A Concise Introduction to Numerical Analysis

Douglas N. Arnold

Institute for Mathematics and its Applications, 400 Lind Hall, 207 Church Street S.E., University of Minnesota, Minneapolis, MN 55455

E-mail address: arnold@ima.umn.edu

URL: http://www.ima.umn.edu/ãrnold/

1991 *Mathematics Subject Classification.* Primary 65-01

©1999, 2000, 2001 by Douglas N. Arnold. All rights reserved. Not to be disseminated without explicit permission of the author.

Preface

These notes were prepared for use in teaching a one-year graduate level introductory course on numerical analysis at Penn State University. The author taught the course during the 1998–1999 academic year (the first offering of the course), and then again during the 2000–2001 academic year. They are still being put into final form, and cannot be used without express permission of the author.

Douglas N. Arnold

Contents

CHAPTER 1

Approximation and Interpolation

1. Introduction and Preliminaries

The problem we deal with in this chapter is the approximation of a given function by a simpler function. This has many possible uses. In the simplest case, we might want to evaluate the given function at a number of points, and an algorithm for this, we construct and evaluate the simpler function. More commonly the approximation problem is only the first step towards developing an algorithm to solve some other problem. For example, an algorithm to compute a definite integral of the given function might consist of first computing the simpler approximate function, and then integrating that.

To be more concrete we must specify what sorts of functions we seek to approximate (i.e., we must describe the space of possible inputs) and what sorts of simpler functions we allow. For both purposes, we shall use a *vector space* of functions. For example, we might use the vector space $C(I)$, the space of all continuous functions on the closed unit interval $I = [0, 1]$, as the source of functions to approximate, and the space $\mathcal{P}_n(I)$, consisting of all polynomial functions on I of degree at most n, as the space of simpler functions in which to seek the approximation. Then, given $f \in C(I)$, and a polynomial degree n, we wish to find $p \in \mathcal{P}_n(I)$ which is close to f.

Of course, we need to describe in what sense the simpler functions is to approximate the given function. This is very dependent on the application we have in mind. For example, if we are concerned about the maximum error across the interval, the dashed line on the left of Figure 1.1 shows the best cubic polynomial approximation to the function plotted with the solid line. However if we are concerned about integrated quantities, the approximation on the right of the figure may be more appropriate (it is the best approximation with respect to the L^2 or root–mean–square norm).

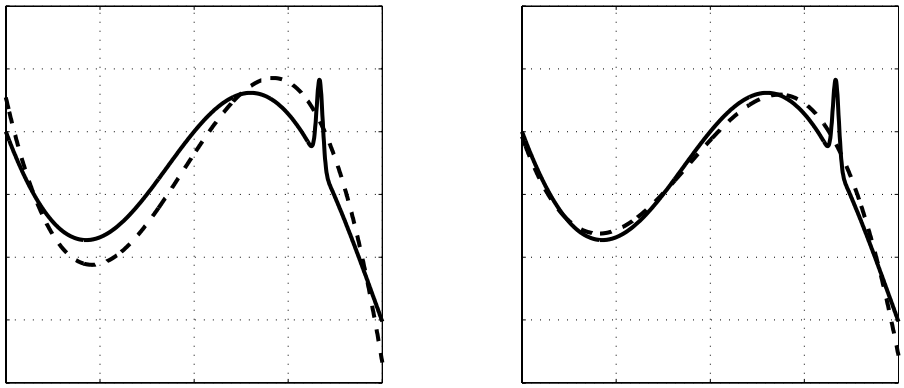
We shall always use a *norm* on the function space to measure the error. Recall that a norm on a vector space V is mapping which associated to any $f \in V$ a real number, often denoted $\|f\|$ which satisfies the homogeneity condition $\|cf\| = |c|\|f\|$ for $c \in \mathbb{R}$ and $f \in V$, the triangle inequality $\|f + g\| \leq \|f\| + \|g\|$, and which is strictly positive for all non-zero f. If we relax the last condition to just $\|f\| \geq 0$, we get a seminorm.

Now we consider some of the most important examples. We begin with the finite dimensional vector space $\mathbb{R}^n$, mostly as motivation for the case of function spaces.

(1) On $\mathbb{R}^n$ we may put the l^p norm, $1 \leq p \leq \infty$

$$\|x\|_{l^p} = \Big(\sum_{i=1}^{n} |x_i|^p\Big)^{1/p}, \quad \|x\|_{l^\infty} = \sup_{1\leq i\leq n} |x_i|.$$

FIGURE 1.1. The best approximation depends on the norm in which we measure the error.



(The triangle inequality for the l^p norm is called Minkowski's inequality.) If $w_i > 0$, $i = 1, \ldots, n$, we can define the weighted l^p norms

$$\|x\|_{w,p} = \Big(\sum_{i=1}^{n} w_i |x_i|^p\Big)^{1/p}, \quad \|x\|_{w,\infty} = \sup_{1\le i\le n} w_i|x_i|.$$

The various l^p norms are *equivalent* in the sense that there is a positive constant C such that

$$\|x\|_{l^p} \le C\|x\|_{l^q}, \quad \|x\|_{l^q} \le C\|x\|_{l^p}, \quad x \in \mathbb{R}^n.$$

Indeed *all* norms on a finite dimensional space are equivalent. Note also that if we extend the weighted l^p norms to allow non-negative weighting functions which are not strictly positive, we get a seminorm rather than a norm.

(2) Let $I = [0, 1]$ be the closed unit interval. We define $C(I)$ to be the space of continuous functions on I with the L^∞ norm,

$$\|f\|_{L^\infty(I)} = \sup_{x\in I} |f(x)|.$$

Obviously we can generalize I to any compact interval, or in fact any compact subset of $\mathbb{R}^n$ (or even more generally). Given a positive bounded weighting function $w : I \to (0, \infty)$ we may define the weighted norm

$$\|f\|_{w,\infty} = \sup_{x\in I}[w(x)|f(x)|].$$

If we allow w to be zero on parts of I this still defines a seminorm. If we allow w to be unbounded, we still get a norm (or perhaps a seminorm if w vanishes somewhere), but only defined on a subspace of $C(I)$.

(3) For $1 \le p < \infty$ we can define the $L^p(I)$ norm on $C(I)$, or, given a positive weight function, a weighted $L^p(I)$ norm. Again the triangle inequality, which is not obvious, is called Minkowski's inequality. For $p < q$, we have $\|f\|_{L^p} \le \|f\|_{L^q}$, but these norms are not equivalent. For $p < \infty$ this space is not *complete* in that there may exist a sequence of functions f_n in $C(I)$ and a function f *not* in $C(I)$ such that $\|f_n - f\|_{L^p}$ goes to zero. *Completing* $C(I)$ in the $L^p(I)$ norm leads to function space $L^p(I)$. It is essentially the space of all functions for which $\|f\|_p < \infty$, but there are some subtleties in defining it rigorously.

(4) On $C^n(I)$, $n \in \mathbb{N}$ the space on n times continuously differentiable functions, we have the seminorm $|f|_{W^n_\infty} = \|f^{(n)}\|_\infty$. The norm in $C^n(I)$ is given by

$$\|f\|_{W^n_\infty} = \sup_{0\le k\le n} |f|_{W^k_\infty}.$$

(5) If $n \in \mathbb{N}$, $1 \le p < \infty$, we define the *Sobolev seminorm* $|f|_{W^n_p} := \|f^{(n)}\|_{L^p}$, and the *Sobolev norm* by

$$\|f\|_{W^n_p} := (\sum_{k=0}^{n} |f|^p_{W^k_p})^{1/p}.$$

We are interested in the approximaton of a given function f, defined, say on the unit interval, by a "simpler" function, namely by a function belonging to some particular subspace S of C^n which we choose. (In particular, we will be interested in the case $S = \mathcal{P}_n(I)$, the vector space of polynomials of degree at most n restricted to I.) We shall be interested in two sorts of questions:

- How good is the best approximation?
- How good are various computable approximation procedures?

In order for either of these questions to make sense, we need to know what we mean by good. We shall always use a norm (or at least a seminorm) to specify the goodness of the approximation. We shall take up the first question, the theory of best approximation, first. Thus we want to know about $\inf_{p\in P}|f - p|$ for some specified norm.

Various questions come immediately to mind:

- Does there exist $p \in P$ minimizing $\|f - p\|$?
- Could there exist more than one minimizer?
- Can the (or a) minimizer be computed?
- What can we say about the error?

The answer to the first question is affirmative under quite weak hypotheses. To see this, we first prove a simple lemma.

LEMMA 1.1. *Let there be given a normed vector space X and $n+1$ elements $f_0, \ldots, f_n$ of X. Then the function $\phi : \mathbb{R}^n \to \mathbb{R}$ given by $\phi(\boldsymbol{a}) = \|f_0 - \sum_{i=1}^n a_i f_i\|$ is continuous.*

PROOF. We easily deduce from the triangle inequality that $|\|f\| - \|g\|| \le \|f - g\|$. Therefore

$$|\phi(\boldsymbol{a}) - \phi(\boldsymbol{b})| \le \Big\|\sum (a_i - b_i) f_i\Big\| \le \sum |a_i - b_i| \|f_i\| \le M \sum |a_i - b_i|,$$

where $M = \max\|f_i\|$. □

THEOREM 1.2. *Let there be given a normed vector space X and a finite dimensional vector subspace P. Then for any $f \in X$ there exists $p \in P$ minimizing $\|f - p\|$.*

PROOF. Let $f_1, \ldots, f_n$ be a basis for P. The map $\boldsymbol{a} \mapsto \|\sum_{i=1}^n a_i f_i\|$ is then a norm on $\mathbb{R}^n$. Hence it is equivalent to any other norm, and so the set

$$S = \{\, \boldsymbol{a} \in \mathbb{R}^n \mid \Big\|\sum a_i f^i\Big\| \le 2\|f\| \,\},$$

is closed and bounded. We wish to show that the function $\phi : \mathbb{R}^n \to \mathbb{R}$, $\phi(\boldsymbol{a}) = \|f - \sum a_i f_i\|$ attains its minimum on $\mathbb{R}^n$. By the lemma this is a continuous function, so it certainly attains a minimum on S, say at $\boldsymbol{a}_0$. But if $\boldsymbol{a} \in \mathbb{R}^n \setminus S$, then

$$\phi(\boldsymbol{a}) \geq \|\sum a_i f_i\| - \|f\| > \|f\| = \phi(\boldsymbol{0}) \geq \phi(\boldsymbol{a}_0).$$

This shows that $\boldsymbol{a}_0$ is a global minimizer. □

A norm is called *strictly convex* if its unit ball is strictly convex. That is, if $\|f\| = \|g\| = 1$, $f \neq g$, and $0 < \theta < 1$ implies that $\|\theta f + (1-\theta)g\| < 1$. The L^p norm is strictly convex for $1 < p < \infty$, but not for $p = 1$ or ∞.

THEOREM 1.3. *Let X be a strictly convex normed vector space, P a subspace, $f \in X$, and suppose that p and q are both best approximations of f in P. Then $p = q$.*

PROOF. By hypothesis $\|f - p\| = \|f - q\| = \inf_{r \in P} \|f - r\|$. By strict convexity, if $p \neq q$, then

$$\|f - (p+q)/2\| = \|(f-p)/2 + (f-q)/2\| < \inf_{r \in P} \|f - r\|,$$

which is impossible. □

Exercise: a) Using the integral $\int (\|f\|_2 g - \|g\|_2 f)^2$, prove the Cauchy-Schwarz inequality: if $f, g \in C(I)$ then $\int_0^1 f(x)g(x)\,dx \leq \|f\|_2 \|g\|_2$ with equality if and only if $f \equiv 0$, $g \equiv 0$, or $f = cg$ for some constant $c > 0$. b) Use this to show that the triangle inequality is satisfied by the 2-norm, and c) that the 2-norm is strictly convex.

2. Minimax Polynomial Approximation

2.1. The Weierstrass Approximation Theorem and the Bernstein polynomials. We shall now focus on the case of best approximation by polynomials of degree at most n measured in the L^∞ norm (minimax approximation). Below we shall look at the case of best approximation by polynomials measured in the L^2 norm (least squares approximation).

We first show that arbitrarily good approximation is possible if the degree is high enough.

THEOREM 1.4 (Weierstrass Approximation Theorem). *Let $f \in C(I)$ and $\epsilon > 0$. Then there exists a polynomial p such that $\|f - p\|_\infty \leq \epsilon$.*

We shall give a constructive proof due to S. Bernstein. For $f \in C(I)$, $n = 1, 2, \ldots$, define $B_n f \in \mathcal{P}_n(I)$ by

$$B_n f(x) = \sum_{k=0}^{n} f\left(\frac{k}{n}\right) \binom{n}{k} x^k (1-x)^{n-k}.$$

Now

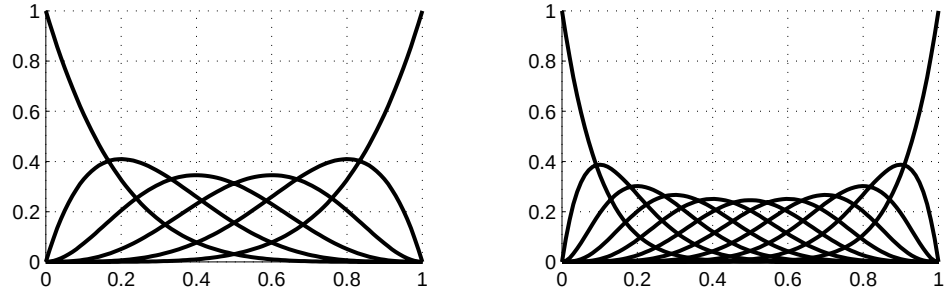
$$\sum_{k=0}^{n} \binom{n}{k} x^k (1-x)^{n-k} = [x + (1-x)]^n = 1,$$

so for each x, $B_n f(x)$ is a weighted average of the $n+1$ values $f(0)$, $f(1/n)$, $\ldots$, $f(1)$. For example,

$$B_2 f(x) = f(0)(1-x)^2 + 2f(1/2)x(x-1) + f(1)x^2.$$

The weighting functions $\binom{n}{k} x^k (1-x)^{n-k}$ entering the definition of $B_n f$ are shown in Figure 1.2. Note that for x near k/n, the weighted average weighs $f(k/n)$ more heavily than

FIGURE 1.2. The Bernstein weighting functions for $n = 5$ and $n = 10$.



other values. Notice also that $B_1 f(x) = f(0)(1-x) + f(1)x$ is just the linear polynomial interpolating f at $x = 0$ and $x = 1$.

Now B_n is a linear map of $C(I)$ into $\mathcal{P}_n$. Moreover, it follows immediately from the positivity of the Bernstein weights that B_n is a positive operator in the sense that $B_n f \geq 0$ on I if $f \geq 0$ on I. Now we wish to show that $B_n f$ converges to f in $C(I)$ for all $f \in C(I)$. Remarkably, just using the fact B_n is a positive linear operator, this follows from the much more elementary fact that $B_n f$ converges to f in $C(I)$ for all $f \in \mathcal{P}_2(I)$. This latter fact we can verify by direct computation. Let $f_i(x) = x^i$, so we need to show that $B_n f_i \to f_i$, $i = 0, 1, 2$. (By linearity the result then extends to all $f \in \mathcal{P}_2(I)$. We know that

$$\sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k} = (a+b)^n,$$

and by differentiating twice with respect to a we get also that

$$\sum_{k=0}^{n} \frac{k}{n} \binom{n}{k} a^k b^{n-k} = a(a+b)^{n-1}, \quad \sum_{k=0}^{n} \frac{k(k-1)}{n(n-1)} \binom{n}{k} a^k b^{n-k} = a^2 (a+b)^{n-2}.$$

Setting $a = x$, $b = 1 - x$, expanding

$$\frac{k(k-1)}{n(n-1)} = \frac{n}{n-1} \frac{k^2}{n^2} - \frac{1}{n-1} \frac{k}{n}$$

in the last equation, and doing a little algebra we get that

$$B_n f_0 = f_0, \quad B_n f_1 = f_1, \quad B_n f_2 = \frac{n-1}{n} f_2 + \frac{1}{n} f_1,$$

for $n = 1, 2, \ldots$.

Now we derive from this convergence for all continuous functions.

THEOREM 1.5. *Let $B_1, B_2, \ldots$ be any sequence of linear positive operators from $C(I)$ into itself such that $B_n f$ converges uniformly to f for $f \in \mathcal{P}_2$. Then $B_n f$ converges uniformly to f for all $f \in C(I)$.*

PROOF. The idea is that for any $f \in C(I)$ and $x_0 \in I$ we can find a quadratic function q that is everywhere greater than f, but for which $q(x_0)$ is close to $f(x_0)$. Then, for n sufficiently large $B_n q(x_0)$ will be close to $q(x_0)$ and so close to $f(x_0)$. But $B_n f$ must be less than $B_n q$. Together these imply that $B_n f(x_0)$ can be at most a little bit larger than $f(x_0)$. Similarly we can show it can be at most a little bit smaller than $f(x_0)$.

Since f is continuous on a compact set it is uniformly continuous. Given $\epsilon > 0$, choose $\delta > 0$ such that $|f(x_1) - f(x_2)| \leq \epsilon$ if $|x_1 - x_2| \leq \delta$. For any x_0, set

$$q(x) = f(x_0) + \epsilon + 2\|f\|_\infty (x - x_0)^2/\delta^2.$$

Then, by checking the cases $|x - x_0| \leq \delta$ and $|x - x_0| \geq \delta$ separately, we see that $q(x) \geq f(x)$ for all $x \in I$.

Writing $q(x) = a + bx + cx^2$ we see that we can write $|a|, |b|, |c| \leq M$ with M depending on $\|f\|$, ϵ, and δ, but not on x_0. Now we can choose N sufficiently large that

$$\|f_i - B_n f_i\| \leq \frac{\epsilon}{M}, \quad i = 0, 1, 2,$$

for $n \geq N$, where $f_i = x^i$. Using the triangle inequality and the bounds on the coefficients of q, we get $\|q - B_n q\| \leq 3\epsilon$. Therefore

$$B_n f(x_0) \leq B_n q(x_0) \leq q(x_0) + 3\epsilon = f(x_0) + 4\epsilon.$$

Thus we have shown: given $f \in C(I)$ and $\epsilon > 0$ there exists $N > 0$ such that $B_n f(x_0) \leq f(x_0) + 4\epsilon$ for all $n \geq N$ and all $x_0 \in I$.

The same reasoning, but using $q(x) = f(x_0) - \epsilon - 2\|f\|_\infty (x - x_0)^2/\delta^2$ implies that $B_n f(x_0) \geq f(x_0) - 4\epsilon$, and together these complete the theorem. □

From a practical point of view the Bernstein polynomials yield an approximation procedure which is very robust but very slow. By robust we refer to the fact that the procedure convergence for any continuous function, no matter how bad (even, say, nowhere differentiable). Moreover if the function is C^1 then not only does $B_n f$ converge uniformly to f, but $(B_n f)'$ converges uniformly to f' (i.e., we have convergence of $B_n f$ in $C^1(I)$, and similarly if f admits more continuous derivatives. However, even for very nice functions the convergence is rather slow. Even for as simple a function as $f(x) = x^2$, we saw that $\|f - B_n f\| = O(1/n)$. In fact, refining the argument of the proof, one can show that this same linear rate of convergence holds for all C^2 functions f:

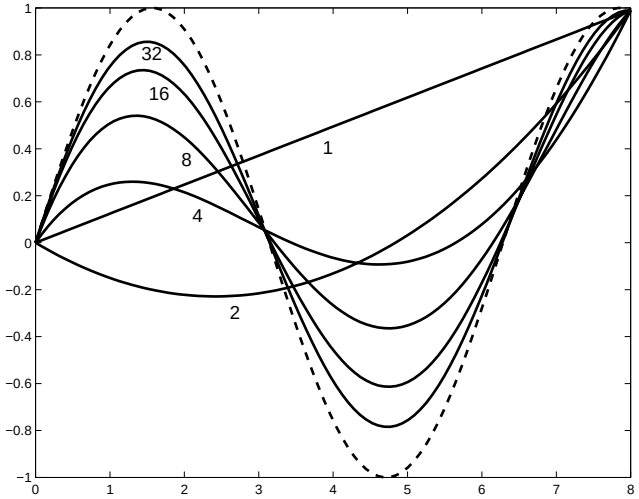
$$\|f - B_n f\| \leq \frac{1}{8n}\|f''\|, \quad f \in C^2(I).$$

this bound holds with equality for $f(x) = x^2$, and so cannot be improved. This slow rate of convergence makes the Bernstein polynomials impractical for most applications. See Figure 1.3 where the linear rate of convergence is quite evident.

