1-1-2005

Reduction of the Gibbs Phenomenon via Interpolation Using Chebyshev Polynomials, Filtering and Chebyshev-Pade' Approximations

Rob-Roy L. Mace

REDUCTION OF THE GIBBS PHENOMENON VIA INTERPOLATION USING CHEBYSHEV POLYNOMIALS, FILTERING AND CHEBYSHEV-PADÉ APPROXIMATIONS

Thesis submitted to

The Graduate College of

Marshall University

In partial fulfillment of the

Requirements for the degree of

Master of Arts

Department of Mathematics

by

Rob-Roy L. Mace

Dr. Scott A. Sarra, Committee Chairman

Dr. Bonita A. Lawrence

Dr. Ari Aluthge

Abstract

In this manuscript, we will examine several methods of interpolation, with an emphasis on Chebyshev polynomials and the removal of the Gibbs Phenomenon. Included as an appendix are the author's MatLab implementations of Lagrange, Chebyshev, and rational interpolation methods.

Acknowledgments

The author wishes to acknowledge the contributions of the following people:

Dr. Bonita Lawrence, without whom I would never have considered an advanced study in mathematics. She has, without a doubt, changed my life. You might say she brought me to my senses and opened the gate of education. Thank you.

Dr. Scott Sarra, for introducing me to this topic and giving me the tools necessary to discover on my own, but always being there to guide. Thank you for your seemingly infinite patience.

Dr. Ari Aluthge, for not only teaching me the required complex analysis, but his concern for my safety in regard to avoiding busses when navigating these dangerous Huntington streets. Thank you.

The entire faculty of the Marshall mathematics department.

My family. I have been blessed with the luck of being born into a family of educators, whose seemingly only goal at times was showing me as much of the world as possible. I love you all.

Contents

List of Tables

List of Figures

Part I

Chebyshev Approximation Methods

Chapter 1

Introduction

Definition 1 (Interpolation) *An interpolating approximation to a function $f(x)$ is an expression $P_{N-1}(x)$, usually an ordinary or trigonometric polynomial, whose N degrees of freedom are determined by the requirement that the interpolant agree with $f(x)$ at each of a set of N interpolation points:*

$$P_{N-1}(x_i) = f(x_i),\ i = 1, 2, \ldots, N.$$

A reasonable question that may arise at this point could be "Why would we want to interpolate?" A reasonable answer would be "Because we have to." When one starts working with numerical analysis or computer programming in general, it is soon learned that many numbers cannot be represented exactly due to the limitations of the computing languages, i.e., binary code. There just isn't enough memory to accurately represent 1/3 as a decimal.

So from the very start, we are faced with some amount of error. If we can keep the error smaller than some predetermined amount, or achieve as many decimal spaces of accuracy that the computer can handle, we will be satisfied. While computer precision varies, in this paper we will consider *machine epsilon* to be nearly 15 decimal places of accuracy. In other words, having the computer work with 15 decimal places of .3333..., is just as good as a million decimal places in our mind and the computer's. Thus, when we start approximating functions with our interpolation methods, if our error is of the size 10^{-15}, or so small that the computer will not know any different, we are doing a good job.

The next goal one might select would be to numerically approximate some function, be it of the nice smooth (boring) variety, or of the more interesting (badly behaved) discontinuous type. In this manuscript we will examine several types of functions, and compare several methods used to approximate them. We will also see exactly how to compare one method to another in terms of which works "better." As it turns out, there is work to be done. We soon see that problems arise in our interpolating methods, such as the Gibbs phenomenon. It is the main focus in this manuscript to discuss ways of combatting such a complication.

Our main tool of choice for approximating functions is interpolation, which was defined above. As for methods of interpolation for some function f, the simplest we could examine would be a linear interpolation method using two interpolating points x_0 and x_1. This would give us the approximation of some function f as

$$f(x) \approx \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1).$$

Clearly, we could do much better. For quadratic interpolation, by specifying three points (instead of two as in the linear case) we could approximate f by a quadratic polynomial, $P_2(x)$. With quadratic interpolation, we will select the three interpolation points x_0, x_1, and x_2 where

$$P_2(x_0) = f(x_0),\ P_2(x_1) = f(x_1),\ P_2(x_2) = f(x_2)$$

giving us

$$\begin{aligned} P_2(x) \ \equiv\ & \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) \\ & + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2). \end{aligned}$$

So in general, we could continue the above method, fitting any $N+1$ points by a polynomial of $N-th$ degree using the Lagrange Interpolating Formula:

$$P_N(x) = \sum_{i=0}^{N} f(x_i) C_i(x),$$

where $C_i(x)$, the "cardinal functions," are polynomials of $N-th$ degree which satisfy the conditions

$$C_i(x_j) = \delta_{ij},$$

where δ_{ij} is the Kronecker δ-function, and

$$C_i(x) = \prod_{j=0, j\neq i}^{N} \frac{x - x_j}{x_i - x_j}.$$

For now, as a general notion, we define the error of an interpolation method to be

$$Er = |P_N(x_i) - f(x_i)|.$$

The above is a point-wise error evaluation that tells us how far away our approximation is from the exact function we are interpolating at each point x_i. Our goal when approximating is to minimize the maximum value of Er, thus such interpolation procedures are usually referred to as "minimax" approximations.

We would expect(or at least hope) $Er \to 0$ as $N \to \infty$. That is to say, as we interpolate with more and more points, we would like our approximation method to become better and better. However, Runge[1] shows this assumption to be untrue. Selecting the function

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5, 5] \tag{1.1}$$

Runge proved that interpolation with evenly spaced points, x_i, converges only within the interval $|x| < 3.63$ and diverges for larger values of x.

Using Matlab, let us visually examine what Runge proved.

As seen in Figure 1.1, as N increases, P_N's oscillations grow wilder. This is a harbinger of things to come. So at this point, we will examine another possibility for interpolation by examining other possible series representations for some function $f(x)$ (or when working with approximating Partial Differential Equation solutions, denoted $u(x)$).

For example, note the improvement achieved using the unequally spaced Chebyshev Gauss-Lobatto points x_j,

$$x_j = -\cos\left(\frac{j\pi}{N}\right) \quad j = 0, 1, \ldots, N,$$

to the Runge function in Figure 1.2. Using just sixteen interpolation points we have removed the oscillations near the endpoints.

In practice, we assume that we can represent a function/solution by a global, interpolating partial sum of the form

$$u_N(x) = \sum_{k=0}^{N} a_k \phi_k(x).$$

The function ϕ can be chosen for various reasons, such as giving a lower error for the approximation of u due to its nature.

When taking ϕ to be the Chebyshev polynomials, which are described in much greater detail later in this paper, we say that we are using a "Chebyshev approximation method." When approximating a solution in regards to numerical PDEs, we term it the "Chebyshev pseudospectral (CPS) method." Given that there are many choices for ϕ, one would want to discover which method works best. To answer that problem, we will now put more detail into what exactly that means.

We want our numerical approximations to have a low error, and to converge quickly. Thus, to be a touch more specific, we shall define rate of convergence[2].

Definition 2 (Rate of Convergence) *Suppose some sequence* $\{\alpha_n\}_{n=1}^{\infty}$

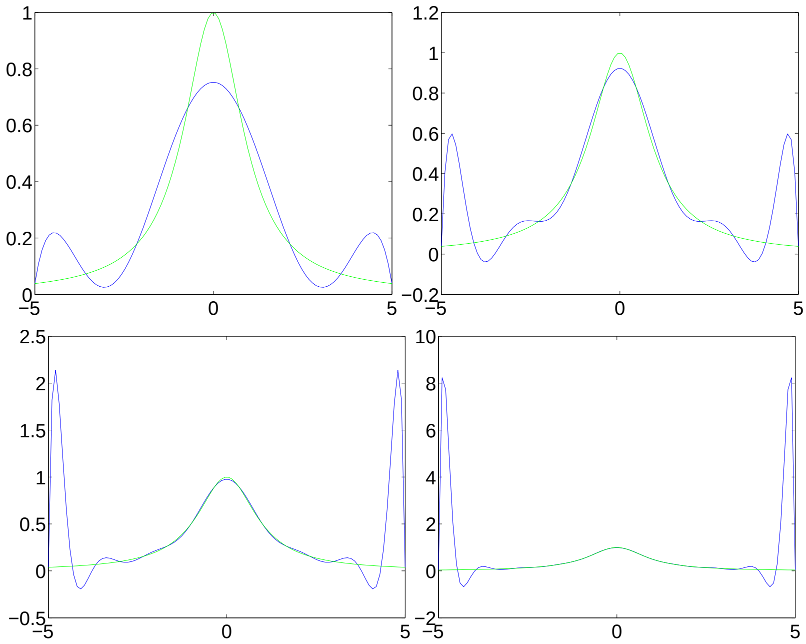

Figure 1.1: Graphs for Lagrange Interpolation of $f(x) = 1/(1 + x^2)$. Top Left: $N = 8$ Top Right: $N = 12$ Bottom Left: $N = 16$ Bottom Right: $N = 20$

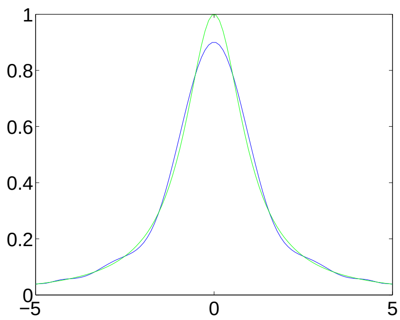

Figure 1.2: Approximation of Runge function with $N = 16$ at the unequally spaced CGL points.

converges to some number x. If $\exists$ some $K > 0 \ni$

$$|\alpha_n - x| \leq K\left(\frac{1}{n^p}\right),$$

for large n, then we say that $\{\alpha_n\}_{n=1}^{\infty}$ converges to x with ***rate of convergence*** *$\mathcal{O}(\frac{1}{n^p})$.*

In general, we are concerned with finding the largest value of p where $\alpha_n = x + O(1/n^p)$.

Now that we have defined the rate of convergence, we will define several names often used to describe the types of convergence[1].

Definition 3 (Algebraic Index of Convergence) *The* ***algebraic index of convergence k*** *is the largest number for which*

$$\lim_{n\to\infty} |a_n| n^k < \infty, \quad n \gg 1;$$

where a_n represents the coefficients of some series.

If the algebraic index is unbounded, that is if the coefficients a_n are decreasing faster than $(1/n)^k$ for any finite k, then we say that the series has exponential (spectral) convergence.

A series whose coefficients have exponential convergence can then be classified as having supergeometric, geometric, or subgeometric convergence.

Definition 4 (Rates of Exponential Convergence) *A series with coefficients a_n has the property of supergeometric, geometric, or subgeo-*

metric convergence depending upon

$$\lim_{n\to\infty} \frac{\log(|a_n|)}{n} = \begin{cases} \infty & \textit{supergeometric} \\ \textit{constant} & \textit{geometric} \\ 0 & \textit{subgeometric.} \end{cases}$$

The Chebyshev series of entire functions have supergeometric convergence(Figure 2.1). For functions with branch points or poles which are a finite distance off the expansion interval, geometric convergence is normal(Figure 2.2). We can expect to see subgeometric convergence for series on infinite or semi-infinite intervals.

