodic pairwise orthogonal functions. Each L-periodic function $u(x,t) = u(x+L,t)$ in $L^2[0,L]$ can be represented by

$$u(x,t) = \sum_{k=-\infty}^{\infty} \hat{u}_k(t)\phi_k(x). \tag{1.8}$$

The series given above is the Fourier series of $u(x,t)$. For numerical work the Fourier series is truncated and we define the interpolating polynomial $I_N(u)$ as

$$I_N(u) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \hat{u}_k(t)\phi_k(x), \qquad N = 0, 1, \ldots. \tag{1.9}$$

The individual terms of this truncated series are the Fourier modes of $u(x,t)$. A crucial property of the interpolating polynomial is the fact that as $N \to \infty$, $\|u - I_N(u)\| \to 0$.

Discretizing the variable x by choosing the grid points $x_j = j\Delta x$, $\Delta x = \frac{L}{N}$, $\quad j = 0, 1, \ldots, N-1$, the interpolating polynomial becomes

$$I_N(u(x_j,t)) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \hat{u}_k(t) e^{\frac{2\pi k i j}{N}}. \tag{1.10}$$

Multiplying the interpolant by $e^{-\frac{2\pi kij}{N}}$ and summing over j yields

$$\hat{u}_k(t) = \frac{1}{N}\sum_{j=0}^{N-1} u(x_j,t)e^{\frac{-2\pi ikj}{N}}, \qquad \frac{-N}{2} \leq k \leq \frac{N}{2} \tag{1.11}$$

where we have used the fact $I_N(u(x_j,t)) = u(x_j,t)$.

Due to the periodicity of the problem, only the points $x_0, x_1, \ldots, x_{N-1}$ carry independent information. Hence, on the grid x_j, only the trial functions given by $\phi_0, \ldots, \phi_{N-1}$ are different, and $\phi_0(x) = \phi_N(x), \phi_{-1}(x) = \phi_{N-1}(x)$ and so on. This implies that the coefficients $\hat{u}_k(t)$ follow the relation $\hat{u}_{-n}(t) = \hat{u}_{N-n}(t)$. These relations are known as aliasing. By using the effect of aliasing, the coefficients $\hat{u}_k(t)$ can be expressed as

$$\hat{u}_k(t) = \frac{1}{N}\sum_{j=0}^{N-1} u(x_j,t)e^{\frac{-2\pi ikj}{N}}, \qquad k = 0,1,2,\ldots,N-1. \tag{1.12}$$

The fact that aliasing requires $\hat{u}_{\frac{-N}{2}} = \hat{u}_{\frac{N}{2}}$ and $\phi_{\frac{-N}{2}} = \phi_{\frac{N}{2}}$ gives the correct choice for the interpolation polynomial

$$I_N(u) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}}{}'' \hat{u}_k(t)\phi_k(x), \qquad N = 0,1,2,\ldots,\ldots. \tag{1.13}$$

The double prime denotes the first and last terms should be halved. Similarly, the derivative of the interpolant is

$$\frac{\partial I_N(u)}{\partial x} = \sum_{k=\frac{-N}{2}}^{\frac{N}{2}}{}'' \hat{u}_k(t)\lambda_k\phi_k(x), \qquad \lambda_k = \frac{2\pi ik}{L}. \tag{1.14}$$

The right hand side of the derivative of the interpolant is

$$\frac{1}{2}(\hat{u}_{\frac{-N}{2}}\lambda_{\frac{-N}{2}}\phi_{\frac{-N}{2}}) + \ldots + \ldots + \hat{u}_{\frac{N}{2}-1}\lambda_{\frac{N}{2}-1}\phi_{\frac{N}{2}-1} + \frac{1}{2}(\hat{u}_{\frac{N}{2}}\lambda_{\frac{N}{2}}\phi_{\frac{N}{2}}).$$

Using $\hat{u}_{\frac{-N}{2}} = \hat{u}_{\frac{N}{2}}$ and $\phi_{\frac{-N}{2}} = \phi_{\frac{N}{2}}$ along with the result

$$\lambda_{\frac{-N}{2}} = \frac{2\pi i(\frac{-N}{2})}{L} = -\frac{2\pi i(\frac{N}{2})}{L} = -\lambda_{\frac{N}{2}}$$

implies that the terms $|k| = \frac{N}{2}$ do not contribute. Hence,

$$\frac{\partial I_N(u)}{\partial x} = \sum_{k=-(\frac{N}{2}-1)}^{(\frac{N}{2}-1)} \hat{u}_k(t)\lambda_k\phi_k(x), \qquad \lambda_k = \frac{2\pi ik}{L}. \tag{1.15}$$

Although the terms $|k| = \frac{N}{2}$ do not contribute because of the first derivative operator, higher derivatives will have different contributions.

The coefficients

$$\hat{u}_k(t) = \frac{1}{N}\sum_{j=0}^{N-1} u(x_j,t)e^{\frac{-2\pi ikj}{N}}, \qquad k = 0,1,\ldots,N-1 \tag{1.16}$$

can be expressed in matrix form as

$$\begin{bmatrix} \hat{u}_0(t) \\ \vdots \\ \hat{u}_{N-1}(t) \end{bmatrix} = \tfrac{1}{N}F_N \begin{bmatrix} u(x_0,t) \\ \vdots \\ u(x_{N-1},t) \end{bmatrix}$$

where the (k,j) element of the matrix F_N, $k,j = 0,1,\ldots N-1$ is $\omega_N = e^{-\frac{2\pi i}{N}}$ and the (k,j) entry is the kj-th power of ω_N.

The N x N matrix given above is denoted by F_N and performs a discrete Fourier transform of $u(x_j, t)$. For example, if $N = 4$ then, $\omega_4 = e^{-\frac{\pi i}{2}} = -i$ and

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}.$$

Recall that the first derivative of the interpolant $I_N(u)$ was given as

$$\frac{\partial(I_N(u))}{\partial x} = \sum_{k=-(\frac{N}{2}-1)}^{\frac{N}{2}-1} \hat{u}_k(t)\lambda_k e^{ikx_j}, \qquad \lambda_k = \frac{2\pi ki}{L}.$$

This process of complex multiplication times the discrete Fourier coefficients can be expressed as the product of a diagonal matrix Λ times PF_N where P is a permutation matrix given by

$$P = \begin{bmatrix} O_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & O_{\frac{N}{2}} \end{bmatrix},$$

$I_{\frac{N}{2}}$ is the $\frac{N}{2}$ x $\frac{N}{2}$ identity matrix, and $O_{\frac{N}{2}}$ is the zero matrix. The permutation matrix P is needed to make sure the complex number λ_k is multiplying the correct discrete Fourier coefficient. Once the permutation P has been

performed on F_N the diagonal matrix Λ which performs the complex multiplication is given by

$$\Lambda = Diag(0, \lambda_{\frac{-N}{2}+1}, \lambda_{\frac{-N}{2}+2}, \ldots, \lambda_{-1}, \lambda_0, \lambda_1, \ldots, \lambda_{\frac{N}{2}-1}). \tag{1.17}$$

Notice the zero term of Λ. This entry ensures there is no contribution from the $|k| = \frac{N}{2}$ terms. The process of finding an IDFT is also linear and F_N^{-1} is given by the simple relation

$$F_N^{-1} = \frac{1}{N}\bar{F}_N \tag{1.18}$$

where the bar over F_N denotes complex conjugation. The first derivative is now approximated by the following formula

$$\frac{\partial u(x_j, t)}{\partial x} \approx \frac{1}{N}\bar{F}_N P \Lambda P F_N \underline{u}(x_j, t). \tag{1.19}$$

The bar under u denotes the fact that $u(x_j, t)$ is really a vector whose entries are data values on the grid x_j. The matrix $\frac{1}{N}\bar{F}_N P \Lambda P F_N$, denoted by D, is called the pseudospectral differentiation matrix. In summary, $\frac{\partial}{\partial x}$ is approximated by D which operates on the vector $\underline{u}$ containing a given set of data values on the grid x_j at some time level t. D is composed of three parts. The first part is PF_N which finds the discrete Fourier transform of $u(x_j, t)$ and permutes the result to perform the correct complex multiplication. Λ

is a diagonal matrix which multiplies the discrete Fourier transform by the complex number λ_k. The last part $\frac{1}{N}F_N P$ permutes the previous result back to its original order and then performs the inverse discrete Fourier transform to bring the result back to physical space. $D\underline{u}$ represents the pseudospectral approximation of the first derivative of $\underline{u}$ at the grid points x_j.

To take spatial derivatives using the pseudospectral matrix D^p we would never actually calculate the product $\bar{F}_N P \Lambda^p P F_N$. Instead, the only matrix-vector multiplication is done by the Fourier transform matrix F_N and the inverse $\frac{1}{N}\bar{F}_N$. The operation by the permutation matrix P can be done by simply rearranging the terms of the vector $F_N \underline{u}(x_j, t)$. Since Λ^p is a diagonal matrix, the product $\Lambda^p P F_N \underline{u}(x_j, t)$ is done componentwise. The permutation of this result is again done by rearranging the components to undo the action of the first permutation. This same procedure would also be advisable using an FFT approach the cost being $O(N)$. A FORTRAN subroutine of this process is included in Appendix A.

The pseudospectral differentiation matrix D is a full matrix. When using finite difference methods to approximate derivatives the process can also be represented by a matrix-vector operation. For example, using a second order

finite difference method

$$\frac{\partial u(x_j,t)}{\partial x} \approx \frac{u(x_{j+1},t)-u(x_{j-1},t)}{2\Delta x}, \qquad j=0,1\ldots,N-1.$$

The right hand side of this approximation can be written as

$$\begin{bmatrix} 0 & \frac{1}{2\Delta x} & 0 & \ldots & \ldots & -\frac{1}{2\Delta x} \\ -\frac{1}{2\Delta x} & 0 & \frac{1}{2\Delta x} & 0 & \ldots & 0 \\ 0 & -\frac{1}{2\Delta x} & 0 & \frac{1}{2\Delta x} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{2\Delta x} & 0 & 0 & \ldots & -\frac{1}{2\Delta x} & 0 \end{bmatrix} \begin{bmatrix} u(x_0,t) \\ u(x_1,t) \\ \vdots \\ \vdots \\ \vdots \\ u(x_{N-1},t) \end{bmatrix}.$$

The use of finite difference methods produces sparse matrices. For the case of periodic problems these matrices are circulant. To increase the accuracy of a finite difference approximation, 4th order or 6th order central difference schemes could be used. The effect of the increased accuracy using matrices would be 4 non-zero or 6 non-zero elements per row and as more and more accuracy is obtained, the number of non-zero elements per row is increased. Pseudospectral methods can be viewed as the limit of finite difference matrices as the order of accuracy increases. Further details on the relationship between pseudospectral and finite difference matricies can be found in Forn-

berg(1996).

One of the advantages of a pseudospectral approach is the treatment of boundary terms. In finite difference methods of increasing order, the cost of evaluating boundary terms also increases since finite difference formulas at the boundary require grid values outside the computational domain. The use of ghost points (Smith 1985) or other such techniques must be used to evaluate boundary terms. Fourier collocation methods do not require any special treatment of the boundary terms because the trial functions automatically satisfy the boundary conditions due to the periodicity of the problem. Fourier collocation methods follow closely the actual conditions of the differential equation.

Higher derivatives, as we saw earlier, are simply found by the appropriate change in the complex multiplication. This is also the case using the pseudospectral differentiation matrix D. The approximation of $\frac{\partial^p}{\partial x^p}$ results in λ_k^p as the complex multiplication. This process is easily implemented by an appropriate change from Λ to Λ^p.

The p-th derivative of the interpolant is

$$\frac{\partial^p I_N(u)}{\partial x^p} = \sum_{k=\frac{-N}{2}}^{\frac{N}{2}}{}'' \hat{u}_k(t)(\lambda_k)^p \phi_k(x). \tag{1.20}$$

For example, the second derivative would be

$$\frac{\partial^2 I_N(u)}{\partial x^2} = \sum_{k=\frac{-N}{2}}^{\frac{N}{2}}{}'' \hat{u}_k(t)(\lambda_k)^2 \phi_k(x). \tag{1.21}$$

Now

$$\lambda^2_{\frac{-N}{2}} = \left(\frac{2\pi i(\frac{-N}{2})}{L}\right)^2 = -\left(\frac{2\pi(\frac{N}{2})}{L}\right)^2 = \left(\frac{2\pi i(\frac{N}{2})}{L}\right)^2 = \lambda^2_{\frac{N}{2}}, \tag{1.22}$$

and

$$\Lambda^2 = Diag(\lambda^2_{\frac{-N}{2}}, \ldots, \lambda^2_{-1}, \lambda^2_0, \lambda^2_1, \ldots, \lambda^2_{\frac{N}{2}-1}). \tag{1.23}$$

Hence, for the second derivative there is a contribution from the $|k| = \frac{N}{2}$ terms. In general, for higher derivatives, the only change in the pseudospectral differentiation matrix D is the diagonal matrix Λ.

The matrix representation for the p-th derivative operator is

$$D^p = \frac{1}{N}\bar{F}_N P \Lambda^p P F_N, \tag{1.24}$$

where

$$\Lambda^p = Diag(\lambda^p_{\frac{-N}{2}}, \ldots, \lambda^p_{-1}, \lambda^p_0, \lambda^p_1, \ldots, \lambda^p_{\frac{N}{2}}). \tag{1.25}$$

A word of caution is in order when evaluating Λ^p. For example, $\Lambda^2 \neq \Lambda\Lambda$ since the first entry of Λ is zero the first entry of the product $\Lambda\Lambda$ would also be zero, but we have just seen this is not the case for Λ^2. As Sanz-Serna(1994)

points out, the difference between Λ^2 and $\Lambda\Lambda$ is small due to the fact that the difference between the two quantities only involves the highest Fourier modes of $u(x_j, t)$.

Unfortunately, pseudospectral methods require that the problem have periodic boundary conditions. Finite difference and finite element methods have no such limitation. For problems that do have periodic boundary conditions, the accuracy of a finite difference or finite element method versus Fourier collocation is not even a close race. For smooth solutions, the errors using a p-th order finite difference method are never better than (Δx^p). For pseudospectral methods the solution error goes to zero as $N \to \infty$. This level of accuracy is typically referred to as spectral accuracy. The rapid rate of convergence of pseudospectral methods is due to the fact that the interpolating polynomial $I_N(u)$ shares the same type of convergence properties as a truncated Fourier series. Both the discrete Fourier coefficients and the coefficients of a truncated Fourier series decaying at a rate faster than algebraically. The relationship between discrete Fourier coefficients and the coefficients u_k of a infinite Fourier series is

$$\hat{u}_k = u_k + \sum_{m=-\infty, m\neq 0}^{\infty} u_{k+Nm}, \qquad k = -\frac{N}{2}, \ldots, \frac{N}{2} - 1. \tag{1.26}$$

This formula shows that the discrete Fourier coefficients depend not only on the k-th mode of $u(x,t)$, but on all the modes of $u(x,t)$ which are indistinguishable on the grid. This is the effect of aliasing and, for k large enough, the errors due to this effect are small. Further details on the effects of aliasing in spectral methods can be found in Canuto et al.(1988).

The rapid rate of convergence due to the decay of the Fourier coefficients allows the use of very coarse grids while still achieving excellent results. By using such coarse grids, many periodic problems can be solved with the same or even less computational cost than finite difference or finite element methods.

As the work proceeds it will become clear that non-periodic and non-smooth data do present problems, but are not reason enough to abandon spectral methods in favor of finite difference or finite element methods. Different analytical and numerical tools can be used on many problems that have non-smooth data or are non-periodic. Examples of problems with non-smooth data are given in Chapters 2 and 3. The use of spectral methods for non-periodic problems is the subject of the next section and is also discussed in Chapters 2 and 3.

1.3 Pseudospectral(Chebyshev Collocation)

For non-periodic problems we replace $u(x,t)$ by the interpolant

$$I_N(u) = \sum_{k=0}^{N} \hat{u}_k(t)\phi_k(x). \tag{1.27}$$

The term pseudospectral still applies since the treatment of non-linear terms will remain the same. What has changed is the choice of trial functions. For non-periodic problems the most popular choice for the trial functions $\phi_k(x)$ are Chebyshev polynomials. Again we require $I_N(u(x_j,t)) = u(x,t)_{x=x_j}$ on some chosen set of grid points x_j. Since the properties of Chebyshev polynomials are not as common as the properties of Fourier series, we will begin with a discussion of Chebyshev polynomials on the interval $[-1,1]$.

Chebyshev polynomials on $[-1,1]$ are defined as

$$T_k(x) = \cos k\theta, \qquad \theta = \arccos x. \tag{1.28}$$

From the trigonometric identity

$$\cos(k+1)\theta + \cos(k-1)\theta = 2\cos\theta\cos k\theta$$

we have the relation

$$T_{k+1}(x) + T_{k-1}(x) = 2xT_k(x)$$

which gives the following three term recurrence relation

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x). \tag{1.29}$$

When $k = 0$, $T_0(x) = 1$ and when $k = 1$, $T_1(x) = \cos(\arccos x) = x$. With these two Chebyshev polynomials and using the recurrence relation above we can generate every Chebyshev polynomial. The next several are given by

$$\begin{aligned}
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
T_5(x) &= 16x^5 - 20x^3 + 5x \\
&\vdots
\end{aligned}$$

The first six Chebyshev polynomials are shown in figure 1. From this figure it is easily seen that $T_k(\pm 1) = (\pm 1)^k$ and $|T_k(x)| \leq 1$. For problems that involve non-linear terms the following is very important:

$$\begin{aligned}
T_s(x)T_r(x) &= \cos(s\theta)\cos(r\theta) \\
&= \frac{1}{2}(\cos(s+r)\theta + \cos(s-r)\theta) \\
&= \frac{1}{2}(T_{s+r}(x) + T_{s-r}(x)).
\end{aligned} \tag{1.30}$$

For numerical work we will be interested in spatial derivatives of the interpolant of u. A very useful property in calculating spatial derivatives of Chebyshev polynomials is found as follows:

$$\begin{aligned}
\int T_k(x)dx &= \int \cos k \arccos x dx \\
&= -\int \cos k\theta \sin\theta d\theta \\
&= -\frac{1}{2}\int (\sin(1-k)\theta + \sin(1+k)\theta)d\theta \\
&= -\frac{1}{2}\int (\sin(k+1)\theta - \sin(k-1)\theta)d\theta \\
&= \frac{1}{2}\left(\frac{1}{k+1}\cos(k+1)\theta - \frac{1}{k-1}\cos(k-1)\theta\right) \\
&= \frac{1}{2}\left(\frac{T_{k+1}(x)}{k+1} - \frac{T_{k-1}(x)}{k-1}\right), k = 2, \ldots, N. \qquad (1.31)
\end{aligned}$$

For the special cases $k = 0$ and $k = 1$ we have

$$\int T_0(x)dx = T_1(x) \qquad (1.32)$$

and

$$\int T_1(x)dx = \int x dx = \frac{x^2}{2},$$

where the constants of integration are provided by the normalization $T_k(1) = 1$. Writing this result in terms of Chebyshev polynomials we get

$$\int T_1(x)dx = \frac{1}{4}\left(T_0(x) + T_2(x)\right). \qquad (1.33)$$

This property allows the calculation of derivatives of $T_k(x)$ in terms of other Chebyshev polynomials.