2.2. Jackson's theorems for trigonometric polynomials. In the next sections we address the question of how quickly a given continuous function can be approximated by a sequence of polynomials of increasing degree. The results were mostly obtained by Dunham Jackson in the first third of the twentieth century and are known collectively as Jackson's theorems. Essentially they say that if a function is in C^k then it can be approximated by a sequence of polynomials of degree n in such a way that the error is at most C/n^k as $n \to \infty$. Thus the smoother a function is, the better the rate of convergence.

Jackson proved this sort of result both for approximation by polynomials and for approximation by trigonometric polynomials (finite Fourier series). The two sets of results are intimately related, as we shall see, but it is easier to get started with the results for trigonometric polynomials, as we do now.

FIGURE 1.3. Approximation to the function $\sin x$ on the interval $[0, 8]$ by Bernstein polynomials of degrees $1, 2, 4, \ldots, 32$.



Let $C_{2\pi}$ be the set of 2π-periodic continuous functions on the real line, and $C^k_{2\pi}$ the set of 2π-periodic functions which belong to $C^k(\mathbb{R})$. We shall investigate the rate of approximation of such functions by trigonometric polynomials of degree n. By these we mean linear combinations of the $n+1$ functions 1, $\cos kx$, $\sin kx$, $k = 1, \ldots, n$, and we denote by $\mathcal{T}_n$ the space of all trigonometric polynomials of degree n, i.e., the span of these of the $2n+1$ functions. Using the relations $\sin x = (e^{ix} - e^{-ix})/(2i)$, $\cos x = (e^{ix} + e^{-ix})/2$, we can equivalently write

$$\mathcal{T}_n = \{ \sum_{k=-n}^{n} c_k e^{ikx} \,|\, c_k \in \mathbb{C}, c_{-k} = \bar{c}_k \}.$$

Our immediate goal is the Jackson Theorem for the approximation of functions in $C^1_{2\pi}$ by trigonometric polynomials.

THEOREM 1.6. *If $f \in C^1_{2\pi}$, then*

$$\inf_{p \in \mathcal{T}_n} \|f - p\| \leq \frac{\pi}{2(n+1)} \|f'\|.$$

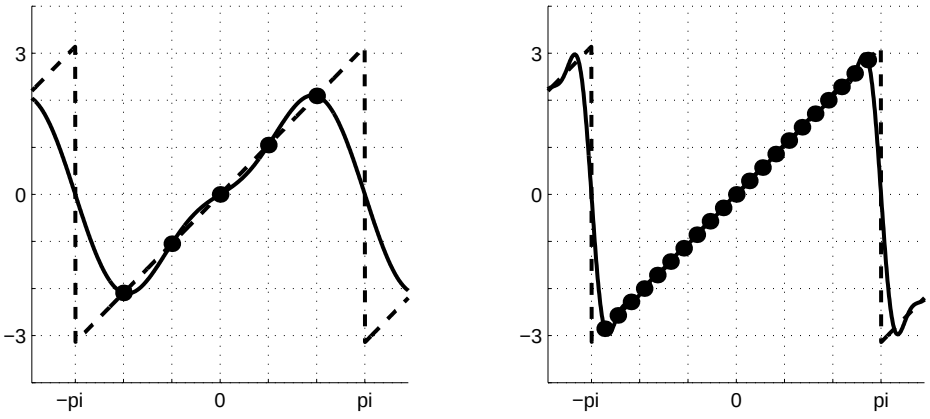
(We are suppressing the subscript on the L^∞ norm, since that is the only norm that we'll be using in this section.) The proof will be quite explicit. We start by writing $f(x)$ as an integral of f' times an appropriate kernel. Consider the integral $\int_{-\pi}^{\pi} y f'(x + \pi + y)\, dy$. Integrating by parts and using the fact that f is 2π-periodic we get

$$\int_{-\pi}^{\pi} y f'(x + \pi + y)\, dy = -\int_{-\pi}^{\pi} f(x + \pi + y)\, dx + 2\pi f(x).$$

The integral on the right-hand side is just the integral of f over one period (and so independent of x), and we can rearrange to get

$$f(x) = \bar{f} + \frac{1}{2\pi} \int_{-\pi}^{\pi} y f'(x + \pi + y)\, dy,$$

FIGURE 1.4. The interpolant $q_n \in \mathcal{T}_n$ of the sawtooth function for $n = 2$ and $n = 10$.



where $\bar{f}$ is the average value of f over any period. Now suppose we replace the function y in the last integral with a trigonometric polynomial $q_n(y) = \sum_{k=-n}^{n} c_k e^{iky}$. This gives

$$\int_{-\pi}^{\pi} q_n(y) f'(x+\pi+y)\,dy = \int_{-\pi}^{\pi} q_n(y-\pi-x) f'(y)\,dy = \sum_{k=-n}^{n} c_k \int_{-\pi}^{\pi} e^{ik(y-\pi)} f'(y)\,dy\, e^{-ikx},$$

which is a trigonometric polynomial of degree at most n in x. Thus

$$p_n(x) := \bar{f} + \frac{1}{2\pi}\int_{-\pi}^{\pi} q_n(y) f'(x+\pi+y)\,dy \in \mathcal{T}_n,$$

and $p_n(x)$ is close to $f(x)$ if $q(y)$ is close to y on $[-\pi,\pi]$. Specifically

$$|f(x) - p_n(x)| = \frac{1}{2\pi}\left|\int_{-\pi}^{\pi} [y - q_n(y)] f'(x+\pi+y)\,dy\right| \le \frac{1}{2\pi}\int_{-\pi}^{\pi} |y - q_n(y)|\,dy \|f'\|. \tag{1.1}$$

Thus to obtain a bound on the error, we need only give a bound on the L^1 error in trigonometric polynomial approximation to the function $g(y) = y$ on $[-\pi,\pi]$. (Note that, since we are working in the realm of 2π periodic functions, g is the sawtooth function.)

LEMMA 1.7. *There exists $q_n \in \mathcal{T}_n$ such that*

$$\int_{-\pi}^{\pi} |x - q_n(x)|\,dx \le \frac{\pi^2}{n+1}.$$

This we shall prove quite explicitly, by exhibiting q_n. Note that the Jackson theorem, Theorem 1.6 follows directly from (1.1) and the lemma.

PROOF. To prove the Lemma, we shall determine $q_n \in \mathcal{T}_n$ by the $2n+1$ equations

$$q_n\left(\frac{\pi k}{n+1}\right) = \frac{\pi k}{n+1}, \quad k = -n, \ldots, n.$$

That, is, q_n interpolates the saw tooth function at the $n+1$ points with abscissas equal to $\pi k/(n+1)$. See Figure 1.4.

This defines q_n uniquely. To see this it is enough to note that if a trigonometric polynomial of degree n vanishes at $2n+1$ distinct points in $[-\pi,\pi)$ it vanishes identically. This is

so because if $\sum_{k=-n}^{n} c_k e^{ikx}$ vanishes at x_j, $j = -n, \dots, n$, then $z^n \sum_{k=-n}^{n} c_k z^k$, which is a polynomial of degree $2n$, vanishes at the $2n+1$ distinct complex numbers e^{ix_j}, so is identically zero, which implies that all the c_k vanish.

Now q_n is odd, since replacing $q_n(x)$ with $-q_n(-x)$ would give another solution, which then must coincide with q_n. Thus $q_n(x) = \sum_{k=1}^{n} b_k \sin kx$.

To get a handle on the error $x - q_n(x)$ we first note that by construction this function has $2n+1$ zeros in $(-\pi, \pi)$, namely at the points $\pi k/(n+1)$. It can't have any other zeros or any double zeros in this interval, for if it did Rolle's Theorem would imply that its derivative $1 - q_n' \in \mathcal{T}_n$, would have $2n+1$ zeros in the interval, and, by the argument above, would vanish identically, which is not possible (it has mean value 1). Thus q_n changes sign exactly at the points $\pi k/(n+1)$.

Define the piecewise constant function

$$s(x) = (-1)^k, \quad \frac{k\pi}{n+1} \le x < \frac{(k+1)\pi}{n+1}, \quad k \in \mathbb{Z}.$$

Then

$$\int_{-\pi}^{\pi} |x - q_n(x)|\, dx = \int [x - q_n(x)] s(x)\, dx.$$

But, as we shall show in a moment,

$$\int \sin kx\, s(x)\, dx = 0, \quad k = 1, \dots, n, \tag{1.2}$$

and it is easy to calculate $\int x s(x)\, dx = \pi^2/(n+1)$. Thus

$$\int_{-\pi}^{\pi} |x - q_n(x)|\, dx = \frac{\pi^2}{n+1},$$

as claimed.

We complete the proof of the lemma by verifying (1.2). Let I denote the integral in question. Then

$$\begin{aligned} I = -\int s(x + \frac{\pi}{n+1}) \sin kx\, dx &= -\int s(x) \sin k(x - \frac{\pi}{n+1})\, dx \\ &= -\cos(\frac{-k\pi}{n+1}) \int s(x) \sin kx\, dx = -\cos(\frac{-k\pi}{n+1}) I. \end{aligned}$$

Since $|\cos(\frac{-k\pi}{n+1})| < 1$, this implies that $I = 0$. □

Having proved a Jackson Theorem in $C^1_{2\pi}$, we can use a bootstrap argument to show that if f is smoother, then the rate of convergence of the best approximation is better. This is the Jackson Theorem in $C^k_{2\pi}$.

THEOREM 1.8. *If $f \in C^k_{2\pi}$, some $k > 0$, then*

$$\inf_{p \in \mathcal{T}_n} \|f - p\| \le \left[\frac{\pi}{2(n+1)}\right]^k \|f^{(k)}\|.$$

PROOF. We shall use induction on k, the case $k = 1$ having been established. Assuming the result, we must show it holds when k is replaced by $k+1$. Now let $q \in \mathcal{T}_n$ be arbitrary. Then

$$\inf_{p\in\mathcal{T}_n} \|f-p\| = \inf_{p\in\mathcal{T}_n} \|f-q-p\| \le \left[\frac{\pi}{2(n+1)}\right]^k \|(f-q)^{(k)}\|,$$

by the inductive hypothesis. Since q is arbitrary and $\{\, p^{(k)} \,|\, p \in \mathcal{T}_n \,\} = \hat{\mathcal{T}}_n$,

$$\inf_{p\in\mathcal{T}_n} \|f-p\| \le \left[\frac{\pi}{2(n+1)}\right]^k \inf_{r\in\hat{\mathcal{T}}_n} \|f^{(k)}-r\| \le \left[\frac{\pi}{2(n+1)}\right]^k \frac{\pi}{2(n+1)} \|f^{(k+1)}\|.$$

□

2.3. Jackson theorems for algebraic polynomials. To obtain the Jackson theorems for algebraic polynomials we use the following transformation. Given $f : [-1,1] \to \mathbb{R}$ define $g : \mathbb{R} \to \mathbb{R}$ by $g(\theta) = f(\cos\theta)$. Then g is 2π-periodic and even. This transformation is a linear isometry ($\|g\| = \|f\|$). Note that if $f \in C^1([-1,1])$ then $g \in C^1_{2\pi}$ and $g'(\theta) = -f'(\cos\theta)\sin\theta$, so $\|g'\| \le \|f'\|$. Also if $f(x) = x^n$, then $g(\theta) = [(e^{ix} + e^{-ix})/2]^n$ which is a trigonometric polynomial of degree at most n. Thus this transformation maps $\mathcal{P}_n([-1,1])$ to the $\mathcal{T}_n^{\text{even}}$, the subspace of even functions in $\mathcal{T}_n$, or, equivalently, the span of $\cos kx$, $k = 0, \dots, n$. Since $\dim \mathcal{T}_n^{\text{even}} = \dim \mathcal{P}_n([-1,1]) = n+1$, the transformation is in fact an isomorphism.

The Jackson theorem in $C^1([-1,1])$ follows immediately from that in $C^1_{2\pi}$:

THEOREM 1.9. *If $f \in C^1([-1,1])$, then*

$$\inf_{p\in\mathcal{P}_n} \|f-p\| \le \frac{\pi}{2(n+1)} \|f'\|.$$

PROOF. Define $g(\theta) = f(\cos\theta)$ so $g \in C^1_{2\pi}$. Since g is even, $\inf_{q\in\mathcal{T}_n^{\text{even}}} \|g-q\| = \inf_{q\in\mathcal{T}_n} \|g-q\|$. (If the second infimum is achieved by $q(\theta)$, then it is also achieved by $q(-\theta)$, then use the triangle inequality to show it is also achieved by the even function $[q(\theta)+q(-\theta)]/2$.) Thus

$$\inf_{p\in\mathcal{P}_n} \|f-p\| = \inf_{q\in\mathcal{T}_n} \|g-q\| \le \frac{\pi}{2(n+1)} \|g'\| \le \frac{\pi}{2(n+1)} \|f'\|.$$

□

You can't derive the Jackson theorem in $C^k([-1,1])$ from that in $C^k_{2\pi}$ (since we can't bound $\|g^{(k)}\|$ by $\|f^{(k)}\|$ for $k \ge 2$), but we can use a bootstrap argument directly. We know that

$$\inf_{p\in\mathcal{P}_n} \|f-p\| = \inf_{q\in\mathcal{P}_n} \inf_{p\in\mathcal{P}_n} \|f-q-p\| \le \inf_{q\in\mathcal{P}_n} \frac{\pi}{2(n+1)} \|f'-q'\|.$$

Assuming $n \ge 1$, q' is an arbitrary element of $\mathcal{P}_{n-1}$ and so we have

$$\inf_{p\in\mathcal{P}_n} \|f-p\| \le \inf_{p\in\mathcal{P}_{n-1}} \frac{\pi}{2(n+1)} \|f'-p\| \le \frac{\pi}{2(n+1)} \frac{\pi}{2n} \|f''\|.$$

But now we can apply the same argument to get

$$\inf_{p\in\mathcal{P}_n} \|f-p\| \le \frac{\pi}{2(n+1)} \frac{\pi}{2n} \frac{\pi}{2(n-1)} \|f'''\|,$$

as long as $n \ge 2$. Continuing in this way we get

$$\inf_{p\in\mathcal{P}_n} \|f-p\| \le \frac{c_k}{(n+1)n(n-1)\dots(n-k+2)} \|f^{(k)}\|$$

if $f \in C^k$ and $n \geq k-1$, with $c_k = (\pi/2)^k$. To state the result a little more compactly we analyze the product $M = (n+1)n(n-1)\dots(n-k+2)$. Now if $n \geq 2(k-2)$ then each factor is at least $n/2$, so $M \geq n^k/2^k$. Also

$$d_k := \max_{k-1 \leq n \leq 2(k-2)} \frac{n^k}{(n+1)n(n-1)\dots(n-k+2)} < \infty,$$

so, in all, $n^k \leq e_k M$ where $e_k = \max(2^k, d_k)$. Thus we have arrived at the Jackson theorem in C^k:

THEOREM 1.10. *Let k be a positive integer, $n \geq k-1$ an integer. Then there exists a constant c depending only on k such that*

$$\inf_{p \in \mathcal{P}_n} \|f - p\| \leq \frac{c}{n^k} \|f^{(k)}\|.$$

for all $f \in C^k([-1,1])$.

2.4. Polynomial approximation of analytic functions. If a function is C^∞ the Jackson theorems show that the best polynomial approximation converges faster than any power of n. If we go one step farther and assume that the function is analytic (i.e., its power series converges at every point of the interval including the end points), we can prove exponential convergence.

We will first do the periodic case and show that the Fourier series for an analytic periodic function converges exponentially, and then use the Chebyshev transform to carry the result over to the algebraic case.

A real function on an open interval is called *real analytic* if the function is C^∞ and for every point in the interval the Taylor series for the function about that point converges to the function in some neighborhood of the point. A real function on a closed interval J is real analytic if it is real analytic on some open interval containing J.

It is easy to see that a real function is real analytic on an interval if and only if it extends to an analytic function on a neighborhood of the interval in the complex plane.

Suppose that $g(z)$ is real analytic and 2π-periodic on $\mathbb{R}$. Since g is smooth and periodic its Fourier series,

$$g(x) = \sum_{n=-\infty}^{\infty} a_n e^{inx}, \quad a_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} g(x) e^{-inx}\, dx,$$

converges absolutely and uniformly on $\mathbb{R}$. Since g is real analytic, it extends to an analytic function on the strip $S_\delta := \{\, x + iy \,|\, x, y \in \mathbb{R}, |y| \leq \delta \,\}$ for some $\delta > 0$. Using analyticity we see that we can shift the segment $[-\pi, \pi]$ on which we integrate to define the Fourier coefficients upward or downward a distance δ in the complex plane:

$$a_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} g(x \pm i\delta) e^{-in(x \pm i\delta)}\, dx = \frac{e^{\pm n\delta}}{2\pi} \int_{-\pi}^{\pi} g(x \pm i\delta) e^{-inx}\, dx.$$

Thus $|a_n| \leq \|g\|_{L^\infty(S_\delta)} e^{-\delta|n|}$ and we have shown that the Fourier coefficients of a real analytic periodic function decay exponentially.

Now consider the truncated Fourier series $q(z) = \sum_{k=-n}^{n} a_k e^{ikz} \in \mathcal{T}_n$. Then $g(z) - q(z) = \sum_{|k|>n} a_k e^{ikz}$, so for any real z

$$|g(z) - q(z)| \leq \sum_{|k|>n} |a_k| \leq 2\|g\|_{L^\infty(S_\delta)} \sum_{k=n+1}^{\infty} e^{-\delta k} = \frac{2e^{-\delta}}{1-e^{-\delta}} \|g\|_{L^\infty(S_\delta)} e^{-\delta n}.$$

Thus we have proven:

THEOREM 1.11. *Let g be 2π-periodic and real analytic. Then there exist positive constants C and δ so that*

$$\inf_{q \in \mathcal{T}_n} \|g - q\|_\infty \leq Ce^{-\delta n}.$$

The algebraic case follows immediately from the periodic one. If f is real analytic on $[-1, 1]$, then $g(\theta) = f(\cos\theta)$ is 2π-periodic and real analytic. Since $\inf_{q \in \mathcal{T}_n} \|g - q\|_\infty = \inf_{q \in \mathcal{T}_n^{\text{even}}} \|g - q\|_\infty = \inf_{p \in \mathcal{P}_n} \|f - p\|_\infty$, we can apply the previous result to bound the latter quantity by $Ce^{-\delta n}$.

THEOREM 1.12. *Let f be real analytic on a closed interval. Then there exist positive constants C and δ so that*

$$\inf_{p \in \mathcal{P}_n} \|f - p\|_\infty \leq Ce^{-\delta n}.$$

2.5. Characterization of the minimax approximant. Having established the rate of approximation afforded by the best polynomial approximation with respect to the L^∞ norm, in this section we derive two conditions that characterize the best approximation. We will use these results to show that the best approximation is unique (recall that our uniqueness theorem in the first section only applied to strictly convex norms, and so excluded the case of L^∞ approximation). The results of this section can also be used to design iterative algorithms which converge to the best approximation, but we shall not pursue that, because there are approximations which yield nearly as good approximation in practice as the best approximation but which are much easier to compute.