Now armed with the ability to determine how our selected method performs, we move to better associate ourselves with this powerful weapon we have.

Chapter 2

Chebyshev Approximation

Definition 5 (Chebyshev Polynomial) *The Chebyshev polynomial of degree n, denoted $T_n(x)$ is defined as*

$$T_n(x) = \cos(n \arccos x) \; \forall \quad x \in [-1, 1]$$

or,

$$T_n(x) = \cos(n\theta) \quad x = \cos\theta \quad \forall \quad \theta \in [-\pi, \pi].$$

Definition 6 (Infinite Continuous Chebyshev Series Expansion) *The* infinite continuous Chebyshev series expansion *is*

$$f(x) \approx {\sum_{n=0}^{\infty}}{}' \alpha_n T_n(x),$$

where

$$\alpha_n = \frac{2}{\pi} \int_{-1}^{1} (1 - x^2)^{-1/2} f(x) T_n(x) dx.$$

Again, as our aims are founded in numerical computations, we will use a truncation of the above series:

$$S_N(x) = \sum_{n=0}^{N}{}' \alpha_n T_n(x).$$

While maybe not quite obvious at first glance, after inspection we see that a Chebyshev polynomial expansion,

$$f(z) = \sum_{n=0}^{\infty} a_n T_n(z),$$

is actually a Fourier cosine series. For non-periodic functions, we can expect exponential convergence. Due to the transform of $z = \cos\theta$, even if $f(z)$ is not periodic, $f(\cos\theta)$ will be periodic. Thus, if we were to vary θ over all real values, z would just oscillate between -1 and 1. Because $f(\cos\theta)$ is periodic, its Fourier series must have exponential convergence, unless $f(z)$ is singular for $z \in [-1, 1]$. The exponential convergence of Fourier series implies equally fast convergence of Chebyshev series since sums are term by term identical.

Also, we shall note that the function $\cos\theta$ is symmetric about $\theta = 0$. Because of this, $f(\cos\theta)$ is also symmetric in θ, even if there is no symmetry for f with respect to z.

If we desire to approximate on an interval other than $[-1, 1]$, we can apply a change of variable. Suppose we wanted to approximate for $y \in [a, b]$. With the change of variable

$$x \equiv \frac{2y - (b + a)}{b - a}$$

we have moved the approximation to the interval $x \in [-1, 1]$.

What can we expect for the error of this method? The following theorems [1] should help us find an answer.

Theorem 1 (Chebyshev Truncation Theorem) *The error in approximating $f(z)$ by the sum of its first N terms is bounded by the sum of the absolute values of all the neglected coefficients. If*

$$f_n(z) \equiv \sum_{n=0}^{N} a_n T_n(z),$$

then

$$E_T(N) \equiv |f(z) - f_N(z)| \leq \sum_{n=N+1}^{\infty} |a_n| \;\forall\; f(z), N, and \;\; z \in [-1, 1]$$

PROOF: We see that from the definition of Chebyshev polynomials,

$$|T_n(z)| \leq 1$$

for all $z \in [-1, 1]$ and for all n. Thus,

$$|a_n(z)||T_n(z)| \leq |a_n(z)|.$$

And we have

$$\begin{aligned} E_T(n) &\equiv |f_n(x) - f_N(x)| \\ &\leq \sum_{n=N+1}^{\infty} |a_n(x)|. \end{aligned}$$

Theorem 2 (Chebyshev Rate of Convergence) *The asymptotic rate* *of convergence of a Chebyshev series for $z \in [-1, 1]$ is equal to μ, the quasi-radial coordinate of the ellipse of convergence. This is related to the location of the convergence limiting singularity at (x_o, y_o) in the complex plane via*

$$\begin{aligned} \mu &= Im\{\arccos[x_o + \imath y_o]\} \\ &= \log|z_o \pm ({z_o}^2 - 1)^{\frac{1}{2}}| \\ &= \log(\alpha + \sqrt{\alpha^2 - 1}) \end{aligned}$$

where the sign in the second line is chosen to make the argument of the logarithm greater than one, so that μ is positive and where

$$\alpha \equiv \frac{1}{2}\sqrt{(x_o + 1)^2 + y_o^2} + \frac{1}{2}\sqrt{(x_o - 1)^2 + y_o^2}.$$

Note that our Chebyshev approximation method will not be evaluated at evenly spaced points on our interval. Instead, it is evaluated at the grid points

$$x_j = -\cos\left(\frac{j\pi}{N}\right)$$

for $j = 0, 1, \ldots, n$. This is known as the "Gauss-Lobatto" grid. To see why one would do such a thing, we will look at the following theorems [1].

Theorem 3 (Interpolation by Quadrature) *Let $P_N(x)$ denote the polynomial of degree N which interpolates to a function $f(x)$ at the $(N + 1)$ Gaussian quadrature points associated with a set of orthogonal* *polynomials $\phi_n(x)$:*

$$P_N(x) = \sum_{n=0}^{N} a_n \phi_n(x) \qquad i = 0, 1, \ldots, N. \tag{2.1}$$

$P_N(x)$ may be expanded without error as the sum of the first $(N+1)$ $\phi_n(x)$ because it is merely a polynomial of degree N. The coefficients a_n of this expansion (2.1) are given without approximation by the discrete inner product

$$a_n = \frac{(f, \phi_n)_G}{(\phi_n, \phi_n)_G}.$$

Theorem 4 (Chebyshev Interpolation and Error Bound) *Let the Gauss-Lobatto (Chebyshev extrema) grid x_j be given by*

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

Let the polynomial $P_N(x)$ which interpolates to $f(x)$ at these grid points be

$$P_N(x) = \sum_{n=0}^{N}{}'' b_n T_n(x)$$

where the $('')$ means the first and last terms are to be taken with a factor of $(1/2)$. The coefficients of the interpolating polynomial are given by

$$b_n = \frac{2}{N} \sum_{j=0}^{N}{}'' f(x_j) T_n(x_j).$$

Let α_n denote the exact spectral coefficients of $f(x)$, such that

$$f(x) = \sum_{n=0}^{\infty}{}' \alpha_n T_n(x).$$

Therefore, the coefficients of the interpolating polynomials are related to those of $f(x)$ by

$$b_n = \alpha_n + \sum_{i=0}^{\infty}(\alpha_{n+2jN} + \alpha_{-n+2jN}).$$

For all N and all real $x \in [-1, 1]$ the error in the interpolating polynomial is bounded by twice the sum of the absolute value of all the neglected coefficients:

$$|f(x) - P_N(x)| \leq \ 2 \sum_{n=N+1}^{\infty} |\alpha_n|.$$

To test the accuracy and to demonstrate the convergence of our approximation methods in this paper we will also use the following test functions:

1. $f_1(x) = |x|$

2. $f_2(x) = \exp(\cos(x^3 + 1))$

3. $f_3(x) = \begin{cases} 1 & for \quad x \geq 0 \\ -1 & for \quad x < 0. \end{cases}$

4. $f_4(x) = \begin{cases} 0 & for \quad 0 < x < .25 \;\; or \;\; .75 < x \leq 1, \\ 1 & for \quad .25 \leq x \leq .75, \\ e^{-400(x+.5)^2} & for \quad -1 \leq x \leq 0. \end{cases}$

We will now show results achieved implementing Chebyshev approximation using the software MatLab. First, we will use an entire function f_2 2. The entire function has no discontinuities and is infinitely differentiable. Therefore, we should expect supergeometric convergence(Figure 2.1).

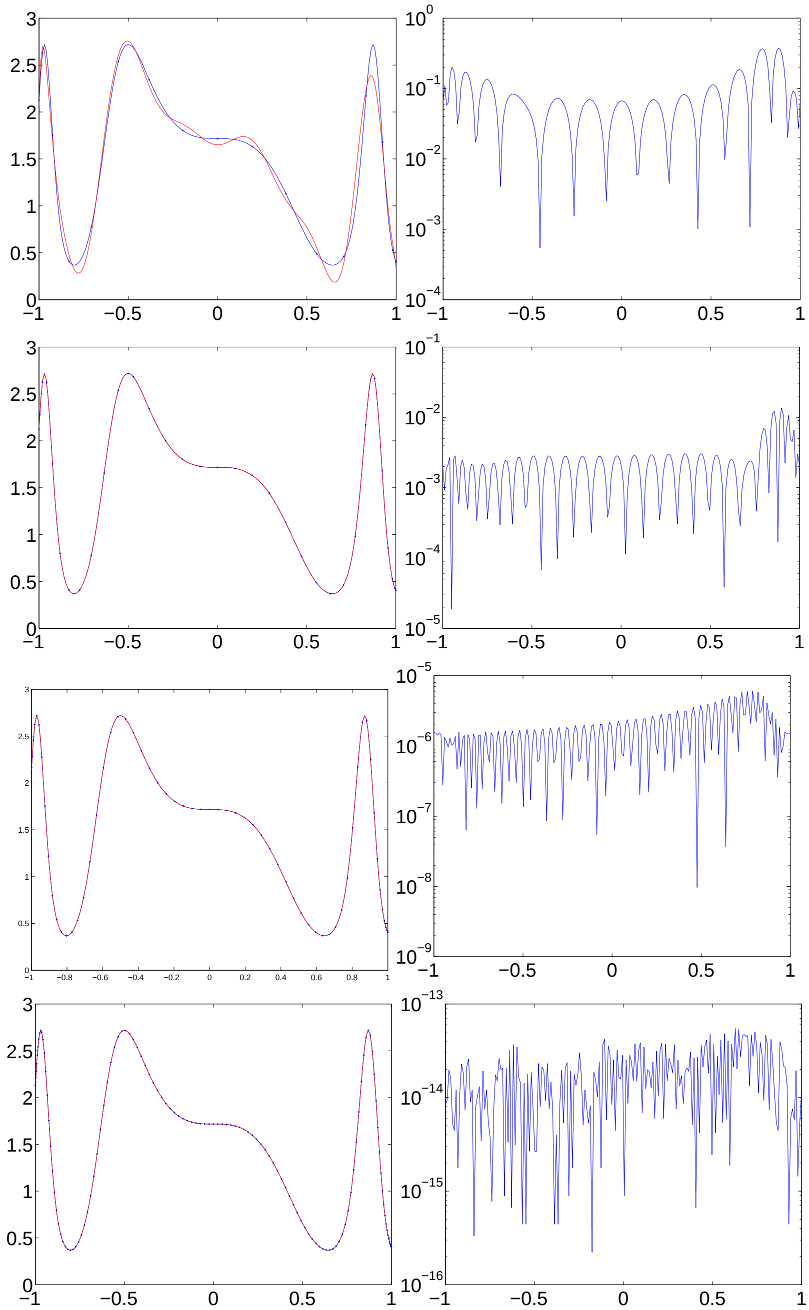

Figure 2.1: Left: From top, we show the Chebyshev approximation of entire function f_2 with $N = 16$, $N = 32$, $N = 64$ and $N = 128$. Right: Error graph. Note the increase of accuracy from graph to graph. As N is doubled, the accuracy is more than doubled. This displays the exponential accuracy we expected for this type of function.

