APPROXIMATION THEORY AND APPROXIMATION PRACTICE

Nick Trefethen, TU Berlin, February 2010

Contents

1. Introduction

Welcome to a beautiful subject! — the constructive approximation of functions. And welcome to a rather unusual book.

Approximation theory is a well-established field, and our aim is to teach you some of its most important ideas and results. The style of this book, however, is quite different from what you will find elsewhere. Everything is illustrated computationally with the help of the chebfun software package in Matlab, from Chebyshev interpolants to Lebesgue constants, from the Weierstrass Approximation Theorem to the Remez algorithm. Everything is practical and fast, so we will routinely compute polynomial interpolants or Gauss quadrature nodes and weights for tens of thousands of points. In fact, each chapter of this book is a single Matlab M-file, and the book has been produced by executing these files with Matlab's "publish" facility. The chapters come from M-files called `chap1.m`, `chap2.m`, ... , and you can download them and use them as templates to be modified for explorations of your own.

Beginners are welcome, and so are experts, who will find familiar topics approached from new angles and familiar conclusions turned on their heads. Indeed, the field of approximation theory came of age in an era of polynomials of degrees perhaps O(10). Now that O(1000) is easy and O(1000000) is not hard, different questions come to the fore. In particular we shall see that "best" approximants are hardly better than "near-best," though they are much harder to compute.

This is a book about approximation, not about chebfun, and for the most part we shall use chebfun tools without explaining them. A brief introduction to chebfun is given in the Appendix, and for much more information, see the Guide and the download page at

`http://www.maths.ox.ac.uk/chebfun/`

In the course of the book we shall use chebfun overloads of the following Matlab functions, among others:

```
CUMSUM, DIFF, INTERP1, NORM, POLY, POLYFIT, SPLINE
```

as well as the additional chebfun commands

```
CF, CHEBPADE, CHEBPOLY, CHEBPOLYPLOT, CHEBPOLYVAL,
CHEBPTS, LEBESGUE, LEGPOLY, LEGPTS, RATINTERP, REMEZ.
```

There are quite a number of excellent books on approximation theory. Three classics are [Cheney 1966], [Davis 1963], and [Meinardus 1967], and a more recent computationally oriented classic is [Powell 1981].

A good deal of our emphasis will be on ideas related to Chebyshev points and polynomials, whose roots go back a century or more to mathematicians including Chebyshev (1821–1894), Zolotarev (1847–1878), de la Vallée Poussin (1866–1962), Bernstein (1880–1968), and Dunham Jackson (1888–1946). In the computer era, some of the early figures who developed "Chebyshev technology," in approximately chronological order, were Lanczos, Clenshaw, Specht, Good, Fox, Elliott, Mason, and Orszag. Two books on Chebyshev polynomials are [Rivlin 1990] and [Mason & Handscomb 2003]. One reason we emphasize Chebyshev technology so much is that in practice, for working with functions on intervals, these methods are unbeatable. For example, we shall see in Chapter 14 that the difference in approximation power between Chebyshev and "optimal" interpolation points is utterly negligible. Another reason is that if you know the Chebyshev material solidly, this is the best possible foundation for work on other approximation ideas.

Our mathematical style is conversational, but that doesn't mean the material is elementary. The book aims to be more readable than most, and the numerical experiments help achieve this. At the same time, theorems are stated and proofs are given, often rather terse, without all the details spelled out. It is assumed that reader is comfortable with rigorous mathematical arguments and familiar with ideas like continuous functions on compact sets, Lipschitz continuity, contour integrals in the complex plane, and norms of matrices and operators. If you are a student, I hope you are an advanced undergraduate or graduate who has taken courses in numerical analysis and complex analysis. If you are a seasoned mathematician, I hope you are also a Matlab user!

This book was produced using `publish` in LaTeX mode: thus this chapter, for example, can be generated with the command `publish('chap1','latex')`. To achieve the desired layout we begin by setting a few default parameters:

```
set(0,'defaultfigureposition',[380 320 540 200],...
'defaultaxeslinewidth',0.9,'defaultaxesfontsize',8,...
'defaultlinelinewidth',1.1,'defaultpatchlinewidth',1.1,...
'defaultlinemarkersize',15), format compact, format long
chebfunpref('factory'); clear all, x = chebfun('x',[-1 1]);
```

To make the chapters independently executable, it is necessary to include these statements at the beginning of each. This would lead to a clutter of text, so instead, at the beginning of each chapter we execute the command

```
ATAPformats
```

which calls an M-file containing the code above. This isn't beautiful, but it works. For convenience, `ATAPformats` is included in the standard distribution of the chebfun package. (For the actual production of the printed book, `publish` was executed not chapter-by-chapter but on a big file concatenating all the chapters, and a few tweaks were made to the resulting LaTeX file.)

The Lagrange interpolation formula was discovered by Waring, the Gibbs phenomenon was discovered by Wilbraham, and the Runge phenomenon was first glimpsed, if perhaps not very clearly, by Meray. These are just some of the instances of Stigler's Law in approximation theory, and the reader will see my interest in history in the references section, where original sources are usually given and the entries stretch back several centuries, each with an editorial comment attached. Often the originals are surprisingly readable and insightful, and in any case, it seems especially important to pay heed to original sources in a book like this that aims to reexamine material that has grown too standardized in the textbooks. Another reason for looking at original sources is that in the last few years, thanks to digitization of journals, it has become far easier to track them down than it used to be, though there are always difficult special cases like [Wilbraham 1848], which I finally found in an elegant leather-bound volume in the Balliol College library.

Perhaps I may add a further personal comment. As an undergraduate and graduate student in the late 1970s and early 1980s, one of my main interests was approximation theory. I regarded this subject as the foundation of my wider field of numerical analysis — but as the years passed, it came to seem dry and academic, and I moved into other areas. Now times have changed, computers have changed, and my perceptions have changed. I now again regard approximation theory as exceedingly close to computing, and this view has been reinforced by new developments including wavelets, radial basis functions, and compressed sensing. The topics discussed here are a bit more classical than those: the foundations of univariate approximation theory. As I hope this book will show, there is scarcely an idea in this area that can't be illustrated compellingly in a few lines of chebfun code, and as I first imagined around 1975, anyone who wants to be expert at numerical computation really does need to know this material.

Exercise 1.1. Chebfun download. Download the current version of the chebfun package from www.maths.ox.ac.uk/chebfun/ and install it in your Matlab path as instructed at the web site. Execute the command `chebtest` to make sure things are working, and note the time taken. Execute `chebtest` again and see how much speedup there is now that various files have been brought into memory.

Exercise 1.2. The publish command. Execute `help publish` and `doc publish` in Matlab to learn the basics of how the `publish` command works. Then download `chap1.m` and `chap2.m` from www.maths.ox.ac.uk/chebfun/ and publish them in HTML with a Matlab command like `open(publish('cheb.1'))`. Now publish them again with `publish('chap2','latex')` followed by appropriate LaTeX commands. (You will probably find that `chap1.tex` and `chap2.tex` appear in a subdirectory on your computer labeled html.) If you are a student taking in a course for which you are expected to turn in writeups of the exercises, then you could hardly do better than to make it a habit of producing them with `publish`.

2. Chebyshev points and interpolants

As always we begin a chapter by setting the default formats:

```
ATAPformats
```

Any interval $[a, b]$ can be scaled to $[-1, 1]$, so most of the time, we shall just talk about $[-1, 1]$.

Let n be a positive integer:

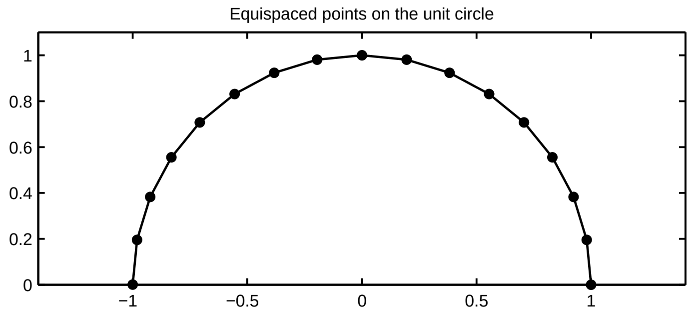
```
n = 16;
```

Consider $n+1$ equally spaced angles $\{\theta_j\}$ from 0 to π:

```
tt = linspace(0,pi,n+1);
```

We can think of these as the arguments of $n+1$ points $\{z_j\}$ on the upper half of the unit circle in the complex plane. These are the $(2n)$th roots of unity lying in the closed upper half-plane:

```
zz = exp(1i*tt);
hold off, plot(zz,'.-k'), axis equal, ylim([0 1.1])
title('Equispaced points on the unit circle')
```



The **Chebyshev points** associated with the parameter n are the real parts of these points,

$$x_j = \mathrm{Re}\, z_j = \frac{1}{2}(z_j + z_j^{-1}), \quad 0 \le j \le n:$$

```
xx = real(zz);
```

Some authors use the terms **Chebyshev-Lobatto points**, **Chebyshev extreme points**, or **Chebyshev points of the second kind**, but as these are the points most often used in practical computation, we shall just say Chebyshev points.

Another way to define the Chebyshev points is in terms of the original angles:

$$x_j = \cos(j\pi/n), \quad 0 \le j \le n,$$

```
xx = cos(tt);
```

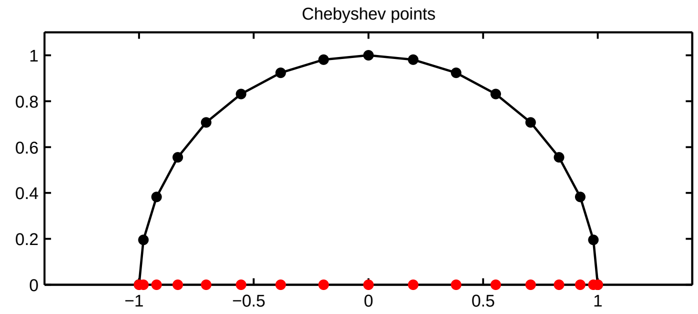

There is also an equivalent chebfun command `chebpts`:

```
xx = chebpts(n+1);
```

Actually this result isn't exactly equivalent, as the ordering is left-to-right rather than right-to-left.

Let us add the Chebyshev points to the plot:

```
hold on, plot(xx,0*xx,'.r'), title('Chebyshev points')
```



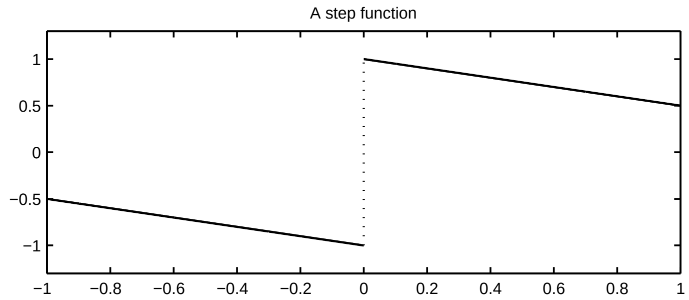
They cluster near 1 and -1, with the average spacing as $n \to \infty$ being given by a density function with square root singularities at both ends (Exercise 2.2).

Let $\{f_j\}$, $0 \le j \le n$, be a set of numbers, which may or may not come from sampling a function $f(x)$ at the Chebyshev points. Then there exists a unique polynomial p of degree n that interpolates these data, i.e., $p(x_j) = f_j$ for each j. When we say "of degree n," we mean of degree less than or equal to n. As we trust the reader already knows, the existence and uniqueness of polynomial interpolants applies for any distinct set of interpolation points. In this case of special interest involving Chebyshev points, we call the polynomial the **Chebyshev interpolant**.

Polynomial interpolants through equally spaced points have terrible properties, and we shall explore this effect in Chapters 11–13. Polynomial interpolants through Chebyshev points, however, are excellent. It is the clustering near the ends of the interval that makes the difference, and other sets of points with similar clustering, like Legendre points (Chapter 15), have similarly good behavior. The explanation of this fact has a lot to do with potential theory [Ransford 1995, Smirnov & Lebedev 1968, Walsh 1969], but we shall not go into that in this book.