The first result applies very generally. Let J be an compact subset of $\mathbb{R}^n$ (or even of a general Hausdorff topological space), and let P be any finite dimensional subspace of $C(J)$. For definiteness you can think of J as a closed interval and P as the space $\mathcal{P}_n(J)$ of polynomials of degree at most n, but the result doesn't require this.

THEOREM 1.13 (Kolmogorov Characterization Theorem). *Let $f \in C(J)$, P a finite dimensional subspace of $C(J)$, $p \in P$. Then p is a best approximation to f in P if and only if no element of P has the same sign as $f - p$ on its extreme set.*

PROOF. First we note that p is a best approximation to f in P if and only if 0 is a best approximation to $g := f - p$ in P. So we need to show that 0 is a best approximation to g if and only if no element q of P has the same sign as g on its extreme set.

First suppose that 0 is not a best approximation. Then there exists $q \in P$ such that $\|g - q\| < \|g\|$. Now let x be a point in the extreme set of g. Unless $\operatorname{sign} q(x) = \operatorname{sign} g(x)$, $|g(x) - q(x)| \geq |g(x)| = \|g\|$, which is impossible. Thus q has the same sign of as g on its extreme set.

The converse direction is trickier, but the idea is simple. If q has the same sign of as g on its extreme set and we subtract a sufficiently small positive multiple of q from g, the

difference will be of strictly smaller magnitude than g near the extreme set. And if the multiple that we subtract is small enough the difference will also stay below the extreme value away from the extreme set. Thus $g - \epsilon q$ will be smaller than g for ϵ small enough, so that 0 is not the best approximation.

Since we may replace q by $q/\|q\|$ we can assume from the outset that $\|q\| = 1$. Let $\delta > 0$ be the minimum of qg on the extreme set, and set $S = \{ x \in J \mid q(x)g(x) > \delta/2\}$, so that S is an open set containing the extreme set. Let $M = \max_{J\setminus S}|g| < \|g\|$.

Now on S, $qg > \delta/2$, so

$$|g - \epsilon q|^2 = |g|^2 - 2\epsilon qg + \epsilon^2|q|^2 \leq \|g\|^2 - \epsilon\delta + \epsilon^2 = \|g\|^2 - \epsilon(\delta - \epsilon),$$

and so if $0 < \epsilon < \delta$ then $\|g - \epsilon q\|_{\infty,S} < \|g\|$.

On $J \setminus S$, $|g - \epsilon q| \leq M + |\epsilon|$, so if $0 < \epsilon < \|g\| - M$, $\|g - \epsilon q\|_{\infty,J\setminus S} < \|g\|$. Thus for any positive ϵ sufficiently small $\|g - \epsilon q\| < \|g\|$ on J. □

While the above result applies to approximation by any finite dimensional subspace P, we now add the extra ingredient that $P = \mathcal{P}_n([a, b])$ the space of polynomial functions of degree at most n on a compact interval $J = [a, b]$.

THEOREM 1.14 (Chebyshev Alternation Theorem). *Let $f \in C([a, b])$, $p \in \mathcal{P}_n$. Then p is a best approximation to f in $\mathcal{P}_n$ if and only if $f - p$ achieves its maximum magnitude at $n + 2$ distinct points with alternating sign.*

PROOF. If $f-p$ achieves its maximum magnitude at $n+2$ distinct points with alternating sign, then certainly no function $q \in \mathcal{P}_n$ has the same sign as $f - p$ on its extreme set (since a nonzero element of $\mathcal{P}_n$ cannot have $n + 1$ zeros). So p is a best approximation to f in $\mathcal{P}_n$.

Conversely, suppose that $f - p$ changes sign at most $n + 1$ times on its extreme set. For definiteness suppose that $f - p$ is positive at its first extreme point. Then we can choose n points $x_1 < x_2 < \cdots x_n$ in $[a, b]$ such that $f - p$ is positive on extreme points less than x_1, negative on extreme points in on $[x_1, x_2]$, positive on extreme points in $[x_2, x_3]$, etc. The function $q(x) = (x_1 - x) \cdots (x_n - x) \in \mathcal{P}_n$ then has the same sign as $f - p$ on its extreme set, and so p is not a best approximation to f in $\mathcal{P}_n$. □

The Chebyshev Alternation Theorem is illustrated in Figure 1.5.

We can now prove uniqueness of the best approximation.

COROLLARY 1.15. *The best L^∞ approximation to a continuous function by a function in $\mathcal{P}_n$ is unique.*

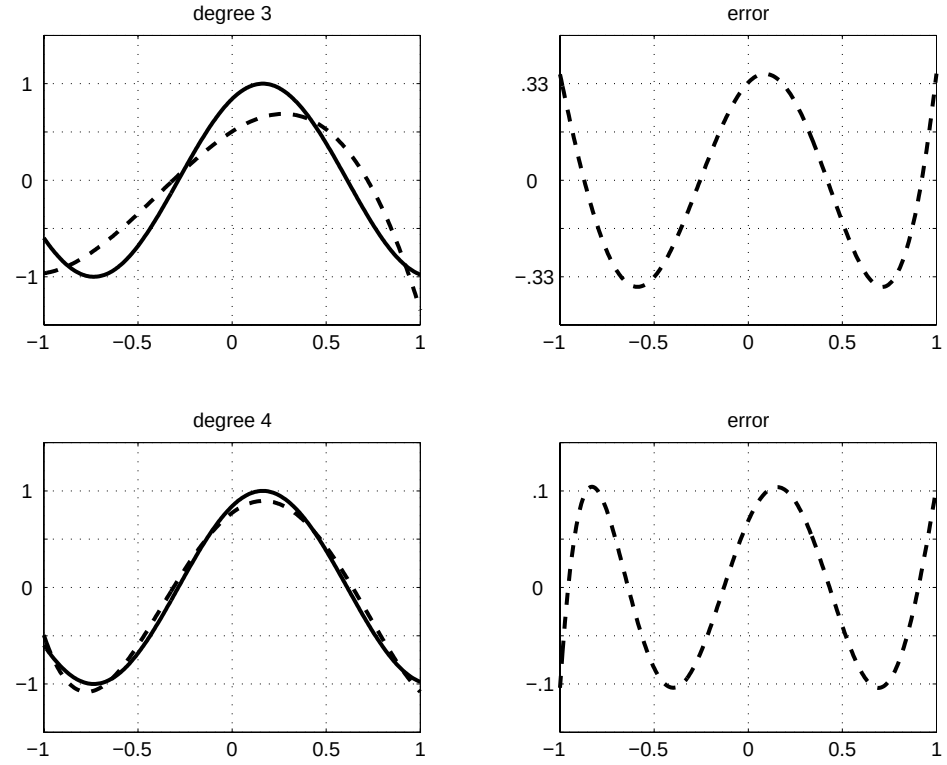
PROOF. Suppose $p, p^* \in \mathcal{P}_n$ are both best approximations to f. Let $e = f-p$, $e^* = f-p^*$, and say $M = \|e\| = \|e^*\|$. Since $(p + p^*)/2$ is also a best approximation, $|f - (p + p^*)/2|$ achieves the value M at $n + 2$ points, $x_0, \ldots, x_{n+1}$. Thus

$$M = |[e(x_i) + e^*(x_i)]/2| \leq |e(x_i)|/2 + |e^*(x_i)|/2 \leq M.$$

Thus equality holds throughout, and $e(x_i) = e^*(x_i)$, so $p(x_i) = p^*(x_i)$ at all $n + 2$ points, which implies that $p = p^*$. □

REMARKS. 1. The only properties we used of the space $\mathcal{P}_n$ are (1) that no non-zero element has more than n zeros, and (2) given any n points there is an element with exactly

FIGURE 1.5. A sinusoidal function and its minimax approximation of degrees 3 and 4. The error curves, on the right, achieve their maximal magnitudes at the requisite 5 and 6 points, respectively.



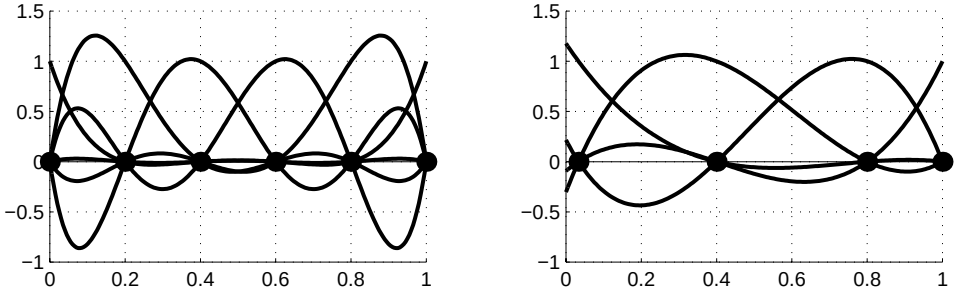
those points as the zeros. These two conditions together (which are equivalent to the existence and uniqueness of an interpolant at $n+1$ points, as discussed in the next section) are referred to as the *Haar property*. Many other subspaces satisfy the Haar property, and so we can obtain a Chebyshev Alternation Theorem for them as well.

2. The Chebyshev alternation characterization of the best approximation can be used as the basis for a computational algorithm to approximate the best approximation, known as the exchange algorithm or Remes algorithm. However in practice something like the interpolant at the Chebyshev points, which, as we shall see, is easy to compute and usually gives something quite near best approximation, is much more used.

3. Lagrange Interpolation

3.1. General results. Now we consider the problem of not just approximating, but *interpolating* a function at given points by a polynomial. That is, we suppose given $n+1$ distinct points $x_0 < x_1 < \cdots < x_n$ and $n+1$ values $y_0, y_1, \ldots, y_n$. We usually think of the y_i as the value $f(x_i)$ of some underlying function f, but this is not necessary. In any case, there exists a unique polynomial $p \in \mathcal{P}_n$ such that $p(x_i) = y_i$. To prove this, we notice that if we write $p = \sum_{i=0}^{n} c_i x^i$, then the interpolation problem is a system of $n+1$ linear equations in the $n+1$ unknowns c_i. We wish to show that this system is non-singular. Were it not, there would exists a non-zero polynomial in $\mathcal{P}_n$ vanishing at the x_i, which is impossible, and

FIGURE 1.6. The Lagrange basis functions for 6 equally space points and 4 unequally spaced points.



the uniqueness and existence is established. (A more algebraic proof consists of writing out the matrix of the linear system explicitly. It is a Vandermonde system, whose determinant can be computed explicitly as $\prod_{i<j}(x_i - x_j)$.)

The polynomial p is called the Lagrange interpolant of the values y_i at the points x_i. If the $y_i = f(x_i)$ for some function f, we call p the Lagrange interpolant of f at the x_i.

While the proof of existence of a Lagrange interpolating polynomial just given was indirect, it is also straightforward to derive a formula for the solution. *Lagrange's formula* states that

$$p(x) = \sum_{k=0}^{n} y_k \prod_{\substack{0 \le m \le n \\ m \ne k}} \frac{x - x_m}{x_k - x_m},$$

and is easily verified. Note that we have expressed the solution not as a linear combination of the monomials x^i, but rather as a linear combination of the *Lagrange basis functions*

$$l_k^n(x) = \prod_{\substack{0 \le m \le n \\ m \ne k}} \frac{x - x_m}{x_k - x_m},$$

plotted in Figure 1.6. We don't have to solve a linear system to find the coefficients in this basis: they are simply the y_i. It is instructive to compare the Lagrange basis functions with the Bernstein weighting functions plotted in Figure 1.2. In each case the sum of all the basis functions is identically 1 (prove this for the Lagrange basis functions).

Our first result is a straightforward application of calculus to obtain an error formula for Lagrange interpolation to a smooth function.

THEOREM 1.16 (Error formula for Lagrange interpolation). *Let x_i, $i = 0, \ldots, n$ be distinct points and let $p \in \mathcal{P}_n$ be the Lagrange interpolant of some function f at the points x_i. Let $x \in \mathbb{R}$ and suppose that $f \in C^{n+1}(J)$ for some interval J containing the x_i and x. Then there exists a point ξ in the interior of J such that*

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x - x_0) \cdots (x - x_n).$$

PROOF. We may assume that x differs from all the x_i, since the theorem is obvious otherwise. Let $\omega(x) = (x - x_0) \cdots (x - x_n)$ and set $G(t) = [f(x) - p(x)]\omega(t) - [f(t) - p(t)]\omega(x)$. Then G has $n + 2$ distinct zeros: the x_i and x. By repeated application of

Rolle's theorem there exists a point ξ strictly between the largest and the smallest of the zeros such that $d^{(n+1)}G/dt^{(n+1)}(\xi) = 0$. Since $p^{(n+1)} \equiv 0$ and $\omega^{(n+1)} \equiv (n+1)!$, this gives $[f(x) - p(x)](n+1)! - f^{(n+1)}(\xi)\omega(x)$ which is the desired result. □

REMARK. If all the x_i tend to the same point a, then p tends to the Taylor polynomial of degree n at a, and the estimate tends to the standard remainder formula for the Taylor polynomial: $f(x) - p(x) = [1/(n+1)!]f^{(n+1)}(\xi)(x-a)^{n+1}$ for some ξ between x and a.

An obvious corollary of the error formula is the estimate:

$$|f(x) - p(x)| \leq \frac{1}{(n+1)!}\|f^{(n+1)}\|_{\infty,J}|\omega(x)|.$$

In particular

$$|f(x) - p(x)| \leq \frac{1}{(n+1)!}\|f^{(n+1)}\|_{\infty,J}k^{n+1},$$

where $k = \max(|x - x_i|, |x_i - x_j|)$. In particular if $a \leq \min x_i$, $b \geq \max x_i$, then

$$\|f - p\|_{\infty,[a,b]} \leq \frac{1}{(n+1)!}\|f^{(n+1)}\|_{\infty,[a,b]}|b-a|^{n+1}, \tag{1.3}$$

no matter what the configuration of the points $x_i \in [a,b]$. This gives a useful estimate if we hold n fixed and let the points x_i tend to the evaluation point x. It establishes a rate of convergence of order $n+1$ with respect to the interval length for Lagrange interpolation at $n+1$ points in the interval.

Another interesting question, closer to the approximation problem we have considered heretofore, asks about the error on a fixed interval as we increase the number of interpolation points and so the degree of the interpolating polynomial. At this point we won't restrict the configuration of the points, so we consider an arbitrary tableau of points

$$\begin{gathered} x_0^0 \\ x_0^1 < x_1^1 \\ x_0^2 < x_1^2 < x_2^2 \\ \vdots \end{gathered}$$

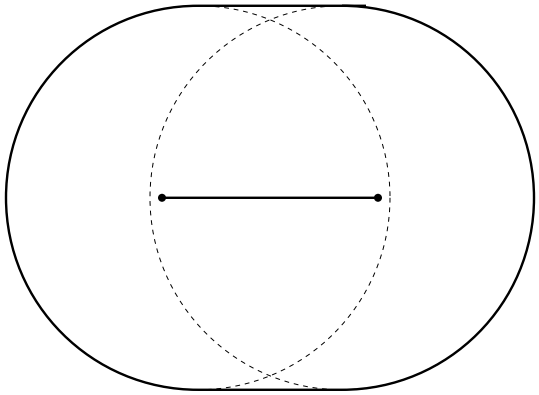
all belonging to $[a,b]$. Then we let $p_n \in \mathcal{P}_n$ interpolate f at x_i^n, $i = 0, \ldots, n$ and inquire about the convergence of p_n to f_n as $n \to \infty$. Whether such convergence occurs, and how fast, depends on the properties of the function f and on the particular arrangement of points.

One possibility is to make a very strong assumption on f, namely that f is real analytic on $[a,b]$. Now if f is analytic on the closed disk $\bar{B}(\xi, R)$ of radius R about ξ, then, by Cauchy's estimate, $|f^{(n+1)}(\xi)| \leq (n+1)!\|f\|_{\infty,\bar{B}(\xi,R)}/R^{n+1}$. Let $O(a,b,R)$ denote the oval $\bigcup_{\xi\in[a,b]} \bar{B}(\xi, R)$. We then have

THEOREM 1.17. *Let $a < b$ and suppose that f extends analytically to the oval $O(a,b,R)$ for some $R > 0$. Let $x_0 < \cdots < x_n$ be any set of $n+1$ distinct points in $[a,b]$ and let p be the Lagrange interpolating polynomial to f at the x_i. Then*

$$\|f - p\|_{\infty,[a,b]} \leq \|f\|_{\infty,O(a,b,R)} \left(\frac{|b-a|}{R}\right)^{n+1}.$$

FIGURE 1.7. If a function on the interval extends analytically to the oval in the complex plane depicted here, then for any choice of interpolation points in the interval, the interpolating polynomials will converge exponentially fast on the interval.



This shows that if the domain of analyticity of f contains $O(a, b, R)$ for some $R > |b-a|$, then for any choice of interpolating tableau, the p_n converge to f exponentially fast in $C([a, b])$. See Figure 1.7 In particular if the function f is entire, this will occur.

However, even if f is real analytic on $[a, b]$, the p_n need not converge to f if a pole lies nearby in the complex plane. A famous example using equally spaced interpolation points was given by Runge: $a = -5$, $b = 5$, $x_i^n = -5 + 10i/n$, $f(x) = 1/(1 + x^2)$. In this case he proved the existence of a number $\kappa \approx 3.63338$ such that $\lim_{n\to\infty} p_n(x) = f(x)$ if and only if $|x| < \kappa$. Figure 1.8 contrasts the striking non-convergence of Lagrange interpolation using equally spaced points in this case, with the convergence that partakes for the entire Gaussian function $f(x) = \exp(2x^2/5)$.

If the function is not smooth, the results may be even worse: in 1918 S. Bernstein proved that equidistant interpolation to $f(x) = |x|$ on $[-1, 1]$ does not converge at any point except $x = -1$, 0, and 1.

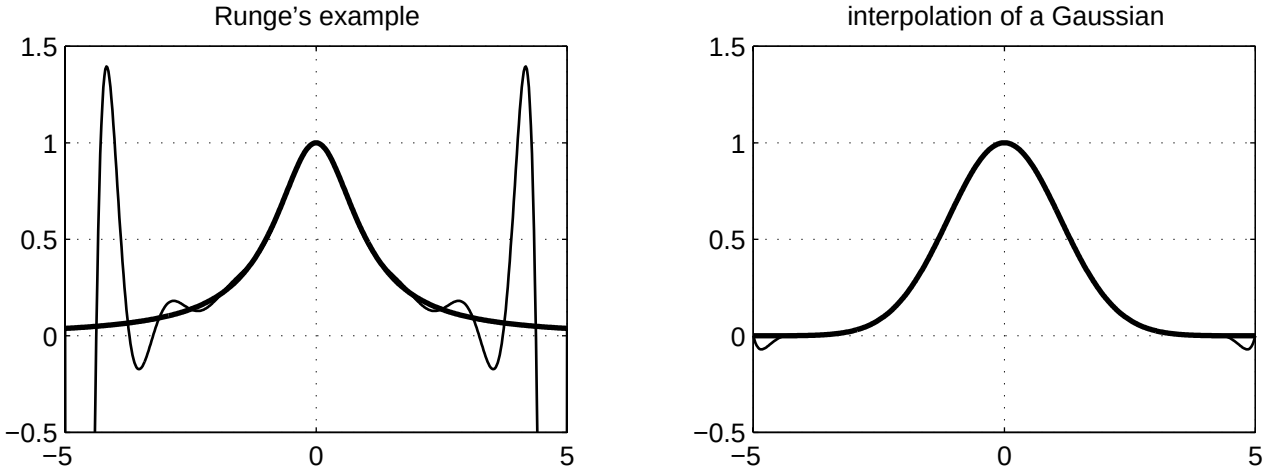
Fortunately, as we shall see in the next subsection, there exist *much* better choices of interpolation points than equally spaced ones. But in 1914 Faber showed that no choice of points works for all continuous functions.