Next, we will use the function from the Runge example (1.1) on the interval $[-1, 1]$. We could approximate on any interval, such as $[-5, 5]$, with a change of variable. This function is smooth, but has poles at the end of the interval at $\pm\imath$, which should make the convergence a bit slower than we had for the entire function f_2.

For f_1, there is a discontinuity in the first derivative at the point $x = 0$. This causes an increase in error near that point as seen in Figure 2.3.

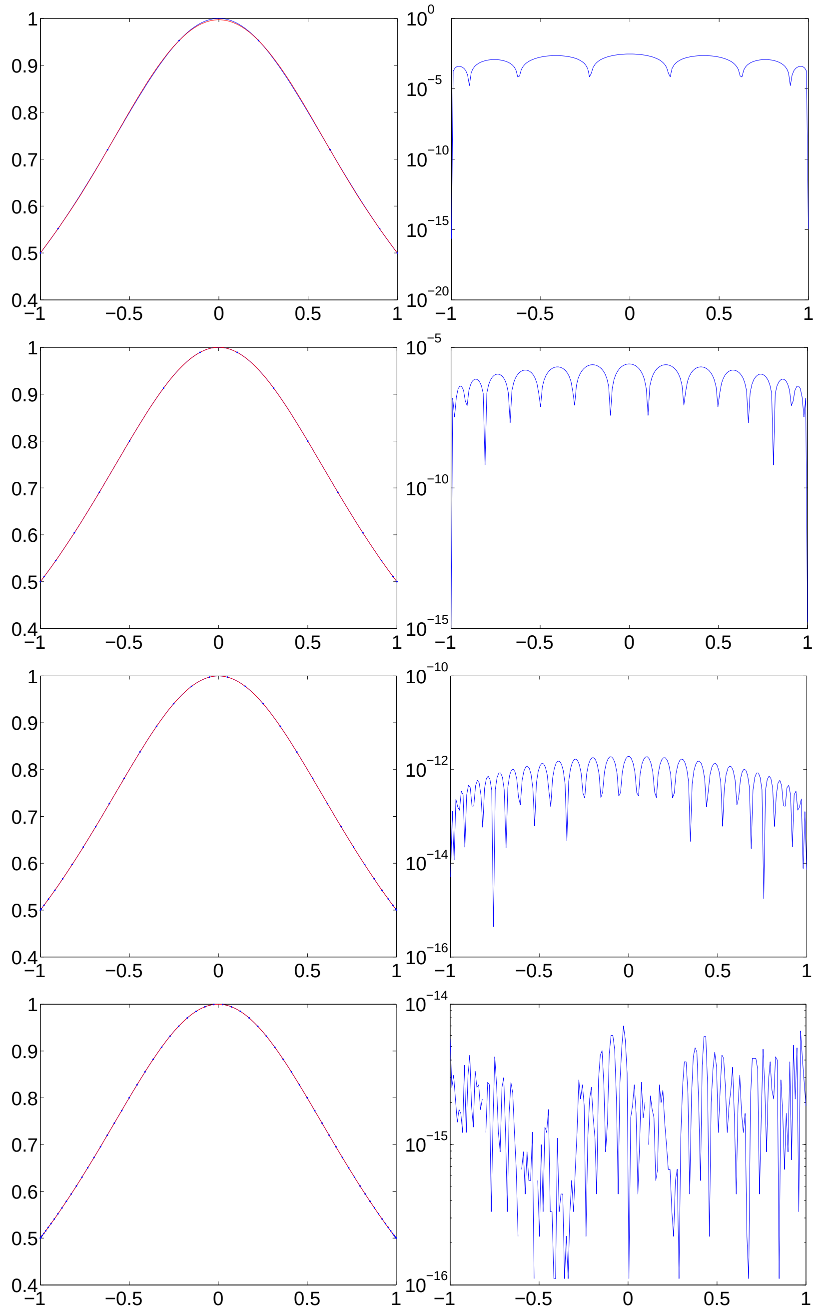

Figure 2.2: Graphs for Chebyshev Approximation of $f(x) = 1/(1 + x^2)$ and error resulting from the method. From top we are using $N = 8$, $N = 16$, $N = 32$, and $N = 64$. It is hard to distinguish the graphs of the approximation from the exact in the figures on the left, but if we look at the error plots, we can see the spectral accuracy. As N is doubled, say from $N = 8$ to $N = 16$, the number of accurate digits in our approximation is doubling.

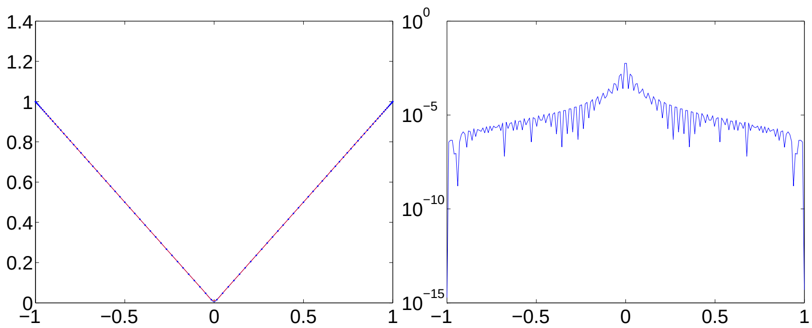

Figure 2.3: Chebyshev approximation of f_1 (2) with $N = 99$ The error is around 10^{-6} away from $x = 0$ and about 10^{-2} at that point, where the first derivative does not exist.

Part II

The Gibbs Phenomenon and its Resolution

Chapter 3

The Gibbs Phenomenon

As we have shown, the Chebyshev Pseudospectral method is spectrally accurate for smooth solutions. Also, we saw how the accuracy of the method is severely degraded by discontinuities in the function's derivatives. Unfortunately, in many applications discontinuities exist, such as fluid flows that contain shock or rarefaction waves. What happens when we apply our CPS method to a discontinuous function?

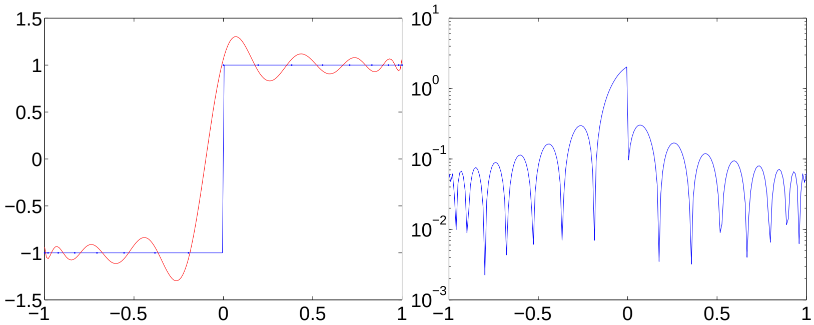

Figure 3.1: Left: Chebyshev approximation with $N = 16$ of sign function. Right: Error graph.

As is clearly evident in Figure 3.1, there are oscillations in the graph of our approximation to the discontinuous sign function. The oscillations grow wilder near the discontinuities. We call these oscillations the

Gibbs phenomenon.

The first thought may be, "what if we just increase N?" Unfortunately, the Gibbs phenomenon is not so easily defeated. Let us look at the graphs, Figure 3.2, of this same sign function with higher values of N. Note that as N increases, the magnitude of the oscillations does not decrease at the discontinuity, but the width of the effected region is reduced.

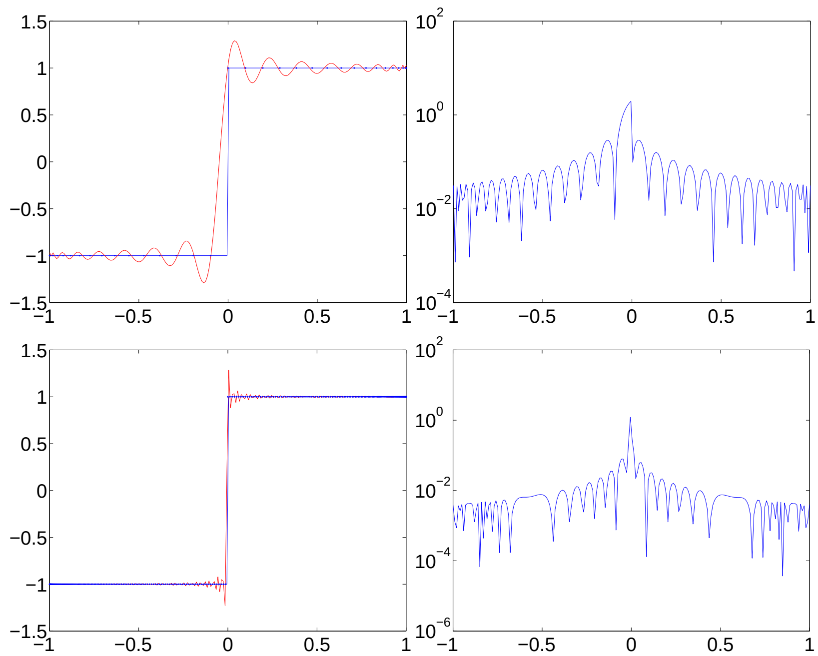

Figure 3.2: Left: From top, Chebyshev approximation with $N = 32$ and $N = 256$ of sign function. Right: Error graphs. Note the reduction of width of affected region but no reduction in magnitude of oscillations.

Another consequence of the Gibbs phenomenon, notable in Figure 3.2, is the lack of convergence at the discontinuity. The overshoot at such a break is approximately 9 percent of the jump size. We can also expect a global $\mathcal{O}(N^{-1})$ convergence rate in mean, and a steepness of the approximation right at the jump being proportional to the length, N, of the polynomial expansion [6].

It seems a reasonable question at this point would be, “Can we get rid of these oscillations in our approximations?” We will see that there are many possible solutions, but some work better than others.

Chapter 4

Removal Overview

The name Gibbs phenomenon was first used by Bôcher in 1906, but the efforts to remove it started over 100 years ago in 1898 when Michelson and Stratton built their harmonic analyzer. It was a mechanical device appealing to Hooke's law that used springs to store the Fourier coefficients of a given curve. A paper they published showed what would later become known as the Gibbs phenomenon in their efforts to reconstruct a square wave function.

It is seemingly unknown how J. Willard Gibbs started working on such a task, but his first writings on the matter in a letter to the editor of *Nature* in December 29, 1898 were not exactly correct. While the letter described the oscillations, he seemingly implied that increasing N would decay the oscillations. It was only a few months later that he published a correction, noting that the oscillations do not decay, but actually that the overshoot tends to a fixed number.