Define $p(x)$ to be a finite Chebyshev series

$$p(x) = \sum_{k=0}^{N} a_k T_k(x). \tag{1.34}$$

When we differentiate a Chebyshev polynomial of degree n the result is an (n-1) degree polynomial. We want to be able to evaluate derivatives of Chebyshev polynomials in terms of other Chebyshev polynomials. Letting

$$p^{'}(x) = \sum_{k=0}^{N-1} c_k T_k(x)$$

and integrating the equation with respect to x yields

$$\int p^{'}(x)dx = \sum_{k=0}^{N-1} c_k \int T_k(x)dx.$$

Using (1.31), the right hand side becomes

$$c_0 \int T_0(x)dx + c_1 \int T_1(x)dx +$$
$$\frac{1}{2}\sum_{k=2}^{N-1} c_k \left(\frac{1}{k+1}T_{k+1}(x) - \frac{1}{k-1}T_{k-1}(x)\right),$$

and, equating this to $p(x)$, we get

$$\sum_{k=0}^{N} a_k T_k(x) = a_0 T_0(x) + c_0 T_1(x) +$$
$$\frac{c_1}{4}T_2(x) + \frac{1}{2}\sum_{k=2}^{N-1} c_k \left(\frac{1}{k+1}T_{k+1}(x) - \frac{1}{k-1}T_{k-1}(x)\right).$$

Matching coefficients we have the following relations for $k = 0, 1$

$$a_0 = a_0, \qquad a_1 = c_0 - \frac{c_2}{2},$$

for $2 < k < N - 1$ we have

$$a_k = \frac{1}{2k}\left(c_{k-1} - c_{k+1}\right),$$

and for $k = N - 1, N$

$$a_{N-1} = \frac{c_{N-2}}{2(N-1)}, \qquad a_N = \frac{c_{N-1}}{2N}.$$

Calculating c_k in succession for decreasing k gives the recurrence relation

$$\begin{aligned} c_{N+1} &= 0 \\ c_N &= 0 \\ c_{k-1} &= c_{k+1} + 2ka_k, \qquad k = N-2, \ldots, 2 \end{aligned}$$

and for $k = 1$, $2c_0 = c_2 + 2a_1$. This enables us to relate derivatives of a Chebyshev polynomial in terms of other Chebyshev polynomials i.e., we can write

$$\frac{\partial}{\partial x}\sum_{k=0}^{N} \hat{u}_k(t)T_k(x) = \sum_{k=0}^{N} \hat{u}_k(t)\frac{d}{dx}T_k(x) = \sum_{k=0}^{N} \hat{u}_k^{(1)}(t)T_k(x),$$

where $\hat{u}_k^{(1)}(t)$ denotes the new coefficients that results in taking the first derivative of $T_k(x)$. Using the recurrence relation for c_k and changing the

subscript from k to $k+1$, we have the following recurrence relation in terms of $\hat{u}_k^{(1)}$

$$\begin{aligned}
\hat{u}_{N+1}^{(1)}(t) &= 0 \\
\hat{u}_N^{(1)}(t) &= 0 \\
c_k \hat{u}_k^{(1)}(t) &= \hat{u}_{k+2}^{(1)}(t) + 2(k+1)\hat{u}_{k+1}(t), \quad k = N-2, \ldots, 1, 0 \quad (1.35)
\end{aligned}$$

where $c_k = 2$ if $k = 0, N$ and $c_k = 1$ if $0 < k < N$. For the second derivative of the interpolant

$$\frac{\partial^2}{\partial x^2} \sum_{k=0}^{N} \hat{u}_k(t) T_k(x) = \sum_{k=0}^{N} \hat{u}_k(t) \frac{d^2}{dx^2} T_k(x) = \sum_{k=0}^{N} \hat{u}_k^{(2)}(t) T_k(x)$$

and using an analogous procedure to that described previously, the coefficients $\hat{u}_k^{(2)}(t)$ are

$$\begin{aligned}
\hat{u}_{N+1}^{(2)}(t) &= 0 \\
\hat{u}_N^{(2)}(t) &= 0 \\
c_k \hat{u}_k^{(2)}(t) &= \hat{u}_{k+2}^{(2)}(t) + 2(k+1)\hat{u}_{k+1}^{(1)}(t), \quad k = N-2, \ldots, 1, 0. \quad (1.36)
\end{aligned}$$

For higher derivatives of order p the recurrence relation for the coefficients is

$$\begin{aligned}
\hat{u}_{N+1}^{(p)}(t) &= 0 \\
\hat{u}_N^{(p)}(t) &= 0 \\
c_k \hat{u}_k^{(p)}(t) &= \hat{u}_{k+2}^{(p)}(t) + 2(k+1)\hat{u}_{k+1}^{(p-1)}(t), \quad k = N-2, \ldots, 1, 0. \quad (1.37)
\end{aligned}$$

Problems involving spatial derivatives of any order can be calculated by these recurrence relations. An important feature of these coefficients is that they can be generated simply, with a single loop of a computer program, once the coefficient $\hat{u}_k(t)$ has been found.

To find the coefficients $\hat{u}_k(t)$ first recall the interpolant using Chebyshev polynomials given by

$$I_N(u) = \sum_{k=0}^{N} \hat{u}_k(t) T_k(x).$$

By requiring the interpolant to satisfy $u(x,t)$ at a chosen set of grid points x_j we have the following expansion of $u(x,t)$ in terms of Chebyshev polynomials

$$I_N(u) = \sum_{k=0}^{N} \hat{u}_k(t) T_k(x_j), \quad x_j = \cos\frac{\pi j}{N}, \quad j = 0, 1, \ldots, N. \tag{1.38}$$

This particular choice of grid points is called Gauss-Lobatto. Gauss-Lobatto points are the most common points used, others can be found in Canuto et al.(1988). An interesting fact is that the Gauss-Lobatto points, as well as the others mentioned above, are not evenly spaced along $[-1, 1]$, but rather the distance between the points decreases quadratically as the grid points approach ± 1. This concentration of points near the boundaries is very helpful in problems with boundary layers or problems with difficult boundary conditions. Unfortunately, this type of spacing of the grid points also plays

an important role in the severe stability restrictions required in some problems when using Chebyshev collocation. As we saw earlier, Fourier collocation uses evenly spaced grid points. Even spacing could result in very poor approximations using Chebyshev polynomials due to Runge phenomena (Issacson and Keller 1966). Under certain circumstances (singularities in the complex plane) Runge phenomena causes the approximation to diverge with increasing N values even though the function is continuously differentiable.

An expression for the discrete Chebyshev coefficients $\hat{u}_k(t)$ can be found from

$$\sum_{j=0}^{N} \frac{1}{c_j} T_k(x_j) T_l(x_j) = \frac{N}{2} c_k \delta_{kl} \tag{1.39}$$

where $c_0 = c_N = 2$, $c_k = 1$, for $0 < k < N$. This property is called the discrete orthogonality property.

Recall $T_k(x) = \cos k\theta, \theta = \arccos x$ at the Gauss-Lobatto points $x_j = \cos \frac{\pi j}{N}, T_k(x_j) = \cos \frac{kj\pi}{N}$. From the discrete orthogonality property

$$\begin{aligned}
\sum_{j=0}^{N} \frac{1}{c_j} T_k(x_j) T_l(x_j) &= \sum_{j=0}^{N} \frac{1}{c_j} \cos \frac{kj\pi}{N} \cos \frac{kl\pi}{N} \\
&= \frac{1}{2} \sum_{j=0}^{N} \frac{1}{c_j} \left(\cos \frac{(k+l)\pi}{N} j + \cos \frac{(k-l)\pi}{N} j \right) \\
&= \frac{1}{2} \sum_{j=0}^{N} \frac{1}{c_j} \cos \theta_1 j + \frac{1}{2} \sum_{j=0}^{N} \frac{1}{c_j} \cos \theta_2 j
\end{aligned}$$

where

$$\theta_1 = \frac{(k+l)\pi}{N}, \qquad \theta_2 = \frac{(k-l)\pi}{N}.$$

Expanding the right hand leads to

$$\begin{aligned} &\tfrac{1}{2} \quad \left[\frac{1}{2} + \cos\theta_1 + \ldots + \cos(N-1)\theta_1 + \frac{1}{2}\cos N\theta_1\right] + \\ &\tfrac{1}{2} \quad \left[\frac{1}{2} + \cos\theta_2 + \ldots + \cos(N-1)\theta_2 + \frac{1}{2}\cos N\theta_2\right]. \end{aligned}$$

Using the trigonometric identity

$$\frac{1}{2} + \cos\theta + \ldots + \cos(N-1)\theta + \frac{1}{2}\cos N\theta = \frac{1}{2}\sin N\theta \cot\frac{\theta}{2}$$

in this expansion yields

$$\frac{1}{4}\left[\sin N\theta_1 \cot\frac{\theta_1}{2} + \sin N\theta_2 \cot\frac{\theta_2}{2}\right].$$

When $k \neq l$, $\sin N\theta_1 = \sin N\theta_2 = 0$. For $k = l$ and $k, l \neq 0, N$ the equation becomes

$$\frac{1}{2}\sum_{j=0}^{N}\frac{1}{c_j}\left(\cos\frac{(k+l)\pi}{N}j + \cos\frac{(k-l)\pi}{N}j\right)$$

$$\begin{aligned} &= \frac{1}{2}\sum_{j=0}^{N}\frac{1}{c_j}\left(\cos\frac{2k\pi}{N}j + 1\right) \\ &= \frac{1}{2}\sum_{j=0}^{N}\frac{1}{c_j}\cos\theta j + \frac{1}{2}\sum_{j=0}^{N}\frac{1}{c_j}, \qquad \theta = \frac{2\pi}{N} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\left[\frac{1}{2} + \cos\theta + \ldots + \cos(N-1)\theta + \frac{1}{2}\cos N\theta\right] \\
&+ \frac{1}{2}\left[\frac{1}{2} + 1 + \ldots + 1 + \frac{1}{2}\right] \\
&= \frac{1}{4}\sin N\theta \cot\frac{\theta}{2} + \frac{N}{2} \\
&= \frac{N}{2},
\end{aligned}
$$

since $\sin N\theta = 0$. When $k = l = 0$, we have

$$
\sum_{j=0}^{N} \frac{1}{c_j} T_0(x_j) T_0(x_j) = \sum_{j=0}^{N} \frac{1}{c_j} = N.
$$

When $k = l = N$, the equation reduces to

$$
\sum_{j=0}^{N} \frac{1}{c_j} T_N(x_j) T_N(x_j) = \sum_{j=0}^{N} \frac{1}{c_j} \cos^2 j\pi = \sum_{j=0}^{N} \frac{1}{c_j} = N,
$$

which proves the discrete orthogonality relation.

Multiplying the interpolant of u by $\frac{1}{c_j}$ gives us

$$
\frac{1}{c_j} u(x_j, t) = \frac{1}{c_j} \sum_{k=0}^{N} \hat{u}_k(t) T_k(x_j).
$$

Multiplying this result by $T_l(x_j)$ and summing over j gives

$$
\begin{aligned}
\sum_{j=0}^{N} \frac{1}{c_j} u(x_j, t) T_l(x_j) &= \sum_{j=0}^{N} \frac{1}{c_j} \sum_{k=0}^{N} \hat{u}_k(t) T_k(x_j) T_l(x_j) \\
&= \sum_{k=0}^{N} \hat{u}_k(t) \sum_{j=0}^{N} \frac{1}{c_j} T_k(x_j) T_l(x_j).
\end{aligned}
$$

Now apply the discrete orthogonality relation for $l = k$,

$$\frac{N}{2} c_k \hat{u}_k(t) = \sum_{j=0}^{N} \frac{1}{c_j} u(x_j, t) T_k(x_j),$$

to yield the formula for the discrete Chebyshev coefficients given by

$$\hat{u}_k(t) = \frac{2}{N c_k} \sum_{j=0}^{N} \frac{1}{c_j} u(x_j, t) T_k(x_j), \qquad k = 0, 1, \ldots, N. \tag{1.40}$$

To summarize the results, we are interested in numerically solving non-periodic differential equations using the method of Chebyshev collocation. For example, the approximation of $\frac{\partial u}{\partial x}$ would be

$$\frac{\partial u}{\partial x} \approx \frac{\partial u(x_j, t)}{\partial x} = \sum_{k=0}^{N} \hat{u}_k^{(1)} T_k(x_j) \tag{1.41}$$

and $\frac{\partial^2 u}{\partial x^2}$ is

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{\partial^2 u(x_j, t)}{\partial x^2} = \sum_{k=0}^{N} \hat{u}_k^{(2)} T_k(x_j). \tag{1.42}$$

For either of the examples above, or even higher derivatives, the first step is to determine the discrete Chebyshev coefficients given by (1.40). This step amounts to transforming the problem to Chebyshev space. Applying the recurrence relation given in (1.37) allows us to find $\hat{u}_k^{(p)}(t)$, $p = 1, 2, \ldots, M$. Once $\hat{u}_k^{(p)}(t)$ has been found, the last step in finding the approximation of $\frac{\partial^p u}{\partial x^p}$ is to evaluate

$$\sum_{k=0}^{N} \hat{u}_k^{(p)}(t) T_k(x_j). \tag{1.43}$$

This step is the inverse, taking the problem from Chebyshev space back to physical space. Again, the differentiation is taking place in the space of the trial functions i.e., Chebyshev space, as opposed to approximating the derivative in physical space. A FORTRAN subroutine for taking derivatives in Chebyshev space is provided in Appendix B.

A very nice feature of approximating spatial derivatives as outlined above using Gauss-Lobatto points is the availability of an FFT to determine $\hat{u}_k(t)$, and to invert the problem back to physical space. This is one of the main reasons for the popularity of Gauss-Lobatto points.

The process of finding spatial derivatives using Chebyshev collocation is also a linear process. Hence, an alternative way to calculate derivatives is by a matrix operation. The matrix we require can be generated by differentiating the Lagrange polynomials ψ_j which are 1 at the grid points x_j and zero at all other collocation points. For the Gauss-Lobatto points, the Lagrange polynomials are

$$\psi_j(x) = \frac{(-1)^{j+1}(1-x^2)T_N^{'}(x)}{c_j N^2(x-x_j)}. \tag{1.44}$$

Solomonoff and Turkel(1989) provide details on differentiating ψ_j. The result,

in matrix form, is

$$D_{\psi_j} = \begin{cases} \frac{c_l}{c_j}\frac{(-1)^{l+j}}{(x_l - x_j)} & l \neq j \\ \frac{-x_j}{2(1-x_j^2)} & 1 \leq l = j \leq N-1 \\ \frac{2N^2+1}{6} & l = j = 0 \\ -\frac{2N^2+1}{6} & l = j = N. \end{cases}$$

This is used to approximate $\frac{\partial u}{\partial x}$ by operating on a vector of grid values, $u(x_j, t)$, by the matrix D_{ψ_j} i.e.,

$$\frac{\partial u}{\partial x} \approx D_{\psi_j}(u(x_j, t)). \tag{1.45}$$

The second derivative $D^2_{\psi_j}$ can be found in Peyret(1986).

Matrix differentiation requires $O(N^2)$ operations compared to $O(N \log N)$ for Chebyshev transform methods using an FFT. This difference in operations makes matrix differentiation practical only for problems where N is small. For such problems, the difference in errors incurred in calculating derivatives by matrix or transform methods is small. Breuer and Everson (1992) point out, that for large N, matrix differentiation magnifies the differentiation errors by an additional factor of N^2.

Using Chebyshev collocation to approximate spatial derivatives results in the magnitude of the error being the most extreme near the boundaries

of the computational domain. The source of these errors is the coupling of coefficients found in the recurrence relation which links coefficients of a Chebyshev polynomial to those of its derivative. This is not the case in Fourier collocation where the entries of Λ represent the coefficients due to differentiation which do not depend on the value of any other coefficient. It is for this reason that errors incurred in approximating derivatives using Fourier collocation are smaller than those using Chebyshev collocation and are evenly distributed throughout the computational domain.

Until now, we have restricted our Chebyshev collocation method to the interval $[-1, 1]$. Any finite interval $a \leq x^{'} \leq b$ can be mapped to the range $-1 \leq x \leq 1$ by

$$x^{'} = \frac{1}{2}(b-a)x + \frac{1}{2}(b+a). \tag{1.46}$$

Since many problems are solved on the interval $[0, 1]$ we will discuss Chebyshev collocation on the interval $[0, 1]$ next. This particular interval is so common that the Chebyshev polynomials defined on $[0, 1]$ are called modified Chebyshev polynomials and are denoted $T_k^*(x)$.

$$T_k^*(x) \quad = \quad T_k(2x-1), \qquad 0 \leq x \leq 1, k = 0, 1, \ldots$$

The first few polynomials are

$$\begin{aligned} T_0^*(x) &= 1 \\ T_1^*(x) &= 2x - 1 \\ T_2^*(x) &= 8x^2 - 8x + 1 \\ T_3^*(x) &= 32x^3 - 48x^2 + 18x - 1 \end{aligned}$$

and with the three term recurrence relation

$$T_{k+1}^*(x) = 2(2x-1)T_k^*(x) - T_{k-1}^*(x) \tag{1.47}$$

we can generate any order modified Chebyshev polynomial we choose. The graph of the first six modified Chebyshev polynomials can be found in Figure 2.

Rather than using the modified Chebyshev polynomials, we could simply map the interval $[0,1]$ to $[-1,1]$ and then use the theory given previously. The difference in approximating functions with the same number of terms between the two intervals can by seen by the following example.

Suppose we want to approximate $x^3 + x^2$ by a polynomial of degree 2. The idea is to express $x^3 + x^2$ in the form $kT_3(x) + q_2(x)$ where $q_2(x)$ is a polynomial of degree 2 and $x^3 + x^2 - q_2(x)$ has the smallest maximum

deviation from zero. Since $T_3(x) = 4x^3 - 3x$, we have

$$x^3 + x^2 = \frac{1}{4}T_3(x) + \frac{3}{4}x + x^2,$$

which implies that

$$q_2(x) = x^2 + \frac{3}{4}x$$

is the best second order polynomial approximation to $x^3 + x^2$ in the range $(-1, 1)$. Using the fact $|T_k(\pm 1)| \leq 1$ implies that the magnitude of the maximum error of $q_2(x)$ is $\frac{1}{4}$. Now suppose we wish to find the best second order approximation of $x^3 + x^2$ on $(0, 1)$. Following the same argument above

$$x^3 + x^2 = kT_3^*(x) + p_2(x)$$

where $T_3^*(x) = 32x^3 - 48x^2 + 18x - 1$. Then,

$$\begin{aligned} x^3 + x^2 &= \frac{1}{32}\left(32x^3 - 48x^2 + 18x - 1\right) + p_2(x) \\ &= x^3 - \frac{3}{2}x^2 + \frac{9}{16}x - \frac{1}{32} + p_2(x) \end{aligned}$$

which implies that

$$p_2(x) = \frac{5}{2}x^2 - \frac{9}{16}x + \frac{1}{32}$$

is the best second order polynomial approximation to $x^3 + x^2$ in the range $(0, 1)$ and the magnitude of the maximum error is $\frac{1}{32}$. To achieve an equivalent

amount of accuracy using Chebyshev polynomials defined on $[-1,1]$ would require a 4-th order polynomial. Figures 3 and 4 compare the polynomials $q_2(x)$ and $p_2(x)$.

For non-linear problems, the evaluation $T_s^*(x)T_r^*(x)$ is very important. To evaluate this product we begin with the fact that

$$\begin{aligned} T_s(T_r(x)) &= \cos\left(s \arccos\left(\cos\left(r\theta\right)\right)\right) \\ &= \cos\left(sr\theta\right) = \cos\left(rs\theta\right) \\ &= T_{sr}(x) = T_{rs}(x) \\ &= T_r(T_s(x)). \end{aligned}$$

For the case $s = 2$ we have

$$T_r(T_2(x)) = T_r(2x^2 - 1) = T_2(T_r(x)) = 2T_r^2(x) - 1 = T_{2r}(x)$$

Replacing x^2 by x gives

$$T_r(2x - 1) = T_r^*(x) = 2T_r^2(x^{\frac{1}{2}}) = T_{2r}(x^{\frac{1}{2}}),$$

and using the above results we obtain

$$\begin{aligned} T_s^*(x)T_r^*(x) &= T_{2s}(x^{\frac{1}{2}})T_{2r}(x^{\frac{1}{2}}) \\ &= \frac{1}{2}\left(T_{2s+2r}(x^{\frac{1}{2}}) + T_{2s-2r}(x^{\frac{1}{2}})\right) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\left(T_{2(s+r)}(x^{\frac{1}{2}}) + T_{2(s-r)}(x^{\frac{1}{2}})\right) \\
&= \frac{1}{2}\left(T^*_{s+r}(x) + T^*_{s-r}(x)\right).
\end{aligned}
$$

To evaluate derivatives of modified Chebyshev polynomials in terms of others we begin with

$$
\begin{aligned}
\int T^*_k(x)dx &= \int T_k(2x-1)dx \\
&= \int \cos k \arccos(2x-1)dx \\
&= -\frac{1}{2}\int \cos k\theta \sin\theta d\theta, \quad \cos\theta = 2x-1 \\
&= -\frac{1}{4}\int (\sin(k+1)\theta - \sin(k-1)\theta)d\theta \\
&= \frac{1}{4}\left(\frac{\cos(k+1)\theta}{k+1} - \frac{\cos(k-1)\theta}{k-1}\right) \\
&= \frac{1}{4}\left(\frac{T^*_{k+1}(x)}{k+1} - \frac{T^*_{k-1}(x)}{k-1}\right).
\end{aligned}
$$

In an analogous procedure to that used before, the recurrence relation that links coefficients of Chebyshev polynomial to those of its derivatives is given by

$$
\begin{aligned}
\hat{u}^{(1)}_{N+1}(t) &= 0 \\
\hat{u}^{(1)}_{N}(t) &= 0 \\
c_k\hat{u}^{(1)}_{k}(t) &= \hat{u}^{(1)}_{k+2}(t) + 4(k+1)\hat{u}_{k+1}(t), \quad k = N-1,\ldots,1,0. \quad (1.48)
\end{aligned}
$$

The coefficients of higher derivatives are

$$\begin{aligned}
\hat{u}_{N+1}^{(p)}(t) &= 0 \\
\hat{u}_{N}^{(p)}(t) &= 0 \\
c_k \hat{u}_k^{(p)}(t) &= \hat{u}_{k+2}^{(p)}(t) + 4(k+1)\hat{u}_{k+1}^{(p-1)}(t), \; k = N-1, \ldots, 1, 0 \quad (1.49)
\end{aligned}$$

for $p = 2, 3, \ldots$.

The Chebyshev transform and inverse transform are given by the same formulas as before, with Chebyshev polynomials replaced by the modified Chebyshev polynomials

$$\hat{u}_k(t) = \frac{2}{N c_k} \sum_{j=0}^{N} \frac{1}{c_j} u(x_j, t) T_k^*(x_j) \tag{1.50}$$

and the inverse given by

$$u_k(x_j, t) = \sum_{j=0}^{N} \hat{u}_k(t) T_k^*(x_j), \quad k = 0, 1, \ldots N. \tag{1.51}$$

A popular choice for the grid points x_j is

$$x_j = \frac{1}{2}\left(1 + \cos\frac{\pi j}{N}\right), \quad j = 0, 1, \ldots, N. \tag{1.52}$$

The modified Chebyshev polynomials approximation, on $[0, 1]$, for spatial derivatives of order p is given by

$$\frac{\partial^p u_k(x_j, t)}{\partial x^p} = \sum_{k=0}^{N} \hat{u}_k^{(p)}(t) T_k^*(x_j), \quad p \geq 1 \tag{1.53}$$

Again, we see the familiar pattern inherent to pseudospectral methods. That is, interpolate $u(x,t)$ on a set of grid points x_j using modified Chebyshev polynomials to transform the problem to Chebyshev space. Then perform the required differentiation in that space, and finally interpolate again to bring the problem back to physical space. A FORTRAN subroutine of this process for modified Chebyshev polynomials is provided in Appendix C.