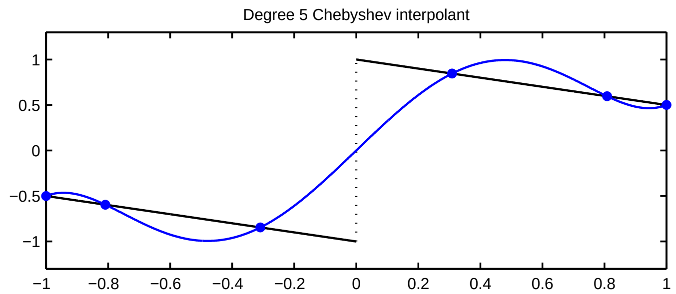
The chebfun system is built on Chebyshev interpolants. For example, here is a certain step function:

```
x = chebfun('x');
f = sign(x) - x/2;
hold off, plot(f,'k'), ylim([-1.3 1.3])
title('A step function')
```



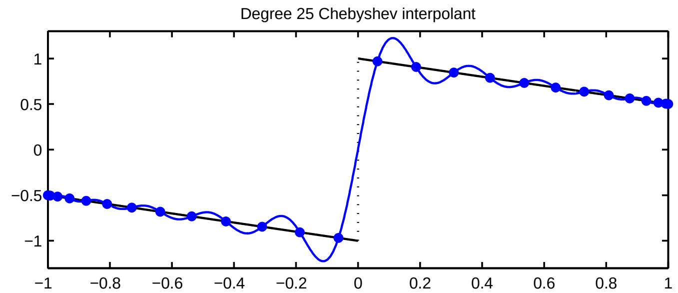
By calling `chebfun` with a second explicit argument of 6, we can construct the Chebyshev interpolant to f through 6 points, that is, of degree 5:

```
p = chebfun(f,6);
hold on, plot(p,'.-'), ylim([-1.3 1.3])
title('Degree 5 Chebyshev interpolant')
```



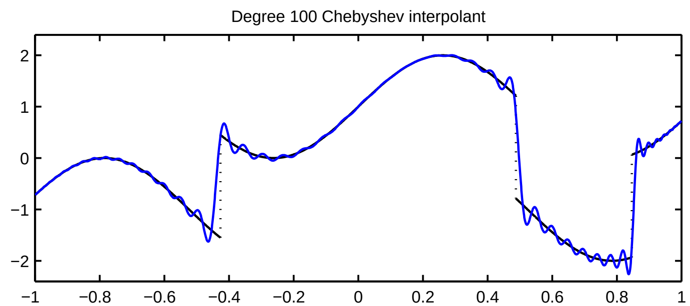
Similarly, here is the Chebyshev interpolant of degree 25:

```
hold off, plot(f,'k')
p = chebfun(f,26);
hold on, plot(p,'.-'), ylim([-1.3 1.3])
title('Degree 25 Chebyshev interpolant')
```

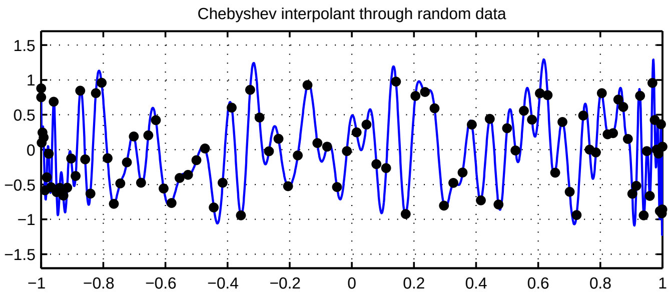
Here's a more complicated function and its Chebyshev interpolant of degree 100:

```
f = sin(6*x) + sign(sin(x+exp(2*x)));
hold off, plot(f,'k')
p = chebfun(f,101);
hold on, plot(p), ylim([-2.4 2.4])
title('Degree 100 Chebyshev interpolant')
```



Another way to use the `chebfun` command is by giving it an explicit vector of data rather than a function to sample, in which case it interprets the vector as data for a Chebyshev interpolant of the appropriate order. Here for example is the interpolant of degree 99 through 100 random data values in $[-1, 1]$:

```
p = chebfun(2*rand(100,1)-1);
hold off, plot(p,'-b')
hold on, plot(p,'.k')
ylim([-1.7 1.7]), grid on
title('Chebyshev interpolant through random data')
```



This experiment illustrates how robust Chebyshev interpolation is. If we had taken a million points instead of 100, the result would not have been much different mathematically, but it would have been a mess to plot. We shall return to this figure in Chapter 13.

For illustrations like these it is interesting to pick data with jumps or wiggles, and Chapter 9 discusses such interpolants more systematically. In the applications where polynomial interpolants are actually useful, however, the data will typically be smooth.

Exercise 2.1. Chebyshev interpolants through random data. Repeat the experiment of interpolation through random data for 10, 100, 1000, and 10000 points. In each case use the command `minandmax(p)` to determine the minimum and maximum values of the interpolant and measure the computer time required for this computation (e.g. using `tic` and `toc`). In addition to the four plots over $[-1, 1]$, use `plot(p,'interval',[0.9999 1])` to produce another plot of the interpolant through 10000 values in the interval $[0.9999, 1]$. How many of the 10000 grid points fall in this interval?

Exercise 2.2. Limiting density as $n \to \infty$. (a) If $-1 \le a < b \le 1$, what fraction of the $n + 1$ Chebyshev points fall in the interval $[a, b]$ in the limit $n \to \infty$? (b) How does this result match the number found in $[0.9999, 1]$ in the last exercise for the case $n = 9999$? (c) Derive the following formula for the density of the Chebyshev points near $x \in (-1, 1)$ in the limit $n \to \infty$: $\rho(x) = (\pi\sqrt{1 - x^2})^{-1/2}$.

Exercise 2.3. Rounding errors in computing Chebyshev points. On a computer in floating point arithmetic, the formula $x_j = \cos(j\pi/n)$ for the Chebyshev points is not so good because it lacks the expected symmetries. (a) Write an elegant Matlab program that finds the smallest even value $n \ge 2$ for which, on your computer as computed by this formula, $x_{n/2} \ne 0$. (b) Write another program that finds the smallest $n \ge 1$ for which the points $\{x_j\}$ do not come out exactly symmetric about 0. Is it the same value of n as in (a)? (c) Derive a mathematically equivalent formula for x_j based on the sine rather than the cosine which achieves perfect symmetry for all n in floating point arithmetic. (You may assume that your computer's sine function and other operations are perfectly symmetric about 0.)

Exercise 2.4. Chebyshev points of the first kind. The Chebyshev points of the first kind, also known as **Gauss-Chebyshev points**, are obtained by taking the real parts of points on the unit circle mid-way between those we have considered, i.e. $x_j = \cos((j + \frac{1}{2})\pi/(n + 1))$ for integers $0 \le j \le n$. Call `help chebpts` and `help legpts` to find out how to generate these points in chebfun and how to generate Legendre points for comparison (these are roots of Legendre polynomials). For $n + 1 = 100$, what is the maximum difference between a Chebyshev point of the first kind and the corresponding Legendre point? Draw a plot to illustrate how close these two sets of points are.

Exercise 2.5. Convergence of Chebyshev interpolants. (a) Use chebfun to produce a plot on a log scale of $\|f - p_n\|$ as a function of n with $f(x) = e^x$ on $[-1, 1]$, where p_n is the degree n Chebyshev interpolant. Take $\|\cdot\|$ to be the supremum norm, which can be computed by `norm(f-p,inf)`. How large must n be for accuracy at the level of machine precision? What happens if n is increased beyond this point? (b) Same questions for $f(x) = 1/(1 + 25x^2)$. Convergence rates like these will be analyzed in Chapters 7 and 8.

3. Chebyshev polynomials and series

`ATAPformats`

One good way to specify a polynomial of degree n on $[-1, 1]$, as we saw in the last chapter, is by its values at $n+1$ Chebyshev points. Another equally good way is by its coefficients in a **Chebyshev expansion**, that is, a linear combination of the Chebyshev polynomials $T_0, \ldots, T_n$. Depending on the application, one or the other of these two representations may be most useful, and one can go back and forth between them rapidly and accurately with an algorithm based on the Fast Fourier Transform (FFT). This duality is exactly analogous to the perhaps more familiar relationship between "space" and "Fourier space" in discrete Fourier analysis.

In Chapter 2 we defined Chebyshev points as the real parts of equally spaced points on the unit circle. Similarly, the kth **Chebyshev polynomial** is the real part of the function z^k on the unit circle:

$$x = \mathrm{Re}(z) = \tfrac{1}{2}(z + z^{-1}) = \cos\theta, \quad \theta = \cos^{-1} x,$$

$$T_k(x) = \mathrm{Re}(z^k) = \tfrac{1}{2}(z^k + z^{-k}) = \cos(k\theta).$$

Chebyshev polynomials were introduced by Chebyshev in the 1850s, though without the connection to z and θ [Chebyshev 1854 & 1859]. The reason they are labelled by the letter T is probably that Chebyshev, de la Vallée Poussin, Bernstein, and other early experts in the subject published in French, and the French transliteration of the Russian name is Tschebyscheff.

It follows immediately from the definition above that T_k satisfies $-1 \leq T_k(x) \leq 1$ for $x \in [-1, 1]$ and takes alternating values ± 1 at the $k+1$ Chebyshev points. What is not so obvious is that T_k is a polynomial. We can verify this property by induction. For example, we can calculate $T_2(x)$ like this:

$$T_2(x) = \tfrac{1}{2}(z^2 + z^{-2}) = \tfrac{1}{2}(z + z^{-1})^2 - 1 = 2x^2 - 1.$$

Similarly we calculate

$$T_3(x) = \tfrac{1}{2}(z^3 + z^{-3}) = \tfrac{1}{2}(z + z^{-1})(z^2 + z^{-2}) - \tfrac{1}{2}(z^1 + z^{-1}) = 2xT_2(x) - T_1(x),$$

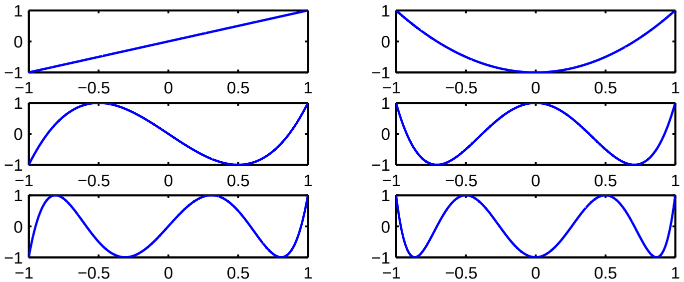
so $T_3(x) = 4x^3 - 3x$. In general we have

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

implying that for each $k \geq 1$, T_k is a polynomial of degree exactly k with leading coefficient 2^{k-1}.

The chebfun command `chebpoly(n)` returns the chebfun corresponding to T_n. Here for example are $T_1, \ldots, T_6$:

```
for n = 1:6
    T{n} = chebpoly(n);
    subplot(3,2,n)
    plot(T{n}), axis([-1 1 -1 1])
end
```



Here are their coefficients with respect to the monomial basis $1, x, x^2, \ldots$. As usual, Matlab orders coefficients from highest degree down to zero.

```
for n = 1:6
    poly(T{n})
end
```



```
ans =
     1     0
ans =
     2     0    -1
ans =
     4     0    -3     0
ans =
     8     0    -8     0     1
ans =
    16     0   -20     0     5     0
ans =
    32     0   -48     0    18     0    -1
```

So, for example,

$$T_5(x) = 16x^5 - 20x^3 + 5x.$$

The monomial basis is familiar and comfortable, but you should never use it for numerical work with functions on an interval. Use the Chebyshev basis instead. (If the domain is $[a, b]$ rather than $[-1, 1]$, the Chebyshev polynomials must be scaled accordingly, and chebfun does this automatically when one works in other intervals.) For example, x^5 has the Chebyshev expansion

$$x^5 = \frac{5}{80}T_5(x) + \frac{5}{16}T_3(x) + \frac{5}{8}T_1(x).$$

We can calculate such expansion coefficients by using the command `chebpoly(p)`, where p is the chebfun whose coefficients we want to know:

```
format short
chebpoly(x.^5)
```

```
ans =
    0.0625         0    0.3125         0    0.6250         0
```

Any polynomial p can be written uniquely like this as a finite Chebyshev series: the functions $T_0(x), T_1(x), \ldots, T_n(x)$ form a *basis* for the space of polynomials of degree $\leq n$. Since p is determined by its values at Chebyshev points, it follows that there is a one-to-one linear mapping between values at Chebyshev points and Chebyshev expansion coefficients. As mentioned at the beginning of this chapter, this mapping can be applied in $O(n \log n)$ operations with the aid of the Fast Fourier Transform (FFT) or the Fast Cosine Transform, an observation perhaps first made by Ahmed and Fisher and Orzsag around 1970 [Ahmed & Fisher 1970, Orszag 1971a and 1971b, Gentleman 1972]. That is what the chebfun system does when you type `chebpoly`. We shall not give details of the FFT here.

Just as a polynomial p has a finite Chebyshev series, a more general function f has an infinite Chebyshev series. Exactly what kind of "more general function" can we allow? For an example like $f(x) = e^x$, everything will turn out to be straightforward, but what if f is merely differentiable rather than analytic? Or what if it is continuous but not differentiable? Analysts have studied such cases carefully, identifying exactly what degrees of smoothness correspond to what kinds of convergence of Chebyshev series. We shall not concern ourselves with trying to state the sharpest possible result but will just make a particular assumption that covers almost every application. We shall assume that f is **Lipschitz continuous** on $[-1,1]$. Recall that this means that there is a constant C such that $|f(x) - f(y)| \le C|x-y|$ for all $x, y \in [-1,1]$. Recall also that a series is **absolutely convergent** if it remains convergent if each term is replaced by its absolute value, and that this implies that one can reorder the terms arbitrarily without changing the result.

Here is our basic theorem about Chebyshev series and their coefficients.

Theorem 3.1: Chebyshev series. *If f is Lipschitz continuous on $[-1,1]$, it has a unique representation as an absolutely and uniformly convergent series*

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

and the coefficients are given by the formula

$$a_k = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} dx,$$

with the special case that for $k=0$, the factor $2/\pi$ changes to $1/\pi$.