There are two classes of resolutions [4] to the Gibbs phenomenon: the first modify expansion coefficients of the approximation in the Fourier space, while the second treats the approximation in the physical space. Alas, except for the Gegenbauer Reconstruction Procedure, both classes improve accuracy only away from discontinuities.

The Hungarian mathematician Fejér is credited for the first attempt at resolving the Gibbs phenomenon in 1900. His method was equivalent to using what is known as a first-order filter. Now, motivated by signal analysis, mathematicians have developed many techniques known as *filtering*. The goal of employing filters is to increase the decay rate of an approximation's coefficients.

Definition 7 (Filtered Chebyshev Approximation) *The* filtered Chebyshev approximation *is defined as*

$$F_N(x) = \sum_{n=0}^{N} \sigma\left(\frac{n}{N}\right) a_n T_n(x),$$

where σ is a spectral filter.

Definition 8 (Spectral Filter) *[9] A pth order spectral filter ($p > 1$) is a sufficiently smooth function satisfying*

$$(i)\ \sigma(0) = 1,$$
$$(ii)\ \sigma^{(m)} = 0, \quad m = 1, 2, \ldots, p-1,$$
$$(iii)\ \sigma^{(0)} = 1, \quad m = 0, 1, \ldots, p-1.$$

How the function behaves away from the discontinuity and the order of filter used in the approximation determine the convergence rate of the filtered approximation.

In Figure 4.1, we have a Chebyshev approximation with $N = 99$ to the sign function. Note the oscillations of the Gibbs phenomenon, and that the error away from the discontinuity approaches 10^{-2}.

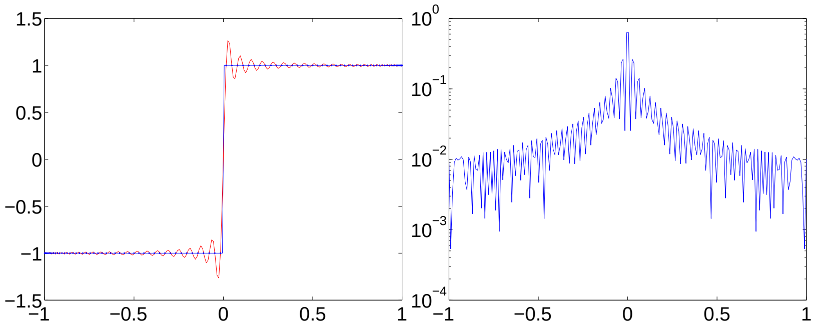

Figure 4.1: Chebyshev approx with $N = 99$ of sign function

Now, if we make a simple change by inserting a *spectral filter* of order $p = 4$, we will be able to achieve an error of machine epsilon away from the discontinuity, as in Figure 4.2.

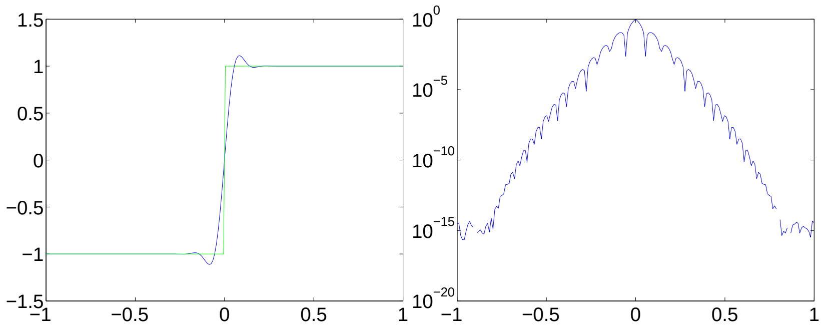

Figure 4.2: Filtered Chebyshev approximation of order $p = 4$ to sign function, with $N = 99$. Note the increase of accuracy and reduction of oscillations as compared to Figure 4.1, which used the same N. Clearly the filter is efffective.

The increase of nearly 13 decimal places of accuracy came from simply adding an exponential filter. That is, transforming the original partial sum of

$$f_n(z) \equiv \sum_{n=0}^{N} a_n T_n(z),$$

into

$$F_N(z) \equiv \sum_{n=0}^{N} \sigma\left(\frac{n}{N}\right) a_n T_n(z),$$

where σ is a spectral filter of order p. In our MatLab code we are implementing an exponential filter, that is

$$\sigma(\eta) = exp(-\alpha|\eta|^p),$$

where p is even, and $\alpha = ln\varepsilon$ for $\varepsilon =$ machine epsilon.

If we change the order of the filter used in the approximation, we get a different behavior at the discontinuity. In Figure 4.3, we see a slight rounding at the discontinuity when using a filter of order $p = 2$. We will see in Figure 4.4 that using a filter of order $p = 12$ gives us an overshoot at the discontinuity.

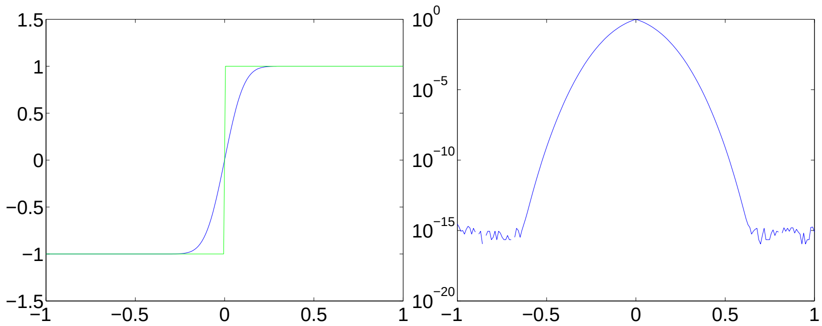

Figure 4.3: Filtered Chebyshev approximation using filter of order $p = 2$. Note the difference in error as compared with Figure 4.2. The filter of order $p = 2$ gives the approximation a bit of rounding at the discontinuity.

Below are results achieved approximating the other various test functions.

For the “super test function” f_4, we can see some improvements using the filtered Chebyshev approximation as compared to the unfiltered, but

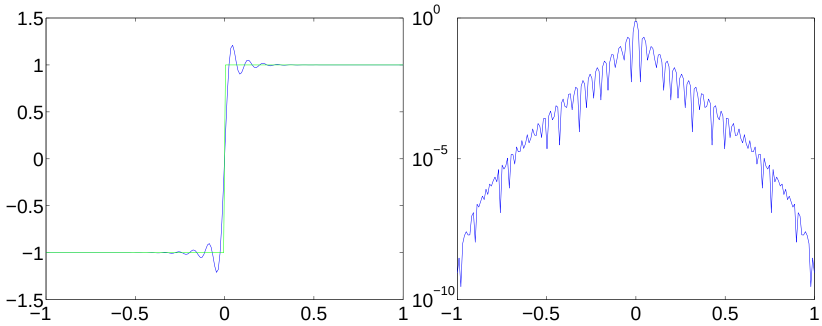

Figure 4.4: Filtered Chebyshev approximation using filter of order $p = 12$. Note the difference in error as compared with Figure 4.3. The filter of order $p = 12$ gives the approximation an overshoot at the discontinuity.

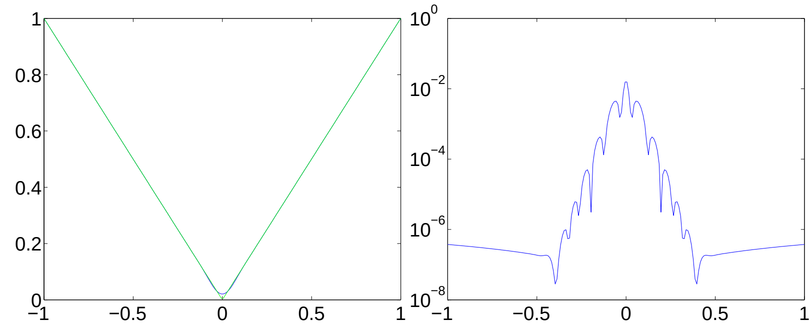

Figure 4.5: Filtered Chebyshev approximation using filter of order $p = 4$. Note the difference in error as compared with Figure 2.3. There is still a spike in error near the point $x = 0$, where the first derivative does not exist. The reduction in error does not seem as impressive though, improving only slightly.

there are still oscillations, and we have severe rounding at the top of the sharp spike. Note Figure 4.6.

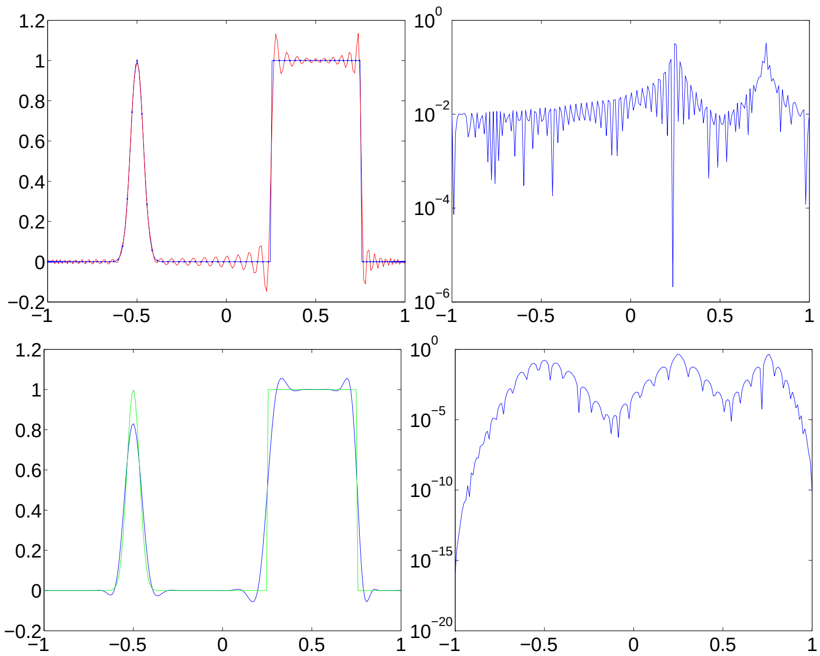

Figure 4.6: Left: From top is an unfiltered Chebyshev approximation to f_4 with $N = 99$, and a filtered Chebyshev approximation using filter of order $p = 4$. Right: Error graphs. Note the clear oscillations of the Gibbs phenomenon in the unfiltered graph. The filtered approximation decreases these oscillations, but we are still left with a bit of rounding at the discontinuity and the top of the sharp spike is severely rounded.

There are many other methods for removal of the Gibbs phenomenon, such as spectral mollification, Gegenbauer Reconstruction Procedure, and digital total variation filtering.

Spectral mollification involves applying a two parameter family of filter to the physical space interpolant. It can recover spectral accuracy outside a neighborhood of the discontinuity, and may or may not incorporate edge detection. The method does give a bit of smearing at the discontinuity.

The Gegenbauer Reconstruction Procedure (GRP) is capable of recovering spectral accuracy up to the discontinuity, but it must know the exact location of the edges. The method is computationally expensive, and prone to round-off errors.