THEOREM 1.18 (Faber's Theorem). *Given a triangular array of points*

$$\begin{array}{ccccc} & & x_0^0 & & \\ & x_0^1 & & x_1^1 & \\ x_0^2 & & x_1^2 & & x_2^2 \\ & & \vdots & & \end{array}$$

in $[a, b]$ and a continuous function $f(x)$ on $[a, b]$, let $p_n(x)$ be the polynomial of degree $\le n$ which interpolates f at the $n + 1$ points x_0^n, x_1^n, ..., x_n^n. Then no matter how the points are chosen, there exists a continuous function f for which the p_n do not *converge uniformly to f.*

FIGURE 1.8. Interpolation by polynomials of degree 16 using equally spaced interpolation points. The first graph shows Runge's example. In the second graph, the function being interpolated is entire, and the graph of the interpolating polynomial nearly coincides with that of the function.



In 1931 Bernstein strengthened this negative theorem to show that there exists a continuous function f and a point c in $[a,b]$ for which $p_n(c)$ does not converge to $f(c)$. In 1980 Erdös and Vértesi showed that in fact there exists a continuous function f such that $p_n(c)$ does not converge to $f(c)$ for almost all c in $[a,b]$.

However, as we shall now show, if the function f is required to have a little smoothness, and if the interpolation points are chosen well, then convergence will be obtained.

3.2. The Lebesgue constant. Given $n+1$ distinct interpolation points in $[a,b]$ and $f \in C([a,b])$, let $P_n f \in \mathcal{P}_n$ be the Lagrange interpolating polynomial. Then P_n is an operator from $C([a,b])$ to itself, and we may consider its norm:

$$\|P_n\| = \sup_{\substack{f \in C([a,b]) \\ \|f\|_\infty \le 1}} \|P_n f\|_\infty.$$

Using this norm, it is easy to relate the error in interpolation to the error in best approximation:

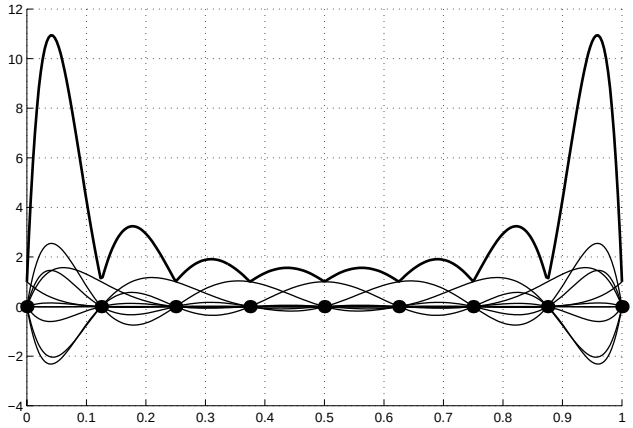
$$\|f - P_n f\| = \inf_{q \in \mathcal{P}_n} \|f - q - P_n(f-q)\| \le (1 + \|P_n\|) \inf_{q \in \mathcal{P}_n} \|f - q\|.$$

Note that the only properties of P_n we have used to get this estimate are linearity and the fact that it preserves $\mathcal{P}_n$.

Thus, if we can bound $\|P_n\|$ we can obtain error estimates for interpolation from those for best approximation (i.e., the Jackson theorems). Now let

$$l_k^n(x) = \prod_{\substack{0 \le m \le n \\ m \ne k}} \frac{x - x_m}{x_k - x_m}$$

FIGURE 1.9. Lebesgue function for degree 8 interpolation at equally spaced points.



denote the Lagrange basis functions. Recall that $\sum_{k=0}^{n} l_k^n(x) = 1$. Set

$$L_n(x) = \sum_{k=0}^{n} |l_k^n(x)|,$$

the *Lebesgue function* for this choice of interpolation points. Then

$$\|P_n\| = \sup_{0\le x\le 1} \sup_{|f|\le 1} \left| \sum_{k=0}^{n} f(x_k) l_k^n(x) \right| = \sup_{0\le x\le 1} \sum_{k=0}^{n} |l_k^n(x)| = \|L_n\|_\infty.$$

Figure 1.9 shows the Lebesgue function for interpolation by a polynomial of degree 8 using equally spaced interpolation points plotted together with the Lagrange basis functions entering into its definition.

The constant $\|P_n\| = \|L_n\|_\infty$ is called the Lebesgue constant of the interpolation operator. Of course it depends on the point placement. However it only depends on the relative configuration: if we linearly scale the points from the interval $[a, b]$ to another interval, then the constant doesn't change. Table 1.1 shows the Lebesgue constants for equally spaced interpolation points ranging from a to b. Note that the constant grows quickly with n reflecting the fact that the approximation afforded by the interpolant may be much worse than the best approximation. The column labelled "Chebyshev" shows the Lebesgue constant if a better choice of points, the Chebyshev points, is used. We study this in the next subsection.

It is not clear from the table whether the Lebesgue constant remains bounded for interpolation at the Chebyshev points, but we know it does not: otherwise the Chebyshev points would give a counterexample to Faber's theorem. In fact Erdös proved a rather precise lower bound on growth rate.

THEOREM 1.19. *[Erdös 1961] For any triangular array of points, there is a constant c such that corresponding Lebesgue constant satisfies*

$$\|P_n\| \ge \frac{2}{\pi} \log n - c.$$

This result was known well earlier, but with a less precise constant. See, e.g., Rivlin's *Introduction to the Approximation of Functions* [**5**] for an elementary argument.

TABLE 1.1. Lebesgue constants for interpolation into $\mathcal{P}_n$ at equally spaced points and Chebyshev points (to three significant digits).

n	Equal	Chebyshev
2	1.25	1.67
4	2.21	1.99
6	4.55	2.20
8	11.0	2.36
10	29.9	2.49
12	89.3	2.60
14	283	2.69
16	935	2.77
18	3,170	2.84
20	11,000	2.90

3.3. The Chebyshev points. Returning to the error formula for Lagrange interpolation we see that a way to reduce the error is to choose the interpolation points x_i so as to decrease $\omega(x) = (x - x_0)\dots(x - x_n)$. Assuming that we are interested in reducing the error on all of $[a, b]$, we are led to the problem of finding $x_0 < \cdots < x_n$ which minimize

$$\sup_{a\le x\le b} |(x - x_0)\dots(x - x_n)|. \tag{1.4}$$

In fact, we can solve this problem in closed form. First consider the case $[a, b] = [-1, 1]$. Define $T_n(x) = \cos(n \arccos x) \in \mathcal{P}_n([-1, 1])$, the polynomial which corresponds to $\cos n\theta$ under the Chebyshev transform. T_n is called the nth Chebyshev polynomial:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1, \quad T_3(x) = 4x^3 - 3x, \quad T_4(x) = 8x^4 - 8x^2 + 1, \dots$$

Using trigonometric identities for $\cos(n \pm 1)x$, we get the *recursion relation*

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),$$

from which it easily follows that $T_n(x)$ is a polynomial of exact degree n with leading coefficient 2^{n-1}.

Now let $x_i^n = \cos[(2i + 1)\pi/(2n + 2)]$. Then it is easy to see that $1 > x_0^n > x_1^n > \cdots > x_n^n > -1$ and that these are precisely the $n + 1$ zeros of T_{n+1}. These are called the $n + 1$ *Chebyshev points* on $[-1, 1]$. The definition is illustrated for $n = 8$ in Figure 1.10. The next theorem shows that the Chebyshev points minimize (1.4).

THEOREM 1.20. *For $n \ge 0$, let $x_0, x_1, \dots x_n \in \mathbb{R}$ and set $\omega(x) = (x - x_0)\dots(x - x_n)$. Then*

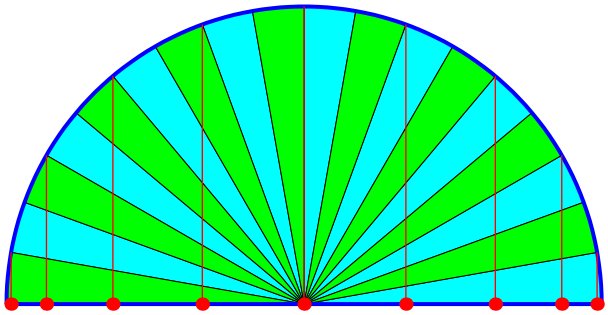
$$\sup_{a\le x\le b} |\omega(x)| \ge 2^{-n},$$

and if $x_i = \cos[(2i + 1)\pi/(2n + 2)]$, then

$$\sup_{a\le x\le b} |\omega(x)| = 2^{-n}.$$

PROOF. First assume that the x_i are the $n+1$ Chebyshev points. Then ω and T_{n+1} are two polynomials of degree $n+1$ with the same roots. Comparing their leading coefficients we see that $\omega(x) = 2^{-n}T_{n+1}(x) = 2^{-n}\cos(n \arccos x)$. The second statement follows immediately.

FIGURE 1.10. The Chebyshev points $x_i^8 = \cos[(2i+1)\pi/18]$.



Note also that for this choice of points, $|\omega(x)|$ achieves its maximum value of 2^{-n} at $n+2$ distinct points in $[-1,1]$, namely at $\cos[j\pi/(n+1)]$, $j = 0,\dots,n+1$, and that the sign of $\omega(x)$ alternates at these points.

Now suppose that some other points $\tilde{x}_i$ are given and set $\tilde{\omega}(x) = (x-\tilde{x}_0)\cdots(x-\tilde{x}_n)$. If $|\tilde{\omega}(x)| < 2^{-n}$ on $[-1,1]$, then $\omega(x) - \tilde{\omega}(x)$ alternates sign at the $n+2$ points $\cos[j\pi/(n+1)]$ and so has at least $n+1$ zeros. But it is a polynomial of degree at most n (since the leading terms cancel), and so must vanish identically, a contradiction. □

Table 1.1 indicates the Lebesgue constant for Chebyshev interpolation grows rather slowly with the degree (although it does not remain bounded). In fact the rate of growth is only logarithmic and can be bounded very explicitly. See [**6**] for a proof.

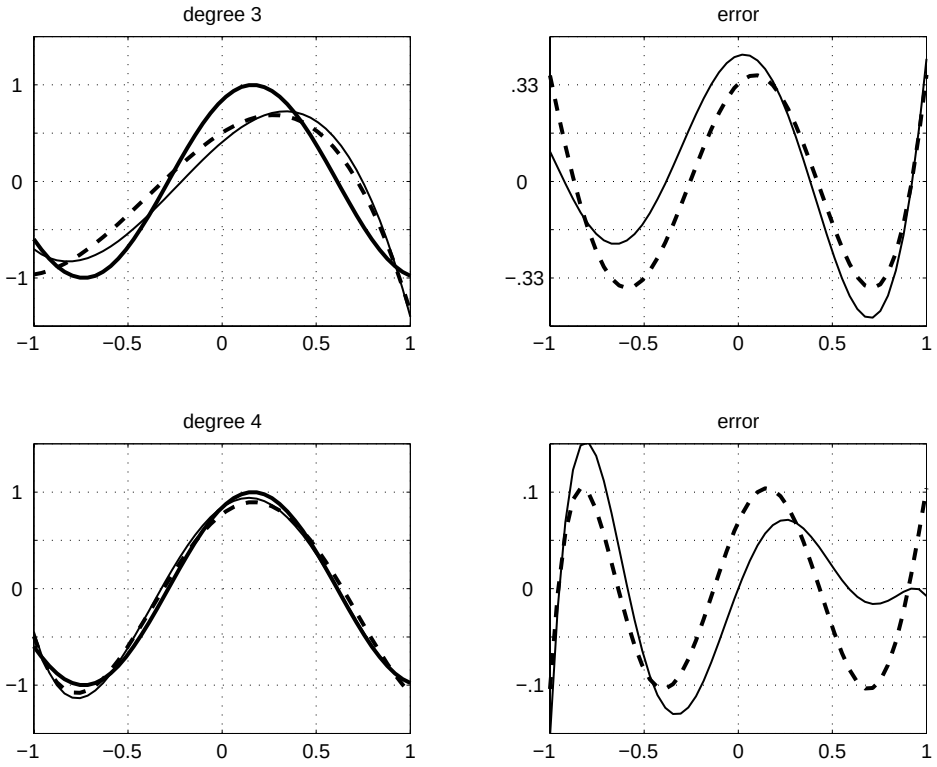
THEOREM 1.21. *If $P_n : C([a,b]) \to \mathcal{P}_n([a,b])$ denotes interpolation at the Chebyshev points, then*

$$\|P_n\| \le \frac{2}{\pi}\log(n+1) + 1, \quad n = 0,1,\dots.$$

Comparing with Theorem 1.19, we see that Chebyshev interpolation gives asymptotically the best results possible. Combining this result with the Jackson theorems we see that Chebyshev interpolation converges for any function C^1, and if $f \in C^k$, then $\|f - P_n\| \le Cn^{-k}\log n$, so the rate of convergence as $n \to \infty$ is barely worse than for the best approximation. Using the Jackson theorem for Hölder continuous functions (given in the exercises), we see that indeed Chebyshev interpolation converges for any Hölder continuous f. Of course, by Faber's theorem, there does exist a continuous function for which it doesn't converge, but that function must be quite unsmooth. We can summarize this by saying that Chebyshev interpolation is a robust approximation procedure (it converges for any "reasonable" continuous function) and an accurate one (it converges quickly if the function is reasonably smooth). Compare this with Bernstein polynomial approximation which is completely robust (it converges for *any* continuous function), but not very accurate.

Figure 1.11 repeats the example of best approximation from Figure 1.5, but adds the Chebyshev interpolant. We see that, indeed, on this example the Chebyshev interpolant is not far from the best approximation and the error not much larger than the error in best approximation.

FIGURE 1.11. The sinusoidal function and its minimax approximation (shown with a dotted line) of degrees 3 and 4 and the corresponding error curves, as in Figure 1.5. The thin lines show the interpolant at the Chebyshev points and the error curves for it.



4. Least Squares Polynomial Approximation

4.1. Introduction and preliminaries. Now we consider best approximation in the space $L^2([a,b])$. The key property of the L^2 norm which distinguishes it from the other L^p norms, is that it is determined by an inner product, $\langle u, v\rangle = \int_a^b u(x)v(x)\,dx$.

DEFINITION. A normed vector space X is an inner product space if there exists a symmetric bilinear form $\langle\,\cdot\,,\,\cdot\,\rangle$ on X such that $\|x\|^2 = \langle x, x\rangle$.

THEOREM 1.22. *Let X be an inner product space, P be a finite dimensional subspace, and $f \in X$. Then there exists a unique $p \in P$ minimizing $\|f-p\|$ over P. It is characterized by the* normal equations

$$\langle p, q\rangle = \langle f, q\rangle, \quad q \in P.$$

PROOF. We know that there exists a best approximation p. To obtain the characterization note that $\|f - p + \epsilon q\|^2 = \|f-p\|^2 + 2\epsilon\langle f-p, q\rangle + \epsilon^2\|q\|^2$ achieves its minimum (as a quadratic polynomial in ϵ) at 0. If p and p^* are both best approximation the normal equations give $\langle p - p^*, q\rangle = 0$ for $q \in P$. Taking $q = p - p^*$ shows $p = p^*$. (Alternative proof of uniqueness: show that an inner product space is always strictly convex.) □

In the course of the proof we showed that the normal equations admit a unique solution. To obtain the solution, we select a basis $\phi_1, \ldots, \phi_n$ of P, write $p = \sum_j a_j \phi_j$, and solve the equations

$$\sum_{j=1}^{n} \langle \phi_j, \phi_i \rangle a_j = \langle f, \phi_i \rangle, \quad i = 1, \ldots, n.$$

This is a nonsingular matrix equation.

Consider as an example the case where $X = L^2([0,1])$ and $P = \mathcal{P}_n$. If we use the monomials as a basis for $\mathcal{P}_n$, then the matrix elements are $\int_0^1 x^j x^i \, dx = 1/(i+j+1)$. In other words the matrix is the Hilbert matrix, famous as an example of a badly conditioned matrix. Thus this is *not* a good way to solve the normal equations.

Suppose we can find another basis for P which is orthogonal: $\langle P_j, P_i \rangle = 0$ if $i \neq j$. In that case the matrix system is diagonal and trivial to solve. But we may construct an orthogonal basis $\{P_i\}$ starting from any basis $\{\phi_i\}$ using the Gram-Schmidt process: $P_1 = \phi_1$,

$$P_j = \phi_j - \sum_{k=1}^{j-1} \frac{\langle \phi_j, P_k \rangle}{\|P_k\|^2} P_k, \quad j = 2, \ldots, n.$$

Note that for an orthogonal basis we have the simple formula

$$p = \sum_{j=1}^{n} c_j P_j, \qquad c_j = \frac{\langle f, P_j \rangle}{\|P_j\|^2}.$$

If we normalize the P_j so that $\|P_j\| = 1$ the formula for c_j simplifies to $c_j = \langle f, P_j \rangle$

4.2. The Legendre polynomials. Consider now the case of $X = L^2([a,b])$, $P = \mathcal{P}_n([a,b])$. Applying the Gram-Schmidt process to the monomial basis we get a sequence of polynomials $p_n = x^n + \text{lower}$ with $p_n \perp \mathcal{P}_{n-1}$. This is easily seen to characterize the p_n uniquely.

The Gram-Schmidt process simplifies for polynomials. For the interval $[-1,1]$ define

$$\begin{aligned} p_0(x) &= 1, \\ p_1(x) &= x, \\ p_n(x) &= x p_{n-1}(x) - \frac{\langle x p_{n-1}, p_{n-2} \rangle}{\|p_{n-2}\|^2} p_{n-2}(x), \quad n \geq 2. \end{aligned}$$

It is easy to check that these polynomials are monic and orthogonal. The numerator in the last equation is equal to $\|p_{n-1}\|^2$. These polynomials, or rather constant multiples of them, are called the Legendre polynomials.

$$\begin{aligned} p_0(x) &= 1, \\ p_1(x) &= x, \\ p_2(x) &= x^2 - 1/3, \\ p_3(x) &= x^3 - 3x/5. \end{aligned}$$

We can obtain the Legendre polynomials on an arbitrary interval by linear scaling.

We normalized the Legendre polynomials by taking their leading coefficient as 1. More commonly the Legendre polynomials are normalized to have value 1 at 1. Then it turns out that the recursion can be written

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x),$$

starting with $P_0 = 1$, $P_1(x) = x$ (see [**5**], Ch. 2.2 for details). Henceforth we shall use P_n to denote the Legendre polynomials so normalized. Now we shall gather some properties of them (see [**5**] Ch. 2, including the exercises, for details).

0) $P_n(1) = 1$, $P_n(-1) = (-1)^n$, P_n is even or odd according to whether n is even or odd.

1) $\|P_n\|^2 = 2/(2n+1)$.

2) The leading coefficient of $P_n = (2n-1)(2n-3)\cdots 1/n! = (2n)!/[2^n(n!)^2]$.

3) It is easy to check, by integration by parts, that the functions $\frac{d^n}{dx^n}[(x^2-1)^n]$ are polynomials of degree n and are mutually orthogonal. Comparing leading coefficients we get Rodrigues's formula for P_n:

$$P_n = \frac{1}{2^n n!}\frac{d^n}{dx^n}[(x^2-1)^n].$$

4) Using Rodrigues's formula one can show, with some tedious manipulations, that

$$\frac{d}{dx}\left[(x^2-1)\frac{dP_n}{dx}\right] = n(n+1)P_n,$$

i.e., the P_n are the eigenfunctions of the given differential operator.