For problems that have a certain degree of smoothness, Chebyshev and modified Chebyshev polynomials both guarantee spectral accuracy. That is, the k-th coefficients of both polynomials decay at a rate faster than any inverse power of k. The rapid convergence of Chebyshev collocation depends only on the smoothness of the problem, where the convergence of the Fourier series depends on the values of the function and its derivatives at the boundaries, as well as the smoothness of the function (Gottlieb and Orszag 1977). It is this rapid decay of the coefficients that makes Chebyshev collocation a good choice for non-periodic problems. Special care should be used before applying Chebyshev collocation to non-periodic problems. Many non-periodic problems with appropriate boundary conditions can be made into an even or odd extension of the initial conditions, allowing one to use Fourier collocation on a non-periodic problem. Details of this process are given in Chapter 3 on

hyperbolic heat transfer.

1.4 Boundary Conditions

We saw earlier that the interpolant in Fourier collocation automatically satisfies the boundary conditions of the differential equation and does not require any special treatment of the boundary conditions. However, this in not the case with Chebyshev collocation. Chebyshev collocation requires the user to decide on two different approaches to evaluate the boundary conditions of the differential equation.

The first approach is to leave the interpolant in the form given by equations (1.38) or (1.51). Using either of these interpolating polynomials at any given time level we must first evaluate any boundary conditions given by the differential equation before interpolating across the computational domain. Notice for explicit time-differencing schemes that have multiple stages, for example p-th order Runge-Kutta, interpolation is done p-times per time step. Hence, boundary conditions will have to be evaluated at each internal stage prior to interpolation. This treatment of boundary conditions can be used in most problems and certainly would be the method of choice for complicated

non-linear boundary conditions.

The other method for handling boundary conditions redefines the trial function $\phi_k(x_j)$ so that the boundary conditions are automatically satisfied during interpolation. For example, suppose we wish to solve the following partial differential equation

$$u_t + u_x = f(x,t), \qquad -1 \leq x \leq 1,\ t > 0$$

$$u(-1,t) = 0, \qquad u(x,0) = g(x).$$

Assuming the problem is non-periodic, and the domain of interest is $[-1,1]$, the appropriate choice for the trial functions would be the Chebyshev polynomials $\phi_k(x) = T_k(x)$. The term we wish to approximate using Chebyshev collocation is u_x on some set of grid points x_j:

$$u_x = \frac{\partial u(x_j,t)}{\partial x} \approx \sum_{k=0}^{N} \hat{u}_k^{(1)}(t) T_k(x_j), \quad j = 0, 1, \ldots N.$$

In evaluating this expression we could avoid evaluating the boundary condition $u(-1,t) = 0$ prior to interpolation by redefining the trial function to be

$$\phi_k(x) = T_k(x) - (-1)^k T_0(x).$$

Now notice that the trial functions individually satisfy the boundary condi-

tions

$$\phi_k(-1) = T_k(-1) - (-1)^k T_0(-1) = (-1)^k - (-1)^k = 0.$$

Hence, we can interpolate at any given time level without having to evaluate the boundary condition.

Obviously, the above procedure is only applicable to problems with simple boundary conditions, for example, Dirichlet or Neumann. This type of procedure is very useful in boundary value problems, examples of which can be found in Karageorghis(1988) and (Karageorghis and Phillips 1989). Problems with complicated boundary conditions would have to be solved using the first procedure.

There are several ways to evaluate boundary conditions using Chebyshev collocation on problems defined on semi-infinite or infinite domains. One method would be to truncate the semi-infinite or infinite interval to a finite one and then apply the previous results. An alternative is to map the semi-infinite or infinite interval to a finite one by algebraic or exponential mappings Canuto et al.(1988) and Boyd(1982, 1987, 1989).

Before closing this chapter, it should be mentioned that other spectral methods exist, as well as other trial functions for the methods already dis-

cussed. To date, the most comprehensive collection of different methods and various trial functions can be found in Canuto et al.(1988), (Gottlieb and Orszag 1977), Boyd(1989), and Fornberg(1996).

Chapter 2

Evolutionary Problems

2.1 Introduction

To demonstrate the effectiveness of pseudospectral techniques in solving partial differential equations, the following problems have been selected: one dimensional wave equation with non-constant coefficients, linear and non-linear forms of the heat equation, third order KdV equation, and Burgers equation. These problems will serve two purposes. First, to demonstrate the superior accuracy of pseudospectral methods over finite element and finite difference methods and to show how the theory presented in Chapter 1 is efficiently implemented. Burgers equation will also demonstrate some of the

difficulties of using pseudospectral methods on problems with non-smooth data.

2.2 Linear Problems

In the first example we apply Chebyshev collocation to the hyperbolic problem $u_t + \frac{x+1}{t+1} u_x = 0$. To advance the solution in time we will use 4-th order Runge-Kutta time differencing. The problem, with initial and boundary conditions, is

$$u_t + \frac{x+1}{t+1} u_x = 0, \qquad -1 \leq x \leq 1, \quad t > 0 \tag{2.1}$$

$$u(x,0) = \sin(\pi(x+1)), \qquad u(-1,t) = 0. \tag{2.2}$$

The exact solution is given by

$$u(x,t) = \sin\left(\frac{\pi(x+1)}{t+1}\right). \tag{2.3}$$

To discretize in space we begin by choosing a set of $N+1$ grid points

$$x_j = \cos\frac{\pi j}{N}, \qquad j = 0, 1, \ldots N. \tag{2.4}$$

We represent the current solution $u(x_j, t^n)$, $n = 0, 1, \ldots$ where $n = 0$ corresponds to the initial data as

$$u(x_j, t^n) = \sum_{k=0}^{N} \hat{u}_k(t^n) \cos \frac{\pi k j}{N}. \tag{2.5}$$

Since we are interested in using Chebyshev collocation to approximate the term $\frac{\partial u}{\partial x}$ the first step would be to invert the above result to obtain the discrete Chebyshev coefficients. For N large this step would use the FFT, but we will consider N small enough not to need the FFT. Under this assumption, the inversion of (2.5) is

$$\hat{u}_k(t^n) = \frac{2}{N c_k} \sum_{j=0}^{N} \frac{1}{c_j} u(x_j, t^n) \cos \frac{\pi k j}{N}, \qquad k = 0, 1, \ldots N. \tag{2.6}$$

Equation (2.6) interpolates across the entire computational domain and this requires the boundary conditions to be evaluated prior to interpolating. Alternatively, we could redefine the trial functions to automatically satisfy the boundary conditions. From the recurrence relations given in Chapter 1 for Chebyshev polynomials on $[-1, 1]$ we can determine the coefficients $\hat{u}_k^{(1)}(t^n)$. The approximation for the first derviative of u is

$$\frac{\partial u}{\partial x} \approx \frac{\partial u(x_j, t^n)}{\partial x} = \sum_{k=0}^{N} \hat{u}_k^{(1)}(t^n) \cos \frac{\pi k j}{N}. \tag{2.7}$$

Next, we evaluate

$$\left(\frac{x_j+1}{t^n+1}\right)\frac{\partial u(x_j,t^n)}{\partial x} \tag{2.8}$$

at each of the grid points and at the appropriate time level. Notice, that by evaluating the right hand side of equation (2.1), we have reduced the partial differential equation to a system of ordinary differential equations, and are now free to select a time integration subroutine. In this case we are using 4-th order Runge-Kutta to advance the solution from time level $t^n \to t^{n+1}$, $n = 0, 1, \ldots$. Table 2.1 gives the magnitude of the maximum error

N	Chebyshev Collocation	Finite Difference
6	1.02 (-2)	.15
8	4.29 (-4)	.09
12	1.88 (-7)	.08
16	2.09 (-11)	.07
32	3.95 (-12)	.06

Table 2.1

(L_∞ error) defined by

$$max_j\|u(x,t) - u(x_j,t^n)\| \tag{2.9}$$

which represents the maximum error between the numerical and exact solution $u(x,t)$ evaluated at the grid points $x = x_j$ and time level $t = t^n$. Results from a finite difference method are also given in Table 2.1 and there is a clear superiority in the accuracy of the Chebyshev collocation.

Our next example is the classical heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}. \tag{2.10}$$

Again we use 4-th order Runge-Kutta in time and Chebyshev collocation in space. The mixed initial boundary value problem is

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \qquad -1 \leq x \leq 1, \quad t > 0 \tag{2.11}$$

$$u(x,0) = \sin \pi x \tag{2.12}$$

with homogeneous Dirichlet boundary conditions

$$u(-1,t) = 0, \qquad u(1,t) = 0. \tag{2.13}$$

The exact solution is

$$u(x,t) = e^{-\pi^2 t} \sin \pi x. \tag{2.14}$$

Once again we choose the grid points to be

$$x_j = \cos \frac{\pi j}{N}, \qquad j = 0, 1, \ldots N \tag{2.15}$$

and the trial function to be Chebyshev polynomials. In this example we are interested in using Chebyshev collocation to approximate the second derivative term. This approximation at a given time level is

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{\partial^2 u(x_j, t^n)}{\partial x^2} = \sum_{k=0}^{N} \hat{u}_k^{(2)}(t^n) \cos \frac{k\pi j}{N}. \tag{2.16}$$

Once the coefficient $\hat{u}_k^{(2)}(t^n)$ has been determined, (2.16) is used to advance the solution from $t^n \to t^{n+1}$, $n = 0, 1, \ldots$. Table 2.2 compares the maximum errors obtained with increasing N and also presents the results obtained from second order finite differences. The superior accuracy of the spectral method over the finite difference method is once again shown dramatically. Doubling the number of grid points from 8 to 16 results in the finite difference error being divided roughly by a factor of 4 as would be expected for second order accuracy. The spectral method error, on the other hand, is divided by a factor on the order of 10^7 when N is increased form 8 to 16.

N	Chebyshev Collocation	Finite Difference
8	4.67 (-5)	.064
10	8.94 (-7)	.035
12	1.22 (-8)	.025
14	1.30 (-10)	.017
16	1.78 (-12)	.013

Table 2.2

Before proceeding to the next example, it is worth mentioning that if the spatial derivatives in the previous two problems had been approximated using the matrix D_{ψ_j} mentioned in Chapter 2, the maximum error is virtually identical to the errors given in Tables 2.1 and 2.2. Again, this is due to the small size of N used in the previous examples.

Problems that have higher and multiple derivatives do not produce any additional complications. For example, a linearized version of the KdV equation is

$$u_t = u_x + u_{xxx}, \qquad 0 \leq x \leq 1, \quad t > 0 \tag{2.17}$$

with initial conditions

$$u(x,0) = \cos 2\pi x. \tag{2.18}$$

The exact solution to this problem is given by

$$u(x,t) = \cos 2\pi(x - (4\pi^2 - 1)t). \tag{2.19}$$

To solve this problem one could use the modified Chebyshev polynomials to approximate the first and third derivatives of u, but this approach does not make use of the fact that the problem is periodic. Hence, a better approach would be to use Fourier collocation to approximate the derivative terms. The first step in using Fourier collocation would be to form the matrix F_N given in Chapter 1. Again, we are assuming that N is small enough not to require the use of the FFT. Notice that F_N can be found once and used at each time level. The next step would be to calculate the diagonal matrices Λ and Λ^3. The approximations for the first and third derivatives are

$$\frac{\partial u}{\partial x} \approx D(\underline{u}(x_j, t^n)) = \frac{1}{N}\bar{F}_N P_N \Lambda P_N F_N \underline{u}(x_j, t^n) \tag{2.20}$$

and

$$\frac{\partial^3 u}{\partial x^3} \approx D^3(\underline{u}(x_j, t^n)) = \frac{1}{N}\bar{F}_N P_N \Lambda^3 P_N F_N \underline{u}(x_j, t^n) \tag{2.21}$$

where the bar under u is used to remind us that we are really treating u as a vector of discrete data values at some time level t^n. Once these approximations have been evaluated, the solution can be advanced in time. This version

of the KdV equation shows a wave profile given by the initial condition propagating to the right as time increases. Figure 5 compares the profiles of the exact and numerical solutions at time $t = .52$. Forth order Runge-Kutta was used to advance the solution in time. This figure clearly shows the lack of any appreciable dissipative or dispersive errors that are so common in other methods. Before we proceed, the fact that the entries of F_N, $\Lambda^{(p)}$, and $\bar{F}_N$ are complex requires extra care in any computer program due to the sensitivity individual compilers might have in performing complex manipulations. To eliminate any problems it is suggested that the problem be split into real and imaginary parts, eliminating the need to perform complex operations.

The problems in this section should demonstrate the ease at which spatial derivatives are approximated using collocation methods. Problems with higher derivatives or variable coefficients do not produce any complications. Spatial derivatives are simply replaced by their pseudospectral approximation and any other terms present in the problem are evaluated at the chosen grid points and time level. Although the choice of Runge-Kutta for time differencing was selected due to its accuracy and ease of implementation, other time differencing schemes such as implicit and predictor-corrector methods can also be used.

The reason that the particular time differencing scheme is not dependent on Chebyshev or Fourier collocation is because the original partial differential equation is discretized in space only using pseudospectral methods. The time derivatives are left continuous. Once the spatial derivatives, and other terms present in the problem, have been evaluated the user is free to choose the time differencing scheme. This freedom to choose any appropriate time stepping routine will be very important in problems with certain stability restrictions. Examples are presented in the next section and in Chapter 3.

2.3 Non-Linear Problems

When considering the use of spectral methods on a non-linear problem it is preferable to choose pseudospectral methods over other spectral methods to solve the problem. The reason for this is due to the way pseudospectral methods approximate the non-linear terms in the differential equation.

Suppose the problem contains the term $\frac{\partial^p u^2}{\partial x^p}$. To approximate this term using pseudospectral methods we would first evaluate u^2 on a chosen set of grid points x_j, $j = 0, 1, \ldots N$. The p-th derivative would then be found following the technique described in section 2.1. For example, suppose Cheby-

shev collocation is used, then

$$\frac{\partial^p u^2}{\partial x^p} \approx \frac{\partial^p (u(x_j, t^n))^2}{\partial x^p} = \sum_{k=0}^{N} \hat{u}_k^{(p)}(t^n) T_k(x_j) \tag{2.22}$$

where $\hat{u}_k^{(p)}(t^n)$ is found using the recurrence relation given by (1.36). The discrete Chebyshev coefficients are

$$\hat{u}_k(t^n) = \frac{2}{N c_k} \sum_{j=0}^{N} \frac{1}{c_j} (u(x_j, t^n))^2 T_k(x_j). \tag{2.23}$$

Had the problem been periodic, Fourier collocation would give

$$\frac{\partial^p u^2}{\partial x^p} \approx D^p (\underline{u}(x_j, t^n))^2 = \frac{1}{N} P_N F_N \Lambda^p P_N F_N (\underline{u}(x_j, t^n))^2. \tag{2.24}$$

The bar under u reminds us that u is used as a vector of grid point values and $(\underline{u}(x_j, t^n))^2$ is found by squaring each of the components of u.

Pseudospectral methods treat combinations of the dependent variable and its derivatives just as they appear in the differential equation. For example, the term $u\frac{\partial u}{\partial x}$ is approximated by first evaluating $\frac{\partial u}{\partial x}$ a collocation methods described earlier, and u is evaluated at the chosen grid points. These terms are then multiplied together to produce the required approximation. Chebyshev collocation gives

$$u\frac{\partial u}{\partial x} \approx u(x_j, t^n) \frac{\partial u(x_j, t^n)}{\partial x} = u(x_j, t^n) \sum_{k=0}^{N} \hat{u}_k^{(1)}(t^n) T_k(x_j) \tag{2.25}$$

and Fourier collocation gives

$$u\frac{\partial u}{\partial x} \approx u(x_j, t^n)\frac{\partial u(x_j, t^n)}{\partial x} = \underline{u}(x_j, t^n)D\underline{u}(x_j, t^n) \tag{2.26}$$

where D is the pseudospectral differentiation matrix.

Suppose we wish to solve a non-linear version of the heat equation

$$\frac{\partial u}{\partial t} = 4\frac{\partial^2 u^2}{\partial x^2}, \qquad -1 \leq x \leq 1, \quad t > 0. \tag{2.27}$$

If we assume $u = u(x - ct)$, and substitute this into the PDE integrating with respect to $x - ct$ we obtain a particular solution given by

$$2u - 3 + \log(u - .5) - 2(2t - \frac{x+1}{2}). \tag{2.28}$$

The solution u can not be solved explicitly, and both the initial and boundary conditions must be determined by iteration.

Since the problem does not possess any periodicity, Chebyshev collocation on $[-1, 1]$ will be used to find a numerical solution. The term we wish to approximate using Chebyshev collocation is $\frac{\partial^2 u^2}{\partial x^2}$ and is given by

$$\frac{\partial^2 u^2}{\partial x^2} \approx \frac{\partial^2 (u(x_j, t^n))^2}{\partial x^2} = \sum_{k=0}^{N} \hat{u}_k^{(2)}(t^n)T_k(x_j) \tag{2.29}$$

where $\hat{u}_k^{(2)}(t^n)$ is found by the recurrence relation (1.36) and

$$\hat{u}_k(t^n) = \frac{2}{Nc_k}\sum_{j=0}^{N}\frac{1}{c_j}(u(x_j, t^n))^2 T_k(x_j). \tag{2.30}$$

The grid points x_j are the standard Gauss-Lobatto points. Newton iteration is used to determine the initial conditions $u(x,0)$ and the boundary conditions $u(-1,t)$ and $u(1,t)$. These boundary conditions must be evaluated at each time level prior to any interpolation.

The results given in Table 2.3 are for 4-th order Runge-Kutta and $N = 10$. The table compares the numerical solution and exact solution at each of the grid points x_j, in $(-1,1)$.

x	Chebyshev Collocation	Exact Solution
-.9510	2.2078968945112	2.2078968944878
-.8090	2.1531632195072	2.1531632195120
-.5877	2.0687523236070	2.0687523236061
-.3090	1.9639420734623	1.9639420734627
0.0	1.8499618380357	1.8499618380355
.3090	1.7385292210325	1.7385292210329
.5877	1.6404123987258	1.6404123987251
.8090	1.5643223500175	1.5643223500212
.9510	1.5163586147859	1.5163586146713

Table 2.3

The maximum error for the results above is 2.33 x 10^{-11}. To achieve these results, a time step of .0001 was needed because of stability restrictions. The Crank-Nicolson scheme with iteration can produce results accurate to 5 decimal places with a time step of .005 and grid spacing of .1. If we try to increase the time step using Chebyshev collocation, even to .0005, overflow occurs. If the number of grid points is increased the stability requirements become even more severe. Can we conclude that, unless extremely accurate results are required, finite difference methods would be the method of choice for this problem?

Gottlieb and Orszag(1977) point out that second derivatives being approximated by Chebyshev collocation using explicit time stepping can have time step restrictions $\Delta t \leq \frac{1}{N^4}$ as $N \to \infty$ due to the high resolution near the boundaries. In fact, when one tries to push the stability requirements, errors first start to increase near the boundaries. Since the methods are global, these errors quickly propagate throughout the computational domain and affect the output in just a few time steps. To relax these prohibitive time step restrictions, implicit time differencing is used. The trapezoidal rule allowed the time step restriction to be relaxed, but the results for $N = 10$ still were not competitive with the Crank-Nicholson scheme described earlier. Various

predictor-corrector methods also have limited success with this problem and in many cases can not improve on the stability requirements found in the 4-th order Runge-Kutta method. To achieve better accuracy than finite difference methods and with a comparable amount of computational cost, the variable time stepping Runge-Kutta-Fehlberg method was used. For $N = 10$ this method still requires 1600 steps to advance the solution to $t = .5$ with a maximum error of 9.54 x 10^{-10}. By decreasing N to 6, only 128 steps are needed to achieve an L_∞ error of 3.83 x 10^{-7}. These results are obtained without the need to iterate the solution at each time level which is necessary in the Crank-Nicolson method. Hence, by choosing an appropriate time stepping routine it is possible to relax the stability restrictions of Chebyshev collocation and still produce accurate results efficiently.

The next non-linear problem we wish to investigate is Burgers equation given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \qquad -1 \leq x \leq 1, \quad t > 0 \tag{2.31}$$

with initial and boundary conditions

$$u(x,0) = -\sin \pi x, \qquad u(-1,t) = u(1,t) = 0. \tag{2.32}$$

The coefficient ν is a positive constant. Burgers equation combines non-

linear propagation and diffusion. With $\nu = \frac{10^{-2}}{\pi}$ the exact solution develops a steep gradient symmetric about the origin. We have already seen how rapidly pseudospectral methods converge for smooth data, but this problem has a region of rapid change over a very small spatial region. The semi-discrete version of Burgers equation using Chebyshev polynomials is

$$\frac{\partial u}{\partial t} = -u(x_j, t^n) \sum_{k=0}^{N} \hat{u}_k^{(1)}(t^n) T_k(x_j) + \nu \sum_{k=0}^{N} \hat{u}_k^{(2)}(t^n) T_k(x_j) \tag{2.33}$$

with Gauss-Lobatto points

$$x_j = \cos \frac{\pi j}{N}, \qquad j = 0, 1, \ldots, N. \tag{2.34}$$

Figures 6-8 for $N = 32, 64, 128$ show the solution using 4th order Runge-Kutta at time $t = 0, .3, .5$. These figures show the development of numerical oscillation near the origin due to the steep gradient which has developed. To resolve this region, and reduce the numerical oscillation, large numbers of Chebyshev polynomials are needed in the approximation. The position of this steep gradient is centered in a region where there is the least amount of spatial resolution. Recall, standard grid point spacing for Chebyshev collocation decreases quadratically towards the boundaries (± 1). The lack of grid points near the origin for small N is not sufficient to produce a sharp profile in

the numerical solution. A large value of N is needed to ensure sufficient resolution in the vicinity of the sharp gradient.