Proof. Throughout this book, our approach to all kinds of results involving Chebyshev polynomials will always be the same: transplant them to the unit circle in the complex plane, where they become results involving powers of z. Integrals over $[-1,1]$ transplant to integrals over the unit circle, where one can generally get the results one wants from the Cauchy integral formula. This method of dealing with Chebyshev mathematics has the advantage that one never has to remember any trigonometric identities!

Here is how it goes for Chebyshev series and their coefficients. We are given a function $f(x)$ on $[-1,1]$. We transplant f by defining a function F on the unit circle whose value at a point z on the circle is the same as the value of f at the corresponding point $x \in [-1,1]$. In other words, $F(z) = F(z^{-1}) = f(x)$, where $x = \operatorname{Re} z = (z + z^{-1})/2$. Notice that each value $x \in (-1,1)$ corresponds to two different values z on the unit circle, one on the upper semicircle and the other on the lower semicircle.

To convert between integrals in x and z, we have to convert between dx and dz.

We can do this by differentiating the formula for x to get

$$dx = \tfrac{1}{2}(1 - z^{-2})\, dz = \tfrac{1}{2} z^{-1}(z - z^{-1})\, dz.$$

Since

$$\tfrac{1}{2}(z - z^{-1}) = i \operatorname{Im} z = \pm i \sqrt{1-x^2},$$

this implies

$$dx = \pm i\, z^{-1} \sqrt{1-x^2}\, dz.$$

In these equations the plus sign applies for $\operatorname{Im} z \ge 0$ and the minus sign for $\operatorname{Im} z \le 0$.

These formulas have implications for smoothness. Since $\sqrt{1-x^2} \le 1$ for all $x \in [-1,1]$, they imply that if $f(x)$ is Lipschitz continuous, then so is $F(z)$. By a standard result in complex variables, this implies that F has a unique representation as an absolutely and uniformly convergent Laurent series on the unit circle,

$$F(z) = \frac{1}{2} \sum_{k=0}^{\infty} a_k (z^k + z^{-k}) = \sum_{k=0}^{\infty} a_k T_k(x).$$

Recall that a **Laurent series** is an infinite series in both positive and negative powers of z, and that such series in general converge in the interior of an annulus. A good treatment of Laurent series can be found in [Markushevich 1985]. Or one can derive results about F by converting them to results about Fourier series, for the Laurent series for F is equivalent to a Fourier series in the variable θ if $z = e^{i\theta}$.

The kth Laurent coefficient of an analytic function $G(z) = \sum_{k=-\infty}^{\infty} b_k z^k$ on the unit circle can be computed by the Cauchy integral formula,

$$b_k = \frac{1}{2\pi i} \int_{|z|=1} z^{-1+k} G(z)\, dz.$$

The notation $|z| = 1$ indicates that the contour consists of the unit circle traversed once in the positive (counterclockwise) direction. Here we have a function F with the special symmetry property $F(z) = F(z^{-1})$, and we also have introduced a factor $1/2$ in front of the series. Accordingly in the case of F we can compute the coefficients a_k from either of two contour integrals,

$$a_k = \frac{1}{\pi i} \int_{|z|=1} z^{-1+k} F(z)\, dz = \frac{1}{\pi i} \int_{|z|=1} z^{-1-k} F(z)\, dz,$$

with πi replaced by $2\pi i$ for $k = 0$.

In particular, we can get a formula for a_k that is symmetric in k and $-k$ by combining the two integrals like this:

$$a_k = \frac{1}{2\pi i} \int_{|z|=1} (z^{-1+k} + z^{-1-k}) F(z)\, dz = \frac{1}{\pi i} \int_{|z|=1} z^{-1}\, T_k(x) F(z)\, dz,$$

with πi replaced by $2\pi i$ for $k = 0$. Replacing $F(z)$ by $f(x)$ and $z^{-1}dz$ by $-i\,dx/(\pm\sqrt{1-x^2})$ gives

$$a_k = -\frac{1}{\pi}\int_{|z|=1}\frac{f(x)T_k(x)}{\pm\sqrt{1-x^2}}\,dx,$$

with π replaced by 2π for $k = 0$. We have now almost entirely converted to the x variable, except that the contour of integration is still the circle $|z| = 1$. When z traverses the unit circle all the way around in the positive direction, x decreases from 1 to -1 and then increases back to 1 again. At the turning point $z = x = -1$, the $\pm$ sign attached to the square root switches from $+$ to $-$. Thus instead of cancelling, the two traverses of $x \in [-1, 1]$ contribute equal halves to a_k. Converting to a single integration from -1 to 1 in the x variable multiplies the integral by $-1/2$, hence multiplies the formula for a_k by -2:

$$a_k = \frac{2}{\pi}\int_{|z|=1}\frac{f(x)T_k(x)}{\sqrt{1-x^2}}\,dx.$$

This is the result stated in the theorem. ∎

The chebfun system represents functions by their values at Chebyshev points. How does it know the right value of n? Given a set of $n+1$ samples, it converts the data to a Chebyshev expansion of degree n and examines the resulting Chebyshev coefficients. If these fall below a relative level of approximately 10^{-15}, then the grid is judged to be fine enough. For example, here are the Chebyshev coefficients of the chebfun corresponding to e^x:

```
f = exp(x);
a = chebpoly(f);
format long
a(end:-1:1)'
```

```
ans =
   1.266065877752008
   1.130318207984970
   0.271495339534077
   0.044336849848664
   0.005474240442094
   0.000542926311914
   0.000044977322954
   0.000003198436463
   0.000000199212481
   0.000000011036772
   0.000000000550590
   0.000000000024980
   0.000000000001039
```

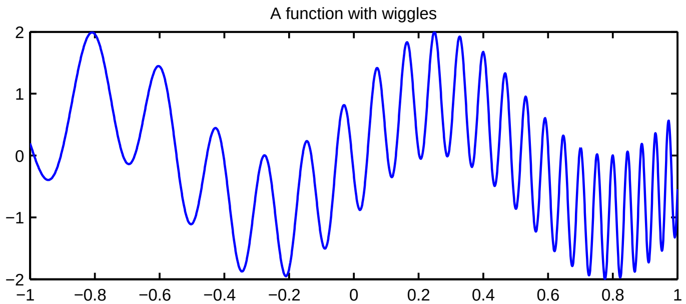


```
   0.000000000000040
   0.000000000000001
```

Notice that the last coefficient is about at the level of machine precision.

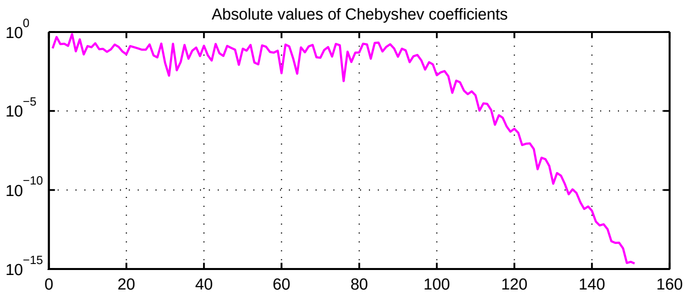
For complicated functions it is often more informative to plot the coefficients than to list them. For example, here is a function with a number of wiggles:

```
f = sin(6*x) + sin(60*exp(x));
clf, plot(f), title('A function with wiggles')
```



If we plot the absolute values of the Chebyshev coefficients, here is what we find:

```
a = chebpoly(f);
semilogy(abs(a(end:-1:1)),'m')
grid on, title('Absolute values of Chebyshev coefficients')
```



One can explain this plot as follows. Up to degree about $k = 80$, a Chebyshev series cannot resolve f accurately, for the oscillations occur on too short wavelengths. After that the series begins to converge rapidly. By the time we reach $k = 150$, the accuracy is about 15 digits, and the computed Chebyshev series is truncated there. We can find out exactly where the truncation took place with the command `length(f)`:

```
length(f)
```

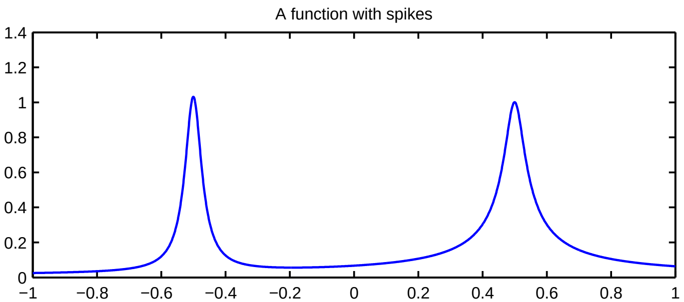
```
ans =
   151
```

This tells us that the chebfun is a polynomial interpolant through 151 points, that is, of degree 150.

Without giving all the engineering details, here is a fuller description of how the chebfun system constructs its approximation. First it calculates the polynomial interpolant through the function sampled at 9 Chebyshev points, i.e., a polynomial of degree 8, and checks whether the Chebyshev coefficients appear to be small enough. For the example just given the answer is no. Then it tries 17 points, then 33, then 65, and so on. In this case the system judges at 257 points that the Chebyshev coefficients have finally fallen to the level of rounding error. At this point it truncates the tail of terms deemed to be negligible, leaving a series of 151 terms. The corresponding degree 150 polynomial is then evaluated at 151 Chebyshev points via FFT, and these 151 numbers become the data defining this particular chebfun.

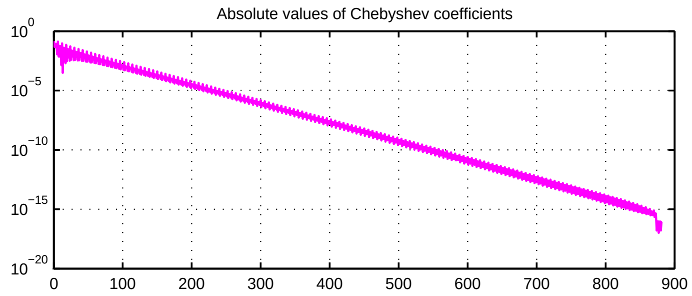
Here is another example, a function with two spikes:

```
f = 1./(1+1000*(x+.5).^2) + 1./sqrt(1+1000*(x-.5).^2);
clf, plot(f), title('A function with spikes')
```



Here are the Chebyshev coefficients of the chebfun. This time instead of `chebpoly` and `semilogy` we execute the special command `chebpolyplot`, which does the same thing.

```
chebpolyplot(f,'m'), grid on
title('Absolute values of Chebyshev coefficients')
```

Note that although it is far less wiggly, this function needs six times as many points to resolve as the previous one.

People often ask, is there anything special about Chebyshev points and Chebyshev polynomials? Could we equally well interpolate in other points and expand in other sets of polynomials? From an approximation point of view, the answer is yes, and in particular, Legendre points and Legendre polynomials have much the same power for representing a general function f, as we shall see in Chapters 15 and 16. Legendre points and polynomials are neither better than Chebyshev for approximating functions, nor worse; they are essentially the same. One can improve both Legendre and Chebyshev—by a factor of up to $\pi/2$—but to do so one must leave the class of polynomials. See Chapter 19.

Nevertheless, there is a big advantage of Chebyshev over Legendre points, and that is that one can use the FFT to go from point values to coefficients and back again. There are fast Legendre transforms that make such computations practicable, but Chebyshev remains much faster and more convenient.

[To be added: (1) Original references for Chebyshev polynomials and Theorem 3.1. (2) In particular, pin down where the notation T_k comes from.]

Exercise 3.1. An expansion coefficient. Determine numerically the coefficient of T_5 in the Chebyshev expansion of $\tan^{-1}(x)$ on $[-1, 1]$.

Exercise 3.2. Chebyshev coefficients and "rat". (a) Use chebfun to determine numerically the coefficients of the Chebyshev series for $1 + x^3 + x^4$. By inspection, identify these rational numbers. Use the Matlab command `[n,d] = rat(c)` to confirm this. (b) Use chebfun and `rat` to make good guesses as to the Chebyshev coefficients of $x^7/7 + x^9/9$.

Exercise 3.3. Dependence on wave number. (a) Calculate the length L_k of the chebfun corresponding to $f(x) = \sin(kx)$ on $[-1, 1]$ for $k = 1, 2, 4, 8, \ldots, 2^{10}$. Make a loglog plot of L_k as a function of k and comment on the result. (b) Do the same for $g(x) = 1/(1+(kx)^2)$.

Exercise 3.4. Chebyshev series of a complicated function. (a) Make chebfuns of the three functions $f(x) = \tanh(x)$, $g(x) = 10^{-5}\tanh(10x)$, $h(x) = 10^{-10}\tanh(100x)$ on $[-1,1]$, and call `chebpolyplot` to show their Chebyshev coefficients. Comment on the results. (b) Now define $s = f+g+h$ and comment on the result of `chebpolyplot` applied to s. Chebfun does not automatically chop the tail of a Chebyshev series, but applying the `simplify` command will do this. What happens with `chebpolyplot(simplify(s))`? (c) Repeat (b) but with the function $t = f + 10^{-5}g + 10^{-10}h$. What does `chebpolyplot` reveal about the difference between `simplify(t)` and `simplify(s)`?

Exercise 3.5. Orthogonality, least-squares.

Exercise 3.6. The Wiener class.