5) Using this equation one can prove that $|P_n| \le 1$ on $[-1,1]$ (see Isaacson & Keller, Ch. 5, Sec. 3, problem 8).

6) P_n has n simple roots, all in $(-1,1)$. Indeed to prove this, it suffices to show that P_n changes sign at n points in $(-1,1)$. If the points where P_n changes sign in $(-1,1)$ are $x_1,\ldots,x_k$, then P_n isn't orthogonal to $(x-x_1)\cdots(x-x_k)$, so $k \ge n$.

Using the Legendre polynomials to compute the least squares approximation has the additional advantage that if we increase the degree the polynomial approximation changes only by adding additional terms: the coefficients of the terms already present don't change.

Now consider the error $f - \sum c_j P_j$. We have

$$\|f - \sum_{j=0}^{n} c_j P_j\|^2 = \|f\|^2 + \sum_{j=0}^{n} c_j^2\|P_j\|^2 - 2\sum_{j=0}^{n} c_j\langle f, P_j\rangle.$$

But $\langle f, P_j\rangle = c_j\|P_j\|^2$, so

$$\|f - \sum c_j P_j\|^2 = \|f\|^2 - \sum_{j=0}^{n} c_j^2\|P_j\|^2$$

In particular, this shows that $\sum_{j=0}^{n} c_j^2\|P_j\|^2$ is bounded above by $\|f\|^2$ for all n, so the limit $\sum_{j=0}^{\infty} c_j^2\|P_j\|^2$ exists. In fact this limit must equal $\|f\|^2$, since if it were strictly less than $\|f\|^2$ we would have $\|f - \sum_{j=0}^{n} c_j P_j\| \ge \delta$ for some $\delta > 0$ and all n, i.e., $\inf_{p\in\mathcal{P}_n}\|f-p\|_2 \ge \delta$ for all n. But, $\inf_{p\in\mathcal{P}_n}\|f-p\|_2^2 \le \inf_{p\in\mathcal{P}_n}\|f-p\|_\infty^2$, and the latter tends to zero by the Weierstrass Approximation Theorem.

THEOREM 1.23. *Let $f \in C([a,b])$ and let P_n be the orthogonal polynomials on $[a,b]$ (with any normalization). Then the best approximation to f in $\mathcal{P}_n$ with respect to the L^2 norm is given by*

$$p_n = \sum_{j=0}^{n} c_j P_j, \qquad c_n = \frac{\langle f, P_j \rangle}{\|P_j\|^2}. \tag{1.5}$$

The L^2 error is given by

$$\|f - p_n\|^2 = \|f\|^2 - \sum_{j=0}^{n} c_j^2 \|P_j\|^2 = \|f\|^2 - \|p_n\|^2$$

and this quantity tends to zero as n tends to infinity. In particular

$$\|f\|^2 = \sum_{j=0}^{\infty} c_j^2 \|P_j\|^2.$$

The last equation is Parseval's equality. It depended only on the orthogonality of the P_n and on the completeness of the polynomials (any continuous function can be represented arbitrarily closely in L^2 by a polynomial).

4.3. Error analysis. We now consider the rate of convergence of p_n, the best L^2 approximation of f, to f. An easy estimate follows from the fact that $\|f\|_{L^2([-1,1]} \le \sqrt{2}\|f\|_{L^\infty([-1,1])}$:

$$\inf_{p \in \mathcal{P}_n} \|f - p\|_2 \le \sqrt{2} \inf_{p \in \mathcal{P}_n} \|f - p\|_\infty.$$

The right-hand side can be bounded Jackson's theorems, e.g., by $c_k n^{-k} \|f^{(k)}\|_\infty$.

Another interesting question is whether p_n converges to f in L^∞. For this we will compute the Lebesgue constant for best L^2 approximation. That is we shall find a number c_n such that $\|p_n\|_\infty \le c_n \|f\|_\infty$ whenever p_n is the best L^2 approximation of f in $\mathcal{P}_n$. It then follows that $\|f - p_n\|_\infty \le (1 + c_n) \inf_{q \in \mathcal{P}_n} \|f - q\|_\infty$, and again we can apply the Jackson theorems to bound the right-hand side. Now we know that all norms on the finite dimensional space $\mathcal{P}_n$ are equivalent, so there is a constant K_n such that $\|q\|_\infty \le K_n \|q\|_2$ for all $q \in \mathcal{P}_n$. Then

$$\|p_n\|_\infty \le K_n \|p_n\|_2 \le K_n \|f\|_2 \le \sqrt{2} K_n \|f\|_\infty,$$

and so the Lebesgue constant is bounded by $\sqrt{2}K_n$. To get a value for K_n, we write an arbitrary element of $\mathcal{P}_n$ as $q = \sum_{k=0}^n a_k P_k$. Then $\|q\|_\infty \le \sum_{k=0}^n |a_k|$, and $\|q\|_2^2 = \sum_{k=0}^n |a_k|^2 \frac{2}{2k+1}$. Thus for the best choice for K_n,

$$K_n^2 = \max \frac{(\sum_{k=0}^n |a_k|)^2}{\sum_{k=0}^n |a_k|^2 \frac{2}{2k+1}}.$$

We can find the maximum using the Cauchy-Schwarz inequality:

$$(\sum_{k=0}^n |a_k|)^2 = (\sum_{k=0}^n (|a_k| \sqrt{\frac{2}{2k+1}}) \sqrt{\frac{2k+1}{2}})^2 \le \left(\sum_{k=0}^n |a_k|^2 \frac{2}{2k+1} \right) \left(\sum_{k=0}^n \frac{2k+1}{2} \right).$$

Thus $K_n^2 \le \sum_{k=0}^n \frac{2k+1}{2} = \frac{(n+1)^2}{2}$. We have thus shown that $\|q\|_\infty \le (n+1)/\sqrt{2}\|q\|_2$ for $q \in \mathcal{P}_n$, the Lebesgue constant of least squares approximation in $\mathcal{P}_n$ is bounded by $n+1$ and, consequently, for any function f for which the best L^∞ approximation converges faster

than $O(1/n)$, the best L^2 approximation converges in L^∞, with at most one lower rate of convergence.

4.4. Weighted least squares. The theory of best approximation in L^2 extends to best approximation in the weighted norm

$$\|f\|_{w,2} := \left(\int_a^b |f(x)|^2 w(x)\,dx\right)^{1/2},$$

where $w : (a,b) \to \mathbb{R}_+$ is an integrable function. The point is, this norm arises from an inner product, and hence most of theory developed above goes through (one thing that does not go through, is that unless the interval is symmetric with respect to the origin and w is even, it will not be the case that the orthogonal polynomials will alternate odd and even parity). Writing $\langle f, g\rangle = \int_a^b f(x)g(x)w(x)\,dx$ for the inner product, we define orthogonal polynomials by a modified Gram-Schmidt procedure as follows. Let $Q_0(x) \equiv 1$. Define $Q_1 = xQ_0 - a_0Q_0$ where a_0 is chosen so that $\langle Q_1, Q_0\rangle = 0$, i.e., $a_0 = \langle xQ_0, Q_0\rangle/\|Q_0\|^2$. Then $Q_2 = xQ_1 - a_1Q_1 - b_1Q_0$ where a_1 is chosen so that $\langle Q_2, Q_1\rangle = 0$ ($a_1 = \langle xQ_1, Q_1\rangle/\|Q_1\|^2$) and b_1 is chosen so that $\langle Q_2, Q_0\rangle = 0$ ($b_1 = \langle (x-a_1)Q_1, Q_0\rangle/\|Q_0\|_2$. We then define $Q_3 = xQ_2 - a_2Q_2 - b_2Q_1$, choosing a_2 and b_2 to get orthogonality to Q_2 and Q_1 respectively. Each of the terms on the right-hand side is individually orthogonal to Q_0, so this procedure works, and can be continued. In summary:

$$Q_{n+1} = (x-a_n)Q_n - b_nQ_{n-1}, \quad a_n = \frac{\langle xQ_n, Q_n\rangle}{\|Q_n\|^2}, \quad b_n = \frac{\langle (x-a_n)Q_n, Q_{n-1}\rangle}{\|Q_{n-1}\|^2}.$$

Actually, $\langle (x-a_n)Q_n, Q_{n-1}\rangle = \|Q_n\|^2$, since $Q_n = (x-a)Q_{n-1} + \cdots$, so we can write $b_n = \|Q_n\|^2/\|Q_{n-1}\|$.

This gives the orthogonal polynomials for the weight w on $[a,b]$ normalized so as to be monic. As for the Legendre polynomials other normalizations may be more convenient.

Probably the most important case is $w(x) = 1/\sqrt{1-x^2}$, in which case we get the Chebyshev polynomials T_n as orthogonal polynomials. To see that these are $L^2([-1,1], w)$ orthogonal make the change of variables $x = \cos\theta$, $0 < \theta < \pi$ to get

$$\int_{-1}^1 T_n(x)T_m(x)\frac{1}{\sqrt{1-x^2}}\,dx = \int_0^\pi \cos n\theta\, \cos m\theta\, d\theta.$$

The recurrence relation is, as we have already seen, is

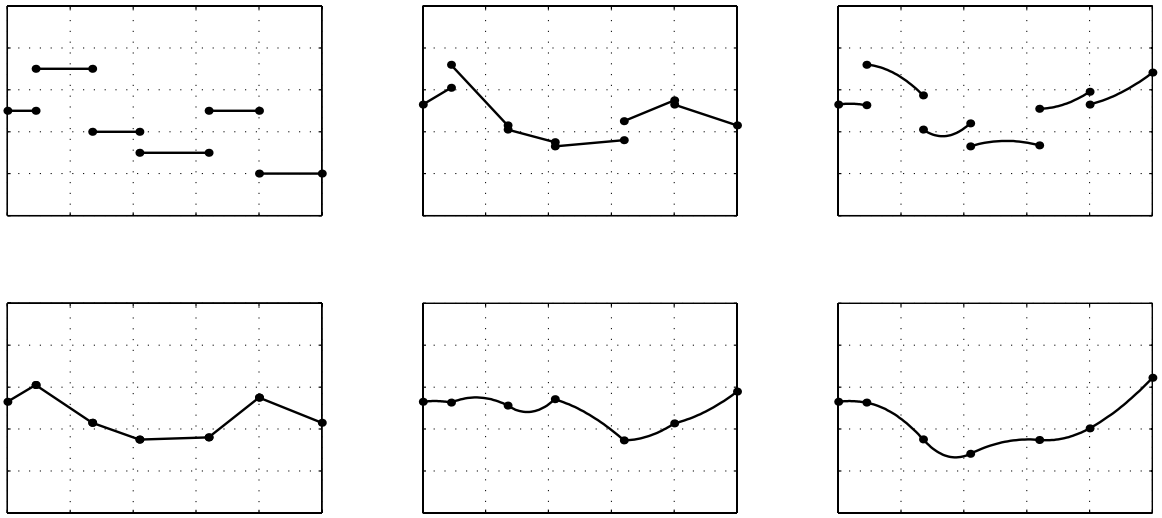
$$T_{n+1} = 2xT_n - T_{n-1}, \quad T_0 = 1, T_1 = x.$$

Other famous families of orthogonal polynomials are: Chebyshev of second kind (weight of $\sqrt{1-x^2}$ on $[-1,1]$), Jacobi (weight of $(1-x)^\alpha(1+x)^\beta$ on $[-1,1]$) which contains the three preceding cases as special cases, Laguerre (weight of e^{-x} on $[0,\infty)$, and Hermite (weight of e^{-x^2} on $\mathbb{R}$).

5. Piecewise polynomial approximation and interpolation

If we are given a set of distinct points on an interval and values to impose at those points, we can compute the corresponding Lagrange interpolating polynomial. However we know, e.g., from Runge's example, that for more than a few points, this polynomial may be highly oscillatory even when the values are taken from a smooth underlying function.

FIGURE 1.12. Piecewise polynomials. In the first row are plotted typical elements of $M^0(\mathcal{T})$, $M^1(\mathcal{T})$, and $M^2(\mathcal{T})$. In the second are shown typical elements of $M_0^1(\mathcal{T})$, $M_0^2(\mathcal{T})$, and $M_1^2(\mathcal{T})$. The mesh $\mathcal{T}$ consists of the same six subintervals in all cases.



In this section we shall explore an alternative to the polynomials for interpolation, namely piecewise polynomials. We shall see that piecewise polynomial give good approximation to smooth functions. Unlike polynomials, they are not infinitely differentiable functions. However we can choose the degree of smoothness (C^0, C^1, ...) according to our needs.

Given an interval $[a, b]$, choose *breakpoints* $a = x_0 < x_1 < \cdots < x_n = b$ and define the subintervals $I_m := [x_{m-1}, x_m]$, $m = 1, 2, \ldots, n$. Then the set $\mathcal{T}$ of these subintervals is a *partition* of the interval $[a, b]$. We denote by $M^k(\mathcal{T})$ the space of functions on $[a, b]$ which restrict to polynomials of degree at most k on each of the subintervals (x_{i-1}, x_i). We don't insist that these piecewise polynomials are continuous so that their value at the breakpoints x_i may not be defined. The subspace of continuous functions in $M^k(\mathcal{T})$ will be denoted by $M_0^k(\mathcal{T})$. More generally for $s = 0, 1, \ldots$ we define

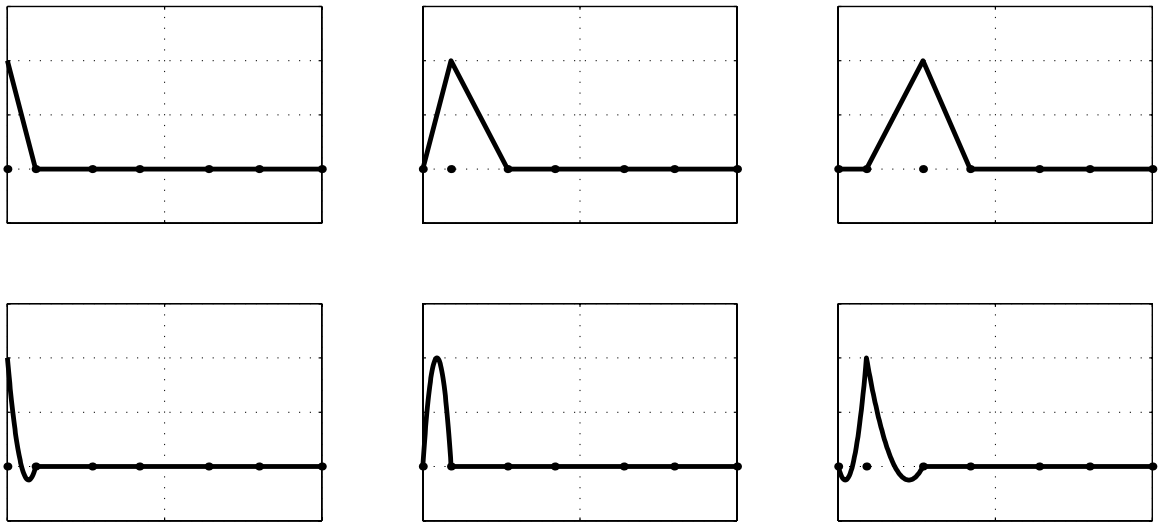
$$M_s^k(\mathcal{T}) = M^k(\mathcal{T}) \cap C^s([a, b]).$$

Figure 1.12 shows typical elements of these spaces for various k and s.

Note that when $s \geq k$, $M_s^k(\mathcal{T}) = \mathcal{P}_k(T)$: there are no piecewise polynomials of degree k in C^k except global polynomials. However, if $s \leq k-1$, then the space $M_s^k(\mathcal{T})$ strictly contains $\mathcal{P}_k(T)$ (as long as the partition contains more than one subinterval), and its dimension grows with the number of subintervals. To determine the dimension explicitly we note that an element of $p \in M_s^k(\mathcal{T})$ can be specified as follows: first choose $p_1 = p|_{I_1}$ as an arbitrary element of $\mathcal{P}_k(I_1)$; then choose $p_2 = p|_{I_2}$ as an arbitrary element of $\mathcal{P}_k(I_2)$ subject to the constraint that $p_2^{(m)}(x_1) = p_1^{(m)}(x_1)$, $m = 0, 1, \cdots, s$; then choose $p_3 = u|_{I_3}$ as an arbitrary element of $\mathcal{P}_k(I_2)$ subject to the constraint that $p_3^{(m)}(x_2) = p_3^{(m)}(x_2)$, $m = 0, 1, \cdots, s$; and so forth. In this we we see that

$$\dim M_s^k(\mathcal{T}) = (k - s)n + s + 1$$

FIGURE 1.13. Some Lagrange basis functions for $M_0^1(\mathcal{T})$ (first row) and $M_0^2(\mathcal{T})$ (second row).

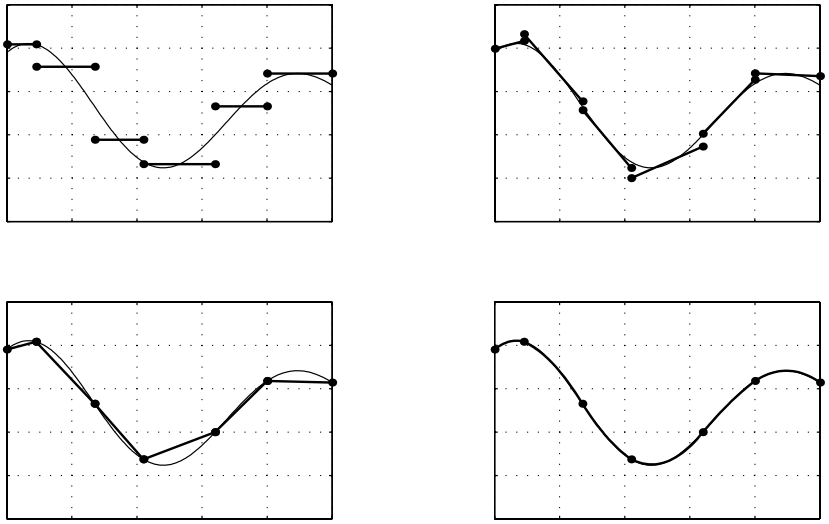


for $0 \le s \le k$ (and even for $s = -1$ if we interpret M_{-1}^k as M^k). We get the same result if we start with the dimension of $M^k(\mathcal{T})$, which is $n(k+1)$, and subtract $(s+1)(n-1)$ for the value and first s derivatives which need to be constrained at each of the $n-1$ interior breakpoints.

Since they are continuous, the functions $p \in M_0^k(\mathcal{T})$ have a well-defined value $p(x)$ at each $x \in [a,b]$, (including the possibility that x is a breakpoint). Consider the set S of points in $[a,b]$ consisting of the $n+1$ breakpoints and an additional $k-1$ distinct points in the interior of each subinterval. For definiteness, when $k=2$ we use the midpoint of each subinterval, when $k=3$, we use the points $1/3$ and $2/3$ of the way across the interval, etc. Thus S contains $nk+1$ points, exactly as many as the dimension of $M_0^k(\mathcal{T})$. An element $p \in M_0^k(\mathcal{T})$ is uniquely determined by its value at these $nk+1$ points, since—according to the uniqueness of Lagrange interpolation—it is uniquely determined on each subinterval by its value at the k+1 points of S in the subinterval (the two end points of the subinterval and $k-1$ points in the interior). Thus the interpolation problem of finding $p \in M_0^k(\mathcal{T})$ taking on given values at each of the points in S has a unique solution. This observation leads us to a useful set of basis of function for $M_0^k(\mathcal{T})$, analogous to the Lagrange basis functions for $\mathcal{P}_n$ discussed in § 3.1, which we shall call a Lagrange basis for $M_0^k(\mathcal{T})$. Namely, for each $s \in S$ we define a basis function $\phi_s(x)$ as the element of $M_0^k(\mathcal{T})$ which is equal to 1 at s and is zero at all the other points of S. Figure 1.13 shows the first few basis functions for $M_0^1(\mathcal{T})$ and $M_0^2(\mathcal{T})$. Notice that this is a *local basis* in the sense that all the basis functions have small supports: they are zero outside one or two subintervals. This is in contrast to the Lagrange basis functions for $\mathcal{P}_n$, and is an advantage of piecewise polynomial spaces.