A naive solution to this dilemma would be to simply add more points near the origin. The problem with this approach is, unlike finite difference schemes, grid point spacing for pseudospectral methods can not be chosen arbitrarily. In the next section coordinate transformations are introduced which can eliminate the resolution problem near the origin.

2.4 Coordinate Transformations

As we saw in the previous section, Burgers equation required a large number of Chebyshev polynomials to accurately resolve the area near the origin. It was determined that for smaller numbers of polynomials, the spacing of Gauss-Lobatto points prevented accurate resolution near the origin.

In an attempt to decrease the number of Chebyshev polynomials and keep the resolution near the origin sharp we introduce the following coordinate transformation, Peyret (1986),

$$x = g(\xi) \tag{2.35}$$

where

$$g(\xi) = (1-\alpha)\xi^3 + \alpha\xi. \tag{2.36}$$

The effect of this odd function is to transform the Gauss-Lobatto points to a grid which has an increased number of points near the origin and less resolution towards the boundaries. Notice $g(\pm 1) = \pm 1$ and α is a parameter to be determined. The effect α has is to increase the resolution near the origin as $\alpha \to 0$ and map the Gauss-Lobatto points to Gauss-Lobatto points as $\alpha \to 1$.

Recall Burgers equation is

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad -1 < x < 1,\ t > 0 \tag{2.37}$$

with initial conditions

$$u(x,0) = -\sin \pi x \tag{2.38}$$

and boundary conditions

$$u(-1,t) = 0, \quad u(1,t) = 0. \tag{2.39}$$

Using the odd function defined above and the chain rule gives us the required derivatives in our transform space i.e.,

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial x} = \frac{1}{g'}\frac{\partial u}{\partial \xi} \tag{2.40}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x}\frac{\partial u}{\partial x} = \frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial x}\left(\frac{1}{g'}\frac{\partial u}{\partial \xi}\right) = \frac{1}{(g')}\frac{\partial}{\partial \xi}\left(\frac{1}{(g')}\frac{\partial u}{\partial \xi}\right) \tag{2.41}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{(g')}\frac{\partial u}{\partial \xi}\frac{\partial}{\partial \xi}\left(\frac{1}{(g')}\right) + \frac{1}{(g')^2}\frac{\partial^2 u}{\partial \xi^2} \tag{2.42}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{(g')}\frac{\partial u}{\partial \xi}\frac{(-g'')}{(g')^2} + \frac{1}{(g')^2}\frac{\partial^2 u}{\partial \xi^2} \tag{2.43}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{(g')^2}\left[\frac{\partial^2 u}{\partial \xi^2} - \frac{g''}{g'}\frac{\partial u}{\partial \xi}\right]. \tag{2.44}$$

Hence, Burgers equation in transformed space is

$$\frac{\partial u}{\partial t} = -\frac{u}{g'}\frac{\partial u}{\partial \xi} + \frac{\nu}{(g')^2}\left[\frac{\partial^2 u}{\partial \xi^2} - \frac{g''}{g'}\frac{\partial u}{\partial \xi}\right], \ -1 < \xi < 1, \ t > 0 \tag{2.45}$$

with initial condition

$$u(\xi, 0) = -\sin\left(\pi(g(\xi))\right) \tag{2.46}$$

and boundary conditions

$$u(-1, t) = 0, \quad u(1, t) = 0. \tag{2.47}$$

For this problem $\nu = \frac{10^{-2}}{\pi}$ and $\alpha = \frac{2}{5}$. Chebyshev collocation was used to approximate the spatial derivatives and 4-th order Runge-Kutta was used to advance the solution. Figures 9-11 show the results for $N = 32, 64, 128$. The parameter α was chosen such that one value could be used over a wide variety of N values and time. For even greater accuracy α could be determined at

each time step so the error under some suitable norm is minimized. Experiments for smaller values of α were also conducted resulting in poorer approximations despite the fact that the resolution near the origin is increased for small values of α. The reason for this is, as α is decreased, resolution away from the origin is also decreased. Since pseudospectral methods are global in character this decrease in resolution could affect the overall accuracy of the method.

Notice that coordinate transformation could be used in problems where sharp gradients propagate over time. In such problems we would have to locate the region of the sharp gradient and then redefine the mapping $g(\xi)$ accordingly.

Before closing this section it should be mentioned that other transformations are possible. A good source for other techniques can be found in Bayliss and Turkel (1992) and Boyd (1992). Domain decomposition which breaks up the computational domain into a finite number of intervals and solves each interval individually is suitable in this problem and many other problems. Details can be found in Canuto et al.(1988).

The non-linear heat equation and Burgers equation demonstrated how pseudospectral methods are used in non-linear partial differential equations.

The straightforward way pseudospectral methods treat non-linear terms makes it clear why these methods are usually chosen first over other spectral methods. Unfortunately, many non-linear problems have complications other than just the approximation of spatial derivatives and non-linear terms. Time-step restrictions and non-smooth data are just some of the complications we face in solving non-linear evolutionary problems. These complications are not just restricted to non-linear problems. Chapter 3 will demonstrate that non-smooth data are also a problem for linear evolutionary problems.

Chapter 3

Hyperbolic Heat Transfer

3.1 Introduction

Despite the frequent use of the heat equation in examples and problems, what is usually not discussed is the invalidity of the heat equation under certain circumstances. The heat equation is not valid for applications where the temperature is near zero or at moderate temperatures when the elapsed time during a transient is extremely small Luikov (1976). Under the latter situation the wave nature of thermal transport becomes dominant. Furthermore, this thermal wave travels through the medium as a steep wave front with a finite speed of propagation. This behavior is in sharp contrast to the

classical parabolic heat equation which predicts infinite speed of propagation and a non-zero temperature gradient throughout the medium.

To model hyperbolic heat transfer we first begin with a modification of Fourier's law, $q = -k\nabla T$, given by

$$\tau \frac{\partial q}{\partial t} + q = -k\nabla T \tag{3.1}$$

where q is the flux and T is the temperature. The constants k and τ are the thermal conductivity and relaxation time. The relaxation time is defined as $\tau = \frac{\alpha}{c^2}$ where α is a thermal diffusive term, and c is the speed of propagation of the thermal wave. The relaxation time τ implies there is a finite build up time before the onset of a thermal response due to an imposed temperature gradient. That is, heat flow does not start instantaneously but grows gradually with a relaxation time of τ. Similarly, heat flow does not cease immediately, but dies out gradually after a temperature gradient is removed.

The one dimensional energy equation is given by

$$-\frac{\partial q(x,t)}{\partial x} = \rho c_v \frac{\partial T(x,t)}{\partial t} \tag{3.2}$$

where ρ is the density and c_v is the specific heat. The following non-dimensional quantities

$$t^* = \frac{c^2 t}{2\alpha}, \qquad x^* = \frac{cx}{2\alpha} \tag{3.3}$$

$$T^*(x^*, t^*) = \frac{T(x,t) - T_0}{\alpha f_r / kc}, \qquad q^*(x^*, t^*) = \frac{q(x,t)}{f_r} \tag{3.4}$$

where T_0 is taken as zero and f_r is a reference heat flux are introduced in equations (3.1) and (3.2) which yield the following non-dimensional hyperbolic system

$$\frac{\partial T}{\partial t} = -\frac{\partial q}{\partial x} \tag{3.5}$$

$$\frac{\partial q}{\partial t} = -2q - \frac{\partial T}{\partial x} \tag{3.6}$$

where starred quantities have been dropped for notational ease. Alternatively, we could substitute (3.1) into (3.2) and again use the non-dimensional variables defined above to yield the non-dimensional hyperbolic heat equation given by

$$\frac{\partial^2 T}{\partial t^2} + 2\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}. \tag{3.7}$$

This equation is also referred to in literature as the telegraph equation (Zauderer 1983) or equation of non-Fourier heat conduction (Glass et al. 1987).

Pseudospectral methods will be used to solve both the hyperbolic system and the hyperbolic heat equation. Both models predict hyperbolic behavior, and the particular choice is usually determined by the boundary conditions. This chapter will demonstrate that pseudospectral methods give superior

results over previous research which used finite difference and finite element methods.

Carey and Tsai (1982) were one of the first to investigate hyperbolic heat transfer numerically. They considered the hyperbolic heat equation with Dirichlet boundary conditions. To approximate spatial derivatives they used a finite element method and tried, commercially available, time integration packages. All of their results were plagued by numerical oscillation especially at the wave front, and their attempts to reduce the oscillation smeared the profile of the wave front. Glass et al.(1985, 1987) also considered this same problem using MacCormack predictor-corrector scheme. In an attempt to reduce the numerical oscillation they considered 1000 grid points on the interval [0,1]. Even with such a fine mesh, numerical oscillation was still present at the wave front.

Glass, Ozisik, and Vick (1985) considered the effects of radiation on hyperbolic heat transfer and compared the results to parabolic behavior. Again, 1000 grid points had to be used to reduce oscillation at the wave front. In addition, very small timesteps also had to be used to prevent instabilities.

This chapter will consider the same problems mentioned above and demonstrate that pseudospectral methods, when used correctly, can produce supe-

rior results without numerical oscillation at the wave front and can do so with coarse grids and large timesteps. Pseudospectral methods will also be used in this chapter to investigate boundary conditions given in terms of a continuous periodic flux with radiation. This generalizes and extends the work of (Glass, Ozisik, and Vick 1985), who only considered boundary conditions with a constant flux with radiation. The results of this investigation will show pseudospectral methods can produce excellent results without the need to introduce fine grids or prohibitively small timesteps.

3.2 Hyperbolic Heat Equation

Consider the hyperbolic heat equation (3.7) defined on the interval $[0, 1]$ with initial and Dirichlet boundary conditions given by

$$T(x, 0) = 0, \qquad \frac{\partial T(x, 0)}{\partial t} = 0 \tag{3.8}$$

$$T(0, t) = 1, \qquad T(1, t) = 0. \tag{3.9}$$

Under these conditions the temperature at the boundary instantaneously jumps to 1 and a thermal wave with a sharp jump at the wave front propagates into the medium reflecting off the boundaries before reaching a steady

state. Fortunately, the problem is linear and an exact solution can easily be reached. Introducing the function $v(x) = 1 - x$ and substituting $T(x,t) = v(x) + w(x,t)$ into the hyperbolic heat equation produces the equivalent problem in terms of $w(x,t)$

$$\frac{\partial^2 w}{\partial t^2} + 2\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2} \tag{3.10}$$

with initial conditions

$$w(x,0) = x - 1, \qquad \frac{\partial w(x,0)}{\partial t} = 0 \tag{3.11}$$

and homogeneous Dirichlet boundary conditions

$$w(0,t) = 0, \qquad w(1,t) = 0. \tag{3.12}$$

Using separation of variables gives the following expression

$$w(x,t) = -\frac{2}{\pi} e^{-t} \sum_{n=1}^{\infty} \frac{\sin\sqrt{\lambda}x}{n} \left(\cos\beta t + \frac{\sin\beta t}{\beta} \right) \tag{3.13}$$

where

$$\sqrt{\lambda} = n\pi, \qquad \beta = \sqrt{\lambda - 1}. \tag{3.14}$$

Hence, our temperature distribution is given by

$$T(x,t) = 1 - x - \frac{2}{\pi} e^{-t} \sum_{n=1}^{\infty} \frac{\sin\sqrt{\lambda}x}{n} \left(\cos\beta t + \frac{\sin\beta t}{\beta} \right). \tag{3.15}$$

To solve this problem numerically a common approach, due the non-periodic nature of the problem, is to reduce the problem to a system of first order equations and use Chebyshev collocation to approximate the second derivative term. This approach accurately resolves the wave front, but severe numerical oscillation is present at the left hand boundary due to the sudden jump in temperature required by the boundary condition $T(0,t) = 1$. Alternatively, finite difference methods could be used, but the results are plagued by numerical oscillation, especially in the vicinity of the wave front. These oscillations can by reduced by the use of artificial viscosity, but only at the expense of smearing the profile of the wave front.

Pseudospectral methods can be used successfully on this problem. To use them correctly, we follow the same procedure as with the exact solution. By introducing the function $v(x) = 1-x$ and substituting $T(x,t) = v(x)+w(x,t)$ results in a partial differential equation in terms of $w(x,t)$,

$$w_{tt} + 2w_t = w_{xx}, \tag{3.16}$$

with initial conditions given by

$$w(x,0) = x - 1, \qquad w_t(x,0) = 0 \tag{3.17}$$

and homogeneous Dirichlet boundary conditions

$$w(0,t) = 0, \qquad w(1,t) = 0. \tag{3.18}$$

Notice these initial conditions have absorbed the sudden change at the left hand boundary. The initial conditions are now oddly extended on the interval $[-1,1]$. This choice of an odd extension is due to the Dirichlet boundary conditions. Had the boundary conditions been Neumann boundary conditions the appropriate extension would be an even extension of the initial conditions. This results in a problem with artificially imposed periodic boundary conditions. The pseudospectral approach now uses Fourier collocation to approximate the spatial derivative w_{xx}. Letting $u(x,t) = w_t(x,t)$ gives the first order system

$$w_t = u \tag{3.19}$$

$$u_t = -2u + w_{xx} \tag{3.20}$$

with initial conditions

$$w(x,0) = \begin{cases} x+1 & -1 \leq x < 0 \\ 0 & x = 0 \\ x-1 & 0 < x \leq 1 \end{cases}$$

$$u(x,0) = 0, \qquad -1 \leq x \leq 1$$

and boundary conditions

$$w(-1,t) = 0, \qquad w(1,t) = 0. \tag{3.21}$$

The first step is to evaluate the right hand side of the system. Since initially $u(x,0) = 0$ the only term we need to evaluate is w_{xx}. This approximation is given by

$$\frac{\partial^2 w(x_j,0)}{\partial x^2} \approx D^2 \underline{w}(x_j,0) = \frac{1}{N} \bar{F}_N P_N \Lambda^2 P_N F_N \underline{w}(x_j,0) \tag{3.22}$$

where $\underline{w}(x_j,0)$ is the odd extension of the initial conditions. Once w_{xx} has been replaced by its pseudospectral approximation, the solution can be advanced in time by an appropriate time integration scheme, which we choose to be 4th order Runge-Kutta. After the dependent quantities u and w have been updated, we repeat the same process of evaluating the right hand side of the system at the new time level and again step in time. This process is repeated until the desired time level is reached. The final temperature distribution at time $t = t^n$ is given by

$$T(x_j,t^n) = v(x_j) + w(x_j,t^n). \tag{3.23}$$

Figures 12-14 compare the exact and numerical solutions of the temperature profile on the interval $[0, 1]$ at times $t = .5, t = .75, t = 1.0$ with $N = 16$ and a timestep of .05. It is clear that even with such a coarse grid this technique of producing homogeneous boundary conditions and artificially imposing periodicity into the new problem produces extremely accurate results, eliminating any numerical oscillation without smoothing the wave front. Furthmore, overshoot at the wave front (the Gibbs phenomenon) is not present in these results despite using a truncated Fourier series as a trial function. The reason Gibbs phenomenon is not present is because Fourier collocation is used only on the transient part of the solution ($w(x, t)$) which does not have any sharp jumps in its solution.

This numerical solution also explains why we chose the hyperbolic heat equation over the hyperbolic system. Had we tried to solve the hyperbolic system, the sudden jump in temperature at the left hand boundary would result in a severe oscillation. This oscillation is caused by trying to approximate the derivative $\frac{\partial T}{\partial x}$ when T is a piecewise function with a discontinuous jump at $x = 0$. The sudden jump in temperature at the boundary also causes a jump in the flux q at the boundary. Hence, any approximation of $\frac{\partial q}{\partial x}$ is plagued by oscillation, further degrading any attempt to solve the system

numerically.

In some problems, the boundary conditions are given in terms of the flux. Since we are interested in boundary conditions with a periodic flux and radiation a closer look at the hyperbolic system with boundary conditions given in terms of the flux is in order. For example, suppose we have the following hyperbolic system defined on the semi-infinite interval $[0, \infty)$ by

$$\frac{\partial T}{\partial t} = -\frac{\partial q}{\partial x} \tag{3.24}$$

$$\frac{\partial q}{\partial t} = -2q - \frac{\partial T}{\partial x} \tag{3.25}$$

with initial conditions

$$q(x, 0) = 0, \qquad T(x, 0) = 0 \tag{3.26}$$

and boundary conditions

$$q(0, t) = f(t), \quad \frac{\partial T(x, t)}{\partial t} = -\frac{\partial q(0, t)}{\partial x} \tag{3.27}$$

$$q(x, t) = 0, \quad T(x, t) = 0, \quad x \to \infty. \tag{3.28}$$

Since the boundary conditions are given in terms of the flux, we eliminate $T(x, t)$ from the hyperbolic system by differentiating (3.25) with respect to time and using (3.24) produces

$$q_{tt} + 2q_t = q_{xx}. \tag{3.29}$$

This second order hyperbolic equation requires two initial conditions and two boundary conditions for $q(x,t)$.

Rather than working with the semi-infinite interval we truncate this to interval $[0, L]$, and require $q(L,t) = 0$, and $T(L,t) = 0$. This now provides the required boundary conditions for the flux

$$q(0,t) = f(t), \qquad q(L,t) = 0. \tag{3.30}$$

Since the flux and temperature are both zero initially, (3.25) provides the other initial conditions for the flux i.e., $q_t(x,0) = 0$. Notice that for $f(t) = 1$ and L=1, we have the same problem as before in terms of the flux. This problem is linear and an exact solution, using separation of variables, is

$$q(x,t) = 1 - x - \frac{2}{\pi} e^{-t} \sum_{n=1}^{\infty} \frac{\sin \sqrt{\lambda} x}{n} \left(\cos \beta t + \frac{1}{\beta} \sin \beta t \right) \tag{3.31}$$

where

$$\sqrt{\lambda} = n\pi, \qquad \beta = \sqrt{\lambda - 1}. \tag{3.32}$$

Using the non-dimensional form of the energy equation provides us with the temperature distribution

$$T(x,t) = t + \frac{2}{\pi^2} \sum_{n=1}^{\infty} \frac{\cos \sqrt{\lambda} x}{n^2} \left(2 - 2e^{-t} \cos \beta t + \frac{\beta^2 - 1}{\beta} e^{-t} \sin \beta t \right). \tag{3.33}$$

If the boundary condition for the flux is given in terms of some periodic function $f(t) = \cos \omega t$, where ω is the frequency of the flux, the exact solution, by separation of variables, in terms of the flux is

$$q(x,t) = \cos \omega t(1-x) +$$

$$\sum_{n=1}^{\infty} \left[\frac{-2e^{-t}}{\sqrt{\lambda}} \left(\cos \beta t + \frac{1}{\beta} \sin \beta t \right) + A \cos \omega t + B \sin \omega t \right] \sin \sqrt{\lambda} x \tag{3.34}$$

where λ and β are given above and

$$A = \frac{-2\omega^4 - 2\omega^2(4-\lambda)}{\sqrt{\lambda}(4\omega^2 + (\lambda - \omega^2)^2}, \quad B = \frac{4\omega\lambda}{\sqrt{\lambda}(4\omega^2 + (\lambda - \omega^2)^2}. \tag{3.35}$$

The temperature distribution is

$$T(x,t) = \frac{1}{\omega} \sin \omega t +$$

$$\sum_{n=1}^{\infty} \left[\frac{2}{\beta^2 + 1} \left(2 + (\beta - \frac{1}{\beta} e^{-t} \sin \beta t - 2e^{-t} \cos \beta t \right) + \hat{A} + \hat{B} \right] \cos \sqrt{\lambda} x. \tag{3.36}$$

where

$$\hat{A} = -\frac{A \sin \omega t}{\omega}, \quad \hat{B} = \frac{B(\cos \omega t - 1)}{\omega}. \tag{3.37}$$

It is interesting to note that in the limit as $\omega \to 0$, the temperature distribution agrees with the solution when $q(0,t) = 1$. When $\omega \to \infty$, the temperature distribution given in (3.36) approaches the solution found when $q(0,t) = 1$. The difference is the term t found in equation (3.33). In other

words, as the frequency of the flux at the boundary increases, the temperature distribution approaches the same profile as in the case when the flux is the constant $q(0,t) = 1$. The profile is shifted vertically down a distance of t.

To see graphically the effect of the periodic flux on the hyperbolic heat equation under different frequencies it is helpful to compare the profile with the parabolic case. The non-dimensional problem using Fourier's law of heat conduction and the one dimensional energy equation is

$$\frac{\partial T}{\partial t} = -\frac{\partial q}{\partial x} \tag{3.38}$$

$$\frac{\partial T}{\partial x} + 2q = 0. \tag{3.39}$$

Eliminating $T(x,t)$ from the parabolic system and using the previous initial and boundary conditions with $L = 1$ gives

$$2\frac{\partial q}{\partial t} = \frac{\partial^2 q}{\partial x^2} \tag{3.40}$$

$$q(x,0) = 0, \quad q(0,t) = \cos \omega t, \quad q(1,t) = 0. \tag{3.41}$$

Figures 15-19 show the profiles of the exact solutions for the hyperbolic and parabolic cases at times t=.5 for $\omega = 0, .5, 1, 10, 100$.