4. Interpolants, truncations, and aliasing

`ATAPformats`

Suppose $f(x)$ is a Lipschitz continuous function on $[-1,1]$ with Chebyshev expansion coefficients $\{a_k\}$ as in Theorem 3.1:

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

One degree n approximation to f is the polynomial obtained by **interpolation** in Chebyshev points:

$$p_n(x) = \sum_{k=0}^{n} c_k T_k(x).$$

Another is the polynomial obtained by **truncation** of the series at term n, whose coefficients through degree n are the same as those of f itself:

$$f_n(x) = \sum_{k=0}^{n} a_k T_k(x).$$

The relationship of the Chebyshev coefficients of f_n to those of f is obvious, and in a moment we shall that the Chebyshev coefficients of p_n have simple expressions too. In computational work generally, and in particular in the chebfun system, the polynomials $\{p_n\}$ are generally nearly as good approximations to f as $\{f_n\}$ and easier to work with, since one does not need to evaluate the integral of Theorem 3.1. The polynomials $\{f_n\}$, on the other hand, are also interesting and have received a great deal of mathematical attention over the years. In this book, most of our computations will make use of $\{p_n\}$, but many of our theorems will treat both cases. A typical example is Theorem 8.2, which

asserts that if f is analytic on $[-1,1]$, then both $\|f-f_n\|$ and $\|f-p_n\|$ decrease geometrically as $n\to\infty$.

The key to understanding $\{c_k\}$ is the phenomenon of **aliasing**, a term which originated among radio engineers early in the 20th century. On the $(n+1)$ -point Chebyshev grid, it is obvious that any function f is indistinguishable from a polynomial of degree n. But something more is true: any Chebyshev polynomial T_N, no matter how big N is, is indistinguishable on the grid from a *single* Chebyshev polynomial T_k for some k with $0\le k\le n$. We state this as a theorem.

Theorem 4.1: Aliasing of Chebyshev polynomials. *For any $n\ge 1$ and $0\le k\le n$, the following Chebyshev polynomials take the same values on the $(n+1)$-point Chebyshev grid:*

$$T_k,\ T_{2n-k},\ T_{2n+k},\ T_{4n-k},\ T_{4n+k},\ T_{6n-k},\dots.$$

Proof. Recall from the last chapter that Chebyshev polynomials on $[-1,1]$ are related to monomials on the unit circle by $T_k(x) = (z^k+z^{-k})/2$ and Chebyshev points are related to $2n$th roots of unity by $x_k = (z_k + z_k^{-1})/2$. It follows that the assertion of the theorem is equivalent to the statement that the following functions take the same values at the $2n$th roots of unity:

$$z^k + z^{-k},\ z^{2n-k}+z^{k-2n},\ z^{2n+k}+z^{-2n-k},\dots.$$

Inspection of the exponents shows that in every case, modulo $2n$, we have one exponent equal to $+k$ and the other to $-k$. The conclusion now follows from the elementary phenomenon of aliasing of monomials on the unit circle: at the $2n$th roots of unity, $z^{2\nu n}=1$ for any integer ν. ∎

Here is a numerical illustration of Theorem 4.1. Taking $n=4$, let `X` be the Chebyshev grid with $n+1$ points and let $T\{1\},\dots,T\{10\}$ be the first 10 Chebyshev polynomials:

```
n = 4; X = chebpts(n+1);
for k = 1:10
    T{k} = chebpoly(k);
end
```

Then T_3 and T_5 are the same on the grid:

```
disp([T{3}(X) T{5}(X)])
```

```
  -1.000000000000000  -1.000000000000000
   0.707106781186548   0.707106781186547
                   0                   0
```

```
  -0.707106781186548  -0.707106781186547
   1.000000000000000   1.000000000000000
```

So are T_1, T_7, and T_9:

```
disp([T{1}(X) T{7}(X) T{9}(X)])
```

```
  -1.000000000000000  -1.000000000000000  -1.000000000000000
  -0.707106781186547  -0.707106781186548  -0.707106781186547
                   0                   0                   0
   0.707106781186547   0.707106781186548   0.707106781186547
   1.000000000000000   1.000000000000000   1.000000000000000
```

As a corollary of Theorem 4.1, we can now derive the connection between $\{a_k\}$ and $\{c_k\}$. The following result can be found in [Tadmor 1986], though that is probably not the earliest reference.

Theorem 4.2: Aliasing formula for Chebyshev coefficients. *Let f be Lipschitz continuous on $[-1,1]$ and let p_n be its Chebyshev interpolant of degree n with $n \geq 1$. Let $\{a_k\}$ and $\{c_k\}$ be the Chebyshev coefficients of f and p_n, respectively. Then*

$$c_0 = a_0 + a_{2n} + a_{4n} + \cdots,$$

$$c_n = a_n + a_{3n} + a_{5n} + \cdots,$$

and for $1 \leq k \leq n-1$,

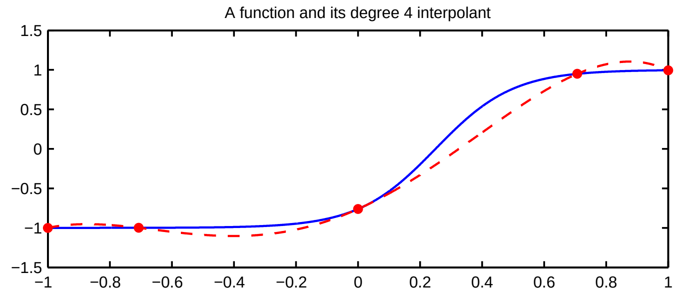
$$c_k = a_k + (a_{k+2n} + a_{k+4n} + \cdots) + (a_{-k+2n} + a_{-k+4n} + \cdots).$$

Proof. By Theorem 3.1, f has a unique Chebyshev series and it converges absolutely. Thus we can rearrange the terms of the series without affecting convergence, and in particular, each of the three series expansions written above converges, so these formulas do indeed define certain numbers $c_0, \ldots, c_n$. Taking these numbers as coefficients multiplied by the corresponding Chebyshev polynomials $T_0, \ldots, T_n$ gives us a polynomial of degree n. By Theorem 4.1, this polynomial takes the same values as f at each point of the Chebyshev grid. Thus it is the unique interpolant p_n. ∎

We can summarize Theorem 4.2 as follows. On the $n+1$ point grid, any function f is indistinguishable from a polynomial of degree n. In particular, the Chebyshev series of the polynomial interpolant to f is obtained by reassigning all the Chebyshev coefficients in the infinite series for f to their aliases of degrees 0 through n.

To illustrate Theorem 4.2, here is a function and its degree 4 Chebyshev interpolant (dashed):


```
f = tanh(4*x-1);
n = 4; pn = chebfun(f,n+1);
hold off, plot(f), hold on, plot(pn,'.--r')
title('A function and its degree 4 interpolant')
```



The first 5 Chebyshev coefficients of f,

```
a = chebpoly(f); a = a(end:-1:1)'; a(1:n+1)
```

```
ans =
  -0.166584582703135
   1.193005991160944
   0.278438064117869
  -0.239362401056012
  -0.176961398392888
```

are different from the Chebyshev coefficients of p_n,

```
c = chebpoly(pn); c = c(end:-1:1)'
```

```
c =
  -0.203351068209675
   1.187719968517890
   0.379583465333916
  -0.190237989543227
  -0.178659622412173
```

As stated in the theorem, the coefficients c_0 and c_n are given by sums of coefficients a_k with a stride of $2n$:

```
c0 = sum(a(1:2*n:end))
```

```
c0 =
  -0.203351068209675
```

```
cn = sum(a(n+1:2*n:end))
```

```
cn =
  -0.178659622412174
```

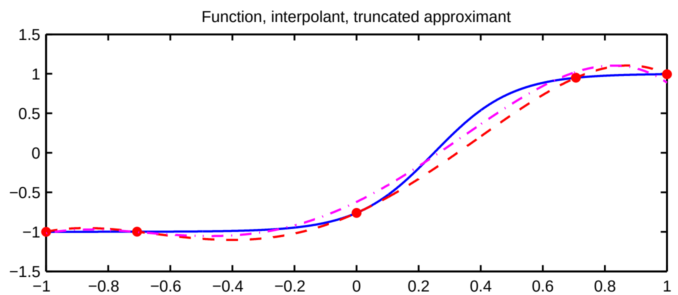
The coefficients c_1 through c_{n-1} are given by formulas involving two such sums:

```
for k = 1:n-1
  ck = sum(a(1+k:2*n:end)) + sum(a(1-k+2*n:2*n:end))
end
```

```
ck =
   1.187719968517890
ck =
   0.379583465333916
ck =
  -0.190237989543227
```

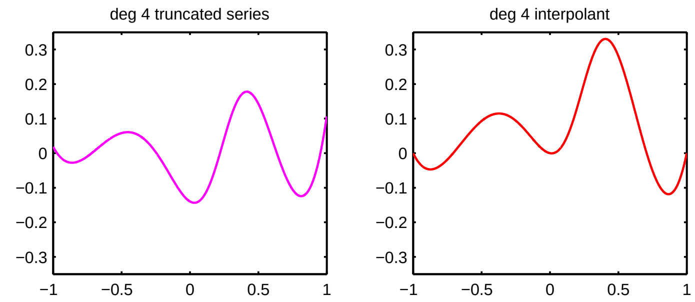
For comparison with the last figure, how does the truncated series f_n compare with the interpolant p_n as an approximation to f? In the chebfun system we can obtain f_n by computing a full set of Chebyshev coefficients down to machine precision, truncating at degree n, and constructing a corresponding chebfun using the `chebpolyval` command. Here f_n is added to the plot as a dot-dash line:

```
a = chebpoly(f);
fn = chebfun(chebpolyval(a(end-n:end)));
plot(fn,'-.m')
title('Function, interpolant, truncated approximant')
```



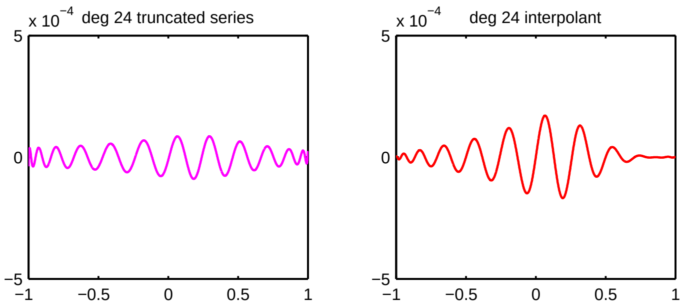
Here are the the two errors $f - f_n$ and $f - p_n$:

```
hold off
subplot(1,2,1), plot(f-fn,'m'), ylim(.35*[-1 1])
title('deg 4 truncated series')
subplot(1,2,2), plot(f-pn,'r'), ylim(.35*[-1 1])
title('deg 4 interpolant')
```



Here is the analogous plot with $n = 4$ increased to 24:

```
n = 24; pn = chebfun(f,n+1);
fn = chebfun(chebpolyval(a(end-n:end)));
subplot(1,2,1), plot(f-fn,'m'), ylim(.0005*[-1 1])
title('deg 24 truncated series')
subplot(1,2,2), plot(f-pn,'r'), ylim(.0005*[-1 1])
title('deg 24 interpolant')
```



On the basis of plots like these, one might speculate that f_n may often be a better approximation than p_n, but that the difference is small. This is indeed the case, as we shall confirm with theorems in Chapters 7 and 8.

Let us summarize where we stand. We have considered Chebyshev interpolants (Chapter 2) and Chebyshev expansions (Chapter 3) for a function $f(x)$ on

$[-1,1]$. Mathematically speaking, each coefficient of a Chebyshev expansion is equal to the value of the integral given in Theorem 3.1. This formula, however, is not needed for effective polynomial approximation, since Chebyshev interpolants are as accurate as truncations. The chebfun system only computes Chebyshev coefficients of polynomial interpolants, and this is done not by the integral but by taking the FFT of the sample values in Chebyshev points. If the degree of the interpolant is high enough that the polynomial matches f to machine precision, then the Chebyshev coefficients will match too.

Exercise 4.1. Aliasing. (a) On the $(n+1)$-point Chebyshev grid with $n = 20$, which Chebyshev polynomials T_k take the same values as T_5? (b) Use chebfun to draw plots illustrating some of these intersections.

Exercise 4.2. Fooling the chebfun constructor. (a) Construct the anonymous function `f = @(M) chebfun(@(x) 1+exp(-(M*(x-0.4)).^4))` and plot `f(10)` and `f(100)`. This function has a narrow spike of width proportional to $1/M$. Confirm this by comparing `sum(f(10))` and `sum(f(100))`. (b) Plot `length(f(M))` as a function of M for $M = 1, 2, 3, \ldots$, going into the region where the length becomes 1. What do you think is happening? (c) Let `Mmax` be the largest value of M for which the constructor behaves normally and execute `semilogy(f(Mmax)-1,'interval',[.37 .43])`. Discuss this plot and relate it to the results from `chebpts(3)`, `chebpts(9)`, `chebpts(17)`.

Exercise 4.3. Relative precision. Try Exercise 4.2 again, but now without the "1+" in the definition of `f`. The value of `Mmax` will be different, and the reason has to do with chebfun's aim of constructing each function to about 15 digits of relative precision, not absolute. Can you figure out what is happening?