To approximate a given function f on $[a,b]$ by a function in $p \in M^k(\mathcal{T})$ we may independently specify p in each subinterval, e.g., by giving $k+1$ interpolation points in the subinterval. In order to obtain a continuous approximation ($p \in M_0^k$) it suffices to include the endpoints among the interpolation points. For example, we may determine a continuous

FIGURE 1.14. Four piecewise polynomial interpolants of the same smooth function. In the first row we see a piecewise constant interpolant determined by interpolation at the interval midpoints, and a piecewise linear interpolant determined by interpolation at points 1/3 and 2/3 of the way across each subinterval. The bottow row shows a continuous piecewise linear interpolant determined by using the breakpoints as interpolation points, and the a continuous piecewise quadratic interpolant in which both the breakpoints and the interval midpoints are taken as interpolation points.



piecewise linear approximation by interpolating f at the breakpoints. We may determine a continuous piecewise quadratic approximation by interpolating at the breakpoints and the midpoint of each subinterval. In terms of the basis functions described in the last paragraph, the formula for the interpolant in $M_0^k(\mathcal{T})$ is simple:

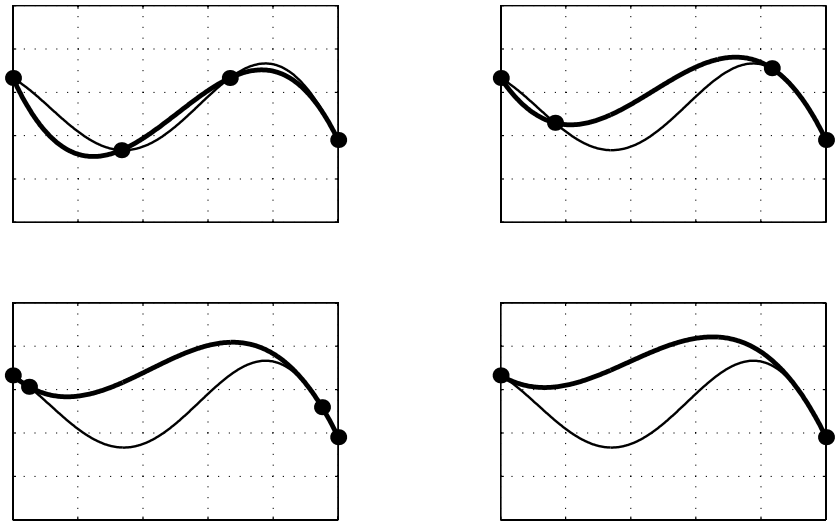
$$p(x) = \sum_{s \in S} f(s)\phi_s(x).$$

Because the basis functions have small support this sum is cheap to evaluate even when there are many subintervals (because for any given x, only a few of the $\phi_s(x)$ are non-zero). Figure 1.14 shows the interpolants of a smooth function using various piecewise polynomial spaces.

It is easy to obtains error bounds for these interpolation procedures because on each subinterval I_i we are simply performing Lagrange interpolation, and so we may apply the estimate (1.3) to bound the $L^\infty(I_i)$ error of the interpolant in terms of $\|f^{(k+1)}\|_{L^\infty(I_i)}$ and $h_i = x_i - x_{i-1}$. Taking the maximum over all the subintervals gives us an L^∞ error bound on the entire interval in terms of $\|f^{(k+1)}\|_{L^\infty([a,b])}$ and the *mesh size* $h := \max_i h_i$. These considerations are summarized in the following theorem.

THEOREM 1.24. *Let $\mathcal{T}$ be a mesh of $[a,b]$, k a non-negative integer, and f a function on $[a,b]$. Given $k+1$ interpolation points in each subinterval of the mesh, there exists a unique function $p \in M^k(\mathcal{T})$ interpolating f at all the interpolation points. Moreover, if*

FIGURE 1.15. As the interior mesh points tend to the endpoints, the Lagrange cubic interpolant tends to the Hermite cubic interpolant.



$f \in C^{(k+1)}([a,b])$, then

$$\|f - p\|_{L^\infty([a,b])} \leq \frac{1}{(k+1)!} \|f^{(k+1)}\|_{L^\infty([a,b])} h^{k+1}$$

where h is the mesh size. Finally, if $k > 0$ and on each subinterval the interpolation points include the endpoints, then $p \in M_0^k(\mathcal{T})$.

REMARK. The constant $1/(k+1)!$ can be improved for particular choices of interpolation points.

Interpolation by smoother piecewise polynomials (elements of $M_s^k(\mathcal{T})$ with $s > 0$) can be trickier. For example, it is not evident what set of $n+2$ interpolation points to use to determine an interpolant in $M_1^2(\mathcal{T})$, nor how to bound the error. The situation for the space $M_1^3(\mathcal{T})$, the space of C^1 piecewise cubic polynomials is simpler. Given a C^1 function f on an interval $[\alpha, \beta]$, there is a unique element of $p \in \mathcal{P}_3([\alpha, \beta])$ such that

$$p(\alpha) = f(\alpha), \quad p(\beta) = f(\beta), \quad p'(\alpha) = f'(\alpha), \quad p'(\beta) = f'(\beta).$$

The cubic p is called the Hermite cubic interpolant to f on $[\alpha, \beta]$. It may be obtained as the limit as $\epsilon \to 0$ of the Lagrange interpolant using interpolation points α, $\alpha + \epsilon$, $\beta - \epsilon$, and β (see Figure 1.15), and so satisfies

$$\|f - p\|_{L^\infty([\alpha,\beta])} \leq \frac{1}{4!} \|f^{(4)}\|_{L^\infty([\alpha,\beta])}.$$

Now if we are given the mesh $\mathcal{T}$, then we may define the piecewise Hermite cubic interpolant p to a C^1 function f by insisting that on each subinterval p be the Hermite cubic interpolant to f. Then p is determined by the interpolation conditions

$$p(x_i) = f(x_i), \quad p'(x_i) = f'(x_i), \quad i = 0, 1, \ldots, n. \tag{1.6}$$

By construction $p \in M_1^3(\mathcal{T})$. When $f \in C^4([a,b])$ we obtain an $O(h^4)$ error estimate just as for the piecewise Lagrange cubic interpolant. We can specify C^1 interpolants of higher degree and order, by supplementing the conditions (1.6) with additional interpolation conditions. Thus to interpolate in $M_1^k(\mathcal{T})$, $k > 3$, we insist that p satisfy (1.6) and as well interpolate f at $k-3$ points interior to each subinterval. It is possible to obtain even smoother interpolants (C^2, C^3, ...) using the same idea. But to obtain an interpolant in C^s in this way it is necessary that the degree k be at least $2s+1$.

5.1. Cubic splines. The space $M_2^3(\mathcal{T})$ of cubic splines has dimension $n+3$. It is therefore reasonable to try to determine an element p of this space by interpolation at the nodes, $p(x_i) = f(x_i)$, and two additional conditions. There are a number of possible choices for the additional conditions. If the values of f' are known at the end points a natural choice is $p'(a) = f'(a)$ and $p'(b) = f'(b)$. We shall mostly consider such *derivative end conditions here.* If the values of f' are not known, one possibility is approximate $f'(a)$ by $r'(a)$ where $r \in \mathcal{P}_3$ agrees with f at x_i, $i = 0,1,2,3$. Another popular possibility is to insist that p''' be continuous at x_1 and x_{n-1}. This means that p belongs to $\mathcal{P}_3([x_0,x_2])$ and $\mathcal{P}_3([x_{n-2},x_n])$. That is, x_1 and x_{n-1} are not true breakpoints or knots. Thus these are called the not-a-knot conditions.

We shall now proceed to proving that there exists a unique cubic spline interpolating f at the breakpoints and f' at the end points. With derivative end conditions it is convenient to define $x_{-1} = x_0 = a$, $x_{n+1} = x_n = b$ and $h_0 = h_{n+1} = 0$. This often saves us the trouble of writing special formulas at the end points.

LEMMA 1.25. *Suppose that e is any function in $C^2([a,b])$ for which $e(x_i) = e'(a) = e'(b) = 0$. Then $e'' \perp M_0^1(\mathcal{T})$ in $L^2([a,b])$.*

PROOF. Let $q \in M_0^1(\mathcal{T})$. Integrating by parts and using the vanishing of the derivatives at the end points we have

$$\int_a^b e''q\,dx = -\int_a^b e'q'\,dx.$$

But, on each subinterval $[x_{i-1},x_i]$, q' is constant and e vanishes at the end points. Thus

$$\int_{x_{i-1}}^{x_i} e'q'\,dx = 0.$$

□

THEOREM 1.26. *Given breakpoints $x_0 < x_1 < \cdots < x_n$ and values $y_0,\dots,y_n,y_a',y_b'$ there exists a unique cubic spline p satisfying $p(x_i) = y_i$, $p'(a) = y_a'$, $p'(b) = y_b'$.*

PROOF. Since the space of cubic splines has dimension $n+3$ and we are given $n+3$ conditions, it suffices to show that if all the values vanish, then p must vanish. In that case, we can take $e = p$ in the lemma, and, since $p'' \in M_0^1(\mathcal{T})$ we find that p'' is orthogonal to itself, hence vanishes. Thus $p \in \mathcal{P}_1([a,b])$. Since it vanishes at a and b, it vanishes identically. □

The lemma also is the key to the error analysis of the cubic spline interpolant.

THEOREM 1.27. *Let $f \in C^2([a,b])$ and let p be its cubic spline interpolant with derivative end conditions. Then p'' is the best least squares approximation to f'' in $M_0^1(\mathcal{T})$.*

If we write $I_{M_2^3} f$ for the cubic spline interpolant of f and $\Pi_{M_0^1} g$ for the L^2 projection (best least squares approximation) of g in M_0^1, we may summarize the result as

$$(I_{M_2^3} f)'' = \Pi_{M_0^1} f'',$$

or by the commutative diagram

$$\begin{array}{ccc} C^2([a,b]) & \xrightarrow{\frac{d^2}{dx^2}} & C^0([a,b]) \\ {\scriptstyle I_{M_2^3}}\downarrow & & \downarrow{\scriptstyle \Pi_{M_0^1}} \\ M_2^3(\mathcal{T}) & \xrightarrow[\frac{d^2}{dx^2}]{} & M_0^1(\mathcal{T}) \end{array}$$

PROOF. We just have to show that the normal equations

$$\langle f'' - p'', q \rangle = 0, \quad q \in M_0^1(\mathcal{T}),$$

are satisfied, where $\langle \cdot, \cdot \rangle$ is the $L^2([a,b])$ inner product. This is exactly the result of the lemma (with $e = f - p$). □

The theorem motivates a digression to study least squares approximation by piecewise linears. We know that if $g \in L^2([a,b])$ there is a unique best approximation $p = \Pi_{M_0^1} g$ to g in $M_0^1(\mathcal{T})$, namely the L^2 projection of g onto $M_0^1(\mathcal{T})$, determined by the normal equations

$$\langle g - p, q \rangle = 0, \quad q \in M_0^1(\mathcal{T}).$$

We now bound the Lebesgue constant of this projection, that is the L^∞ operator norm

$$\|\Pi_{M_0^1}\|_\infty := \sup_{\substack{f \in C([a,b]) \\ \|f\|_\infty \le 1}} \|\Pi_{M_0^1} f\|_{L^\infty}.$$

THEOREM 1.28. $\|\Pi_{M_0^1}\|_\infty \le 3$.

PROOF. Let $\phi_i \in M_0^1(\mathcal{T})$ denote the nodal basis function at the breakpoint x_i. Then $r = \Pi_{M_0^1} f = \sum_j \alpha_j \phi_j$ where

$$\sum_{j=0}^{n} \alpha_j \langle \phi_j, \phi_i \rangle = \langle f, \phi_i \rangle, \quad i = 0, \ldots, n.$$

Now we can directly compute

$$\begin{aligned} \langle \phi_j, \phi_j \rangle &= \frac{h_j + h_{j+1}}{3}, \\ \langle \phi_{j-1}, \phi_j \rangle &= \frac{h_j}{6}, \\ \langle \phi_j, \phi_k \rangle &= 0, \quad |j - k| \ge 2, \end{aligned}$$

(recall that by convention $h_0 = h_{n+1} = 0$). Thus the ith normal equation in this basis is

$$h_i \alpha_{i-1} + 2(h_i + h_{i+1})\alpha_i + h_{i+1}\alpha_{i+1} = 6\langle f, \phi_i \rangle,$$

where again we define $h_0 = h_{n+1} = 0$. We remark in passing that the matrix representing the normal equations in this basis is symmetric, tridiagonal, positive definite, and strictly

diagonally dominant, so easy to solve. (Gaussian elimination without pivoting works in $O(n)$ operations.)

Let $\alpha = \max_k |\alpha_k| = \|r\|_\infty$. Choose i such that $|\alpha_i| = \alpha$ and write the ith equation as

$$2(h_i + h_{i+1})\alpha_i = 6\langle f, \phi_i\rangle - h_i\alpha_{i-1} - h_{i+1}\alpha_{i+1}.$$

Taking absolute values we get

$$2(h_i + h_{i+1})\|r\|_\infty \le 6|\langle f, \phi_i\rangle| + h_i\|r\|_\infty + h_{i+1}\|r\|_\infty.$$

Now $|6\langle f, \phi_i\rangle| \le 6\|f\|_\infty\langle 1, \phi_i\rangle = 3(h_{i+1} + h_i)\|f\|_\infty$ and the result follows. □

It follows that $\|f - \Pi_{M_0^1} f\|_\infty \le 4 \inf_{q\in M_0^1}\|f - q\|_\infty$, and, for $f \in C^2$, $\|f - \Pi_{M_0^1} f\|_\infty \le \frac{1}{2}h^2\|f''\|_\infty$.

Returning now to the case of p the cubic spline interpolant of $f \in C^4$, we know that $p'' = \Pi_{M_0^1} f''$, so

$$\|f'' - p''\|_\infty \le \frac{1}{2}h^2\|f^{(4)}\|.$$

This gives a bound on the W_∞^2 seminorm. To move down to the L^∞ norm, we note that $f - p$ vanishes at the breakpoints, and hence its piecewise linear interpolant is 0. Applying the error estimate for piecewise linear interpolation gives

$$\|f - p\|_\infty \le \frac{1}{8}h^2\|f'' - p''\|_\infty.$$

Combining the last two estimates gives us an error bound for cubic spline interpolation:

$$\|f - p\|_\infty \le \frac{1}{16}h^4\|f^{(4)}\|.$$

The relation $(I_{M_2^3} f)'' = \Pi_{M_0^1} f''$ can be used to compute the cubic spline interpolant as well. We know that

$$p''(x) = \alpha_{j-1}\frac{x_j - x}{h_j} + \alpha_j\frac{x - x_{j-1}}{h_j}, \quad x \in [x_{j-1}, x_j],$$

where the α_j are determined by the tridiagonal system

$$h_j\alpha_{j-1} + 2(h_j + h_{j+1})\alpha_j + h_{j+1}\alpha_{j+1} = 6\langle f'', \phi_j\rangle.$$

Since integration by parts gives

$$\langle f'', \phi_j\rangle = (h_j + h_{j+1})f[x_{j-1}, x_j, x_{j+1}],$$

the right-hand side of the tridiagonal system can be written in terms of the data $f(x_i)$, $f'(a)$, $f'(b)$, so the α_i can be computed.

Integrating twice we then get

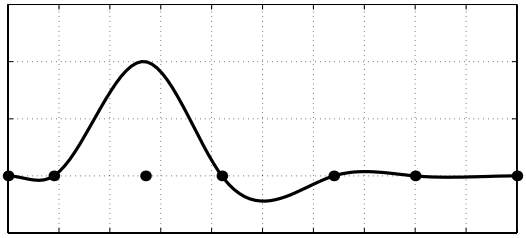
$$p(x) = \alpha_{j-1}\frac{(x_j - x)^3}{6h_j} + \alpha_j\frac{(x - x_{j-1})^3}{6h_j} + A_j\frac{x_j - x}{h_j} + B_j\frac{x - x_{j-1}}{h_j}, \quad x \in [x_{j-1}, x_j].$$

Finally we may apply the interpolation conditions to find that

$$A_j = f(x_{j-1}) - \alpha_{j-1}\frac{h_j^2}{6}, \quad B_j = f(x_j) - \alpha_j\frac{h_j^2}{6},$$

and so we have an explicit formula for p on each subinterval $[x_{j-1}, x_j]$.

FIGURE 1.16. A Lagrange basis function for cubic spline interpolation. This cubic spline is equal to 1 at the third breakpoint and to 0 at the others, and satisfies zero derivative endpoint conditions. Notice that, although its magnitude decreases quickly away from the third breakpoint, it is non-zero on every subinterval.



While this procedure allows one to efficiently compute the cubic spline interpolant to given data, it is more complicated than computing a merely continuous piecewise polynomial interpolant or a Hermite cubic interpolant. This is because, through the linear system which must be solved, the cubic spline on each subinterval depends on all the interpolation data, not just the data on the subinterval. Otherwise put, if we define a cubic spline by setting its value equal to one at one of the breakpoints, and its other values and end conditions equal to zero (i.e., a Lagrange basis function for the space of cubic splines, see Figure 1.16), this function will not vanish on any subinterval. (It does, however, tend to decrease quickly away from the breakpoint where it is equal to one, as suggested by the figure.)

6. Piecewise polynomials in more than one dimension

Much of the theory of Lagrange interpolation and best L^2 approximation does not extend from one to more dimensions. However the theory of continuous piecewise polynomial approximation and interpolation extends rather directly and has many important applications. One of the most important of these is the finite element method for the numerical solution of partial differential equations which will be studied later in the course.

Here we will consider the case of a plane domain, that is, $n = 2$, although the extension to $n > 2$ is quite similar. Let Ω be a polygon. By a *triangulation* of Ω we mean a set $\mathcal{T}$ of closed triangles T with the properties that

(1) $\bigcup_{T\in\mathcal{T}} T = \bar{\Omega}$
(2) any two distinct elements $T_1, T_2 \in \mathcal{T}$ are either disjoint or meet in a common edge or vertex

The second point means we exclude configurations such as shown in Figure 1.17 from triangulations.

Figure 1.18 shows a triangulation of a polygonal domain.

As in one-dimension, we write $M^k(\mathcal{T})$ as the space of functions on Ω whose restriction to T is a polynomial function of degree at most k. These functions may be discontinuous (and for our purposes their values on the edges of the triangles are irrelevant). The space $\mathcal{P}_k(T)$ of polynomial functions of degree at most k on T has dimension $(k+1)(k+2)/2$ when T is a 2-dimensional domain. Therefore $\dim M^k(\mathcal{T}) = N_{\mathrm{T}}(k+1)(k+2)/2$ where N_{T} where N_{T} is the number of triangles in the triangulation.

FIGURE 1.17. Inadmissible configurations of triangles for a triangulation.