Figures 20-22 for $\omega = 1, 10, 100$ show the surface response (temperture at the boundary $x = 0$) of the hyperbolic and parabolic cases. The first fundamental difference between the hyperbolic and parabolic cases is the difference

in the magnitude of the temperature at the surface. The parabolic case implies a gradual increase in surface temperature and a faster convergence to periodic behavior. The hyperbolic case implies an immediate response to the flux resulting in a much higher prediction for the surface temperature. Each of the Figures for the hyperbolic case show a jump in surface temperature at time $t \approx 2.0$. This is due to the time it takes the thermal wave to propagate across the medium, reflect off the right hand boundary, and then hit the left boundary at $t \approx 2.0$. When this reflecting thermal wave hits the boundary it causes a sudden jump in the surface temperature. This shows how hyperbolic heat transfer travels as a wave with finite speed of propagation. This also shows that the speed of the propagating thermal wave is independent of the frequency of the periodic flux being applied to the surface. In fact, for $\omega = 0$ (constant flux), the speed of propagation of the thermal wave is the same as the case for a periodic flux. Finally, these figures demonstrate that for this problem, hyperbolic heat transfer with a fast enough changing flux at the boundary, the surface temperature never drops below the initial levels.

The case of periodic flux with radiation effects at the boundary has not been investigated, and one of our goals in this work is to study the effects that radiation, combined with a periodic flux, has on the temperature distri-

bution for an opaque medium. Radiation introduces a non-linear term into the boundary condition, eliminating the possibility of an analytical solution. Hence a numerical solution must be found to investigate this problem. To develop a numerical method, which will produce accurate results, we first consider the hyperbolic system

$$\frac{\partial T}{\partial t} = -\frac{\partial q}{\partial x} \tag{3.42}$$

$$\frac{\partial q}{\partial t} = -2q - \frac{\partial T}{\partial x} \tag{3.43}$$

with initial conditions

$$q(x,0) = 0, \quad T(x,0) = 0 \tag{3.44}$$

and boundary conditions

$$q(0,t) = \cos \omega t, \quad \frac{\partial T(0,t)}{\partial t} = -\frac{\partial q(0,t)}{\partial x} \tag{3.45}$$

$$q(1,t) = 0, \quad T(1,t) = 0. \tag{3.46}$$

To solve this problem numerically we could use the fact that it is linear and use Fourier collocation as we did earlier, but if the effects of radiation are included which will make the problem non-linear, Fourier collocation is no longer applicable. Hence, we will disregard solving the system numerically

by artificially imposing periodic boundary conditions and using Fourier collocation and try to solve the system directly, since we want to eventually consider the effects of radiation.

The first difficulty we must address is the jump in the flux given by the boundary conditions. A large time step initially would certainly reduce the magnitude of the jump , but then stability constraints of the numerical scheme make this effort useless. As we pointed out earlier, this jump will produce numerical oscillation when we try to approximate the spatial derivatives. To reduce the oscillation caused by the jumps in both dependent quantities we will eliminate $T(x,t)$ from the system so we only have to worry about the jump in the flux. The problem then becomes

$$\frac{\partial^2 q}{\partial t^2} + 2\frac{\partial q}{\partial t} = \frac{\partial^2 q}{\partial x^2}. \tag{3.47}$$

The initial and boundary conditions have not changed, and we will take the frequency to be $\omega = .5$.

We first reduce the second order problem to a system of first order equations, then apply Chebyshev collocation to approximate q_{xx}. We will use 4-th order Runge-Kutta to advance the solution from q^n to q^{n+1}. Once we

have q^{n+1} we use the non-dimensional energy equation

$$\frac{\partial T}{\partial t} = -\frac{\partial q}{\partial x} \tag{3.48}$$

to find the temperature distribution.

Figures 23 and 24 show that the sudden jump in the flux results in severe numerical oscillation in the temperature profile. Various numerical schemes including operator splitting, predictor-corrector methods, and fully implicit schemes were tried. The results were all plagued by numerical oscillation due to the jump in the dependent quantities and the effects this had on Chebyshev collocation. Increasing the number of grid points does help to reduce the amplitude of the oscillation, but the time step restrictions due to further increasing the number of grid points makes this method impractical. Other spectral methods were also used in an attempt to reduce the oscillation. Tau methods which are spectral methods that use the same trial functions as pseudospectral methods in a fully discrete manner (Gottlieb and Orszag 1977), were also used. Tau methods did a better job handling the oscillation due to the jump at the boundary but the profile at the wave front was smeared. Despite the poor results, tau methods should not be overlooked. Excellent results can be found in Karageorghis (1988) and (Karageorghis and

Phillips 1989). In addition to these papers (Gottlieb and Orszag 1977) and Peyret (1986) also provide very useful information on using tau methods.

In the next sections we introduce some of the tools one can use to handle non-smooth data when using pseudospectral techniques. By using these techniques to reduce the oscillation to an acceptable level and doing so with a realistic number of grid points, will give us confidence in the numerical scheme and allow us to go on to the more interesting problem of the effects of radiation.

3.3 Perturbing the Initial Conditions

A simple, easy to implement, technique that attempts to damp out numerical oscillation is to perturb the initial conditions. In the case of the hyperbolic heat equation with a periodic flux, eliminating the temperature from the hyperbolic system gives

$$\frac{\partial^2 q}{\partial t^2} + 2\frac{\partial q}{\partial t} = \frac{\partial^2 q}{\partial x^2} \tag{3.49}$$

with initial conditions

$$q(x,0) = 0, \qquad q_t(x,0) = 0 \tag{3.50}$$

and boundary conditions

$$q(0,t) = \cos \omega t, \qquad q(1,t) = 0. \tag{3.51}$$

The initial conditions are zero and ,instantaneously, the flux at the surface jumps. To help smooth out this transition we define

$$q(x,0) = e^{-\alpha x} \tag{3.52}$$

where α is a parameter to be determined. Redefining the initial conditions in this way helps to absorb the sudden change at the boundary into the initial conditions. Figure 25 shows the temperature profile at time $t = .5$ with $q(x,0) = e^{-100x}$. Recall, the temperature distribution is given by

$$\frac{\partial T}{\partial t} = -\frac{\partial q}{\partial x}. \tag{3.53}$$

Figure 25 shows that this technique is ineffective in reducing any of the numerical oscillation. Experiments were run over a wide variety of values for the parameters α and ω, and the results were all disappointing. Tests were also done keeping the initial conditions fixed at zero and perturbing the boundary condition at $x = 0$ using the same function in (3.52). This was done during the first time step in the first internal stage of the Runge-Kutta scheme. Results of this did not produce any improvement. Variable

time stepping and finer grids were also used in addition to these techniques without any significant improvement.

3.4 Conservative Smoothing

Artificial viscosity is one of the most widely used techniques to suppress numerical oscillation. Unfortunately, it is also smears sharp gradients. In shock computations, artificial viscosity creates a wider spatial region over which the shock is defined Guenther et al.(1994). As a result of this smearing artificial viscosity was not used in this paper. Hicks (1969) developed a technique he termed conservative smoothing to handle numerical oscillation. Fortunately, the term smoothing is not due to the smoothing of gradients, but rather how the technique smoothes out the spikes or numerical oscillation. It can be shown (Guenther et al. 1994) that artificial viscosity of the type developed by von Neumann and Richtmyer is a special case of conservative smoothing. One of the main advantages of conservative smoothing over artificial viscosity is the ability to selectively use it on any part of the computational domain. In addition to being able to use it selectively in space, it can also be used selectively in time.

For the hyperbolic heat problem we will use conservative smoothing selectively in both space and time. Before using it on the hyperbolic heat problem it is helpful to demonstrate how it is used on a simple problem. Consider the conservation of volume given by

$$\frac{\partial V}{\partial t} = \frac{\partial u}{\partial x} \tag{3.54}$$

where V is volume and u is momentum. The discrete analogue of the equation given by von Neumann and Richtmyer is

$$(V_t)_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{V_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^{n}}{t^{n+1} - t^{n}} \tag{3.55}$$

and

$$(u_x)_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{u_{j+1}^{n+\frac{1}{2}} - u_{j}^{n+\frac{1}{2}}}{x_{j+1} - x_j}. \tag{3.56}$$

For simplicity we will consider a uniform mesh size

$$x_j = j\Delta x, \qquad 0 \leq j \leq J \tag{3.57}$$

with boundaries

$$x_L = x_0, \qquad x_R = x_J. \tag{3.58}$$

It should be pointed out that conservative smoothing is not restricted to the one dimensional case or limited to uniform mesh size. We can use this fact to

our advantage due to the grid spacing required by the Gauss-Lobatto points. Define a midpoint mesh by

$$x_{j+\frac{1}{2}} = \frac{x_j + x_{j+1}}{2}, \qquad 0 \leq j \leq J-1. \tag{3.59}$$

Next, conservatively rezone the volume distribution V on the original mesh to a $\hat{V}$ distribution on the midpoint mesh

$$\hat{V}_j = \frac{V_{j-\frac{1}{2}} + V_{j+\frac{1}{2}}}{2}, \qquad 1 \leq j \leq J-1, \tag{3.60}$$

with boundary conditions

$$\hat{V}_0 = V_{\frac{1}{2}}, \qquad \hat{V}_J = V_{J-\frac{1}{2}}. \tag{3.61}$$

The term "conservatively rezone" implies the total volume before is equal to the total volume after. Proof of this fact is given by Guenther et al.(1994). Next, we conservatively rezone the $\hat{V}$ distribution on the midpoint mesh to the $\widetilde{V}$ distribution on the original mesh

$$\widetilde{V}_{j+\frac{1}{2}} = \frac{\hat{V}_{j+1} + \hat{V}_j}{2}, \qquad 0 \leq j \leq J-1. \tag{3.62}$$

In terms of the original V distribution

$$\widetilde{V}_{j+\frac{1}{2}} = \frac{V_{j+\frac{3}{2}} + 2V_{j+\frac{1}{2}} + V_{j-\frac{1}{2}}}{4}, \quad 1 \leq j \leq J-2 \tag{3.63}$$

with boundary conditions

$$\widetilde{V}_{\frac{1}{2}} = \frac{V_{\frac{3}{2}} + 2V_{\frac{1}{2}} + V_{-\frac{1}{2}}}{4}, \ \widetilde{V}_{J-\frac{1}{2}} = \frac{V_{J+\frac{1}{2}} + 2V_{J-\frac{1}{2}} + V_{J-\frac{3}{2}}}{4}. \tag{3.64}$$

Notice the terms $V_{-\frac{1}{2}}$ and $V_{J+\frac{1}{2}}$ both are outside the computational domain. By defining these points by

$$V_{-\frac{1}{2}} = V_{\frac{1}{2}}, \qquad V_{J+\frac{1}{2}} = V_{J-\frac{1}{2}} \tag{3.65}$$

we have a single recipe for the conservatively smoothed volume $\widetilde{V}$ i.e.,

$$\widetilde{V}_{j+\frac{1}{2}} = \frac{V_{j+\frac{3}{2}} + 2V_{j+\frac{1}{2}} + V_{j-\frac{1}{2}}}{4}, \quad 0 \leq j \leq J-1. \tag{3.66}$$

This conservatively smoothed variable $\widetilde{V}$ can now be advanced in time. Notice that this method could have first advanced the volume V, and then conservatively smoothed this quantity at the new time level. In addition to smoothing the volume V, we could have also smoothed the momentum u.

Our problem first solves for the flux q given by the mixed initial boundary problem

$$\frac{\partial^2 q}{\partial t^2} + 2\frac{\partial q}{\partial t} = \frac{\partial^2 q}{\partial x^2} \tag{3.67}$$

with initial conditions

$$q(x,0) = 0, \qquad q_t(x,0) = 0 \tag{3.68}$$

and boundary conditions

$$q(0,t) = \cos \omega t, \qquad q(1,t) = 0. \tag{3.69}$$

Once q is found we can use the one dimensional energy equation

$$\frac{\partial T}{\partial t} = -\frac{\partial q}{\partial x} \tag{3.70}$$

to solve for the temperature.

The results given earlier demonstrated that as time advanced, the numerical method using Chebyshev collocation did and excellent job resolving the area in the vicinity of the wave front, but the sudden jump due to the boundary conditions at $x = 0$ caused severe oscillation. By locating the wave front at the time levels when conservative smoothing is used allows us to use conservative smoothing on the flux, or temperature, or both, behind the wave front. This selective use of conservative smoothing is one of the main advantages in using it over artificial viscosity of the von Neumann-Richtmyer type.

Figure 26 shows the results using conservative smoothing on the problem given above. Smoothing was performed on the flux after it was advanced in time. This smoothed flux was then used to find $\frac{\partial q}{\partial x}$. The temperature is then advanced to the new time level explicitly in terms of conservatively smoothed

quantities. Smoothing was also performed prior to advancing the flux in time, but the results of this did not offer any significant improvement over smoothing after advancing the flux in time. Smoothing was done on the first 5 time steps and then turned off. Prior to the final time, in this case $t = .5$, smoothing was turned back on for the final several time steps. The location of the smoothing was always selected behind the wave front, preventing any significant smearing of the wave front. As this figure shows, conservative smoothing could not remove all the oscillation near the boundary and, in fact, this oscillation is even more pronounced if the smoothing is kept off after the initial amount of smoothing has been performed. Experiments were performed for various times and values of ω, $0 \leq \omega \leq 1$. The smoothing routine was identical to above recipe. In each case the wave front remained sharp and the oscillation at the boundary $x = 0$ was always within the levels found in Figure 26.

With a numerical method in place that can successfully model the hyperbolic system directly we can now investigate the case of a periodic flux with radiation. The problem is

$$\frac{\partial^2 q}{\partial t^2} + 2\frac{\partial q}{\partial t} = \frac{\partial^2 q}{\partial x^2} \tag{3.71}$$

with initial conditions

$$q(x,0) = 0, \qquad q_t(x,0) = 0 \tag{3.72}$$

and boundary conditions

$$q(0,t) = \cos\omega t - \epsilon T^4(0,t), \qquad q(1,t) = 0. \tag{3.73}$$

The parameter ϵ is a non-dimensional surface absorptivity constant (Glass, Ozisik, and Vick 1985) which in this problem we take as $0 \leq \epsilon \leq .5$. Under these conditions the surface is completely absorbing and the flux at the boundary remains positive. Again, we first solve for the flux q and then use the energy equation to solve for the temperature. Recall that Chebyshev collocation requires us to interpolate across the entire domain and this requires the boundary conditions to be known prior to interpolating. It is obvious that the present boundary conditions will prevent this requirement from being explicitly satisfied. Hence, iteration will be performed to evaluate the boundary conditions and advance the flux in time.

To evaluate the boundary conditions, the previous value for $T(0,t)$ was used to begin the internal stages of the Runge-Kutta subroutine. Once the Runge-Kutta subroutine advanced the flux q using conservative smoothing we could then calculate $\frac{\partial q}{\partial x}$ and use the energy equation to determine the

temperature at the boundary. This new value for the temperature at the boundary was then used to repeat the process again. Five iterations were found to be satisfactory to evaluate the boundary conditions and advance the flux to the next time level.

Figure 27 shows the profile of the temperature at time $t = .5$ for various values of ϵ. This figure also shows the temperature profile for the parabolic case when $\epsilon = .5$. As ϵ is increased the effects of radiation become more pronounced causing the magnitude of the flux to decrease at the boundary, which reduces the temperature response. As we saw before, hyperbolic heat transfer results in a larger response to an imposed temperature gradient, and the difference between hyperbolic behavior and parabolic behavior is quite pronounced. When radiation effects are considered, the behavior between hyperbolic and parabolic behavior begins to converge. This is shown in Figure 27 by the fact the temperature response when $\epsilon = .5$ is quite similar near the surface between the hyperbolic and parabolic cases. This convergence can be explained by the fact that hyperbolic behavior results in a higher surface temperature which then loses more energy due to the effects of radiation than the case of parabolic behavior. The results shown in Figure 27 used 40 grid points and a fixed timestep of .001. These values differ dramatically from

Glass, Ozisik, and Vick (1985), where 1000 grid points and variable timestepping with extremely small initial steps were needed to prevent instabilities and reduce numerical oscillation at the wave front.

Summary and Recommendations

The results in Chapter 3 demonstrate pseudospectral methods can be used successfully in hyperbolic heat transfer. When radiation effects are neglected, and the problem is linear, artificially imposing periodic boundary conditions and using Fourier collocation significantly improves previous results which used finite difference and finite element methods. By using Fourier collocation only on the transient part of the solution, numerical oscillation is prevented at the wave front without smearing its profile. Furthermore, these results are obtained with a mesh of only 16 grid points and a fixed timestep of .05.

When radiation effects were considered Chebyshev collocation with conservative smoothing had to be used, because of the non-linear effects of the radiation term and the sudden jump in dependent quantities at the boundary. These results dramatically improved the computational effort previously needed to investigate the effects of radiation. Pseudospectral methods allowed for a much coarser grid and larger timestep than previous methods.

For a very fast fluctuating flux the methods presented in this work, as they stand, are not sufficient to model this behavior. This is the reason why omega was restricted to a small value. Certainly Figures 20-22 are motivation enough to numerically investigate this problem for larger values of omega and the effects radiation would have on the temperature response. The challenge of

this problem is in developing a technique that when used withChebyshev collocation would capture the rapidly changing conditions near the surface. Other trial functions could be considered, and rational functions would certainly be a logical beginning. In addition to different trial functions, domain decomposition that would split the problem into fast and slow varying regions might also prove helpful.

Bibliography

Abramowitz, M. and A. Stegun, Eds., (1965). Handbook of Mathematical Functions. New York: Dover publishing.

Baumeister, K. and T., Hamill (1969). "Hyperbolic Heat-Conduction Equation-A Solution for the Semi-Infinite Body Problem," J. of Heat Transfer Nov. 543-548 .

Bayliss, A. and E. Turkel (1992). "Mappings and Accuracy for Chebyshev Pseudospectral Approximations," J. of Computational Physics 101, 349-359.

Bird, R., W. Stewart, and E. Lightfoot (1960). Transport Phenomena. New York: John Wiley & Sons.

Boyd, J., (1982). "The Optimization of Convergence for Chebyshev Polynomial Meth ods in an Unbounded Domain," J. of Computational Physics 45, 43-79.

Boyd, J., (1987). "Spectral Methods Using Rational Basis Functions on an Infinite Interval," J. of Computational Physics 69, 112-142.

Boyd, J., (1989). Chebyshev and Fourier Spectral Methods. New York: Springer-Verlag.

Boyd, J., (1992). "Multiple Expansions and Pseudospectral Cardinal Functions," J. of Computational Physics 103, 184-186.

Breuer, K., and R. Everson (1992). "On The Errors Incurred Calculating Derivatives Using Chebyshev Polynomials," Journal of Computational Physics 99, 56-

67.

Burden, R. and D., Faires (1985). Numerical Analysis 3rd ed. Massachusetts: Prindle Weber and Schmidt.

Canuto, C., M., Hussaini, A., Quarteroni, and A., Zang (1988). Spectral Methods in Fluid Dynamics. New York: Springer-Verlag.

Carey, G., M., Tsai (1982). "Hyperbolic Heat Transfer With Reflection," Numerical Heat Transfer, 5, 309-327.

Conte, S., and C., deBoor (1972). Elementary Numerical Analysis. New York: McGraw-Hill.

Fornberg, B., (1987). "The Pseudospectral Method: Comparison With Finite Difference for the Elastic Wave Equation," Geophysics, 52, 483-501.

Fornberg, B., (1996). A Practical Guide to Pseudospectral Methods. New York: Cambridge University Press.

Fox, L., and I., Parker (1968). Chebyshev Polynomials in Numerical Analysis. New York: Oxford University Press.

Frankel, J., and B., Vick (1985). "J. Applied Physics," Flux Formulation of Hyperbolic Heat Conduction," 58 no. 9 3340-3345.

Glass, D., M., Ozisik, and D. McRae (1987). "Hyperbolic Heat Conduction With Radiation in an Absorbing and Emitting Medium," Numerical Heat Transfer 12, 321-333.

Glass, D., M., Ozisik, D., McRae, and B. Vick (1985). "On The Numerical Solution of Hyperbolic Heat Conduction," Numerical Heat Transfer 8, 497-504.

Glass, D., M., Ozisik, and B. Vick (1985). "Hyperbolic Heat Conduction With Surface Radiation," Int. J. Heat and Mass Transfer 28, 10, 1823-1830.

Glass, D., M., Ozisik, and B. Vick (1987). "Non-Fourier Effects on Transient Tem perature Resulting From Periodic On-Off Heat Flux," Int. J. Heat and Mass Transfer 30, 8, 1623-1631.

Gottlieb, D., and S. Orszag (1977). Numerical Analysis of Spectral Methods: Theory and Applications. Philadelphia: SIAM-CBMS

Gottlieb, D., and E., Turkel. (1985). Topics in Spectral Methods for Time Dependent Problems. New York: Springer-Verlag.

Guenther, C., D., Hicks, and J., Swegle (1994). "Conservative Smoothing Versus Artificial Viscosity," Sandia Report Sand94-1853 UC-705.