5. Barycentric interpolation formula

`ATAPformats`

How does one evaluate a Chebyshev interpolant? One approach, involving $O(n \log n)$ work for a single point evaluation, would be to compute Chebyshev coefficients and use the Chebyshev series. However, there is a direct method requiring just $O(n)$ work, not based on the series expansion, that is both elegant and numerically stable. It also has the virtue of generalizing to sets of points other than Chebyshev. It is called the **barycentric formula**, and it was introduced by Salzer in 1972 [Salzer 1972, Berrut & Trefethen 2004]. We first state the formula, then illustrate its use, then give the proof.

Theorem 5.1: Barycentric interpolation in Chebyshev points. *The polynomial interpolant through data $\{f_j\}$ in Chebyshev points $\{x_j\}$ is given by*

$$p(x) = \sum_{j=0}^{n}{}' \frac{(-1)^j f_j}{x - x_j} \Bigg/ \sum_{j=0}^{n}{}' \frac{(-1)^j}{x - x_j},$$

with the special case $p(x) = f_j$ if $x = x_j$ for some j. The primes on the summation symbols signify that the terms $j = 0$ and $j = n$ are multiplied by $1/2$.

If you look at the barycentric formula, it is obvious that the function it defines interpolates the data. As x approaches one of the values x_j, one term in the numerator blows up and so does one term in the denominator. Their ratio is f_j, so this is clearly the value approached as x approaches x_j. On the other hand note that if x is equal to x_j, we can't use the formula: that would be a division of ∞ by ∞. That's why the theorem is stated with the qualification for the special case $x = x_j$.

What is not obvious is that the function defined by this formula is a polynomial, let alone a polynomial of degree n. In fact it is, as we shall prove below, but the proof takes a little work. For polynomial interpolation in points other than Chebyshev, there are other barycentric interpolation formulas with coefficients different from $(-1)^j$, going back to [Taylor 1945] and [Dupuy 1948]. The various cases are reviewed in [Berrut & Trefethen 2004], and the general formula is implemented in the chebfun overload of Matlab's `interp1` command, which we shall use in Chapters 12 and 13.

It is also not obvious that the barycentric formula is numerically stable. One might especially wonder, won't cancellation errors on a computer cause trouble if x is close to some x_j but not equal to it? In fact they do not, and the formula has been proved stable in floating point arithmetic for all $x \in [-1,1]$ [Rack & Reimer 1982, Higham 2004]. This is in marked contrast to the more familiar algorithm for polynomial interpolation via solution of a Vandermonde linear system of equations, which is exponentially unstable (Exercise 5.1).

Here is an illustration of the speed and accuracy of the barycentric formula even when n is large. Let p be the Chebyshev interpolant of degree 10^6 to the function $\sin(10^5 x)$ on $[-1,1]$:

```
ff = @(x) sin(1e5*x);
p = chebfun(ff,1000001);
```

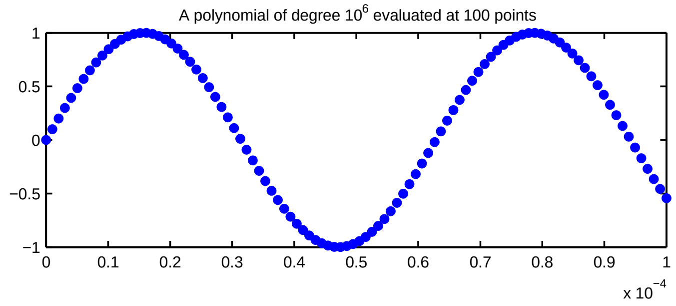
How long does it take to evaluate this interpolant in 100 points?

```
xx = linspace(0,0.0001);
tic, pp = p(xx); toc
```

```
Elapsed time is 2.203393 seconds.
```

Not bad for a million-degree polynomial! The result looks fine,

```
clf, plot(xx,pp,'.'), axis([0 0.0001 -1 1])
title('A polynomial of degree 10^6 evaluated at 100 points')
```



and it matches the target function closely.

```
format long
for j = 1:5
    r = rand;
    disp([ff(r) p(r) ff(r)-p(r)])
end
```

```
  -0.038787951022018  -0.038787951020191  -0.000000000001827
   0.997459577166873   0.997459577165562   0.000000000001311
  -0.864212757304804  -0.864212757300479  -0.000000000004326
  -0.105230083868529  -0.105230083879794   0.000000000011265
   0.911653660186257   0.911653660185483   0.000000000000774
```

The apparent loss of 4 or 5 digits of accuracy is to be expected since the derivative of this function is of order 10^5.

Now, using transplantation to the unit circle as in the proofs of Theorems 3.1 and 4.1, we derive Salzer's barycentric formula.

Proof of Theorem 5.1. We start with the observation that the function $z^n - z^{-n}$ has simple roots at the $(2n)$th roots of unity. Multiplying by $z^2 - 1$ gives a function $z^{n+2} - z^n - z^{2-n} + z^{-n}$ with simple roots at these roots of unity except double roots at ± 1. Now for z_j equal to any of the roots of unity, let us divide by $(z - z_j)(z - \bar{z}_j)$ to get

$$r_{j,n}(z) = \frac{z^{n+2} - z^n - z^{2-n} + z^{-n}}{(z - z_j)(z - \bar{z}_j)}.$$

If $z_j = \pm 1$, the division cancels the double root and we are left with a function equal to zero at all the others. If z_j is one of the roots of unity other than ± 1, the division cancels a conjugate pair of roots and again we have a function that is zero at all the other roots of unity. In fact, $r_{j,n}(z) = 2n(-1)^{n+j+1}$ if z is equal to z_j or $\bar{z}_j$ and these are $\neq \pm 1$, $r_{j,n}(z) = 4n(-1)^{n+j+1}$ if $z = z_j = \bar{z}_j = \pm 1$, and $r_{j,n}(z) = 0$ if z is one of the roots of unity other than z_j and $\bar{z}_j$ (Exercise 5.2).

Let us now define the weights

$$w_j = (2n)^{-1}(-1)^{n+j+1}$$

for $z_j = \pm 1$ and half this value for $z_j = \pm 1$. This choice implies that the function

$$w_j \frac{z^{n+2} - z^n - z^{2-n} + z^{-n}}{(z - z_j)(z - \bar{z}_j)}$$

equals 1 if z is z_j or $\bar{z}_j$ and 0 if it is one of the other roots of unity. By taking a linear combination of these functions for j from 0 to n with coefficients $\{f_j\}$, we get an interpolant through data $\{f_j\}$ satisfying the symmetry condition $f_j = f_{-j}$,

$$\sum_{j=0}^{n} w_j f_j \frac{z^{n+2} - z^n - z^{2-n} + z^{-n}}{(z - z_j)(z - \bar{z}_j)}.$$

This interpolant takes the value f_j at both the points z_j and $\bar{z}_j$, which is just what we need for transplantation to Chebyshev points in the unit interval. In fact, since $T_k(x) = \frac{1}{2}(z^k + z^{-k})$, this sum is the same as

$$p_n(x) = \sum_{j=0}^{n} w_j f_j \frac{T_{n+2}(x) - T_n(x)}{x - x_j}.$$

This equation is a representation in Lagrange form of the unique polynomial of degree $\leq n$ that interpolates the data $\{f_j\}$ in the Chebyshev points $\{x_j\}$.

A final observation completes the proof. Let the expression just given be divided by the constant function 1 expressed in the same form. This will not change its value, and the interpolant becomes

$$p_n(x) = \sum_{j=0}^{n} w_j f_j \frac{T_{n+2}(x) - T_n(x)}{x - x_j} \Bigg/ \sum_{j=0}^{n} w_j \frac{T_{n+2}(x) - T_n(x)}{x - x_j}.$$

Cancelling the common factor $T_{n+2}(x) - T_n(x)$ gives

$$p_n(x) = \sum_{j=0}^{n} \frac{w_j f_j}{x - x_j} \bigg/ \sum_{j=0}^{n} \frac{w_j}{x - x_j} \, .$$

A common factor $(2n)^{-1}(-1)^{n+1}$ still remains in the weights w_j. If this is cancelled, the summation turns into a summation with a prime because the points $j = 0$ and n have half weights, and we are left with the formula stated in the theorem. ∎

Polynomial interpolation is an old subject, going back at least to Newton, who devised an interpolation formula based on divided differences. The barycentric formula is an example of a *Lagrange interpolation formula,* in which the interpolant is written as a linear combination of cardinal functions that are zero at all the interpolation points except one. Lagrange considered such interpolations in 1795, but the same idea had been treated by Waring in 1779 and Euler in 1783 [Waring 1779].

[To be added: (1) Barycentric formula for general points.]

Exercise 5.1. Instability of Vandermonde interpolation. The best-known algorithm for polynomial interpolation, unlike the barycentric formula, is unstable. This is the method implemented in Matlab's `polyfit` command, in which one forms a Vandermonde matrix of sampled powers of x and solves a corresponding linear system of equations. (In [Trefethen 2000], for example, this unstable method is used repeatedly, forcing the values of n employed to be kept not too large.) (a) Explore this instability by comparing a chebfun evaluation of $p(0)$ with the result of `polyval(polyfit(xx,f(xx),n),0)` where `f = @(x) cos(k*x)` for $k = 0, 10, 20, \ldots, 100$ and n is the degree of the corresponding chebfun. (b) Examining Matlab's `polyfit` code as appropriate, construct the Vandermonde matrices V for each of these 11 problems and compute their condition numbers. By contrast, the underlying Chebyshev interpolation problem is well-conditioned.

Exercise 5.2. Confirmation of values in proof of Theorem 5.1. In the proof of Theorem 5.1, values were stated for the function $r_{j,n}$ at the $(2n)$th roots of unity. (a) Show that $r_{j,n}(z) = 0$ if z is one of the roots of unity other than z_j and $\bar{z}_j$. (b) Use L'Hopital's rule and the fact that $z_j^n = (-1)^j$ to show that $r_{j,n}(z) = 2n(-1)^{n+j+1}$ if z is equal to z_j or $\bar{z}_j$ and these are $\neq \pm 1$. (c) Show that $r_{j,n}(z) = 4n(-1)^{n+j+1}$ if $z = z_j = \bar{z}_j = \pm 1$.

Exercise 5.3. Interpolating the sign function. Use `x = chebfun('x')`, `f = sign(x)` to construct the sign function on $[-1, 1]$ and `p = chebfun('sign(x)',10000)` to construct its interpolant in 10000 Chebyshev points. Explore the difference in the interesting region by defining `d = f-p`, `d = d{-0.002,0.002}`. What is the maximum value of `d`? In what subset of $[-1, 1]$ is it smaller than 0.5 in absolute value?

6. The Weierstrass Approximation Theorem

`ATAPformats`

Every continuous function on a bounded interval can be approximated to arbitrary accuracy by polynomials. This is the famous Weierstrass Approximation Theorem, proved by Karl Weierstrass when he was 70 years old [Weierstrass 1885]. The theorem was independently discovered at about the same time, nearly, by Carl Runge: as pointed out by Phragmén and Mittag-Leffler, it can be derived as a consequence of results Runge published in a pair of papers in 1885 and 1886 [Runge 1885 & 1885/1886].

Here and throughout this book, except where indicated otherwise, $\|\cdot\|$ denotes the supremum norm on $[-1, 1]$.

Theorem 6.1: Weierstrass Approximation Theorem. *Let f be a continuous function on $[-1, 1]$ and let $\varepsilon > 0$ be arbitrary. Then there exists a polynomial p such that*

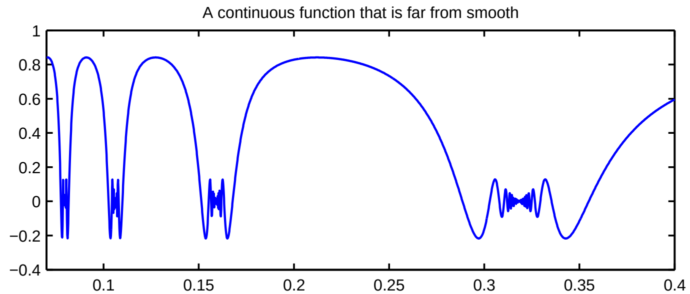
$$\|f - p\| < \varepsilon.$$

Proof. We shall not give a proof in detail. However, here is an outline of the beautiful proof from Weierstrass's original paper. First, extend $f(x)$ to a continuous function $\tilde{f}$ with compact support on the whole real line. Now, take $\tilde{f}$ as initial data at $t = 0$ for the diffusion equation $\partial u/\partial t = \partial^2 u/\partial x^2$ on the real line. It is known that by convolving $\tilde{f}$ with the Gaussian kernel $\phi(x) = e^{-x^2/4t}/\sqrt{4\pi t}$, we get a solution to this partial differential equation that converges uniformly to f as $t \to 0$, and thus can be made arbitrarily close to f on $[-1, 1]$ by taking t small enough. On the other hand, since $\tilde{f}$ has compact support, for each $t > 0$ this solution is an integral over a bounded interval of entire functions and thus itself an entire function, that is, analytic throughout the complex plane. Therefore it has a convergent Taylor series on $[-1, 1]$, which can be truncated to give polynomial approximations of arbitrary accuracy. ∎

For a fuller presentation of the argument just given as "one of the most amusing applications of the Gaussian kernel," where the result is stated for the more general case of a function of several variables approximated by multivariate polynomials, see Chapter 4 of [Folland 1995]. Many other proofs are also known, including early ones due to Picard (1891), Lerch (1892), Volterra (1897), Lebesgue (1898), Mittag-Leffler (1900), Landau (1908), Jackson (1911), Bernstein (1912), and Montel (1918). This long list gives an idea of the great amount of mathematics stimulated by Weierstrass's theorem and the significant role it played in the development of analysis in the early 20th century.