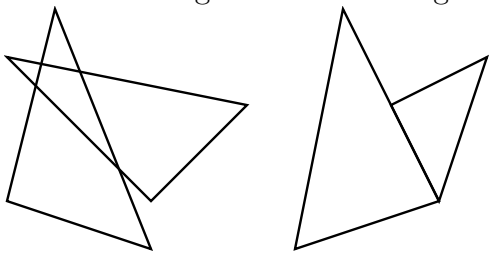
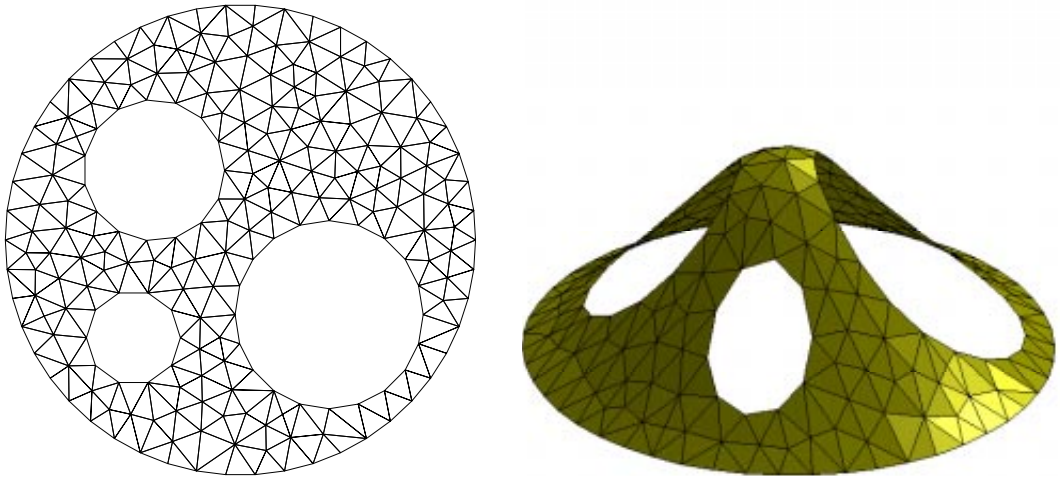


FIGURE 1.18. A triangulation and a continuous piecewise linear function.



Of more interest to us is the space

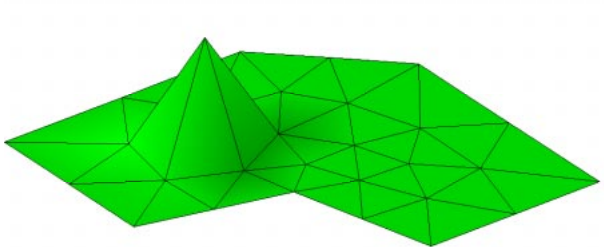
$$M_0^k(\mathcal{T}) = M^k(\mathcal{T}) \cap C(\Omega),$$

of continuous piecewise polynomials of degree k with respect to the triangulation $\mathcal{T}$. An element of $M_1^k(\mathcal{T})$ is plotted along with the mesh in Figure 1.18. Our first task will be to determine the dimension of this space and exhibit a local basis. First, let $k = 1$. If we associate to each vertex of the triangulation a value, then on each triangle we may uniquely determine a linear polynomial taking the given values at the vertices. In this way we define a piecewise linear function. The function is continuous. To show this we need only show that if two triangles share a common edge, then the linear polynomials determined by interpolation of the vertex values on the two triangles agree on the common edge. But the restriction of these polynomials to the edge is a linear polynomial in one variable (say in the distance along the edge from one of the vertices), and they agree at the two endpoints of the edge, so they must agree on the entire edge. In this way we obtain a formula for the dimension of the space of piecewise linears with respect to the given triangulation:

$$\dim M_0^1(\mathcal{T}) = V_{\mathcal{T}},$$

where $V_{\mathcal{T}}$ is the number of vertices in the triangulation. We also obtain a Lagrange basis for $M_0^1(\mathcal{T})$, a typical element of which is shown in Figure 1.19. For obvious reasons, these are called *hat functions*. Notice that the basis is local: each basis function is zero outside the

FIGURE 1.19. A typical Lagrange basis function for the space $M_0^1(\mathcal{T})$.



union of triangles containing one vertex.

Similar considerations allow us to determine the dimension and a local basis for $M_0^k(\mathcal{T})$ for $k > 1$. This time we assign values at the vertices and the midpoints of edges. Our first claim is that for T a triangle we can determine $p \in \mathcal{P}_2(T)$ by giving the values at the vertices of T and at the edge midpoints. There is something to prove here, since *a priori* there might exist a non-zero quadratic polynomial vanishing at all six of these points. To see that this doesn't happen we notice that the restriction of the polynomial to each edge is a quadratic function on the edge which vanishes at three points on the edge. This certainly implies that the restriction to each edge vanishes identically. But if a polynomial vanishes on a line, then it must be divisible by the linear polynomial defining the line. If the three lines through the edges of the triangle are given by $l_i(x) = 0$, $i = 1, 2, 3$, which $l_i \in \mathcal{P}_2(\mathbb{R}^2)$, we conclude that the polynomial is divisible by the product $l_1 l_2 l_3$. Since this product has degree 3 and the polynomial in question has degree at most 2, the only possibility is that the polynomial is identically 0.

Thus we can indeed determine an element of $M^2(\mathcal{T})$ by giving arbitrary values at vertices and edge midpoints. To see that the resulting function is in $M_0^2(\mathcal{T})$ we again observe that on a common edge to two triangles, the restriction to the edge is fully determined by the values at the endpoints and midpoint of the edge. We conclude that

$$\dim M_0^2(\mathcal{T}) = V_{\mathcal{T}} + E_{\mathcal{T}},$$

with $E_{\mathcal{T}}$ the number of edges of triangles. Figure 1.20 shows a typical basis element associated with a vertex and another one associates with and edge midpoint.

Similar considerations apply to continuous piecewise polynomials of any degree. The appropriate interpolation points for $\mathcal{P}_k$ on a single triangle T are shown for degrees 1 through 4 in Figure 1.21

6.1. The Bramble–Hilbert Lemma. Our next goals is to obtain error estimates for piecewise polynomial interpolation in two dimensions. In this section we prove a lemma which will be key to obtaining the estimates.

FIGURE 1.20. Two Lagrange basis functions for the space $M_0^2(\mathcal{T})$.

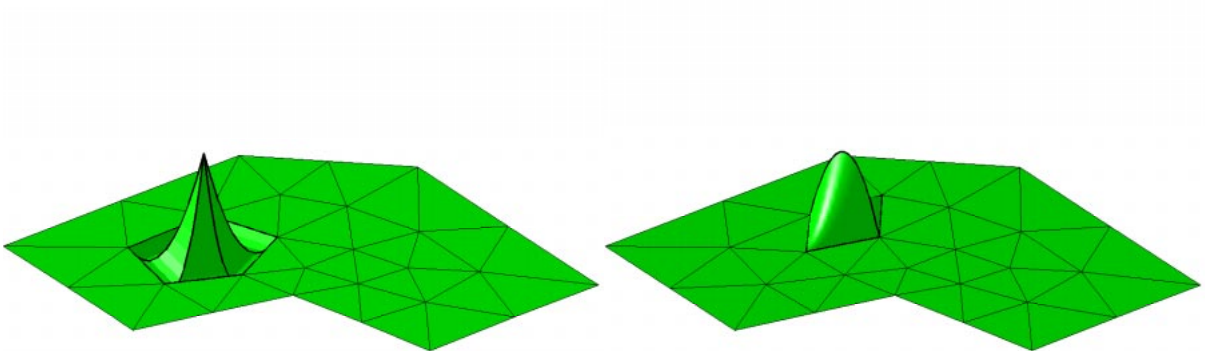
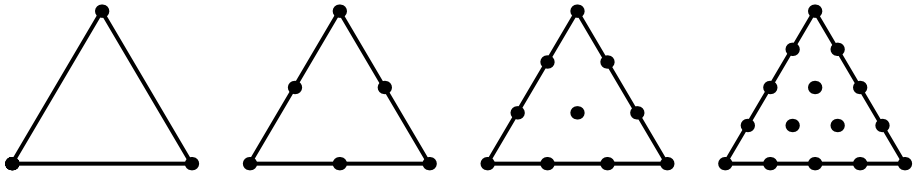


FIGURE 1.21. Interpolation points for $\mathcal{P}_k(T)$, $k = 1, 2, 3, 4$.



THEOREM 1.29 (Bramble–Hilbert Lemma). *Let $\Omega \subset \mathbb{R}^N$ be a bounded open convex set and let n be a non-negative integer. Then there exists a constant C depending only on Ω and n, such that for $1 \le p \le \infty$,*

$$\inf_{p \in \mathcal{P}_n(\Omega)} \|u - p\|_{W_p^{n+1}(\Omega)} \le C|u|_{W_p^{n+1}(\Omega)}$$

for all $u \in C^{n+1}(\Omega)$.

Here, we shall prove the Bramble–Hilbert Lemma in the case of plane domains ($N = 2$), but essentially the same proof works in any number of dimensions.

We first introduce *multi-index notation*. Since we are working in $\mathbb{R}^2$ a multi-index is just a pair $\alpha = (\alpha_1, \alpha_2)$ of non-negative integers. We write $|\alpha|$ for $\alpha_1 + \alpha_2$, $D^\alpha u$ for $\partial^{|\alpha|} u / \partial x_1^{\alpha_1} \partial x_2^{\alpha_2}$, x^α for $x_1^{\alpha_1} x_2^{\alpha_2}$, and $\alpha!$ for $\alpha_1! \alpha_2!$. With this notation we can state Taylor's theorem with remainder for functions of two variables as follows.

THEOREM 1.30. *Let u be $n+1$ times continuously differentiable on a neighborhood of the line segment connecting two points x and y in $\mathbb{R}^2$. Then*

$$u(x) = T_y^n u(x) + R(x, y), \tag{1.7}$$

where

$$T_y^n u(x) = \sum_{|\alpha| \le n} \frac{1}{\alpha!} D^\alpha u(y)(x - y)^\alpha$$

is the Taylor polynomial of degree n for u about y and

$$R(x, y) = \sum_{|\alpha| = n+1} \frac{n+1}{\alpha!} \int_0^1 s^n D^\alpha u\big(x + s(y - x)\big)(y - x)^\alpha \, ds.$$

PROOF. Let $F(s) = u(y + s(x - y))$. By Taylor's theorem in one dimension,

$$F(t) = \sum_{m=0}^{n} \frac{1}{m!} F^{(m)}(0) t^m + \frac{1}{n!} \int_0^t (t-s)^n F^{(n+1)}(s)\, ds.$$

Taking $t = 1$ and substituting $s \mapsto 1 - s$ in the last integral gives the result. □

Now we suppose that Ω is a bounded open convex set in $\mathbb{R}^2$ and proceed to the proof of the Bramble–Hilbert lemma. Translating and dilating we can assume that Ω contains the unit ball B.

Integrating (1.7) over $y \in B$ and dividing by π (the area of B), we find that

$$u(x) = P^n u(x) + E(x),$$

where

$$P^n u(x) = \frac{1}{\pi} \int_B T_y^n u(x)\, dy$$

is the *averaged Taylor polynomial* (note that $P^n u \in \mathcal{P}_n(\Omega)$), and

$$E(x) = \sum_{|\alpha|=n+1} \frac{n+1}{\pi \alpha!} \int_{\mathbb{R}^2} \int_0^1 s^n D^\alpha u\big(x + s(y-x)\big) \chi_B(y) (y-x)^\alpha\, ds\, dy.$$

We have used the characteristic function χ_B of the unit ball to enable us to write the outer integral over $\mathbb{R}^2$. Next we change variable in the double integral from (s, y) to (t, z) with $t = s$ and $z = x + s(y - x)$. This gives

$$E(x) = \sum_{|\alpha|=n+1} \frac{n+1}{\pi \alpha!} \int_{\mathbb{R}^2} \int_0^1 t^{-3} D^\alpha u(z) \chi_B\big(x + t^{-1}(z-x)\big) (z-x)^\alpha\, dt\, dz$$

Now if $z \notin \Omega$, then also $x + t^{-1}(z - x) \notin \Omega$, since z lies on the segment joining x to this point. Thus $\chi_B\big(x + t^{-1}(z-x)\big)$ vanishes whenever $z \notin \Omega$, and so the outer integral may be taken over Ω. Rearranging slightly we have

$$\begin{aligned} E(x) &= \sum_{|\alpha|=n+1} \int_\Omega D^\alpha u(z) \left[\frac{n+1}{\pi \alpha!} (z-x)^\alpha \int_0^1 t^{-3} \chi_B\big(x + t^{-1}(z-x)\big)\, dt \right] dz \\ &= \sum_{|\alpha|=n+1} \int_\Omega D^\alpha u(z) K_\alpha(x, z)\, dz, \end{aligned}$$

where $K_\alpha(x, z)$ is defined to be the term in brackets. We shall prove below:

LEMMA 1.31. *There exists a constant C depending only on Ω and n such that*

$$K_\alpha(x, z) \le \frac{C}{|x - z|}, \quad x, z \in \Omega.$$

In particular,

$$\int_\Omega |K_\alpha(x, z)|\, dx, \int_\Omega |K_\alpha(x, z)|\, dz \le C$$

for all $x, z \in \Omega$ (with a possibly different constant C, but still just depending on Ω and n). It follows[1] that

$$\|E\|_{L^p(\Omega)} \le C \sum_{|\alpha|=n+1} \|D^\alpha u\|_{L^p(\Omega)}.$$

Thus (modulo the proof of Lemma 1.31) we have proven the following theorem, which is a part of the Bramble–Hilbert theorem.

THEOREM 1.32. *Let $\Omega \subset \mathbb{R}^N$ be a bounded open convex set and let n be a non-negative integer. Then there exists a constant C depending only on Ω and n, such that for $1 \le p \le \infty$,*

$$\|u - P^n u\|_{L^p(\Omega)} \le C|u|_{W_p^{n+1}(\Omega)},$$

for all $u \in C^{n+1}(\Omega)$.

We now proceed to bound $D^\beta(u - P^n u)$. For the Taylor polynomial

$$D^\beta T_y^n u(x) = T_y^{n-|\beta|} D^\beta u(x)$$

for any multi-index β with $|\beta| \le n$. Averaging over $y \in B$ we get

$$D^\beta P^n u(x) = P^{n-|\beta|} D^\beta u(x).$$

Thus,

$$\|D^\beta(u - P^n u)\|_{L^p(\Omega)} = \|D^\beta u - P^{n-|\beta|} D^\beta u\|_{L^p(\Omega)} \le C|D^\beta u|_{W_p^{n-|\beta|+1}(\Omega)} \le C|u|_{W_p^{n+1}(\Omega)},$$

where we have used Theorem 1.32 in the second step. If $|\beta| = n+1$, then

$$\|D^\beta(u - P^n u)\|_{L^p(\Omega)} = \|D^\beta u\|_{L^p(\Omega)} \le |u|_{W_p^{n+1}(\Omega)}.$$

Thus we have bounded $\|D^\beta(u - P^n u)\|_{L^p(\Omega)}$ by $C|u|_{W_p^{n+1}(\Omega)}$ whenever $|\beta| \le n+1$, which establishes the Bramble–Hilbert Lemma.

It remains to prove Lemma 1.31. From its definition

$$|K_\alpha(x,z)| \le C|z-x|^{n+1} \int_0^1 t^{-3} \chi_B\big(x + t^{-1}(z-x)\big)\, dt,$$

and $n \ge 0$, so it suffices to prove that

$$\int_0^1 t^{-3} \chi_B\big(x + t^{-1}(z-x)\big)\, dt \le \frac{C}{|z-x|^2}.$$

Now $\chi_B\big(x + t^{-1}(z-x)\big)$ vanishes if $|x + t^{-1}(z-x)| \ge 1$ which happens if $t < |z-x|/(1+|x|)$ and, *a fortiori*, if $t \le |z-x|/(2d)$ where $d = \max_{x\in\Omega} |x| \ge 1$. Therefore,

$$\int_0^1 t^{-3} \chi_B\big(x + t^{-1}(z-x)\big)\, dt \le \int_{|z-x|/(2d)}^\infty t^{-3}\, dt = 2\frac{d^2}{|z-x|^2}.$$

[1] This uses the *generalized Young inequality*: if K is a function on $\Omega \times \Omega$ for which $\int_\Omega |K(x,z)|\, dx, \int_\Omega |K(x,z)|\, dz \le C$ and $f \in L^p(\Omega)$, then $g(x) := \int_\Omega K(x,z) f(z)\, dz$ belongs to $L^p(\Omega)$ and $\|g\|_{L^p(\Omega)} \le C\|f\|_{L^p(\Omega)}$.

REMARKS. 1. We have given a constructive proof in that we have exhibited a particular $p \in \mathcal{P}_n$ for which the bound holds. Using functional analysis it is possible to give a much shorter, but non-constructive, proof. See, for example, [**2**], Theorem 3.1.1. The proof presented here is along the lines of that in [**4**]. 2. Usually the result is stated for all $u \in W_p^{n+1}(\Omega)$ rather than in $C^{n+1}(\Omega)$. That result follows from this one, using the density of the latter space in the former. 3. This result became a key tool in approximation theory after the 1970 paper of J. Bramble and S. Hilbert [**1**]. However it can already be found in a much earlier paper of Deny and Lions [**3**].

6.2. Error estimates for piecewise polynomial interpolation. Let T be a triangle of diameter h_T, and define the linear interpolant

$$I_T : C(T) \to \mathcal{P}_1(T)$$

by $I_T f(v) = f(v)$ for each vertex v of T. We shall establish the following L^∞ error estimate:

THEOREM 1.33. *There exists an absolute constant C such that*

$$\|f - I_T f\|_{L^\infty(T)} \le C h_T^2 |f|_{W^2_\infty(T)} \text{ for all } f \in C^2(T).$$

The proof consists of two main steps. First we prove an estimate of the same form, except without any indication of how the error depends on the triangle T. Namely:

THEOREM 1.34. *For each triangle T there exists a constant C_T such that*

$$\|f - I_T f\|_{L^\infty(T)} \le C_T |f|_{W^2_\infty(T)} \text{ for all } f \in C^2(T).$$

In the second step we shall apply Theorem 1.34 for one particular triangle $\hat{T}$ and use an affine map from an arbitrary T to $\hat{T}$ deduce Theorem 1.33.

PROOF OF THEOREM 1.34. If $q \in P_1(T)$, then $I_T q = q$, so $f - I_T f = (f-q) - I_T(f-q)$. Now clearly $\|I_T g\|_{L^\infty} \le \|g\|_{L^\infty}$ for all continuous functions g. Applying this with $g = f - q$ gives

$$\|f - I_T f\|_{L^\infty} = \|(f-q) - I_T(f-q)\|_{L^\infty} \le 2\|f - q\|_{L^\infty}.$$

Since this is true for all $q \in \mathcal{P}_1(T)$, by the Bramble–Hilbert lemma (actually only the part of it given in Theorem 1.32), we get

$$\|f - I_T f\|_{L^\infty(T)} \le C_T |f|_{W^2_\infty(T)},$$

for some constant C_T, as claimed. □

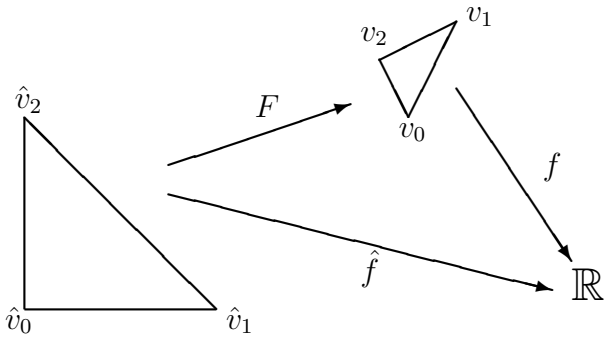
Notice the main ingredients of the proof: first the fact that I_T preserve $\mathcal{P}_1$ ($I_T q = q$ for $q \in \mathcal{P}_1(T)$); second, boundedness of the interpolant I_T ($\|I_T g\|_{L^\infty} \le \|g\|_{L^\infty}$ for all continuous functions g); and finally the Bramble–Hilbert lemma.

To go from the estimate of Theorem 1.34, which does not specify dependence of the constant on T, to that of Theorem 1.33, which shows a dependence of $O(h_T^2)$, we shall apply the former theorem only a single fixed reference triangle, and then use an affine map of the reference triangle onto an arbitrary triangle to deduce the result.