Haberman, R., (1987). Elementary Applied Partial Differential Equations. New Jersey: Prentice Hall.

Hamming, R., (1962). Numerical Methods for Scientists and Engineers. New York: McGraw Hill.

Hicks, D., (1969). "The Convergence of Numerical Solutions of Hydrodynamic Shock Problems," Ph.D. Thesis, University of New Mexico.

Isaacson, E. and H. Keller (1966). Analysis of Numerical Methods. New York: John Wiley and Sons.

Karageorghis, A., (1988). "Chebyshev Spectral Methods for Solving Two-Point Boundary Value Problems Arising in Heat Transfer," Computational Methods in Applied Mechanics and Engineering 70, 103-121.

Karageorghis, A. and T. Phillips (1989). "Spectral Collocation Methods for Stokes Flow in Contraction Geometries and Unbounded Domains," Journal of Compu tational Physics 80, 314-330.

Kreiss, H. and J. Oliger (1972). "Comparison of Accurate Methods for the Integration of Hyperbolic Equations," Tellus, 24, 199-215.

Kreith, F., (1973). Principles of Heat Transfer. New York: Harper and Row.

Luikov, A., V. Bubnov, and I. Soloviev (1976). "On Wave Solutions of the Heat Conduction Equation," International Journal of Heat and Mass Transfer 19, 245-248.

Majda, A., J., McDonough, and S., Osher (1978). "The Fourier Method for Nonsmooth Initial Data," Mathematics of Computation 32 no. 144, 1041-1081.

Mitchell, A., and D. Griffiths (1980). The Finite Difference Methods in Partial Differential Equations. New York: John Wiley and Sons.

Orszag, S., (1972). "Comparison of Pseudospectral and Spectral Approximations," Studies in Applied Mathematics, 51, 253-259.

Peyret, R., (1986). "Introduction To Spectral Methods With Application to Fluid Mechanics," von-Karman Institute for Fluid Dynamics Lecture Series 1986-04.

Press, W., S., Teukolsky, W., Vetterling, and W., Flannery (1992). Numerical Recipes in Fortran 2nd Eds. New York: Cambridge University Press.

Rivlin, T., (1974). Pure and Applied Mathematics. New York: John Wiley and Sons.

Sansone, G., (1959). Orthogonal Functions. New York: Interscience Publishers Inc.

Sanz-Serna, J., (1994). "Fourier Techniques in Numerical Methods for Evolutionary Problems," Universidad De Valladolid Applied Mathematics and Computational Report 1994/10.

Smith, G., (1985). Numerical Solution of Partial Differential Equations. Oxford, Eng land: Clarendon Press.

Solomonoff, A., and E. Turkel (1989). "Global Properties of Pseudospectral Methods," Journal of Computational Physics 81, 239-276.

Spiegel, M., (1968). Schaum's Oulines Mathematical Handbook. New York: McGraw-Hill.

Strang, G., (1986). Introduction to Applied Mathematics. Massachusetts: Wellesley-Cambridge Press.

Tang, D., and N. Araki (1996). "Non-Fourier Heat Conduction in a Finite Medium Under Periodic Surface Therma Disturbance," Int. J. Heat and Mass Transfer 39 no. 8, 1585-1590.

Voigt, R., D., Gottlieb, and Y. Hussaini, Eds. (1984). "Spectral Methods for Partial Differential Equations," Philadelphia: SIAM.

Zauderer, E., (1983). Partial Differential Equations of Applied Mathematics. New York: John Wiley and Sons.

Appendix A

```
      subroutine fmatrix(fr,fi)
c     this subroutine calculates the n x n Fourier transform matrix
c     recall this matrix is complex this subroutine calculates both
c     the real fr and complex fi entries
      double precision fr(n,n),fi(n,n)
      double precision dim,pi,val1,val2
      integer r,s,i,j
      dim=n
      pi=3.141592653589793d0
      val1=2d0*pi/dim
      do 1 j=1,n
      do 2 i=1,n
      r=j-1
      s=i-1
      val2=val1*r*s
      fr(i,j)=dCOS(val2)
      fi(i,j)=-dSIN(val2)
2     continue
1     continue
      return
      end

c     This subroutine transforms the vector real part vinr and
c     complex part vini to Fourier space
c     and takes the required spatial derivative.
      subroutine fouriercolloc(fr,fi,vinr,vini,yr,yi)
      double precision fr(n,n),fi(n,n),vinr(n),vini(n),yr(n),yi(n)
      double precision sqlam(n)
      double precision st2r(n),st2i(n)
      double precision sumr,sumi,dim,al,pi,val1
      integer m
      pi=3.141592653589793d0
c     the problem is periodic with period al
      al=2.0d0
      dim=n
      val1=2d0*pi/al
      m=n/2
c     lambda matrix(this matrix is a diagonal matrix)
      do 10 i=1,n
      sqlam(i)=-((val1)*(i-1-n))**2
10    continue
c     Calculate the second derivative using the spectral
c     differentiation matrix in terms of real and imaginary parts
      do 20 i=1,n
      sumr=0.0d0
```

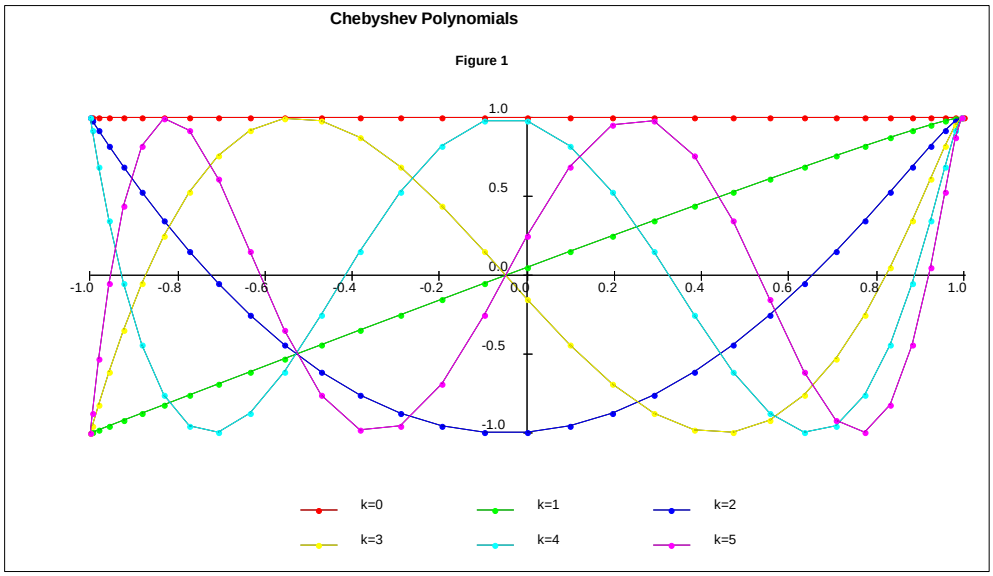
```
      sumi=0.0d0
c     The transform of the given vector in terms or real
c     and imaginary parts.
      do 30 k=1,n
      sumr=sumr+fr(i,k)*vinr(k)-fi(i,k)*vini(k)
      sumi=sumi+fi(i,k)*vinr(k)+fr(i,k)*vini(k)
30    continue
      yr(i)=sumr
      yi(i)=sumi
      st2r(i)=yr(i)
      st2i(i)=yi(i)
20    continue
c     Permute the transformed vector
      do 40 i=1,m
      yr(i)=st2r(i+m)
      yr(i+n)=st2r(i)
      yi(i)=st2i(i+m)
      yi(i+n)=st2i(i)
40    continue
c     Multiply the vector by lambda squared
      do 50 i=1,n
      st2r(i)=sqlam(i)*yr(i)
      st2i(i)=sqlam(i)*yi(i)
50    continue
c     Permute back
      do 60 i=1,m
      yr(i)=st2r(i+m)
      yr(i+n)=st2r(i)
      yi(i)=st2i(i+m)
      yi(i+n)=st2i(i)
60    continue
c     Invert back out of Fourier space
      do 70 i=1,n
      sumr=0d0
      sumi=0d0
      do 80 k=1,n
      sumr=sumr+fr(i,k)*yr(k)+fi(i,k)*yi(k)
      sumi=sumi-fi(i,k)*yr(k)+fr(i,k)*yi(k)
80    continue
      st2r(i)=sumr
      st2i(i)=sumi
70    continue
c     The second derivative is approximated by:
c     yr=real part
c     yi=imaginary part
      do 90 i=1,n
      yr(i)=st2r(i)/dim
      yi(i)=st2i(i)/dim
90    continue
```

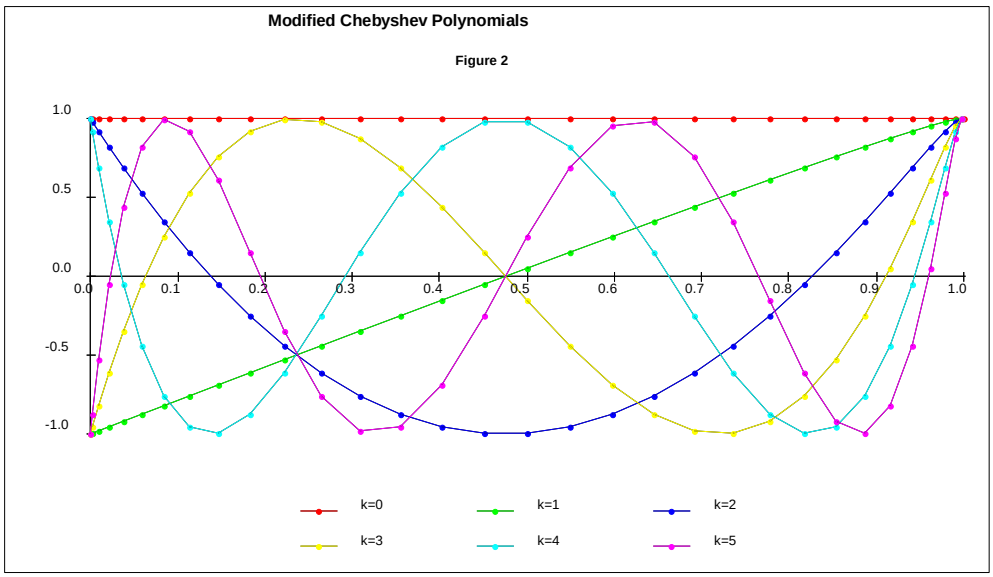
Appendix B

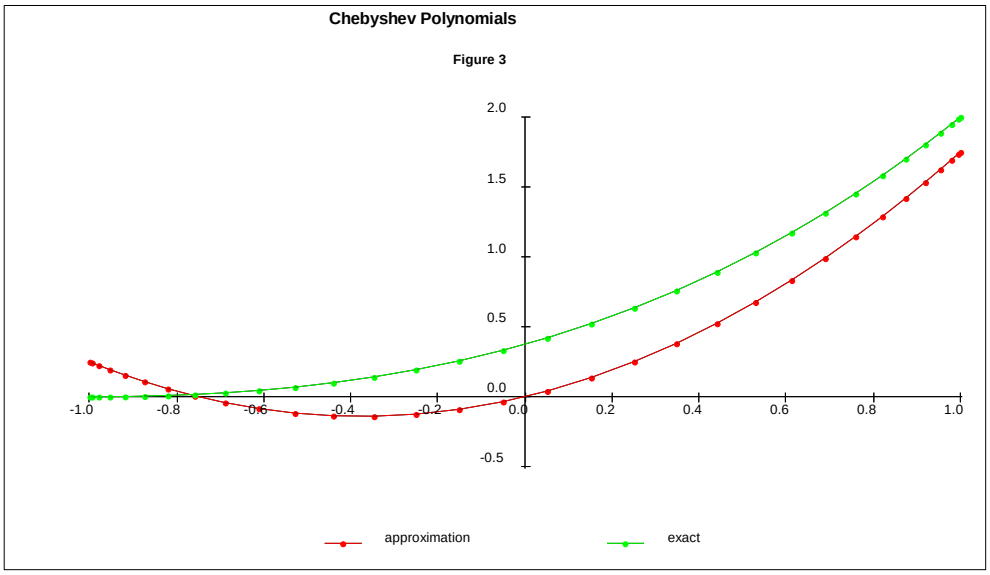
```
      Subroutine Cheby(uhat,u1,u2,uinit,diffu,diff2u)
      double precision uhat(n),u1(n),u2(n),uinit(n)
      double precision diffu(n),diff2u(n)
c     calculates the first and second derivatives of a
c     vector of n-data values uinit(n). The results are stored
c     in diffu(n), diff2u(n)
      double precision c(n)
      double precision pi,sum,sum2
      pi=3.141592653589793d0
      do 1 i=1,n
      if(i.eq.1.or.i.eq.n)then
      c(i)=2.0d0
      else
      c(i)=1.0d0
      endif
1     continue
c     calculates the discrete Chebyshev coefficients
      do 2 i=1,n
      sum=0.0d0
      do 3 j=1,n
      sum=1.0d0/c(j)*uinit(j)*dCOS(pi*(j-1.0d0)*(i-1.0d0)/n)+sum
3     continue
      uhat(i)=(2.0d0/((n-1.0d0)*c(i)))*sum
2     continue
c     recurrence relation to find derivative coefficients
      u2(n+1)=0.0d0
      u2(n)=0.0d0
      u1(n+2)=0.0d0
      u1(n)=0.0d0
      do 4 i=n-1,1,-1
      u1(i)=1.0d0/c(i)*(u1(i+2)+2.0d0*(i*u(i+1))
      u2(i)=1.0d0/c(i)*(u2(i+2)+2.0d0*(i*u1(i+1))
4     continue
      do 5 i=1,n
      sum=0.0d0
      sum2=0.0d0
      do 6 j=1,n
      sum=u1(j)*dCOS(pi*(i-1.0d0)*j-1.0d0)/(n-1))+sum
      sum2=u2(j)*dCOS(pi*(i-1.0d0)*j-1.0d0)/(n-1))+sum2
6     continue
c     the first and second derivatives are below
      diffu(i)=sum
      diff2u(i)=sum2
5     continue
      return
      end
```

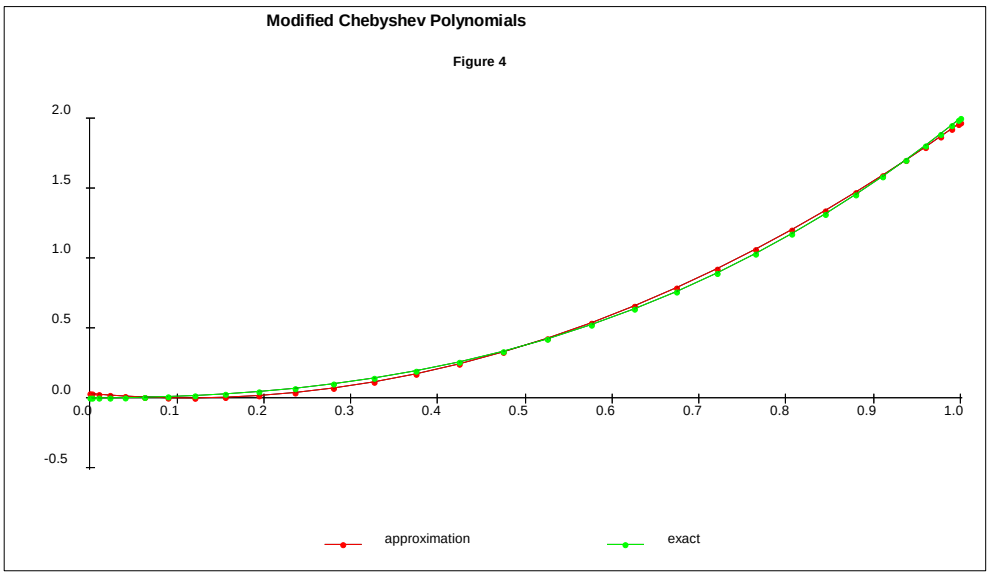
Appendix C

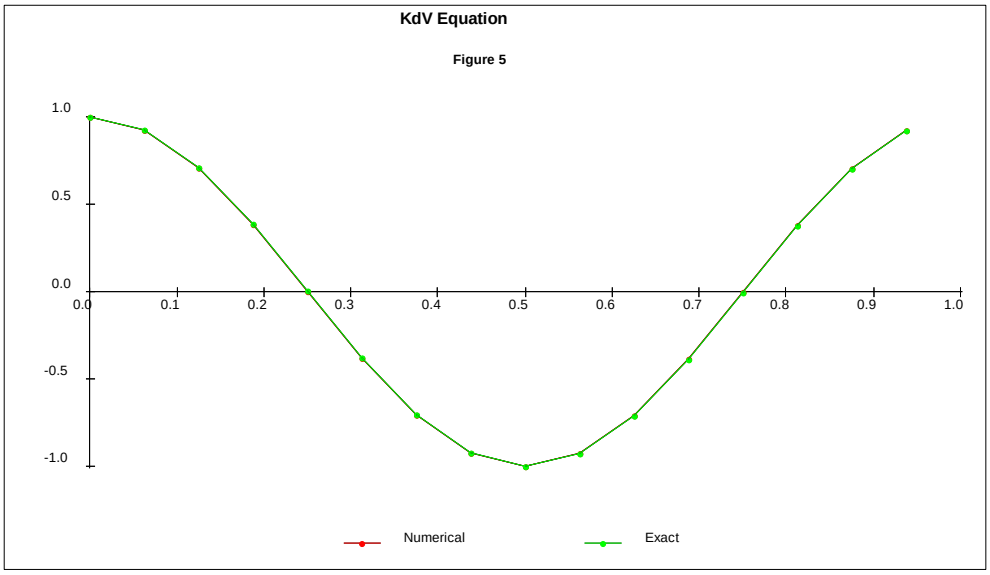
```
      Subroutine Cheby(uhat,u1,u2,uinit,diffu,diff2u)
      double precision uhat(n),u1(n),u2(n),uinit(n)
      double precision diffu(n),diff2u(n)
      double precision c(n)
      double precision pi,sum,sum2
      pi=3.141592653589793d0
      do 1 i=1,n
      if(i.eq.1.or.i.eq.n)then
      c(i)=2.0d0
      else
      c(i)=1.0d0
      endif
1     continue
c     calculates the discrete Chebyshev coefficients
      do 2 i=1,n
      sum=0.0d0
      do 3 j=1,n
      sum=1.0d0/c(j)*uinit(j)*dCOS((i-1.0d0)*dACOS(-dCOS(pi*
            (j-1.0d0)/(n-1))))+sum
3     continue
      uhat(i)=(2.0d0/((n-1.0d0)*c(i)))*sum
2     continue
c     recurrence relation to find derivative coefficients
      u2(n+1)=0.0d0
      u2(n)=0.0d0
      u1(n+2)=0.0d0
      u1(n)=0.0d0
      do 4 i=n-1,1,-1
      u1(i)=1.0d0/c(i)*(u1(i+2)+4.0d0*(i*u(i+1))
      u2(i)=1.0d0/c(i)*(u2(i+2)+4.0d0*(i*u1(i+1))
4     continue
      do 5 i=1,n
      sum=0.0d0
      sum2=0.0d0
      do 6 j=1,n
      sum=u1(j)*dCOS((j-1.0d0)*dACOS(-dCOS(pi*
      (i-1.0d0/(n-1.0d0))))+sum
      sum2=u2(j)*dCOS((j-1.0d0)*dACOS(-dCOS(pi*
      (i-1.0d0/(n-1.0d0))))+sum2
6     continue
c     the first and second derivatives are below
      diffu(i)=sum
      diff2u(i)=sum2
5     continue
      return
```

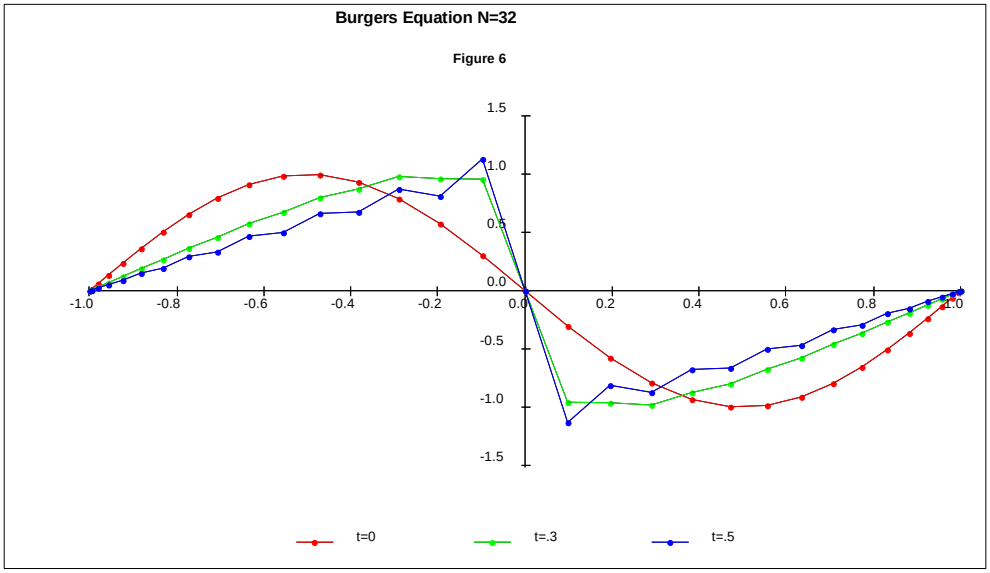

Chebyshev Polynomials
Figure 1
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
k=0
k=1
k=2
k=3
k=4
k=5