Digital total variation filtering was designed to create clear images in pictures that are affected by random noise. The method begins with a discrete variational problem, using data on a general discrete domain. When the GRP is too difficult to implement, DTV filtering is a more computationally efficient choice for post-processing. Applying this process to the Gibbs phenomenon, it has been shown that a DTV filter can give a sharp resolution at discontinuities and an accelerated convergence overall, without the knowledge of the discontinuities [9].

At this point, we may ask the question "Is there a method to sharply resolve the discontinuities, unlike the smearing effect of filters that does not need the edge detection of a GRP?" This leads us to the Chebyshev rational approximation.

Part III

Rational Approximation Methods

Chapter 5

Rational Approximation

While there are advantages to using algebraic polynomials, such as ease of evaluation, derivation, and integration, polynomial approximations tend to create problems in the error bounds due to their oscillations. Also, polynomial approximations might not be the fastest methods in terms of convergence. We can see this from Markov's inequality, which states that on $[-1,1]$

$$\| p_n' \|_\infty \leq n^2 \| p \|_\infty$$

for any polynomial p_n of degree $\leq n$. It follows that no function f with its derivative larger than $n^2 \| f \|_\infty$ at some point can be approximated very well by a polynomial of degree n. This leads us to our attempts of approximating functions with a rational approximation. The following methods involving rational functions are designed to distribute error over the interval on which we are approximating. Note there is no convergence theory for rational approximations in this manuscript because the convergence theory does not exist. We will accept numerical results displaying convergence as an indication to how well the method works.

Definition 9 (Rational Function) *A **rational function** r of degree N has the form*

$$r(x) = \frac{p(x)}{q(x)},$$

where $p(x)$ and $q(x)$ are polynomials whose degrees sum to N.

We can see that a rational function where $q(x) \equiv 1$ is simply a polynomial function. Therefore, we expect to have similar results when approximating. It is when we approximate with the degree of the numerator and denominator close or equal to each other that the approximation results are more successful than polynomial methods. As we will see, polynomial approximations are also better suited when approximating discontinuous functions.

If we were approximating some function on an interval containing zero with

$$r(x) = \frac{p(x)}{q(x)} = \frac{p_o + p_1 x + \cdots + p_n x^n}{q_o + q_1 x + \cdots + q_m x^m},$$

the rational function of degree $n+m$, we must have $q_o \neq 0$ to ensure that r is defined. In practice, we will assume $q_o = 1$, or replace $p(x)$ by $p(x)/q_o$ and $q(x)$ by $q(x)/q_o$. This gives us $N+1$ parameters $q_1, q_2, \ldots, q_m, p_o, p_1, \ldots, p_n$ for approximation.

From [3], we learn that a general approach is to assume a formal series expansion of f in terms of ϕ_k, where ϕ_k is any function that satisfies the relation

$$\phi_i \phi_j = \sum_k A_{ijk} \phi_k.$$

The rational function U_m/V_n is obtained by equating the leading terms in the series expansion of

$$V_n f - U_m,$$

to zero. This process leads us to calculations where Chebyshev polynomials lend themselves as a top candidate for ϕ, because of their relation that

$$T_i T_k = \frac{1}{2}(T_{i+k} + T_{|i-k|}).$$

Definition 10 (Padé Approximation) *A Padé approximating function is a rational function*

$$R_{mk}(x) = \frac{P_m(x)}{Q_k(x)}$$

where

$$P_m(x) = \sum_{j=0}^{n} p_j x^j$$

$$Q_k(x) = \sum_{j=0}^{k} q_j x^j \quad \textit{and} \quad q_o = 1$$

with $m + k + 1$ coefficients p_j, q_j are chosen so that $R_{mk}(x)$ agrees with the approximated function and as many derivatives as possible at the point $x = \alpha$.

In practice, we will assume $\alpha = 0$, which can be achieved by a change of variable. To determine coefficients a_j, b_j, expand approximated function, f, in terms of its Maclaurin series.

So we would take

$$f(x) = \sum_{j=0}^{\infty} c_j x^j \qquad c_j = f^{(j)}(0)/j!$$

Then

$$f(x) - R_{mk}(x) = \sum_{j=0}^{\infty} c_j x^j - \frac{\sum_{j=0}^{m} p_j x^j}{\sum_{j=0}^{k} q_j x^j}$$

can be written as

$$f(x) - R_{mk}(x) = \frac{\sum_{j=0}^{\infty} c_j x^j \sum_{j=0}^{k} q_j x^j - \sum_{j=0}^{m} p_j x^j}{\sum_{j=0}^{k} q_j x^j}.$$

For example, suppose we wanted to approximate the function $f(x) = \log(1 + x)$. The Taylor series of f gives

$$\log(1+x) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^k}{k}.$$

Then we have that

$$p_o = 0,$$

$$p_1 = q_o,$$

$$p_2 = -\frac{1}{2} q_0 + q_1,$$

$$-\frac{1}{4} q_o + \frac{1}{3} q_1 - \frac{1}{2} q_2 = 0.$$

Solving the above system, we find that

$$f(x) \approx R(x) = \frac{x + \frac{1}{2}x^2}{1 + x + \frac{1}{6}x^2}.$$

Graphically, the approximations look like Figure 5.1.

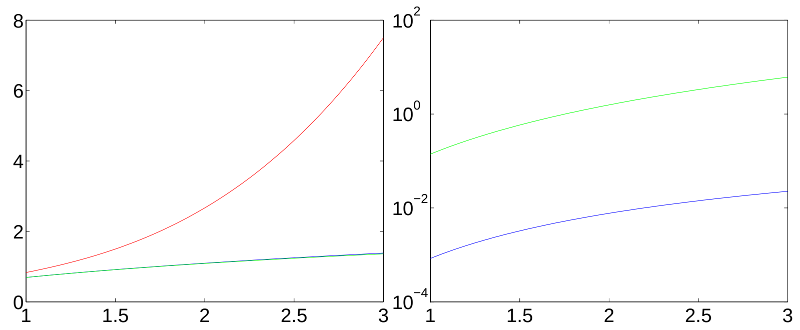

Figure 5.1: Left: Graph of $f(x) = log(1 + x)$, its Taylor series approximation and its rational approximation R(x). The graphs clearly show the divergence of the Taylor series, while the rational approximation performs much better. Right: Error graph

We took the Taylor approximations only to three terms, because as the graph shows, it quickly starts diverging. The graph would diverge even faster the more terms we would add. While the above rational method is fairly successful with an error around 10^{-2}, it certainly isn't giving us our machine epsilon. Thus, we would like to find a better method. We discover that we obtain more uniformly accurate approximations when replacing the x^k term by the kth-degree Chebyshev polynomial $T_k(x)$.

Definition 11 (Chebyshev Rational Approximation) *A **Chebyshev rational approximation** [3]is an N-th degree rational function $r_T(x)$ written in the form*

$$r_T(x) = \frac{\sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)} = \frac{P_n}{Q_m},$$

where $N = n + m$, $q_o = 1$, and T_k is the k-th Chebyshev polynomial.

Suppose we are to approximate some function $f(x)$. We will first write $f(x)$ as a series of Chebyshev polynomials, that is

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x).$$

We then have

$$f(x) - r_T(x) = \sum_{k=0}^{\infty} a_k T_k(x) - \frac{\sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)}$$

or

$$f(x) - r_T(x) = \frac{\sum_{k=0}^{\infty} a_k T_k(x) \sum_{k=0}^{m} q_k T_k(x) - \sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)}.$$

We will choose the $q_1, q_2, \ldots, q_m$ and $p_o, p_1, \ldots, p_n$ coefficients in such a way that there are no terms of degree less than or equal to N in the numerator.

Notice that the approximation as defined has us calculating the product of Chebyshev polynomials. Fortunately, we can implement the relationship

$$T_i(x)T_j(x) = \frac{1}{2}[T_{i+j}(x) + T_{|i-j|}(x)].$$

Another botheration arises when computing the Chebyshev series for $f(x)$. The integrals can rarely be evaluated in closed form, thus we must use a numerical integration technique for each evaluation.

We will now look at results achieved with MatLab code approximating various functions with the Chebyshev Rational method. In

Figure 5.2, Figure 5.3, and Figure 5.4 we are approximating the sign function with fixed numerator degree higher than various denominator degrees. We will also try approximating with a fixed denominator degree and various numerator degrees in Figure 5.5, Figure 5.6,and Figure 5.7.

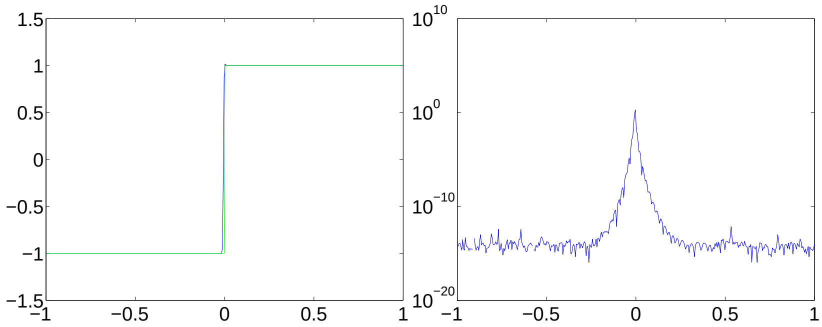

Figure 5.2: Left: Graph of sign function and its Chebyshev rational approximation with $n = 99$ and $m = 48$. Right: Error graph

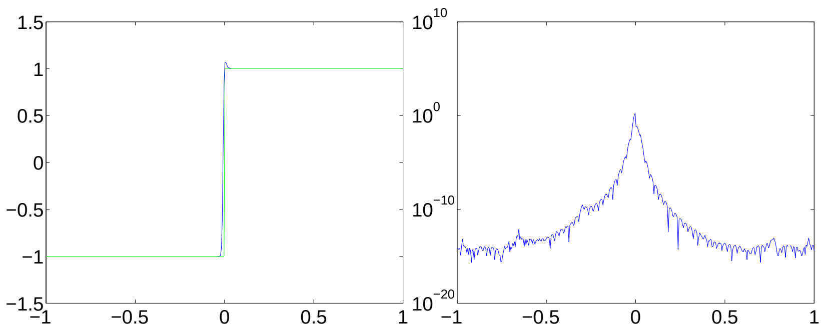

Figure 5.3: Left: Graph of sign function and its Chebyshev rational approximation with $n = 99$ and $m = 12$. Right: Error graph

An inspection of the graphs for the approximations show that there are still errors at the discontinuity, but the approximation in Figure 5.4 is a much sharper resolution than the Chebyshev approximation in Figure 4.1 or the filtered approximation in Figure 4.2. If we look at the error for points away from the discontinuity (Table 5.1), perhaps near $x = .5$, we can see the error decaying for increasing N.