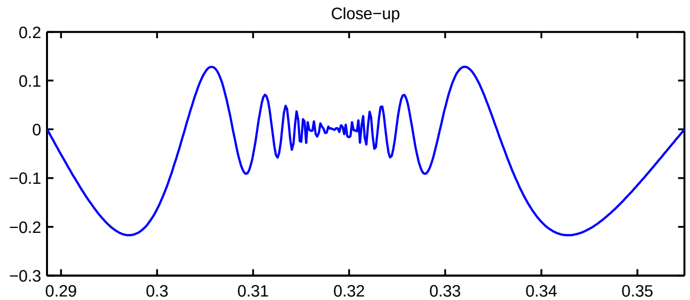
Weierstrass's theorem establishes that even extremely non-smooth functions can be approximated by polynomials, functions like $x\sin(x^{-1})$ or even $\sin(x^{-1})\sin(1/\sin(x^{-1}))$. The latter function has an infinite number of points near which it oscillates infinitely often, as we begin to see from this plot over the range $[0.07, 0.4]$. In this calculation the chebfun system is called with a user-prescribed number of interpolation points, 30000, since the usual adaptive procedure has no chance of resolving the function to machine precision with a practicable number of points.

```
f = chebfun(@(x) sin(1./x).*sin(1./sin(1./x)),[.07 .4],3e4);
plot(f), xlim([.07 .4])
title('A continuous function that is far from smooth')
```



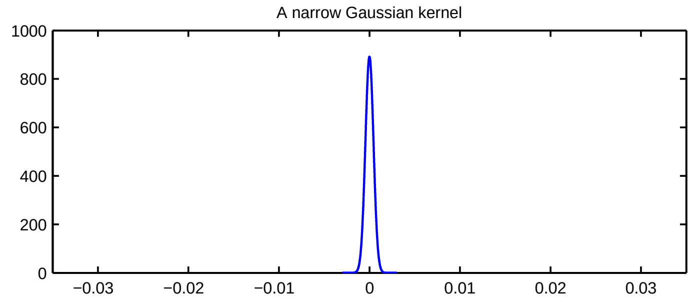
We can illustrate the idea of Weierstrass's proof by showing the convolution of this complicated function with a Gaussian. Here is the same function f recomputed over a subinterval extending from one of its zeros to another:

```
r = roots(f{.27,.37});
a = min(r); b = max(r);
f2 = chebfun(@(x) sin(1./x).*sin(1./sin(1./x)),[a,b],2e3);
plot(f), xlim([a b]), title('Close-up')
```



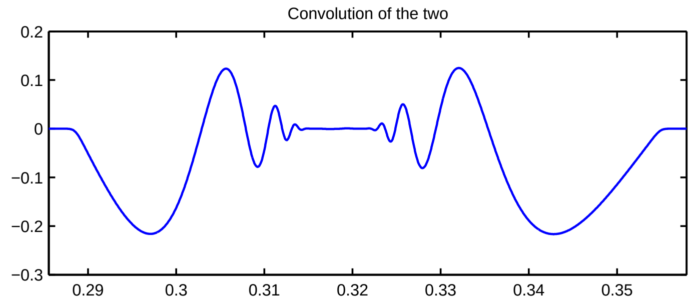
Here is a narrow Gaussian.

```
t = 1e-7;
phi = chebfun(@(x) exp(-x.^2/(4*t))/sqrt(4*pi*t),[-.003,.003]);
plot(phi), xlim([-.035 .035])
title('A narrow Gaussian kernel')
```



Convolving the two gives a smoothed version of f.

```
f3 = conv(f2,phi);
plot(f3), xlim([a-.003,b+.003])
title('Convolution of the two')
```



This is an entire function, readily approximated by polynomials.

For all its beauty, power, and importance, Weierstrass's theorem has in some respects served as an unfortunate distraction. Since we know that even troublesome functions can be approximated by polynomials, it is hard to resist asking, how can we do it? A famous result of Faber in 1914 asserts that there is no set of interpolation points, Chebyshev or otherwise, that achieves convergence as $n \to \infty$ for all f [Faber 1914]. So it becomes tempting to look at approximation methods that go beyond interpolation, and to warn people that interpolation is not enough, and to try to characterize exactly what minimal properties of f

suffice to ensure that interpolation will work after all. A great deal is known about these subjects. The trouble with this line of research is, for almost all the functions encountered in practice, Chebyshev interpolation works beautifully! Weierstrass's theorem has encouraged mathematicians over the years to pay too much attention to pathological functions at the edge of discontinuity, leading to the bizarre and unfortunate situation where many books on numerical analysis caution their readers that interpolation may fail without mentioning that for functions with a bit of smoothness, it succeeds outstandingly. For a discussion of the history of such misrepresentations and misconceptions, see Chapter 12.

[To be added: (1) Can we speed up `conv`?]

Exercise 6.1. A pathological function of Weierstrass. Weierstrass was one of the first to give an example of a function continuous but nowhere differentiable on $[-1,1]$, and it is one of the early examples of a fractal [Weierstrass 1872]:

$$w(x) = \sum_{k=0}^{\infty} 2^{-k}\cos(3^k x).$$

(a) Construct a chebfun `w7` corresponding to this series truncated at $k=7$. Plot `w7`, its derivative (use `diff`), and its indefinite integral (`cumsum`). What is the degree of the polynomial defining this chebfun? (b) Prove that w is continuous. (You can use the Weierstrass M-test. In this and the next part, you are free to look up literature for help.) (c) Prove that w is nondifferentiable at every point $x\in[-1,1]$.

7. Convergence for differentiable functions

`ATAPformats`

The principle mentioned at the end of the last chapter might be regarded as the fundamental fact of approximation theory: the smoother a function, the faster its approximants converge as $n\to\infty$. Connections of this kind were considered in the early years of the 20th century by three of the founders of approximation theory: Charles de la Vallée Poussin (1866–1962), a mathematician at Louvain in Belgium, Serge Bernstein (1880–1968), a Ukrainian mathematician who had studied with Hilbert in Göttingen, and Dunham Jackson (1888–1946), an American student of Landau's also at Göttingen. (Henri Lebesgue in France (1875–1941) also proved some of the early results. For comments on the history see [Goncharov 2000, Steffens 2006].) Bernstein made the following comment concerning best approximation errors in his summary article for the International Congress of Mathematicians in 1912 [Bernstein 1912a].

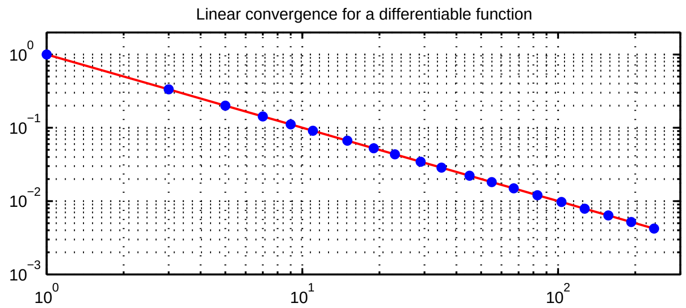
Le fait général qui se dégage de cette étude est l'existence d'une liaison des plus intimes entre les propriétés différentielles de la fonction $f(x)$ et la loi asymptotique de la decroissance des nombres positifs $E_n[f(x)]$.

[*The general fact which emerges from this study is the existence of a very intimate connection between the differential properties of the function $f(x)$ and the asymptotic rate of decrease of the positive numbers $E_n[f(x)]$.*]

In this and the next chapter our aim is to make the smoothness–approximability link precise in the context of Chebyshev truncations and interpolants. Everything here is analogous to results for Fourier analysis of periodic functions, and indeed, the whole theory of Chebyshev interpolation can be regarded as a transplant to nonperiodic functions on $[-1,1]$ of the theory of trigonometric interpolation of periodic functions on $[-\pi,\pi]$.

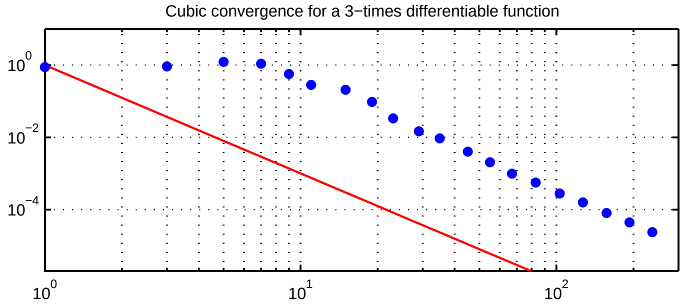
Suppose a function f is k times differentiable on $[-1,1]$, possibly with jumps in the kth derivative, and you look at the convergence of its Chebyshev interpolants as n approaches ∞, measuring error in the ∞-norm. You will typically see convergence at the rate $O(n^{-k})$. We can explore this effect readily in the chebfun system. For example, the function $f(x)=|x|$ is once differentiable with a jump in the derivative at $x=0$, and the convergence curve nicely matches n^{-1} (shown as a straight line). Actually the match is more than just "nice" in this case—it is exact, with p_n taking its maximal error at the value $p(0)=1/n$ for odd n. (For even n the error is somewhat smaller.)

```
f = abs(x);
nn = 2*round(2.^(0:.3:7))-1;
ee = 0*nn;
for j = 1:length(nn)
    n = nn(j); fn = chebfun(f,n+1); ee(j) = norm(f-fn,inf);
end
hold off, loglog(nn,1./nn,'r')
grid on, axis([1 300 1e-3 2])
hold on, loglog(nn,ee,'.')
title('Linear convergence for a differentiable function')
```



Similarly, we get cubic convergence for the function $f(x) = |\sin(5x)|^3$, which is three times differentiable with jumps in the third derivative at $x = 0$ and $x = \pm\pi/5$.

```
f = abs(sin(5*x)).^3;
for j = 1:length(nn)
    n = nn(j); fn = chebfun(f,n+1); ee(j) = norm(f-fn,inf);
end
hold off, loglog(nn,nn.^-3,'r')
grid on, axis([1 300 2e-6 10])
hold on, loglog(nn,ee,'.')
title('Cubic convergence for a 3-times differentiable function')
```



Encouraged by such experiments, you might look in a book to try to find theorems about $O(n^{-k})$. If you do, you'll run into two difficulties. First, it's hard to find theorems about Chebyshev interpolants, for most of the literature is about other approximations such as best approximations (see Chapters 10 and 14) or interpolants in Chebyshev polynomial roots rather than extrema. Second, you will probably fall one power of n short! In particular, the most commonly quoted of the **Jackson theorems** asserts that if f is k times *continuously* differentiable on $[-1, 1]$, then its best polynomial approximations converge at the rate $O(n^{-k})$ [Jackson 1911; Cheney 1966, sec. 4.6]. But the first and third derivatives of the functions we just looked at, respectively, are not continuous. Thus we must settle for the zeroth and second derivatives, respectively, if we insist on continuity, so the theorem would ensure only $O(n^0)$ and $O(n^{-2})$ convergence, not the $O(n^{-1})$ and $O(n^{-3})$ that are actually observed. And it would apply to best approximations, not Chebyshev interpolants.

We can get the result we want by recognizing that most functions encountered in applications have a property that is not assumed in most theorems: **bounded variation**. A function, whether continuous or not, has bounded variation if its total variation is finite. The **total variation** is the 1-norm of the derivative (as defined if necessary in the distributional sense; see [Ziemer 1989, chap. 5] or [Evans & Gariepy 1991, sec. 5.10]). We can compute this number conveniently

with chebfuns by writing an anonymous function:

```
tv = @(f) norm(diff(f),1);
```

Here are two examples:

```
tv(x)

ans =
     2

tv(sin(10*pi*x))

ans =
  39.999999999999986
```

Here is the total variation of the derivative of $|x|$:

```
tv(diff(abs(x)))

ans =
     2
```

Here is the total variation of the third derivative of the function f from the plot above.

```
tv(diff(f,3))

ans =
    1.652783663421985e+004
```

It is the finiteness of this number that allowed the Chebyshev interpolants to this function f to converge at least as fast as $O(n^{-3})$.

To get to a precise theorem we begin with a bound on Chebyshev coefficients, an improvement (in the definition of V) of a similar result in [Trefethen 2008]. The condition of *absolute continuity* is a standard one which we shall not make detailed use of, so we will not discuss. An absolutely continuous function is equal to the integral of its derivative, which exists almost everywhere and is Lebesgue integrable.