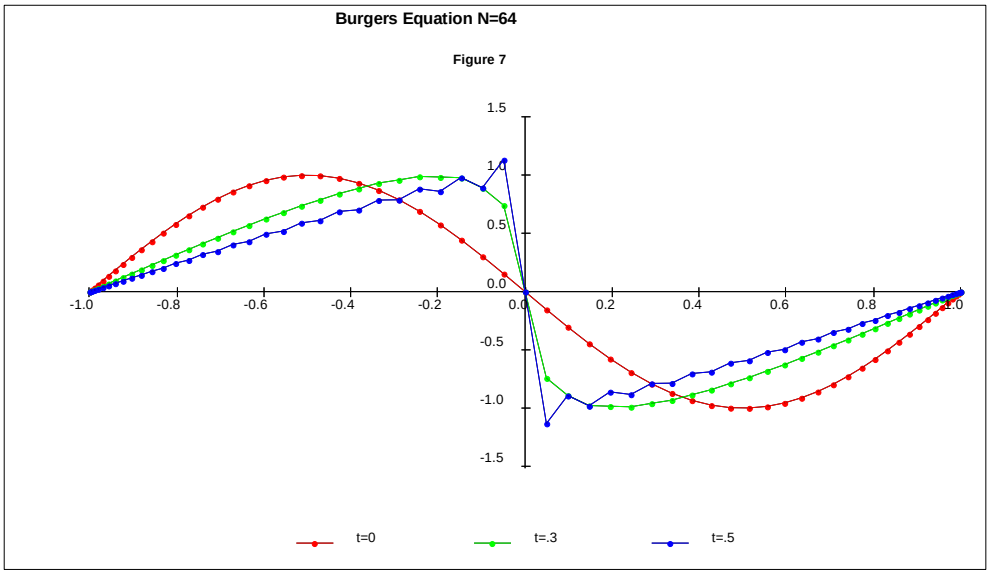

Modified Chebyshev Polynomials
Figure 2
1.0
0.5
0.0
-0.5
-1.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
k=0
k=1
k=2
k=3
k=4
k=5

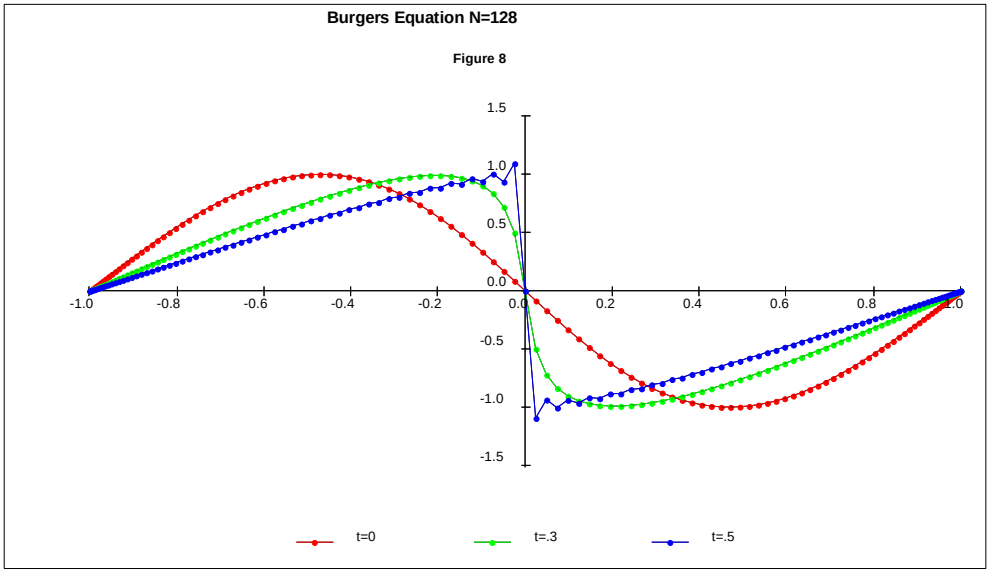

Chebyshev Polynomials
Figure 3
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
approximation
exact

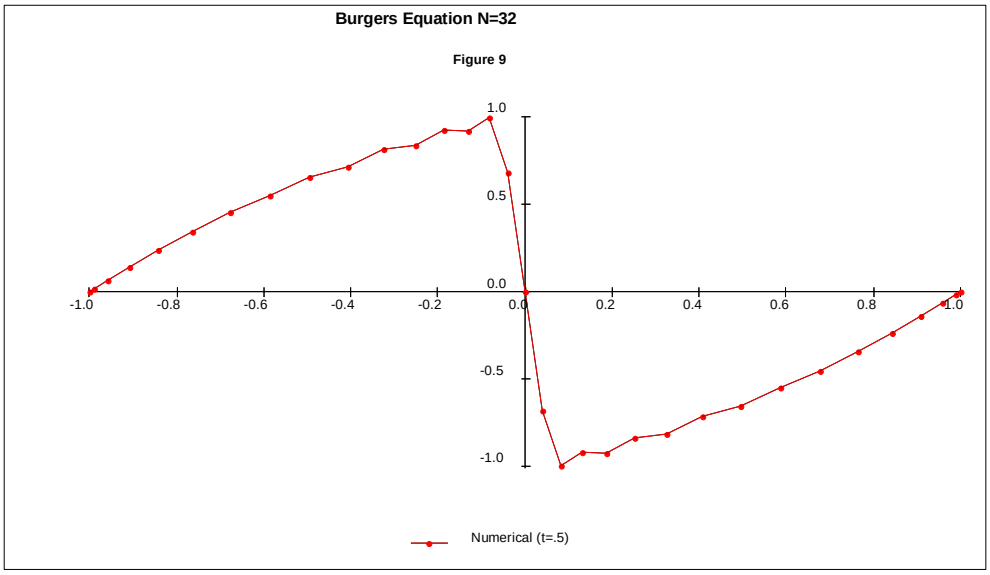

Modified Chebyshev Polynomials
Figure 4
2.0
1.5
1.0
0.5
0.0
-0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
approximation
exact

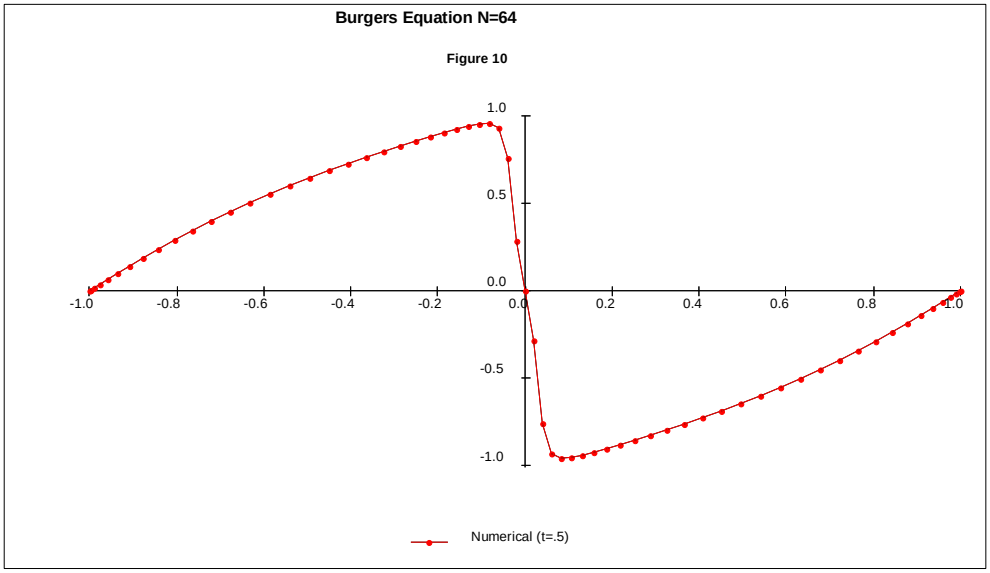

KdV Equation
Figure 5
1.0
0.5
0.0
-0.5
-1.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Numerical
Exact

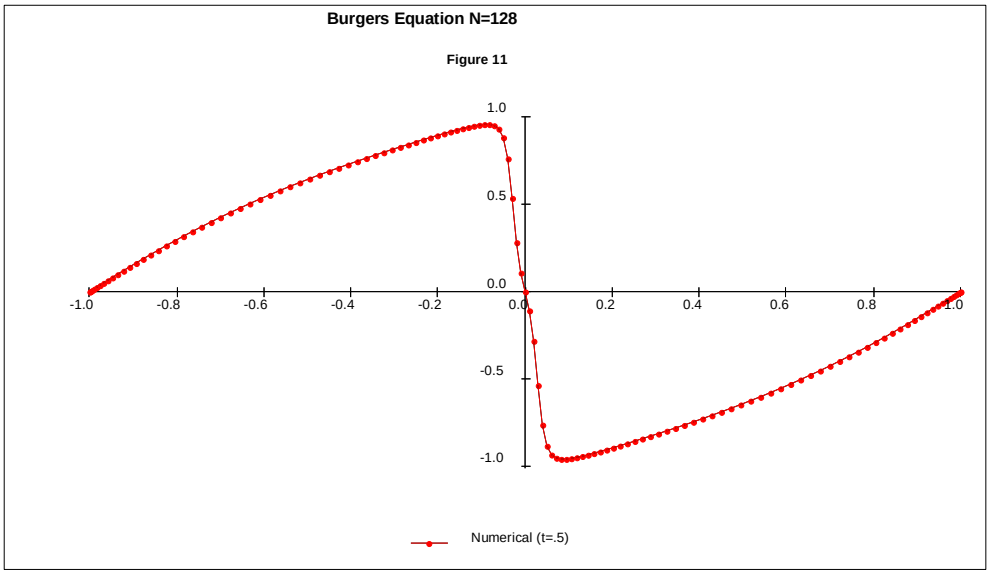

Burgers Equation N=32
Figure 6
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
t=0
t=.3
t=.5

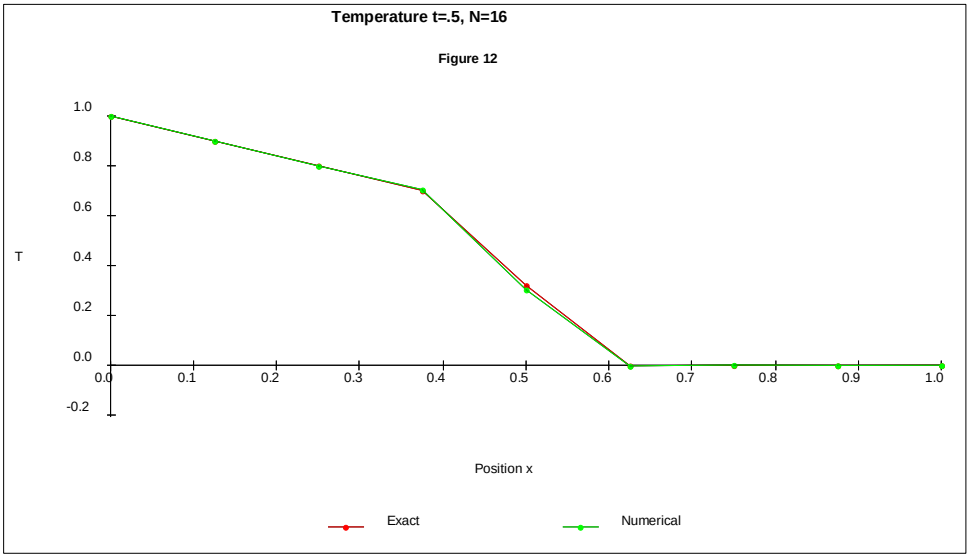

Burgers Equation N=64
Figure 7
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
t=0
t=.3
t=.5

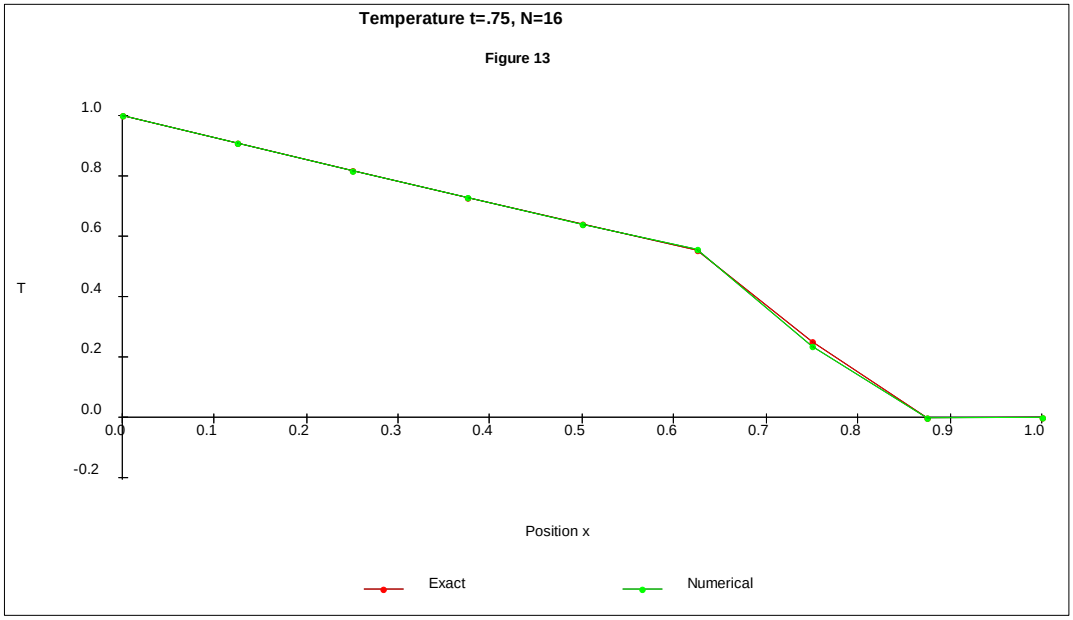

Burgers Equation N=128
Figure 8
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
t=0
t=.3
t=.5

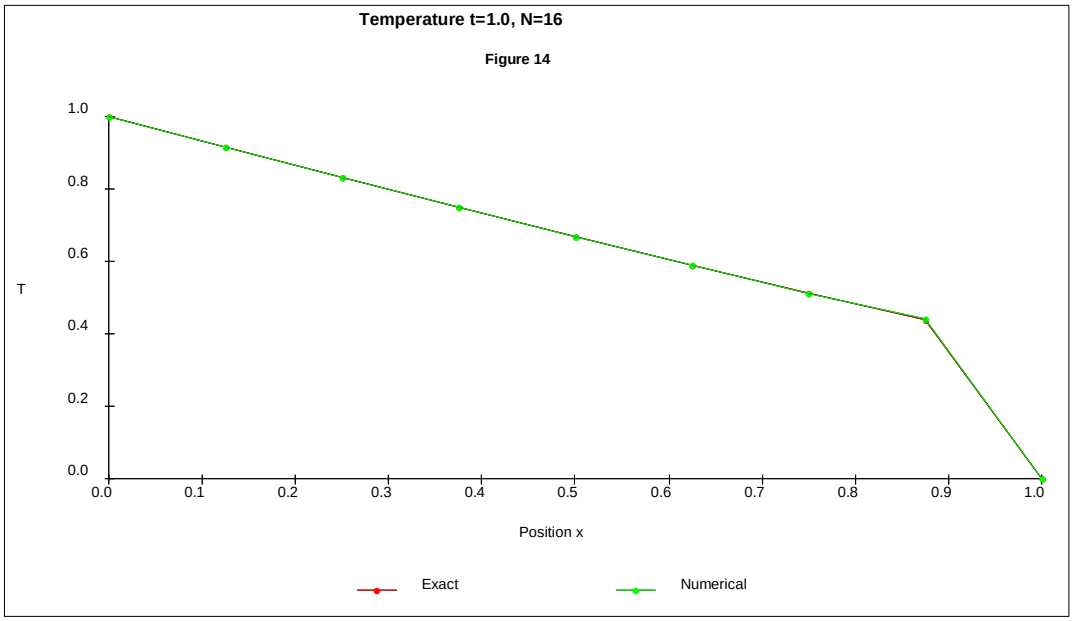

Burgers Equation N=32
Figure 9
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
Numerical (t=.5)

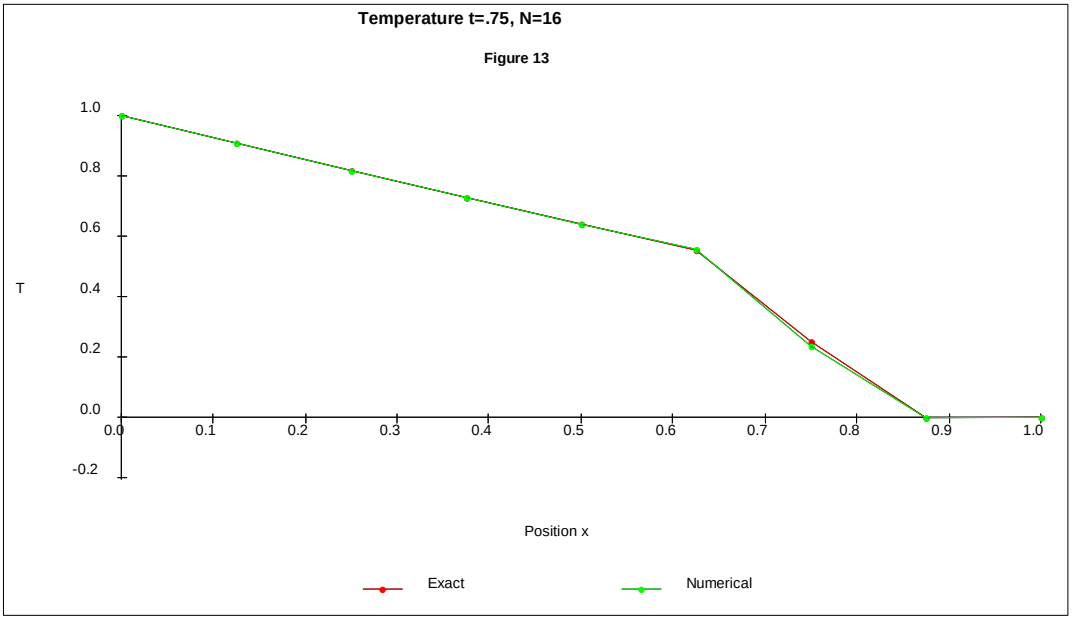

Burgers Equation N=64
Figure 10
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
Numerical (t=.5)

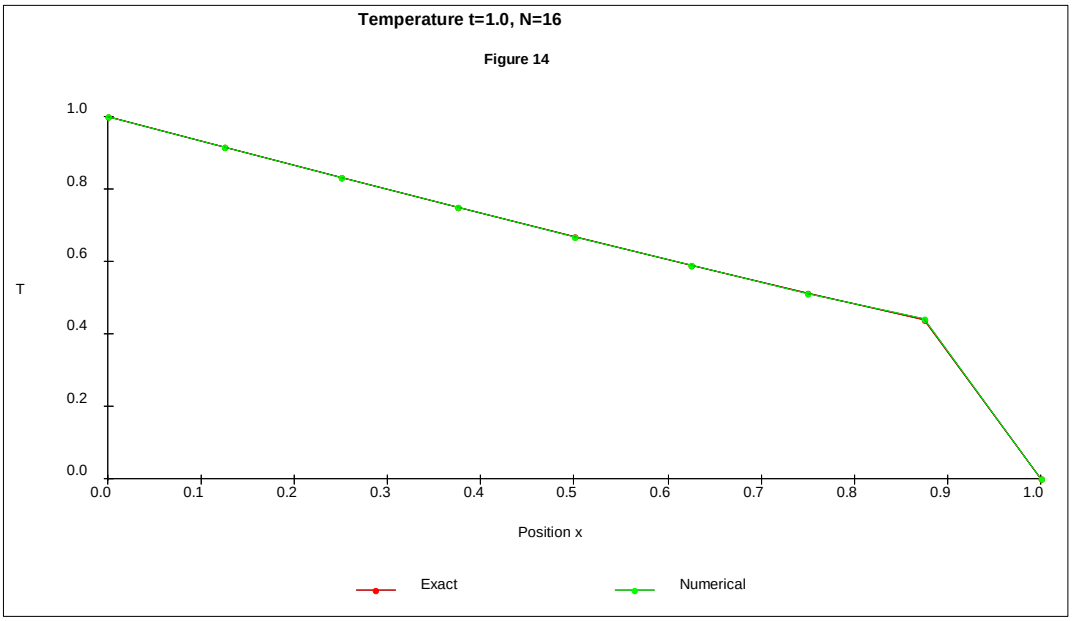

Burgers Equation N=128
Figure 11
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
Numerical (t=.5)