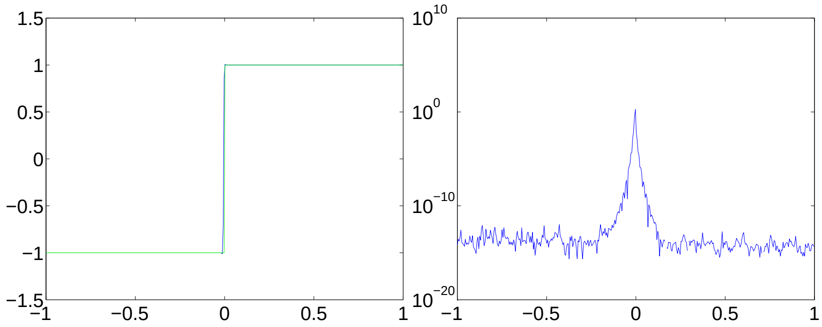

Figure 5.4: Left: Graph of sign function and its Chebyshev rational approximation with $n = 99$ and $m = 98$. Right: Error graph

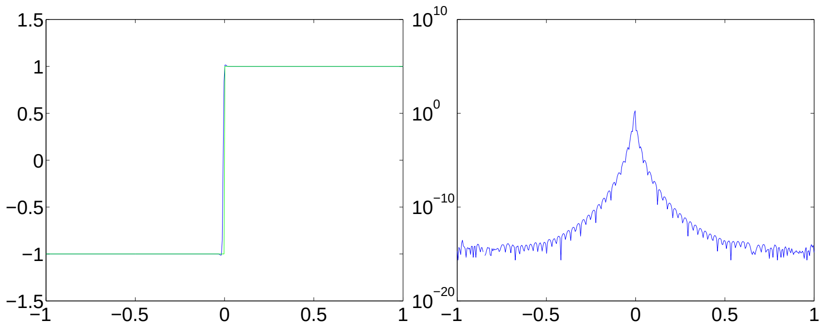

Figure 5.5: Left: Graph of sign function and its Chebyshev rational approximation with $n = 11$ and $m = 98$. Right: Error graph

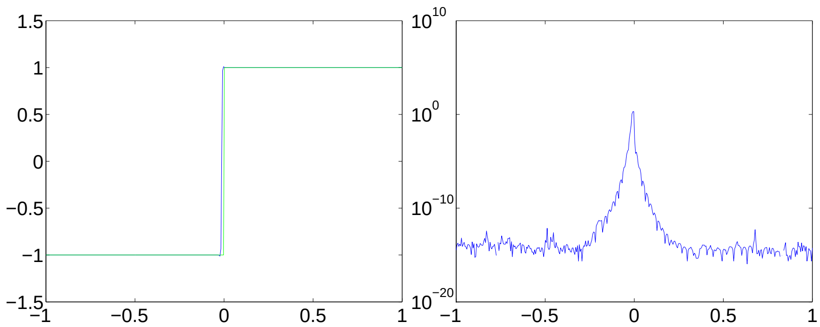

Figure 5.6: Left: Graph of sign function and its Chebyshev rational approximation with $n = 48$ and $m = 98$. Right: Error graph

$\frac{N}{M}$	8	16	32	64	99
8	$2.1471e-005$	$3.3488e-009$	$2.0441e-011$	$6.7435e-013$	$2.3093e-014$
16	$6.2709e-010$	$4.2930e-012$	$1.6720e-013$	$8.1046e-015$	$3.1086e-015$
32	$3.4412e-011$	$1.5499e-013$	$1.0991e-014$	$1.7764e-015$	$2.1094e-015$
64	$4.2299e-013$	$1.1102e-015$	$9.3259e-015$	$3.0642e-014$	$9.1038e-015$
99	$2.4059e-013$	$4.6629e-015$	$2.6645e-015$	$1.1102e-015$	$1.6098e-014$

Table 5.1: Error for Chebyshev Rational Approx to f_5 near the point $x = .5$

Also, note that the matrices can become ill-conditioned with increasing M in the denominator. Thus, we will only work with sufficiently small M.

Let us now see how the Chebyshev rational approximation method works on the other test functions. There seems to be a small improvement when working with the absolute value function f_1 in Figure 5.8 at the point $x = 0$, and a general smoothing of the error throughout the interval on which we are interpolating. But, the true test of the Chebyshev rational approximation is seen when approximating f_4. In Figure 5.9, we see that the Chebyshev rational approximation removes the Gibbs phenomenon as compared to the Chebyshev approximation in Figure 5.10. Also, the resolution at the discontinuities is again much sharper than the filtered approximation in Figure 5.11.

So, with our Chebyshev-Padé approximation we have reached our desired machine epsilon very close to any breaks there may be in a discontinuous function. The Gibbs phenomenon has been relegated to a very small interval, but has not been completely defeated.

We set out to rid the world of the nonuniform pointwise convergence of polynomial approximations to discontinuous functions, other-

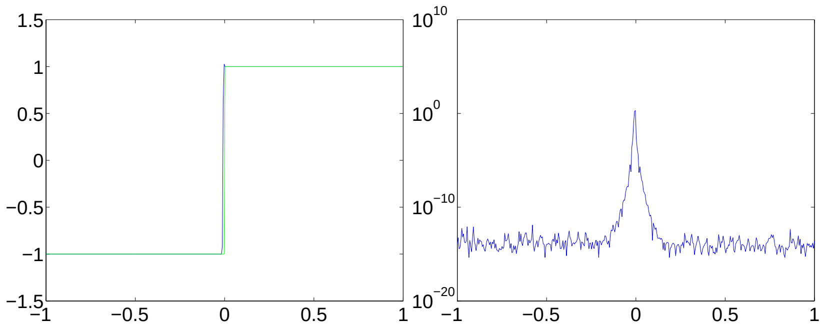

Figure 5.7: Left: Graph of sign function and its Chebyshev rational approximation with $n = 98$ and $m = 98$. Right: Error graph

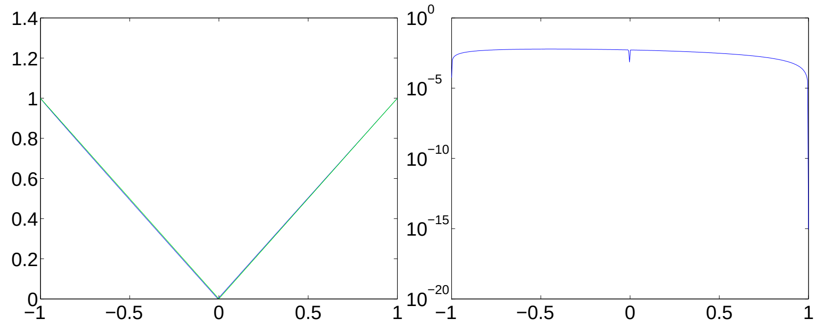

Figure 5.8: Left: Graph of f_1 and its Chebyshev rational approximation with $n = 99$ and $m = 99$. Right: Error graph. Recall the approximations of f_1 in Figure 2.3 and Figure 4.5. The Chebyshev rational approximation has a much smoother error and smaller disturbance at the point $x = 0$, where the first derivative does not exist.

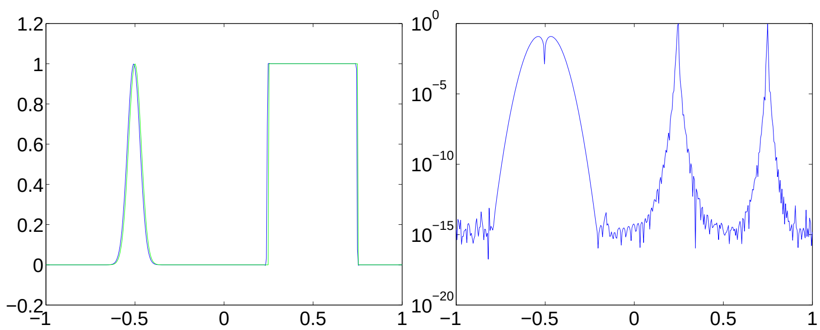

Figure 5.9: Left: Graph of f_4 and its Chebyshev rational approximation with $N = 128$ and $M = 64$. Right: Error graph

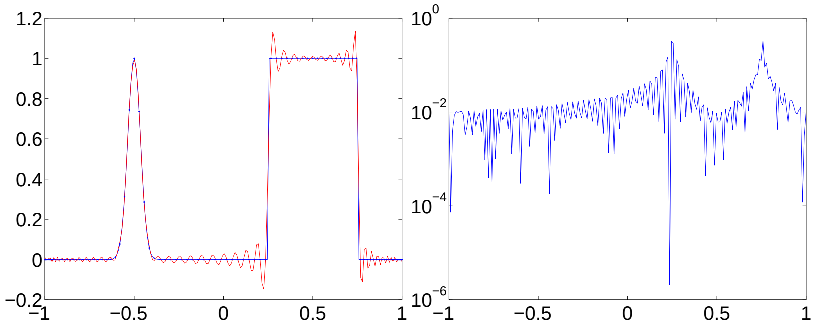

Figure 5.10: Left: Graph of f_4 and its Chebyshev approximation with $N = 99$. Note the Gibbs phenomenon at the discontinuities. Right: Error graph

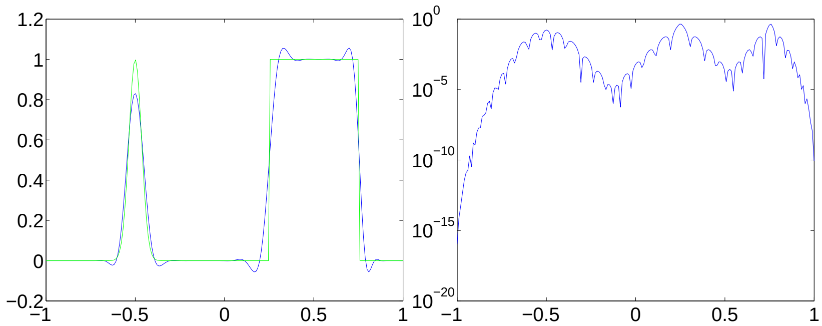

Figure 5.11: Left: Graph of f_4 and its filtered Chebyshev approximation with filter order $p = 4$. Note the rounding at the discontinuities. Right: Error graph.

wise known as the Gibbs phenomenon. Can we say that we are satisfied with our results? What are some other methods current researchers are trying? Is the fight against the Gibbs phenomenon over? At this point we shall examine some other methods currently being worked on to see if they are doing any better.

In a paper by J.S. Hesthaven and S.M. Kaber [6], the use of Jacobi polynomials and expansions, as opposed to our choice of Chebyshev, is investigated. Developing an approximation of the sign function, they are able to reduce the overshoot at the discontinuity and recover high order accuracy away from the jump, just as we have done. It does not seem to have any great benefits that would make it a better choice. However, their paper is left open with the possibilities of future research in generalizing results of Padé-Jacobi approximations for postprocessing. That lends us the interesting problem of approximating Partial Differential Equation solutions, $u(x)$, not particular known function values, with our Chebyshev approximations.

In another paper by Hesthaven, Kaber, and L. Lurati [7], the use of Padé-Legendre expansions are used to achieve almost non-oscillatory behavior without knowledge of the location of discontinuities. Again, testing on various functions, their numerical results show the reduction of the Gibbs phenomenon. Future possibilities left open from this paper include generaliztion to multi-dimensional problems.