Theorem 7.1: Chebyshev coefficients of differentiable functions. *For any integer $\nu \geq 0$, let $f, f', \ldots, f^{(\nu-1)}$ be absolutely continuous on $[-1, 1]$ with*

$f^{(\nu)}$ *of bounded variation* V*. Then for* $k \geq \nu + 1$*, the Chebyshev coefficients of* f *satisfy*

$$|a_k| \leq \frac{2V}{\pi k(k-1)\cdots(k-\nu)} \leq \frac{2V}{\pi(k-\nu)^{\nu+1}}.$$

Proof. As in the proof of Theorem 3.1, setting $x = \frac{1}{2}(z + z^{-1})$ with z on the unit circle gives

$$a_k = \frac{1}{\pi i}\int_{|z|=1} f(\tfrac{1}{2}(z+z^{-1}))\, z^{k-1}\, dz,$$

and integrating by parts with respect to z converts this to

$$a_k = \frac{-1}{\pi i}\int_{|z|=1} f'(\tfrac{1}{2}(z+z^{-1}))\, \frac{z^k}{k}\frac{dx}{dz}\, dz\,;$$

the factor dx/dz appears since f' denotes the derivative with respect to x rather than z. Suppose now $\nu = 0$, so that all we are assuming about f is that it is of bounded varation $V = \|f'\|_1$. Then we note that this integral over the upper half of the unit circle is equivalent to an integral in x; the integral over the lower half gives another such integral. Combining the two gives

$$a_k = \frac{1}{\pi i}\int_{-1}^{1} f'(x)\,\frac{z^k - \bar{z}^k}{k}\, dx = \frac{2}{\pi}\int_{-1}^{1} f'(x)\,\mathrm{Im}\,\frac{z^k}{k}\, dx,$$

and since $|z^k/k| \leq 1/k$ for $x \in [-1,1]$ and $V = \|f'\|_1$, this implies $a_k \leq 2V/\pi k$, as claimed.

If $\nu > 0$, we replace dx/dz by $\frac{1}{2}(1 - z^{-2})$ in the second formula for a_k above, obtaining

$$a_k = -\frac{1}{\pi i}\int_{|z|=1} f'(\tfrac{1}{2}(z+z^{-1}))\,\left[\frac{z^k}{2k} - \frac{z^{k-2}}{2k}\right] dz.$$

Integrating by parts again with respect to z converts this to

$$a_k = \frac{1}{\pi i}\int_{|z|=1} f''(\tfrac{1}{2}(z+z^{-1}))\,\left[\frac{z^{k+1}}{2k(k+1)} - \frac{z^{k-1}}{2k(k-1)}\right]\frac{dx}{dz}\, dz.$$

Suppose now $\nu = 1$ so that we are assuming f' has bounded variation $V = \|f''\|_1$. Then again this integral is equivalent to an integral in x,

$$a_k = \frac{-2}{\pi}\int_{-1}^{1} f''(x)\,\mathrm{Im}\left[\frac{z^{k+1}}{2k(k+1)} - \frac{z^{k-1}}{2k(k-1)}\right] dx.$$

Since the term in square brackets is bounded by $1/k(k-1)$ for $x \in [-1,1]$ and $V = \|f''\|_1$, this implies $a_k \leq 2V/\pi k(k-1)$, as claimed.

If $\nu > 1$, we continue in this fashion with a total of $\nu + 1$ integrations by parts with respect to z, in each case first replacing dx/dz by $\frac{1}{2}(1 - z^{-2})$. At the next step the term that appears in square brackets is

$$\left[\frac{z^{k+2}}{4k(k+1)(k+2)} - \frac{z^k}{4k^2(k+1)} - \frac{z^k}{4k^2(k-1)} + \frac{z^{k-2}}{4k(k-1)(k-2)}\right],$$

which is bounded by $1/k(k-1)(k-2)$ for $x \in [-1,1]$. And so on. ∎

From Theorems 3.1 and 7.1 we can derive consequences about the accuracy of Chebyshev truncations and interpolants. The second statement of the following theorem can be found as Corollary 2 in [Mastroianni & Szabados 1995], though with a bound of the form $O(n^{-\nu}V)$ rather than an explicit constant, whose appearance here so far as we know is new. The analogous result for best approximations as opposed to Chebyshev interpolants or truncations was announced in [Bernstein 1911] and proved in [Bernstein 1912c].

Theorem 7.2: Convergence for differentiable functions. *If* f *satisfies the conditions of Theorem 7.1, with* V *again denoting the total variation of* $f^{(\nu)}$*, then for any* $n > \nu$ *its Chebyshev truncations satisfy*

$$\|f - f_n\| \leq \frac{2V}{\pi\nu(n-\nu)^{\nu}}$$

and its Chebyshev interpolants satisfy

$$\|f - p_n\| \leq \frac{4V}{\pi\nu(n-\nu)^{\nu}}.$$

Proof. For the first estimate, Theorem 7.1 gives us

$$\|f - f_n\| \;\leq\; \sum_{k=n+1}^{\infty} |a_k| \;\leq\; \frac{2V}{\pi}\sum_{k=n+1}^{\infty} (k-\nu)^{-\nu-1}$$

and this sum can in turn be bounded by

$$\int_n^{\infty} (s-\nu)^{-\nu-1} ds = \frac{1}{\nu(n-\nu)^{\nu}}.$$

For the second estimate, we note that by Theorem 4.2, the Chebyshev interpolants satisfy the same bound except with coefficients $2|a_k|$ rather than $|a_k|$. ∎

Here is a way to remember the $O(n^{-\nu})$ message of Theorem 7.2. Suppose we try to approximate the step function $\mathrm{sign}(x)$ by polynomials. There is no hope of convergence, since polynomials are continuous and $\mathrm{sign}(x)$ is not, so all we can achieve is accuracy $O(1)$ as $n \to \infty$. That's the case $\nu = 0$. But now, each time we make the function "one derivative smoother," ν increases by 1 and so does the order of convergence.

How sharp is Theorem 7.2 for our example functions? In the case of $f(x) = |x|$, with $\nu = 1$ and $V = 2$, it predicts $\|f - f_n\| \leq 4/\pi(n-1)$ and $\|f - p_n\| \leq 8/\pi(n-1) \approx 2.55/(n-1)$. As mentioned above, the actual value for Chebyshev interpolation is $\|f - p_n\| = 1/n$ for odd n. The minimal possible error in polynomial approximation, with p_n replaced by the best approximation p_n^* (Chapter

10), is $\|f - p_n^*\| \sim 0.280169\ldots n^{-1}$ as $n \to \infty$ [Varga & Carpenter 1985]. So we see that the range from best approximant, to Chebyshev interpolant, to bound on Chebyshev interpolant is less than a factor of 10. The approximation of $|x|$ was a central problem studied by de la Vallée Poussin, Bernstein, and Jackson in the 1910s.

The results are similar for the other example, $f(x) = |\sin(5x)|^3$, whose third derivative, we saw, has variation $V \approx 16528$. Theorem 7.2 implies that the Chebyshev interpolants satisfy $\|f - p_n\| < 7020/(n-1)^3$, whereas in fact, we have $\|f - p_n\| \approx 309/n^3$ for large odd n and $\|f - p_n^*\| \approx 80/n^3$.

We close with a comment about Theorem 7.2. We have assumed in this theorem that $f^{(\nu)}$ is of bounded variation. A similar but weaker condition would be that $f^{(\nu-1)}$ is Lipschitz continuous (Exercise 7.2). This weaker assumption is enough to ensure $\|f - p_n^*\| = O(n^{-\nu})$ for the best approximations $\{p_n^*\}$; this is one of the Jackson theorems. On the other hand it is not enough to ensure $O(n^{-\nu})$ convergence of Chebyshev truncations and interpolants. The reason we emphasize the stronger condition with the stronger conclusion is that in practice one rarely deals with a function that is Lipschitz continuous while lacking a derivative of bounded variation, whereas one constantly deals with truncations and interpolants rather than best approximations.

Incidentally it was de la Vallée Poussin in 1908 who first showed that the strong hypothesis is enough to reach the weak conclusion: if $f^{(\nu)}$ is of bounded variation, then $\|f - p_n^*\| = O(n^{-\nu})$ for the best approximation p_n^* [de la Vallée Poussin 1908]. Three years later Jackson sharpened the result by weakening the hypothesis [Jackson 1911].

[To be added: (1) Converse of Thm 7.2. (2) Jackson and other literature?]

Exercise 7.1. Total variation. Determine numerically the total variation of $f(x) = \sin(100x)/(1 + x^2)$ on $[-1, 1]$.

Exercise 7.2. Lipschitz continuous vs. derivative of bounded variation. (a) Show that if the derivative f' of a function f has bounded variation, then f is Lipschitz continuous. (b) Show that the converse does not hold.

Exercise 7.3. Convergence for Weierstrass's function. Exercise 6.1 considered a "pathological function of Weierstrass" $w(x)$ which is continuous but nowhere differentiable on $[-1, 1]$. Use chebfun to produce plots of $\|f - f_n\|$ and $\|f - p_n\|$ accurate enough and for high enough values of n to confirm visually that convergence appears to take place as $n \to \infty$. Thus w is not one of the functions for which interpolants fail to converge, a fact we shall prove in Chapter 13 while also showing how such troublesome functions can be constructed.

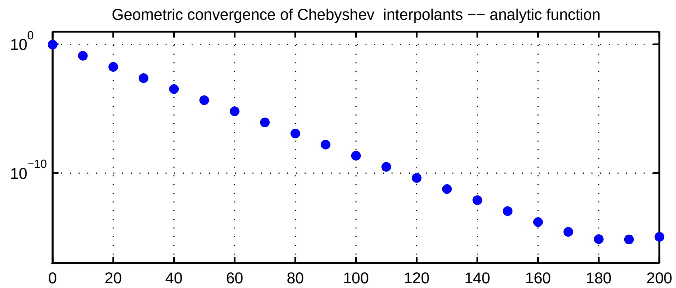
8. Convergence for analytic functions

```
ATAPformats
```

Suppose f is not just k times differentiable but infinitely differentiable and in fact analytic on $[-1, 1]$. (Recall that this means that for any $s \in [-1, 1]$, f has a Taylor series about s that converges to f in a neighborhood of s.) Then without any further assumptions we may conclude that the Chebyshev truncations and interpolants converge **geometrically**, that is, at the rate $O(C^{-n})$ for some constant $C > 1$. This means the errors will look like straight lines (or better) on a semilog scale rather than a loglog scale. This kind of connection was first announced by Bernstein in 1911, who showed that the best approximations to a function f on $[-1, 1]$ converge geometrically as $n \to \infty$ if and only if f is analytic [Bernstein 1911 & 1912c].

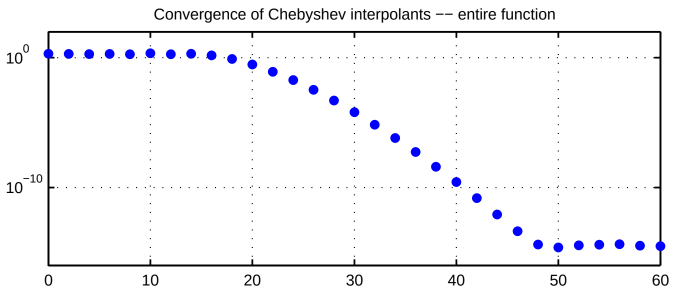
For example, for Chebyshev interpolants of the function $(1 + 25x^2)^{-1}$, often known as the **Runge function**, we get steady geometric convergence down to the level of rounding errors:

```
f = 1./(1+25*x.^2);
nn = 0:10:200;
ee = 0*nn;
for j = 1:length(nn)
    n = nn(j); fn = chebfun(f,n+1); ee(j) = norm(f-fn,inf);
end
hold off, semilogy(nn,ee,'.'), grid on, axis([0 200 1e-17 10])
title(['Geometric convergence of Chebyshev ' ...
      ' interpolants -- analytic function'])
```



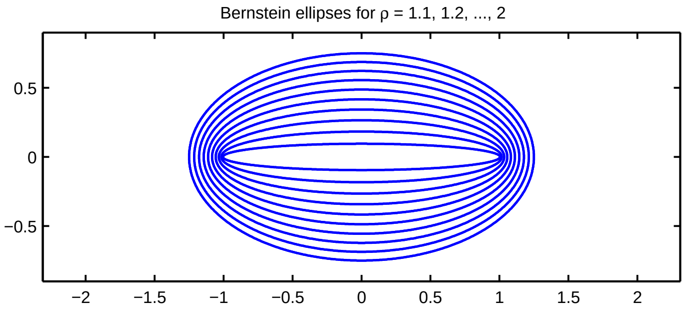
If f is analytic not just on $[-1, 1]$ but in the whole complex plane—such a function is said to be **entire**—then the convergence is even faster than geometric. Here, for the function $\cos(20x)$, the dots are not approaching a fixed straight line but a curve that gets steeper as n increases, until rounding error cuts off the progress.

```
f = cos(20*x);
nn = 0:2:60;
ee = 0*nn;
for j = 1:length(nn)
    n = nn(j); fn = chebfun(f,n+1); ee(j) = norm(f-fn,inf);
end
semilogy(nn,ee,'.'), grid on, axis([0 60 1e-16 100])
title('Convergence of Chebyshev interpolants -- entire function')
```



There are elegant theorems that explain these effects. If f is analytic on $[-1,1]$, then it can be analytically continued to a neighborhood of $[-1,1]$ in the complex plane. The bigger the neighborhood, the faster the convergence. In particular, for polynomial approximations, the neighborhoods that matter are the regions in the complex plane bounded by ellipses with foci at -1 and 1. We call these **Bernstein ellipses**, for they were introduced into approximation theory by Bernstein in 1912 [Bernstein 1912b & 1914]. It is easy to plot Bernstein ellipses: pick a number $\rho > 1$ and plot the image in the complex z-plane of the circle of radius ρ in the z-plane under the Joukowsky map $x = (z + z^{-1})/2$. We let E_r denote the open region bounded by this ellipse. Here for example are the Bernstein ellipses corresponding to $\rho = 1.1, 1.2, \ldots, 2$:

```
z = exp(2i*pi*x);
for rho = 1.1:0.1:2
    e = (rho*z+(rho*z).^(-1))/2;
    plot(e), hold on
end
ylim([-.9 .9]), axis equal
title('Bernstein ellipses for \rho = 1.1, 1.2, ..., 2')
```

It is not hard to verify that the length of the semimajor axis of E_ρ plus the length of the semiminor axis is equal to ρ.