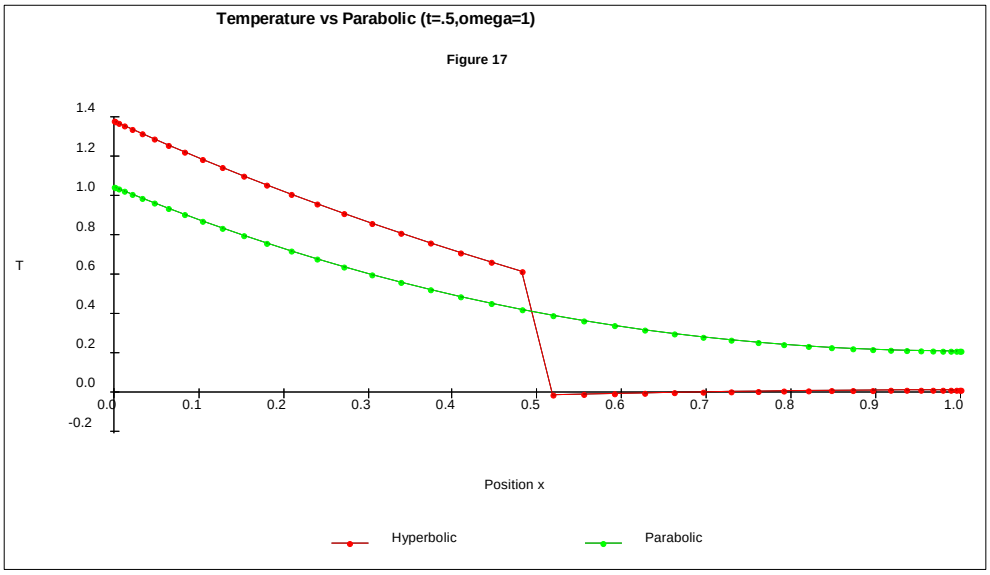

Temperature t=.5, N=16
Figure 12
T
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Exact
Numerical

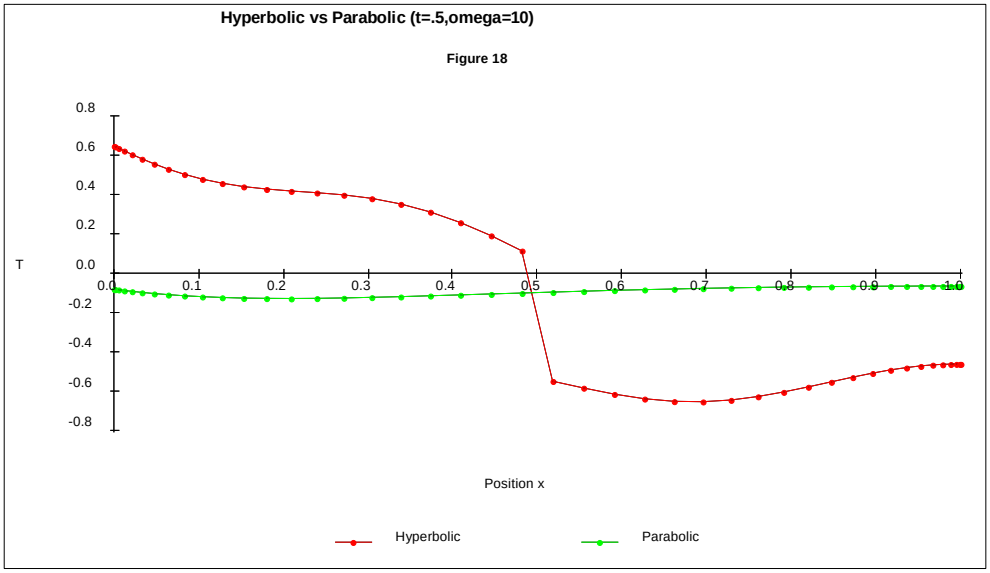

Temperature t=.75, N=16
Figure 13
1.0
0.8
0.6
T
0.4
0.2
0.0
-0.2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Exact
Numerical

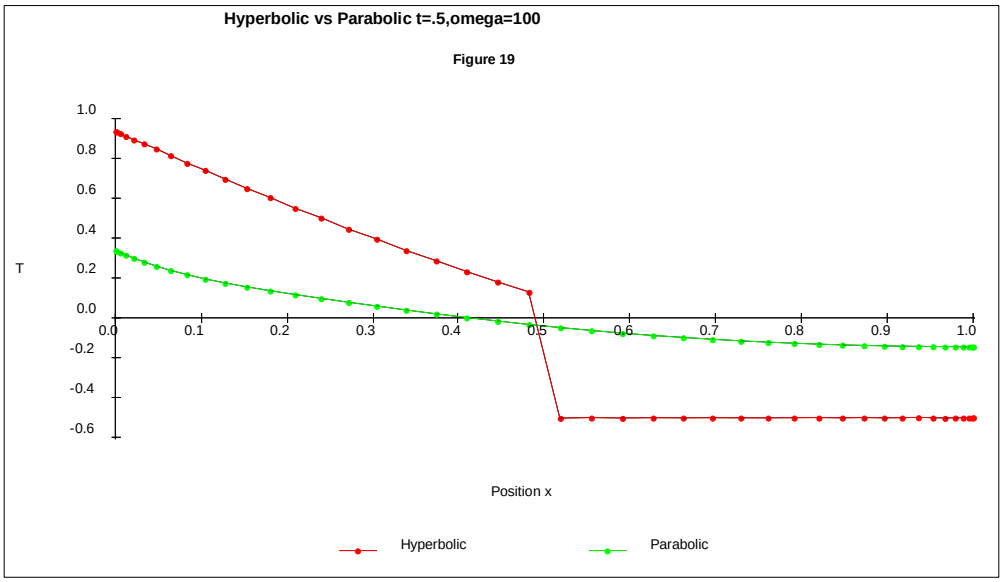

Temperature t=1.0, N=16
Figure 14
1.0
0.8
0.6
T
0.4
0.2
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Exact
Numerical

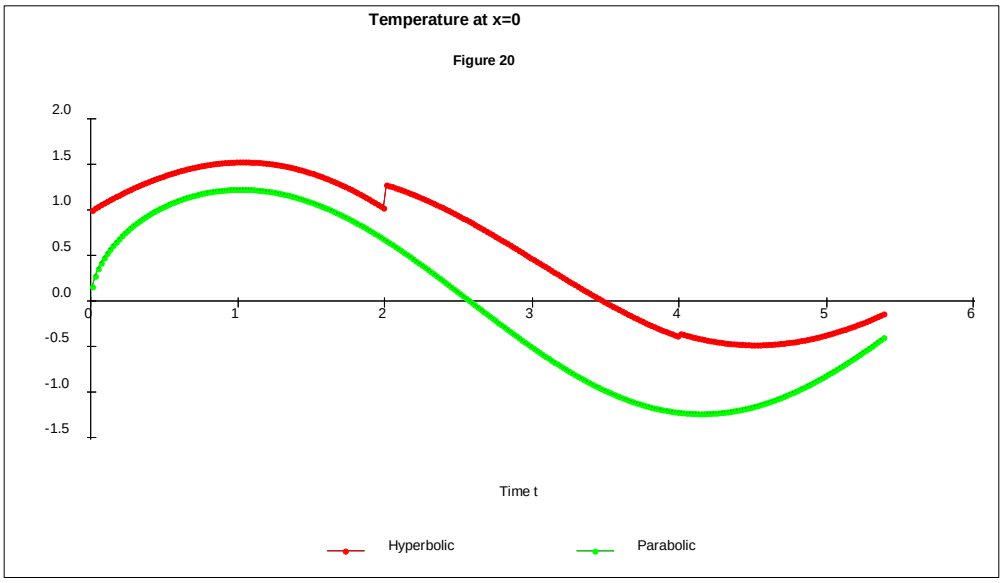

Temperature t=.75, N=16
Figure 13
1.0
0.8
0.6
T
0.4
0.2
0.0
-0.2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Exact
Numerical

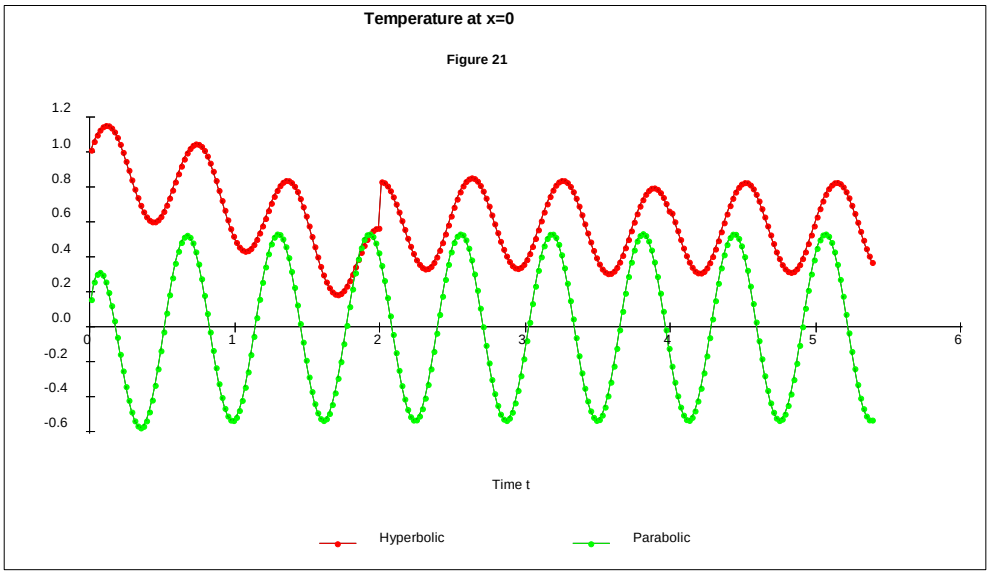

Temperature t=1.0, N=16
Figure 14
1.0
0.8
0.6
T
0.4
0.2
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Exact
Numerical

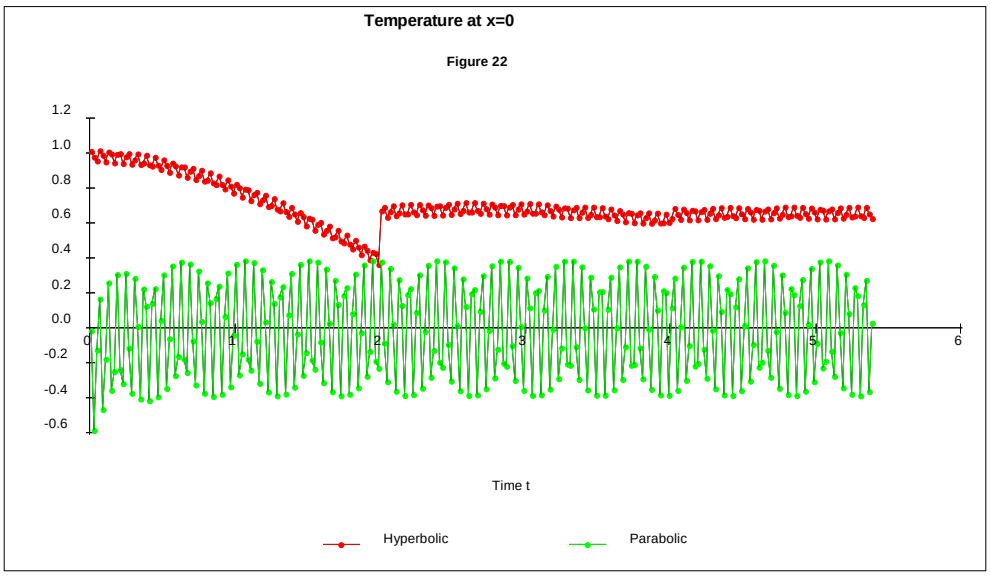

Temperature vs Parabolic (t=.5,omega=1)
Figure 17
T
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Hyperbolic
Parabolic

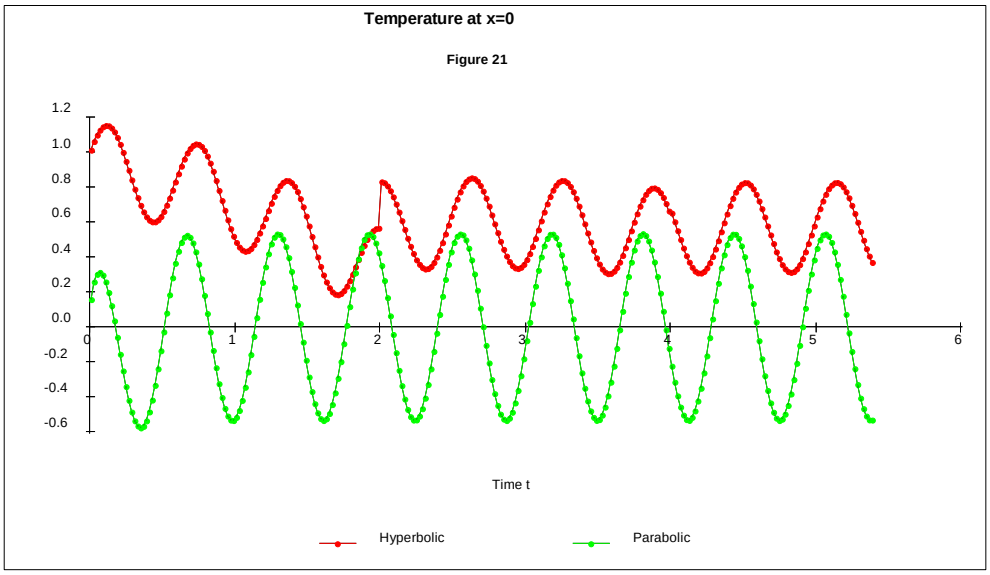

Hyperbolic vs Parabolic (t=.5,omega=10)
Figure 18
T
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Hyperbolic
Parabolic

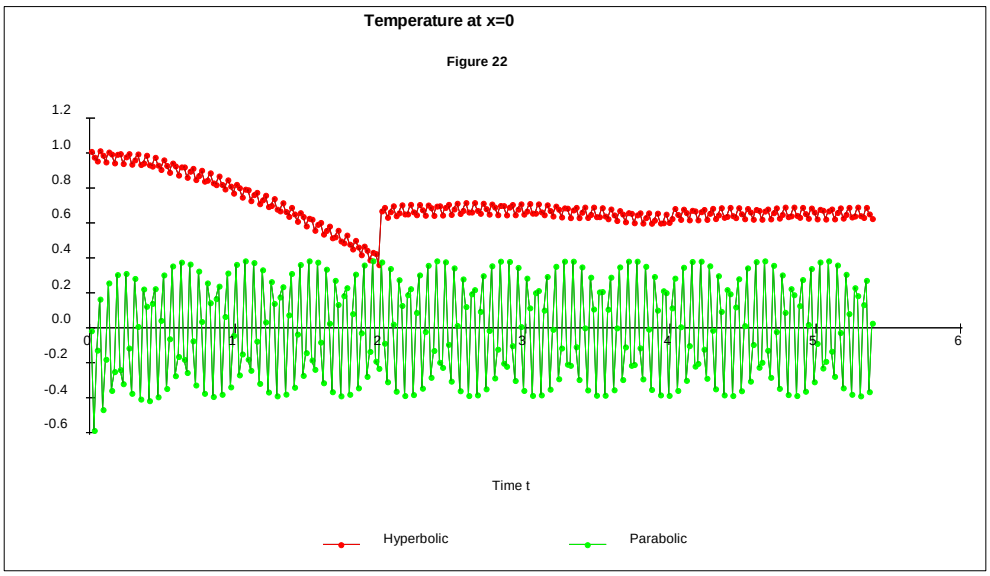

Hyperbolic vs Parabolic t=.5,omega=100
Figure 19
T
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Hyperbolic
Parabolic

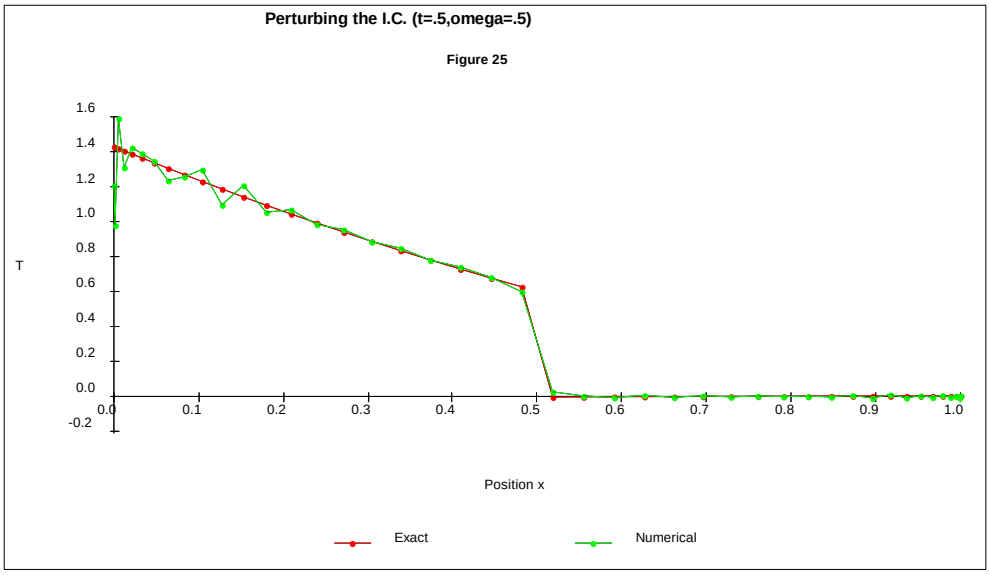

Temperature at x=0
Figure 20
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
0
1
2
3
4
5
6
Time t
Hyperbolic
Parabolic

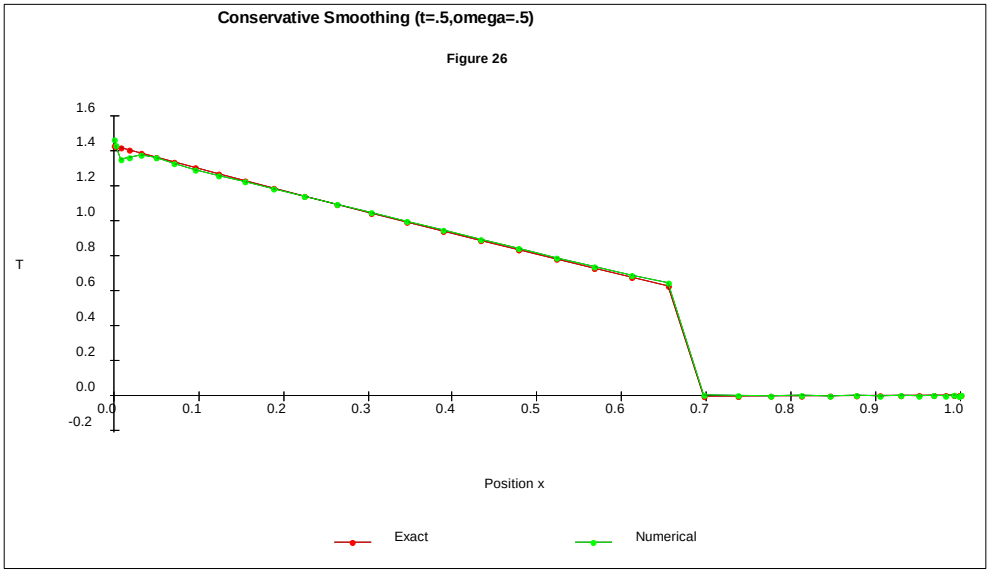

Temperature at x=0
Figure 21
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
0
1
2
3
4
5
6
Time t
Hyperbolic
Parabolic

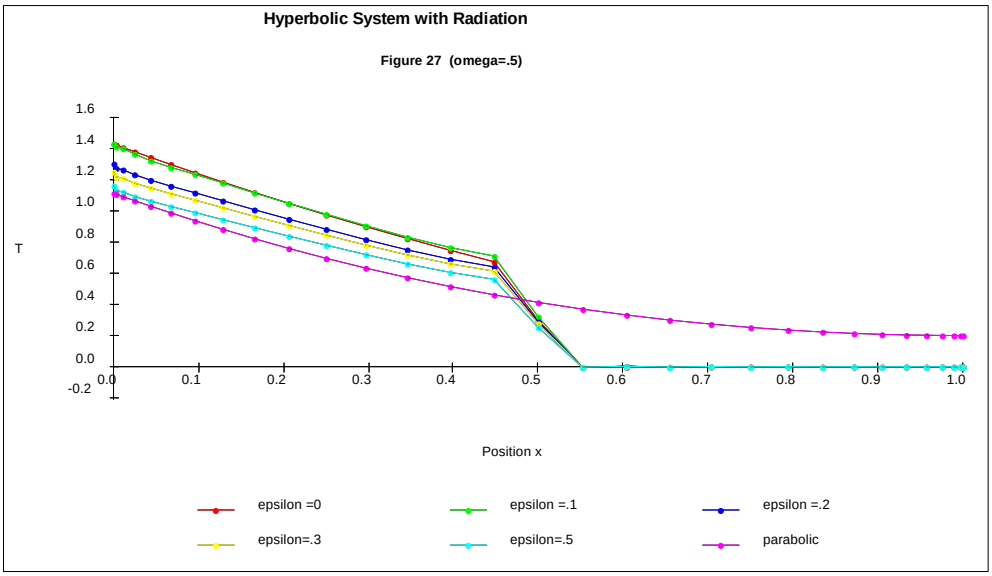

Temperature at x=0
Figure 22
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
0
1
2
3
4
5
6
Time t
Hyperbolic
Parabolic


Temperature at x=0
Figure 21
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
0
1
2
3
4
5
6
Time t
Hyperbolic
Parabolic


Temperature at x=0
Figure 22
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
0
1
2
3
4
5
6
Time t
Hyperbolic
Parabolic


Perturbing the I.C. (t=.5,omega=.5)
Figure 25
T
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Exact
Numerical


Conservative Smoothing (t=.5,omega=.5)
Figure 26
T
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
Exact
Numerical


Hyperbolic System with Radiation
Figure 27 (omega=.5)
T
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Position x
epsilon =0
epsilon =.1
epsilon =.2
epsilon=.3
epsilon=.5
parabolic