We see that there is work left to be done. In the beginning of this work, we saw the failure of simple polynomial approximations to discontinuous functions. This led us to use rational approximations in the hope that discontinuity would have less of an impact. We have demonstrated that rational approximation methods work very well on discontinuous functions without any knowledge of edges. While we have mentioned some areas for future research such as applying our method to a two-dimensional problem, another might include developing some convergence theory.

After examining the results of other choices of polynomials such as the Legendre and Jacobi in rational approximations, the Chebyshev polynomials in our rational approximations seem as good a choice, if not better. As seen in the appendix, the Chebyshev polynomials have some lovely properties that lend to their integration and other computations in general. We have shown similar reductions of the Gibbs phenomenon at discontinuities without knowledge of their location. And lastly, machine epsilon - the driving force of our efforts - has been achieved away from such breaks.

Part IV

Appendix

Appendix A

Chebyshev Polynomials

There are various ways to describe the derivation of the Chebyshev polynomials, we will start with the solution of the Chebyshev differential equation by contour integrals [10].

The general Chebyshev equation is given as

$$L(u) = (1 - z^2)\frac{d^2u}{dz^2} - z\frac{du}{dz} + \lambda^2 u = 0,$$

with $Re(\lambda) \geq 0$. Our goal is to express solutions of the above equation by contour integrals of the form

$$u(z) = \int_C K(z, \xi)v(\xi)d\xi,$$

where the kernel function K, the function v, and the contour C are to be determined.

Substituting in the assumed form of the solution into Chebyshev's equation, we have

$$L(u) = \int_C \{(1-z^2)K_{zz} - zK_z + \lambda^2 K\} v(\xi) d\xi.$$

We require $K(z,\xi)$ to satisfy the partial differential equation

$$(1-z^2)K_{zz} - zK_z + \xi(\xi K_\xi)_\xi = 0,$$

and take as a solution

$$K(z,\xi) = \frac{\xi^2 - 1}{1 - 2z\xi + \xi^2}.$$

With $K(z,\xi)$ restricted to satisfy the above partial differential equation, we now have

$$L(u) = \int_C \{-\xi(\xi K_\xi)_\xi + \lambda^2 K\} v(\xi) d\xi.$$

Integrating the first term by parts, we can obtain an equation for the function $v(\xi)$ and a determination of the contour C.

$$L(u) = \xi\{\xi K v_\xi + Kv - \xi K_\xi v\}]_C - \int_C \{\xi^2 v_{\xi\xi} + 3\xi v_\xi + (1-\lambda^2)v\} K d\xi.$$

Finally, we impose the condition that $v(\xi)$ satisfy the ordinary differential equation

$$\xi^2 \frac{d^2 v}{d\xi^2} + 3\xi \frac{dv}{d\xi} + (1-\lambda^2)v = 0.$$

This equation is known as an Euler equation, thus we will choose as a solution

$$v(\xi) = \xi^{\lambda - 1}.$$

For the contour C, we can choose either one of the two circles enclosing a pole of the kernel $K(z,\xi)$. Such a choice causes the integrated terms to vanish. The poles are located at

$$\xi^{\pm} = z \pm \sqrt{z^2 - 1}.$$

We have now obtained two independent solutions of Chebyshev's equation, which we will denote by

$$T_{\lambda}^{\pm}(z) = \frac{1}{2\pi i} \int_{C\pm} \frac{\xi^2 - 1}{1 = 2z\xi + \xi^2} \xi^{\lambda - 1} d\xi.$$

Using Cauchy's residue theorem, we find

$$T_{\lambda}^{\pm}(z) = (z \pm \sqrt{z^2 - 1})^{\lambda}.$$

Thus, we will define the so-called Chebyshev function by the relation

$$T_{\lambda}(z) = \frac{1}{2}[T_{\lambda}^{+}(z) + T_{\lambda}^{-}(z)].$$

If $\lambda = n$, an integer, then $T_n(z)$ is the nth Chebyshev polynomial usually associated with this symbol. Then we have the representation

$$T_n(z) = \frac{1}{2}[(z + \sqrt{z^2 - 1})^n + (z - \sqrt{z^2 - 1})^n].$$

The simplest, and perhaps most common characterization of the

Chebyshev polynomials $T_n(z)$ is the formula

$$T_n(\cos\theta) = \cos n\theta.$$

This result follows directly from setting $z = \cos\theta$ in the equation

$$T_n(z) = \frac{1}{2}[(z + \sqrt{z^2 - 1})^n + (z - \sqrt{z^2 - 1})^n],$$

and applying De Moivre's theorem. In terms of the original variable z, we may write

$$T_n(z) = \cos(n\cos^{-1} z).$$

The Chebyshev polynomials as defined above give us a handy recurrence relation that allows a fairly simple calculation [8] of T_n.

For $n = 0$ and $n = 1$ we simply have

$$T_0(x) = \cos 0 = 1$$

and

$$T_1(x) = \cos\theta = x.$$

Using the sum of cosine formula,

$$\cos[(n+1)\theta] + \cos[(n-1)\theta] = 2\cos n\theta\cos\theta,$$

we find that

$$T_{n+1}(x) = 2x\,T_n(x) - T_{n-1}(x) \quad n = 1, 2, \ldots$$

with $x = \cos\theta$.

The Chebyshev polynomials have some noteworthy properties that we will now state.

The polynomial $T_n(x)$ is a polynomial of degree n with leading coefficient 2^{n-1} for all $n > 0$. When n is even, the polynomial of degree n is even. For n odd, the polynomial of degree n is odd. For example:

$$T_3(x) = 2xT_2(x) - T_1(x) = 4x^3 - 3x$$

and

$$T_4(x) = 2xT_3(x) - T_2(x) = 8x^4 - 8x^2 + 1.$$

The Chebyshev polynomials are also orthogonal on the interval $[-1, 1]$ with respect to the weight function

$$w(x) = \frac{1}{\sqrt{1 - x^2}}.$$

To verify this we must show that the integral of the product of $w(x)$,$T_m(x)$ and $T_n(x)$ over the interval $[-1, 1]$ is equal to zero when $m = n$ and equal to some positive value when $m \neq n$. Thus starting off, we have

$$\int_{-1}^{1} \frac{T_m(x)T_n(x)}{\sqrt{1 - x^2}} dx = \int_{-1}^{1} \frac{\cos(m \arccos x)\cos(n \arccos x)}{\sqrt{1 - x^2}} dx.$$

Making the substitution given in the definition we have $\theta = \arccos x$, which gives us

$$d\theta = -\frac{1}{\sqrt{1 - x^2}}.$$

Thus our integral now becomes

$$\int_{-1}^{1} \frac{T_m(x)T_n(x)}{\sqrt{1-x^2}}dx = -\int_{\pi}^{0} \cos(m\theta)\cos(n\theta)\; d\theta = \int_{0}^{\pi} \cos(m\theta)\cos(n\theta)\; d\theta.$$

We now have the two cases of either $m = n$ or $m \neq n$.

Let us first suppose that $m \neq n$. From the trigonometric formula for the product of cosines

$$\cos(m\theta)\cos(n\theta) = \frac{1}{2}[\cos(m+n)\theta + \cos(m-n)\theta],$$

we can rewrite the integral as

$$\begin{aligned}\int_{-1}^{1} \frac{T_m(x)T_n(x)}{\sqrt{1-x^2}}dx &= \frac{1}{2}\int_{0}^{\pi} \cos((m+n)\theta)d\theta + \frac{1}{2}\int_{0}^{\pi} \cos((m-n)\theta)d\theta \\ &= \left[\frac{1}{2(m+n)}\sin((m+n)\theta) + \frac{1}{2(m-n)}\sin((m-n)\theta)\right]_0^{\pi} \\ &= 0.\end{aligned}$$

Similarly, for $m = n$ we have

$$\int_{-1}^{1} \frac{[T_m(x)]^2}{\sqrt{1-x^2}}dx = \frac{\pi}{2}, \;\; \forall \;\; m \geq 1.$$

The polynomial of degree n attains its maximum and minimum values of ± 1, alternately, at the points

$$x_j = \cos\frac{j\pi}{n}.$$

Because of this, we will chose the points x_j as our interpolation points in Chebyshev approximations.

If we wish to differentiate Chebyshev polynomials [5], we can begin with

$$T_{n+1}(x) = \cos[(n_1)\arccos x]$$

$$\frac{1}{n+1}\frac{d[T_{n+1}(x)]}{dx} = \frac{-\sin[(n+1)\arccos x]}{-\sqrt{1-x^2}}.$$

Now, subtracting the corresponding equation for $n-1$ we have

$$\frac{1}{n+1}\frac{d[T_{n+1}(x)]}{dx} - \frac{1}{n-1}\frac{d[T_{n-1}(x)]}{dx} = \frac{\sin(n+1)\theta - \sin(n-1)\theta}{\sin\theta}$$

or

$$\frac{T'_{n+1}(x)}{n+1} - \frac{T'_{n-1}(x)}{n-1} = \frac{2\cos n\theta \sin\theta}{\sin\theta} = 2T_n(x) \quad n \geq 2,$$

and

$$T'_2(x) = 4T_1$$

$$T'_1(x) = T_o$$

$$T'_o(x) = 0.$$

We can use the above differentiation formulas to develop integration formulas for Chebshev polynomials. Thus, we have

$$\int T_n(x)dx = \frac{1}{2}\left[\frac{T_{n+1}(x)}{n+1} - \frac{T_{n-1}(x)}{n-1}\right] + C \quad n \geq 2$$

$$\int T_1(x)dx = \frac{T_2(x)}{4} + C$$

$$\int T_0(x)dx = T_1(x) + C.$$

If we want to approximate on the interval $[0, 1]$ instead of $[-1, 1]$ we will use what are known as *shifted Chebyshev polynomials*.

Definition 12 (Shifted Chebyshev Polynomials) *The* ***shifted Chebyshev polynomials***

$$T_n^*(x) = T_n(2x - 1)$$

are used to approximate on the interval $[0, 1]$.