Here is the basic bound on Chebyshev coefficients of analytic functions from which many other things follow. It first appeared in Section 61 of [Bernstein 1912c].

Theorem 8.1: Chebyshev coefficients of analytic functions. *Let a function f analytic in $[-1,1]$ be analytically continuable to the open ρ-ellipse* $E\rho$, where it satisfies $|f(z)| \le M$ for some M. Then its Chebyshev coefficients satisfy_

$$|a_k| \le 2M\rho^{-k},$$

with $|a_0| \le M$ in the case $k = 0$.

Proof. As in the proofs of Theorems 3.1, 4.1, and 5.1, we make use of the transplantation from $f(x)$ and $T_k(x)$ on $[-1,1]$ in the x-plane to $F(z)$ and $(z^k + z^{-k})/2$ on the unit circle in the z-plane, with $x = (z + z^{-1})/2$ and $F(z) = F(z^{-1}) = f(x)$. The ellipse E_ρ in the x-plane corresponds under this formula in a 1-to-2 fashion to the annulus $\rho^{-1} < |z| < \rho$ in the z-plane. By this we mean that for each x in $E_\rho \backslash [-1,1]$ there are two corresponding values of z which are inverses of one another, and both the circles $|z| = \rho$ and $|z| = \rho^{-1}$ map onto the ellipse itself. (We can no longer use the formula $x = \mathrm{Re}\, z$, which is valid only for $|z| = 1$.) The first thing to note is that if f is analytic in the ellipse, then F is analytic in the annulus since it is the composition of the two analytic functions $z \mapsto (z + z^{-1})/2$ and $x \mapsto f(x)$. Now we make use of the contour integral formula from the proof of Theorem 3.1,

$$a_k = \frac{1}{\pi i} \int_{|z|=1} z^{-1-k} F(z)\, dz,$$

with πi replaced by $2\pi i$ for $k = 0$. Suppose for a moment that F is analytic not just in the annulus but in its closure $\rho^{-1} \le |z| \le \rho$. Then we can expand the contour to $|z| = r$ without changing the value of the integral, giving

$$a_k = \frac{1}{\pi i} \int_{|z|=\rho} z^{-1-k} F(z)\, dz,$$

again with πi replaced by $2\pi i$ for $k = 0$. Since the circumference is $2\pi\rho$ and $|F(z)| \le M$, the required bound now follows from an elementary estimate. If F is analytic only in the open annulus, we can move the contour to $|z| = s$ for any $s < \rho$, leading to the same bound for any $s < \rho$ and hence also for $s = \rho$. ∎

Here are two of the consequences of Theorem 8.1. The first bound first appeared in Section 61 of [Bernstein 1912c]. I do not know where the second may have appeared.

Theorem 8.2: Convergence for analytic functions. *If f has the properties of Theorem 8.1, then for each $n \ge 0$ its Chebyshev truncations satisfy*

$$\|f - f_n\| \le \frac{2M\rho^{-n}}{\rho - 1}$$

and its Chebyshev interpolants satisfy

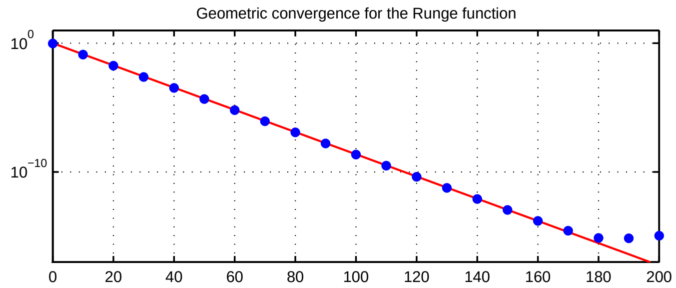
$$\|f - p_n\| \le \frac{4M\rho^{-n}}{\rho - 1}.$$

Proof. The first bound follows by estimating the sum of the coefficients $a_{n+1}, a_{n+2}, \ldots$ using Theorem 8.1. The second bound follows in the same way using also Theorem 4.2, which implies that in Chebyshev interpolation, each coefficient $a_{n+1}, a_{n+2}, \ldots$ contributes to $f - p_n$ not once but twice. ∎

We can apply Theorem 8.2 directly if f is analytic and bounded in E_ρ. If it is analytic but unbounded in E_ρ, then it will be analytic and bounded in E_s for any $s < \rho$, so we still get convergence at the rate $O(s^{-n})$ for any $s < \rho$.

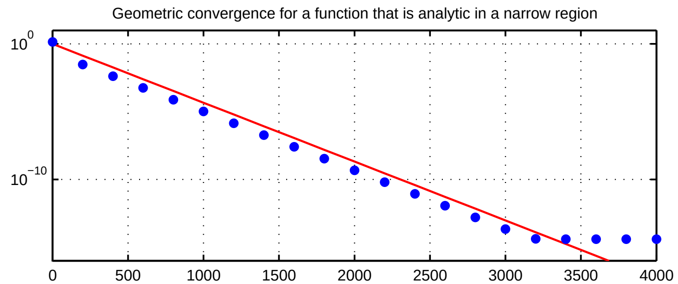
For example, the function $(1 + 25x^2)^{-1}$ considered above has poles at $\pm i/5$. The corresponding value of ρ is $(1 + \sqrt{26})/5 \approx 1.220$. The errors in Chebyshev interpolation match this rate beautifully:

```
f = 1./(1+25*x.^2);
nn = 0:10:200; ee = 0*nn;
for j = 1:length(nn)
    n = nn(j); fn = chebfun(f,n+1);
    ee(j) = norm(f-fn,inf);
end
rho = (1+sqrt(26))/5;
hold off, semilogy(nn,rho.^(-nn),'-r')
hold on, semilogy(nn,ee,'.')
grid on, axis([0 200 1e-17 10])
title('Geometric convergence for the Runge function')
```

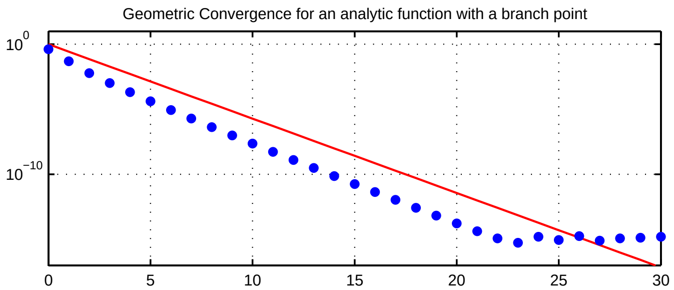
Here is a more extreme but entirely analogous example: $\tanh(50\pi x)$, with poles at $\pm 0.01i$. These poles are so close to $[-1, 1]$ that the convergence is much slower, but it is still robust. The only difference in this code segment is that `norm(f-fn,inf)`, a relatively slow chebfun operation that depends on finding zeros of the derivative of `f-fn`, has been replaced by the default 2-norm `norm(f-fn)`, which is quick. The exponential decay rates are the same.

```
f = tanh(50*pi*x);
nn = 0:200:4000; ee = 0*nn;
for j = 1:length(nn)
    n = nn(j); fn = chebfun(f,n+1); ee(j) = norm(f-fn);
end
rho = (1+sqrt(10001))/100;
hold off, semilogy(nn,rho.^(-nn),'-r')
hold on, semilogy(nn,ee,'.')
grid on, axis([0 4000 1e-16 10])
title(['Geometric convergence for a function ' ...
       'that is analytic in a narrow region'])
```



For another example, the function $\sqrt{2 - x}$ has a branch point at $x = 2$, corresponding to $\rho = 2 + \sqrt{3}$. Again we see a good match, with the curve gradually bending over to the expected slope as $n \to \infty$.

```
f = sqrt(2-x);
nn = 0:30; ee = 0*nn;
for j = 1:length(nn)
    n = nn(j); fn = chebfun(f,n+1); ee(j) = norm(f-fn,inf);
end
rho = 2+sqrt(3);
hold off, semilogy(nn,rho.^(-nn),'-r')
hold on, semilogy(nn,ee,'.')
grid on, axis([0 30 1e-17 10])
title(['Geometric Convergence for an analytic ' ...
       'function with a branch point'])
```



We conclude this section by stating a converse of Theorem 8.2, also due to Bernstein [Bernstein 1912c, Section 9]. The converse is not quite exact: Theorem 8.2 assumes analyticity and boundedness in E_r, whereas the conclusion of Theorem 8.3 is analyticity in E_r but not necessarily boundedness.

Theorem 8.3: Converse of Theorem 8.2. *Suppose f is a function on $[-1,1]$ for which there exist polynomial approximations $\{q_n\}$ satisfying*

$$\|f - q_n\| \le C\rho^{-n}, \quad n \ge 0$$

for some constants $\rho > 1$ and $C > 0$. Then f can be analytically continued to an analytic function in the open ρ-ellipse E_ρ.

Proof. The assumption implies that the polynomials $\{q_n\}$ satisfy $\|q_n - q_{n-1}\| \le 2\,C\,\rho^{1-n}$ on $[-1,1]$. Since $q_n - q_{n-1}$ is a polynomial of degree n, it can be shown that this implies $\|q_n - q_{n-1}\|_{E_s} \le 2\,Cs^n\rho^{1-n}$ for any $s > 1$, where $\|\cdot\|_{E_s}$ is the supremum norm on the s-ellipse E_s (this estimate is one of **Bernstein's inequalities**, from Section 9 of [Bernstein 1912c]). For $s < \rho$, this gives us a representation for f in E_s as a series of analytic functions,

$$f = q_0 + (q_1 - q_0) + (q_2 - q_1) + \cdots,$$

which according to the Weierstrass M test is uniformly convergent. According to another well-known theorem of Weierstrass, this implies that the limit is a bounded analytic function [Ahlfors 1953, Markushevich 1985]. Since this is true for any $s < \rho$, the analyticity applies throughout E_ρ. ∎

[To be added: (1) Reference for Thm 8.2(b). (2) Hermite integral formula as an alternative proof.]

Exercise 8.1. A Chebyshev series. With `x = chebfun('x')`, execute `chebpolyplot(sin(100*(x-.1))+.01*tanh(20*x))`. Explain the various features of the resulting plot as quantitatively as you can.

Exercise 8.2. Interpolation of an entire function. The function $f(x) = \exp(-x^2)$ is analytic throughout the complex x-plane, so Theorem 8.2 can be applied for any value of the parameter $\rho > 1$. Produce a semilog plot of $\|f - p_n\|$ as a function of n together with lines corresponding to the upper bound of the theorem for $r = 1.1, 1.2, 1.3, \ldots, 5$. How well do your data fit the bounds?

Exercise 8.3. Convergence rates for different functions. Based on the theorems of this section, what can you say about the convergence as $n \to \infty$ of the Chebyshev interpolants to (a) $\log((x+3)/4)/(x-1)$, (b) $\int_0^x \cos(t^2)\,dt$, (c) $\tan(\tan(x))$, (d) $(1+x)\log(1+x)$?

9. The Gibbs phenomenon

`ATAPformats`

Polynomial interpolants and truncations oscillate and overshoot near discontinuities. We have observed this **Gibbs phenomenon** already in Chapter 2, and now we shall look at it more carefully. We shall see that the Gibbs effect for interpolants can be regarded as a consequence of the oscillating inverse-linear tails of cardinal polynomials, i.e., interpolants of Kronecker delta functions. Chapter 13 will show that these same tails, combined together in a different manner, are also the origin of Lebesgue constants of size $O(\log n)$, with implications throughout approximation theory.

To start with let us consider the function `sign(x)`, which we interpolate in $n+1 = 10$ or 20 Chebyshev points. We take n to be odd to avoid including a value 0 at the middle of the step.

```
f = sign(x);
subplot(1,2,1), hold off, plot(f,'k'), hold on, grid on
f9 = chebfun(f,10); plot(f9,'.-'), title('n = 9')
subplot(1,2,2), hold off, plot(f,'k'), hold on, grid on
f19 = chebfun(f,20); plot(f19,'.-'), title('n = 19');
```