The recurrence relation for shifted polynomials is given by

$$T_{n+1}^*(x) = (4x - 2)T_n^*(x) - T_{n-1}^*(x), \quad T_o^* = 1.$$

Appendix B

MatLab Code

B.1 Lagrange Interpolation Program

```
%Rob-Roy Mace
%Marshall University

function LagrangeTrial(r)

xi = linspace(-5,5,r); x = linspace(-5,5,100);
fEx = 1./(1 + xi.^2);
z = 1./(1 + x.^2);


n = size(xi,2); k = size(x,2);

for j = 1:k
   for i = 1:n
     xr = xi; xr(i) = [];
     card(i) = prod(x(j) - xr)/prod(xi(i) - xr);
   end
P(j) = fEx * card'; end

plot(x,P,'b',x,z,'g');

 pause

for c = 1:length(x)
  err(c) = abs(z(c) - P(c));
end
 semilogy(x,err)
```

B.2 Chebyshev Interpolation of $1/(1+x^2)$

```
% Rob-Roy Mace
% Marshall University

function ChebSmooth(N) x = linspace(-1,1,200);

for j = 1:N+1
  xi(j) = cos((j-1)*pi/N);
end

f = 1./(1 + xi.^2); fex = 1./(1 + x.^2);

a = zeros(N+1,1); for k = 0:N
    n = 0:N;
      temp = f(n+1).*cos(pi.*k*n./N);
      temp(1) = temp(1)*0.5;
      temp(N+1) = temp(N+1)*0.5;
      a(k+1) = 2.0*sum(temp)./N;
end

for s = 1:length(x)
 for p = 1:N+1
   tempb(p) = a(p)*cos((p-1)*acos(x(s)));
 end
Cheb(s) = sum(tempb)-.5*a(1); end

plot(x,fex,'b',x,Cheb,'r',xi,f,'k*')

 pause

for c = 1:length(x)
  err(c) = abs(fex(c) - Cheb(c));
end
 semilogy(x,err)
```

B.3 Chebyshev Interpolation of Sign Function

```
% Rob-Roy Mace
```

```
% Marshall University

function ChebDisct(N) x = linspace(-1,1,200);

for j = 1:N+1
  xi(j) = cos((j-1)*pi/N);
end

f = -1.*(xi<0) + 1.0.*(xi>=0);
fex = -1.*(x<0) + 1.0.*(x>=0);

a = zeros(N+1,1); for k = 0:N
    n = 0:N;
      temp = f(n+1).*cos(pi.*k*n./N);
      temp(1) = temp(1)*0.5;
      temp(N+1) = temp(N+1)*0.5;
      a(k+1) = 2.0*sum(temp)./N;
end

for s = 1:length(x)
 for p = 1:N+1
   tempb(p) = a(p)*cos((p-1)*acos(x(s)));
 end
Cheb(s) = sum(tempb)-.5*a(1); end

plot(x,fex,'b',x,Cheb,'r',xi,f,'k*')

 pause

for c = 1:length(x)
  err(c) = abs(fex(c) - Cheb(c));
end
 semilogy(x,err)
```

B.4 Filtered Chebyshev Interpolation of Sign Function

```
% Rob-Roy Mace
% Marshall University
```

```
% MyFilterChebyTrial
% function  MyFilterChebyTrial

x = linspace(-1,1,200); N = 99;

for j = 1:N+1
    xi(j) = cos((j-1)*pi/N);
end

f = -1.*(xi<0) + 1.0.*(xi>=0);
fex = -1.*(x<0) + 1.0.*(x>=0);

a = zeros(N+1,1);

for k = 0:N;
    n = 0:N;
  temp = f(n+1).*cos(pi.*k*n./N);
  temp(1) = temp(1)*0.5;
  temp(N+1) = temp(N+1)*0.5;
  a(k+1) = 2.0*sum(temp)./N;
end


a(1) = 0.5*a(1); a(end) = 0.5*a(end);

Cheb = zeros(1,length(x)); for s = 1:length(x)
  for p = 1:N+1
     Cheb(s) = Cheb(s) +
  exp(-36*((p-1)/(N))^(4))*a(p)*cos((p-1)*acos(x(s)));
  end
end

plot(x,fex,'b',x,Cheb,'r')

 pause

for c = 1:length(x)
  err(c) = abs(fex(c) - Cheb(c));
```

```
end
 semilogy(x,err)
```

B.5 Chebyshev Rational Interpolation of Test Functions

```
%Rob-Roy Mace
%Marshall University

%Chebyshev Rational Approximation Code
%LN denotes degree of numerator
%LM denotes degree of denominator
%Choice determines which function will be approximated
%1: 7th degree polynomial  2: square root
%3: absolute value   4: exp(cos)
%5: sign function


function ChebRatTrial(LN,LM,choice)

TRUE = 1; FALSE = 0;

OK = TRUE;
%LM = 20;
%LN = 55;
BN = LM + LN; AA = zeros(1,BN+LM+1);
NROW =  zeros(1,BN+1);
P=zeros(1,LN+1);
Q = zeros(1,LM+1);
A = zeros(BN+1,BN+2); Na = BN +
LM + 1; Np = 200;

x = linspace(-1,1,Np);

for j = 1:(Na+1)
   xi(j) = cos((j-1)*pi/(Na+1));
end

if choice == 1
```

```
  f = xi.^7 - 2*xi.^6 + xi + 3;
  fex = x.^7 - 2*x.^6 + x + 3;
elseif choice == 2
  f = sqrt(1-xi.^2);
  fex = sqrt(1-x.^2);
elseif choice == 3
  f = abs(xi);
  fex = abs(x);
elseif choice == 4
  f = exp(cos(8*(xi.^3)+1));
  fex = exp(cos(8*(x.^3)+1));
elseif choice == 5
  f = -1.*(xi<0) + 1.0.*(xi>=0);
  fex = -1.*(x<0) + 1.0.*(x>=0);
elseif choice == 6
   d = 0.025;
  f =  ( xi >= 0.25 & xi <= 0.75 ).*1 +
  ( xi > 0 & xi < 0.25 | xi <= 1.0 & xi > 0.75 ).*0
  + (xi<=0).*exp(-((xi+0.5).^2)/(4*d^2));
  fex =  ( x >= 0.25 & x <= 0.75 ).*1 +
  ( x > 0 & x < 0.25 | x <= 1.0 & x > 0.75 ).*0 +
  (x<=0).*exp(-((x+0.5).^2)/(4*d^2));
end

ac = zeros(Na+1,1); for k = 0:Na
    l = 0:Na;
       temp = f(l+1).*cos(pi.*k*l./Na);
       temp(1) = temp(1)*0.5;
       temp(Na+1) = temp(Na+1)*0.5;
       ac(k+1) = 2.0*sum(temp)./Na;
end

for I=0:Na-1
     AA(I+1) = ac(I+1);
end

N = BN; M = N+1;

for I = 1 : M
```

```
  NROW(I) = I;
end

NN = N-1; Q(1) = 1.0;

for I = 0 : N
    for J = 0 : I
      if J <= LN
        A(I+1,J+1) = 0;
      end
    end
  if I <= LN
    A(I+1,I+1) = 1.0;
  end
  for J = I+1 : LN
    A(I+1,J+1) = 0;
  end
  for J = LN+1 : N
    if I ~= 0
      PP = I-J+LN;
      if PP < 0
        PP = -PP;
      end
      A(I+1,J+1) = -(AA(I+J-LN+1)+AA(PP+1))/2.0;
     else
        A(I+1,J+1) = -AA(J-LN+1)/2.0;
    end
  end
 A(I+1,N+2) = AA(I+1);
end

A(1,N+2) = A(1,N+2)/2.0; I = LN+2;

while OK == TRUE & I <= N
 IMAX = NROW(I);
 AMAX = abs(A(IMAX,I));
 IMAX = I;
 JJ = I+1;
 for IP = JJ : N + 1
```

```
    JP = NROW(IP);
   if abs(A(JP,I)) > AMAX
    AMAX = abs(A(JP,I));
    IMAX = IP;
   end
 end
 if AMAX <= 1.0e-20
    OK = FALSE;
   else
     if NROW(I) ~= NROW(IMAX)
       NCOPY = NROW(I);
       NROW(I) = NROW(IMAX);
       NROW(IMAX) = NCOPY;
     end
   I1 = NROW(I);
   for J = JJ : M
     J1 = NROW(J);
     XM = A(J1,I)/A(I1,I);
     for K = JJ : M + 1
      A(J1,K) = A(J1,K)-XM*A(I1,K);
     end
     A(J1,I) = 0;
   end
  end
 I = I+1;
end
  if OK == TRUE
   N1 = NROW(N+1);
    if abs(A(N1,N+1)) <= 1.0e-20
     OK = FALSE;
    else
     if LM > 0
      Q(LM+1) = A(N1,M+1)/A(N1,N+1);
      A(N1,M+1) = Q(LM+1);
     end
  PP = 1;
   for K = LN+2 : N
    I = N-K+LN+2;
    JJ = I+1;
```

```
    N2 = NROW(I);
    SM = A(N2,M+1);
      for KK = JJ : N + 1
       LL = NROW(KK);
       SM = SM - A(N2,KK) * A(LL,M+1);
      end
     A(N2,M+1) = SM / A(N2,I);
     Q(LM-PP+1) = A(N2,M+1);
     PP = PP+1;
    end

    for K = 1 : LN + 1
     I = LN+1-K+1;
     N2 = NROW(I);
     SM = A(N2,M+1);
      for KK = LN+2 : N + 1
       LL = NROW(KK);
       SM = SM-A(N2,KK)*A(LL,M+1);
      end
     A(N2,M+1) = SM ;
     P(LN-K+2) = A(N2,M+1);
    end

rat = zeros(length(x),1);

for k = 1:length(x)
    num = 0;
    for j = 0:length(P)-1
        num = num + P(j+1)*cos(j*acos(x(k)));
    end

    dem = 0;
    for j = 0:length(Q)-1
        dem = dem + Q(j+1)*cos(j*acos(x(k)));
    end
    rat(k) = num/dem;
end

end end
```

```
set(gcf,'unit','inch','pos',[0 0 3 2.25]) plot(x,rat,'b',x,fex,'g')

pause for c = 1:length(x)
  err(c) = abs(fex(c) - rat(c));
end semilogy(x,err) norm(err,inf)
```

Bibliography

[1] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Dover, 2001, 2nd ed. ISBN 0-486-41183-4. 1, 1, 2, 2

[2] R. Burden and D. Faires, *Numerical Analysis*, Brooks and Cole, 2001, 7th ed. ISBN 0-534-38216-9. 1

[3] C.W. Clenshaw and K. Lord,"Rational approximations from Chebyshev series",In: B.K.P. Scaife(Ed.),*Studies in Numerical Analysis*,Academic Press, London and New York, 1974, pp.95-113. 5, 11

[4] David Gottlieb and Chi-Wang Shu, *On the Gibbs Phenomenon and Its Resolution*, SIAM review, 39, pp.644-668, 1997. 4

[5] R. W. Hamming, *Numerical Methods for Scientists and Engineers*,McGraw Hill, 1973, 2nd ed. ISBN 07-025887-2. A

[6] J.S. Hesthaven and S.M. Kaber, *Padé-Jacobi Approximants*, Brown University. 3, 5

[7] J.S. Hesthaven, S.M. Kaber, and L. Lurati, *Padé-Legendre Interpolants for Gibbs Reconstruction*, Brown University. 5

[8] J. Ll. Morris, *Computational Methods in Elementary Numerical Analysis*, Wiley, 1983, ISBN 0-471-10420-5. A

[9] Scott A. Sarra, *Digital Total Variation Filtering as Postprocessing for Pseudospectral Methods for Conservation Laws.* To appear in Numerical Algorithms 2005/6. 8, 4

[10] Martin Avery Snyder, *Chebyshev Methods in Numerical Approximation*,Prentice-Hall, 1966. A