Let $\hat{T}$ be the reference triangle with vertices $\hat{v}_0 = (0,0)$, $\hat{v}_1 = (1,0)$, and $\hat{v}_2 = (0,1)$, and let the vertices of T be denoted $v_0, v_1, v_2 \in \mathbb{R}^2$. There is a unique affine map F taking $\hat{v}_i$ to v_i, $i = 0, 1, 2$, and mapping $\hat{T}$ one-to-one and onto T. Indeed

$$F\hat{x} = v_0 + B\hat{x}, \quad B = \big(v_1 - v_0 \,|\, v_2 - v_0\big),$$

FIGURE 1.22. Mapping between the reference triangle and an arbitrary triangle.



(the last notation means that B is the 2×2 matrix whose columns are the vectors $v_1 - v_0$ and $v_2 - v_0$). Since the columns of B are both vectors of length at most h_T, certainly the four components b_{ij} of B are bounded by h_T. Now to any function f on T we may associate the pulled-back function $\hat{f}$ on $\hat{T}$ where

$$\hat{f}(\hat{x}) = f(x) \quad \text{with } x = F\hat{x}.$$

I.e., $\hat{f} = f \circ F$. See Figure 1.22.

Now, the pull-back of the linear interpolant is the linear interpolant of the pull-back: $I_{\hat{T}}\hat{f} = \widehat{I_T f}$. To verify this, it suffices to note that $\widehat{I_T f u}$ is a linear polynomial and it agrees with $\hat{f}$ at the vertices of $\hat{T}$. Moreover, the pull-back operation clearly doesn't change the L^∞ norm. Thus

$$\|f - I_T f\|_{L^\infty(T)} = \|\hat{f} - I_{\hat{T}}\hat{f}\|_{L^\infty(\hat{T})} \leq C_{\hat{T}} |\hat{f}|_{W^2_\infty(\hat{T})}. \tag{1.8}$$

Since $\hat{T}$ is a fixed triangle, $C_{\hat{T}}$ is an absolute constant: it doesn't depend on T. To finish the argument we need to show that

$$|\hat{f}|_{W^2_\infty(\hat{T})} \leq C h_T^2 |f|_{W^2_\infty(T)}.$$

This just comes from the chain rule for differentiation. We have

$$\frac{\partial \hat{f}}{\partial \hat{x}^i}(\hat{x}) = \sum_{j=1}^{2} \frac{\partial f}{\partial x^j}(x) \frac{\partial x^j}{\partial \hat{x}^i} = \sum_{j=1}^{2} b_{ji} \frac{\partial f}{\partial x^j}(x).$$

Differentiating again,

$$\frac{\partial^2 \hat{f}}{\partial \hat{x}^i \partial \hat{x}^k}(\hat{x}) = \sum_{j=1}^{2} \sum_{k=1}^{2} b_{ji} \frac{\partial^2 f}{\partial x^j \partial x^l}(x) b_{lk}.$$

Thus

$$\left|\frac{\partial^2 \hat{f}}{\partial \hat{x}^i \partial \hat{x}^k}(\hat{x})\right| \leq 4h_T^2 \max_{j,l} \left|\frac{\partial^2 f}{\partial x^j \partial x^l}(x)\right|$$

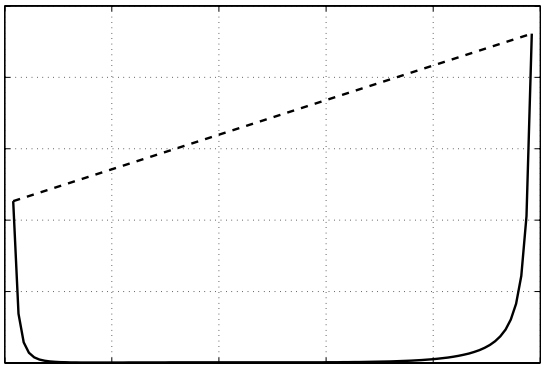
and

$$|\hat{f}|_{W^2_\infty(\hat{T})} \leq 4h_T^2 |f|_{W^2_\infty(T)}. \tag{1.9}$$

Combining (1.8) and (1.9) we complete the proof of Theorem 1.33.

We may use the same sort of arguments to obtain estimates in other norms. First consider the L^p norm. In order to obtain Theorem 1.34 we used the boundedness of the interpolation operator with respect to the L^∞ norm: $\|I_T g\|_{L^\infty} \le \|g\|_{L^\infty}$. This result is not true if we replace L^∞ with L^p for $p < \infty$. The difficulty is not with the norm on the left-hand side. Indeed $I_T g$ lies in the 3-dimensional space $\mathcal{P}_1(T)$ and all norms are equivalent on a finite dimensional space. The difficulty is that a continuous function on a triangle may have arbitrarily large vertex values without having a large L^p norm. See Figure 1.23 for the idea in one dimension. Fortunately we really only need boundedness of the interpolant with respect

FIGURE 1.23. A function with a small L^p norm (solid line), but large interpolant (dashed line).



to the W^2_p norm. That is we need the existence of a constant C such that $\|I_T g\| \le C\|g\|_{W^2_p}$ for all $v \in C(T)$. (Again, the norm on the left-hand side doesn't matter in view of the equivalence of norms on the space of polynomials.) Since the interpolant is given in terms of the values of the function at the vertices ($I_T g(x) = \sum g(v_i) l_i(x)$ where the l_i are the hat functions), the bound on the interpolant reduces to showing that $\|g\|_{L^\infty} \le C\|g\|_{W^2_p}$. This is a special case of the famous *Sobolev embedding theorem.*

THEOREM 1.35 (Sobolev embedding). *Let Ω be an n-dimensional domain, $p \ge 1$, and m an integer greater than n/p. Then there exists a constant C such that*

$$\|g\|_{L^\infty} \le C\|g\|_{W^m_p} \qquad \text{for all } g \in C^m(\Omega).$$

Thus from the Sobolev embedding theorem we know that, in two dimensions, if $p > 1$, then $\|I_T g\|_{L^\infty(T)} \le C_T \|g\|_{W^2_p}$, where the constant C_T may depend on T, but not on $g \in C^2(T)$. We may then argue just as in the proof of Theorem 1.34

$$\|f - I_T f\|_{L^p} = \|(f-q) - I_T(f-q)\|_{L^p} \le C_T \|f-q\|_{W^2_p} \le C'_T |f|_{W^2_p},$$

where we have used polynomial preservation, boundedness, and the Bramble–Hilbert lemmas, respectively.

To obtain a useful L^p error estimate, we apply this result only for the reference triangle $\hat{T}$ and scale that estimate to an arbitrary triangle. Let $|T|$ denote the area of T. Changing variables from $\hat{x}$ to $x = Fx$, we get

$$\int_T |(f - I_T f)(x)|^p \, dx = \frac{|T|}{2} \int_{\hat{T}} |(\hat{f} - I_{\hat{T}} \hat{f})(\hat{x})|^p \, d\hat{x} \le C \frac{|T|}{2} \sum_{|\beta|=2} \int_{\hat{T}} |D^\beta \hat{f}(\hat{x})|^p \, d\hat{x},$$

where D^β ranges over the second partial derivatives. Using the chain rule we have

$$|D^\beta \hat{f}(\hat{x})| \le Ch_T^2 \sum_{|\gamma|=2} |D^\gamma f(x)|$$

for each β with $|\beta| = 2$. Thus

$$\int_T |(f - I_T f)(x)|^p \, dx \le Ch_T^{2p} \frac{|T|}{2} \sum_{|\gamma|=2} \int_{\hat{T}} |D^\gamma f(x)|^p \, d\hat{x} = Ch_T^{2p} \sum_{|\gamma|=2} \int_T |D^\gamma f(x)|^p \, dx.$$

In other words

$$\|f - I_T f\|_{L^p} \le Ch_T^2 |f|_{W_p^2(T)}.$$

Note that the factor of $|T|/2$, which came from the change of variables $x \mapsto \hat{x}$ disappeared when we changed back to the original variable.

Finally we consider estimates of the error in the first derivatives of the interpolants; that is, we bound $|f - I_T f|_{W_p^1}$. As we remarked above, in the boundedness result $\|I_T g\| \le C_T \|g\|_{W_p^2}$ (which holds for any $p > 1$), it does not matter what norm we take on the left-hand side. Consequently it is straightforward use the Bramble–Hilbert lemma and show that

$$|\hat{f} - I_{\hat{T}} \hat{f}|_{W_p^1(\hat{T})} \le C |\hat{f}|_{W_p^2(\hat{T})}. \tag{1.10}$$

The next step is to use affine scaling to relate $|f - I_T f|_{W_p^1(T)}$ to $|\hat{f} - I_{\hat{T}} \hat{f}|_{W_p^1(\hat{T})}$ and $|f|_{W_p^2(T)}$ to $|\hat{f}|_{W_p^2(\hat{T})}$. We have already made the second relation:

$$|\hat{f}|_{W_p^2(\hat{T})} \le Ch_T^2 |T|^{-1/p} |f|_{W_p^2(T)}. \tag{1.11}$$

(The factor of h_T^2 comes from applying the chain rule to calculate the second derivatives, and the factor of $|T|^{-1/p}$ comes from the change of variable in the integral.) To compute $|f - I_T f|_{W_p^1(T)}$ we will again use the chain rule and a change of variables, but this time, since $f = \hat{f} \circ F^{-1}$ with $F^{-1}x = -B^{-1}v_0 + B^{-1}x$, a factor of $\|B^{-1}\|$ will appear in our estimate (rather than factors of $\|B\|$. Just as we were able to bound $\|B\|$ by a geometric quantity, namely by $h_T = \operatorname{diam}(T)$, we can bound $\|F^{-1}\|$ by a geometric quantity. Let ρ_T denote the diameter of the largest disk contained in T (the circumscribed disk). Then any vector of length ρ_T is the vector connecting two points in the circumscribed disk, and this vector is mapped by B^{-1} to the difference of two points in $\hat{T}$, i.e., to a vector of length at most $\sqrt{2}$. Thus B^{-1} maps any vector of length ρ_T to a vector of length at most $\sqrt{2}$, which is the equivalent to saying that $\|B^{-1}\| \le \sqrt{2}/\rho_T$. When we use this result together with the chain rule and change of variable, we find that

$$|g|_{W_p^1(T)} \le \frac{C}{\rho_T} |T|^{1/p} |\hat{g}|_{W_p^1(T)}.$$

In particular

$$|f - I_T f|_{W_p^1(T)} \le \frac{C}{\rho_T} |T|^{1/p} |\hat{f} - I_{\hat{T}} \hat{f}|_{W_p^1(\hat{T})}. \tag{1.12}$$

Combining (1.12), (1.10), and (1.11), we obtain the estimate

$$|f - I_T f|_{W_p^1(T)} \le C \frac{h_T^2}{\rho_T} |\hat{f}|_{W_p^2(\hat{T})}.$$

This argument is almost complete. I need to now define shape regularity and conclude a first order estimate. Then state a theorem with all the estimates for piecewise linear interpolation. Then remark on extension to higher order elements. Perhaps also remark on case of zero Dirichlet boundary conditions. State final theorem in form to be used in section on FEMs.

7. The Fast Fourier Transform

The discrete Fourier transform (DFT) is the linear operator $\mathcal{F}_N : \mathbb{C}^N \to \mathbb{C}^N$ given by

$$(\mathcal{F}_n y)_k = \sum_{j=0}^{N-1} e^{-2\pi ijk/N} y_j, \quad k = 0, 1, 2, \ldots, N-1,$$

(note that we index the vectors starting from 0). We can use the above formula to define $(\mathcal{F}_N y)_k$ for any k, not just $0 \le k < N$, but it is an N-periodic sequence. If you recall that the kth Fourier coefficient of a function f on $[0,1]$ is defined by $\hat{f}(k) = \int_0^1 f(x) e^{-2\pi ikx}\, dx$, we see that $N^{-1}(\mathcal{F}_N y)_k$ is just the approximation to $\hat{f}(k)$ obtained by using the trapezoidal rule with N equal subintervals to compute the integral, expressed in terms of the values $y_j = f(j/N)$.

Note that application of the DFT is simply multiplication of an N-vector by a particular $N \times N$ matrix. This would suggest that N^2 multiplications and additions are needed to compute it. The fast Fourier transform (FFT) is a clever algorithm to compute the DFT much more quickly (for large N). Because large DFTs arise in many contexts, the FFT proved to be one of the most important algorithms of the twentieth century. In particular, it has played a tremendous role in signal processing.

The FFT can also be used as a fast means of computing the coefficients of the a trigonometric interpolating polynomial with equally spaced interpolation points (or for the closely related problem of computing the algebraic polynomial interpolating polynomial at the Chebyshev points). In this last section of the chapter we introduce the FFT in that context.

We defined the space $\mathcal{T}_n$ of real-valued trigonometric polynomials of degree at most n as the span of the $2n+1$ functions $1, \cos x, \sin x, \ldots, \cos nx, \sin nx$. Recall that

$$\mathcal{T}_n = \{ \sum_{k=-n}^{n} c_k e^{ikx} \,|\, c_k \in \mathbb{C}, c_{-k} = \bar{c}_k \}.$$

This space has dimension $2n+1$. However, for reasons that will soon be clear, it is more convenient to work with a space of even dimension at this point. Hence we define $\mathcal{T}_n'$ as the span of the $2n$ functions $1, \cos x, \sin x, \ldots, \cos nx$, or, equivalently,

$$\mathcal{T}_n' = \{ \sum_{k=-n}^{n} c_k e^{ikx} \in \mathcal{T}_n \,|\, c_{-n} = c_n \} = \operatorname{span}[1, \cos x, \sin x, \ldots, \cos nx].$$

This space has dimension $N = 2n$.

We now consider the problem of finding a function in $\mathcal{T}_n'$ interpolating given data y_j at N equally spaced points x_j in $[0, 2\pi]$. Since the trigonometric polynomials are 2π-periodic, we include only one of the endpoints, 0, but not 2π, among the interpolation points. Thus

the interpolation points are $x_j = 2\pi j/N$, $j = 0, \dots, N-1$. The interpolation problem is thus to find coefficients $c_k \in \mathbb{C}$, $k = -n, \dots, n$, with $c_n = c_{-n}$, such that

$$\sum_{k=-n}^{n} c_k e^{2\pi i j k/N} = y_j, \quad j = 0, \dots, N-1.$$

(If the y_j are real, we may conjugate this equation to find that if (c_k) is a solution, then so is $(\bar{c}_{-k})$. Since—as we shall see—the solution is unique, we see that the condition $c_k = \bar{c}_{-k}$ is automatic.)

It is possible to write this system of equations in a somewhat more convenient fashion. If $-n \le k \le -1$, we use the identity $e^{2\pi i j k/N} = e^{2\pi i j(k+N)/N}$. Then $n \le k + N \le N - 1$. In this way we find

$$\sum_{k=-n}^{n} c_k e^{2\pi i j k/N} = \sum_{k=0}^{N-1} d_k e^{2\pi i j k/N} \tag{1.13}$$

where

$$d_k = \begin{cases} c_k, & 0 \le k < n, \\ c_n + c_{-n}, & k = n, \\ c_{k-N}, & n < k < N. \end{cases}$$

Note that, since $c_n = c_{-n}$, we can recover the c_k from the d_k.

Thus our problem is to find the coefficients d_k such that

$$\sum_{k=0}^{N-1} d_k e^{2\pi i j k/N} = y_j. \tag{1.14}$$

Once this is done, we can define

$$c_k = \begin{cases} d_k, & 0 \le k < n, \\ d_n/2, & k = -n \text{ or } n, \\ d_{k+N}, & -n < k < 0, \end{cases} \tag{1.15}$$

to obtain the coefficients of the interpolating trigonometric polynomial.

Notice that (1.14) is just an $n \times n$ linear system. The matrix is

$$M_N = (e^{2\pi i j k/N})_{0 \le j,k < N} = (\omega^{jk})_{0 \le j,k < N} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{pmatrix}$$

Thus M_N is a Vandermonde matrix based on the powers of $\omega = \omega_N = e^{2\pi i/N}$, an Nth root of unity. Notice the close connection with the DFT: $\bar{M}_N$ is exactly the matrix of the DFT.

The very special form of the matrix M_N allows us to write down its inverse explicitly. First note that $\bar{\omega} = \omega^{-1}$. Also, by summing the geometric series, we find

$$\sum_{k=0}^{N-1} \omega^{jk} = \begin{cases} 0, & j \not\equiv 0 \pmod N, \\ N, & j \equiv 0 \pmod N. \end{cases}$$

Now consider the product $\bar{M}_N M_N$. For $0 \le j, m < N$, the (j, m) element is

$$\sum_{k=0}^{N-1} \omega^{-jk} \omega^{km} = \sum_{k=0}^{N-1} \omega^{(m-j)k} = N\delta_{jm}.$$

Thus $M_N^{-1} = \frac{1}{N}\bar{M}_N$.

To summarize: given a vector of values y_j, we take its discrete Fourier transform to get a vector d_j, and then rearrange the d_j according to (1.15) to get the coefficients c_j of the trigonmetric polynomial of degree at most n interpolating the y_j at x_j.

In 1965 Cooley and Tukey published a clever way to multiply by $\mathcal{F}_N$ that exploits its special structure. The resulting algorithm, which exists in many variants, is the FFT. With it, it is quite practical to compute DFTs of size in the tens of thousands, or even millions.

Let $x \in \mathbb{R}^N$ be given with $N = 2n$. Let $\bar{x}, \tilde{x} \in \mathbb{R}^n$ be the odd and even index elements of x: $\bar{x}_j = x_{2j}$, $\tilde{x}_j = x_{2j+1}$. Then

$$\begin{aligned}
(\mathcal{F}_N x)_k &= \sum_{j=0}^{N-1} \omega_N^{-jk} x_j = \sum_{j=0}^{n-1} \omega_N^{-2jk} x_{2j} + \sum_{j=0}^{n-1} \omega_N^{-(2j+1)k} x_{2j+1} \\
&= \sum_{j=0}^{n-1} \omega_n^{-jk} \bar{x}_j + \omega_N^{-k} \sum_{j=0}^{n-1} \omega_n^{-jk} \tilde{x}_j \\
&= (\mathcal{F}_n \bar{x})_k + \omega_N^{-k} (\mathcal{F}_n \tilde{x})_k.
\end{aligned}$$

This equation shows how to reduce the evaluation of the size N discrete Fourier transform $\mathcal{F}_N x$ to the two size n discrete Fourier transforms $\mathcal{F}_n \bar{x}$ and $\mathcal{F}_n \tilde{x}$. In matrix terms it shows

$$\begin{pmatrix} \\ \mathcal{F}_N x \\ \\ \end{pmatrix} = \begin{pmatrix} \mathcal{F}_n \bar{x} \\ \cdots\cdots \\ \mathcal{F}_n \bar{x} \end{pmatrix} + \begin{pmatrix} 1 & & & \\ & \omega_N^{-1} & & \\ & & \ddots & \\ & & & \omega_N^{1-N} \end{pmatrix} \begin{pmatrix} \mathcal{F}_n \tilde{x} \\ \cdots\cdots \\ \mathcal{F}_n \tilde{x} \end{pmatrix}.$$

Now if n is itself an even number, then we can reduce each of the discrete Fourier transforms of size n to two transforms of size $n/2$, etc. If N is a power of 2, we can continue in this way until we have reduced the work to N transforms of size 1, which of course are trivial ($\mathcal{F}_1 = 1$). Algorithm 1.1 shows, in metacode, an algorithm for computing the FFT in this case. Notice that the use of recursion makes the algorithm statement quite brief.

$\boldsymbol{y} = \mathrm{FFT}(\boldsymbol{x})$
input: $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{C}^n$ where n is a power of 2
output: $\boldsymbol{y} = \mathcal{F}_n \boldsymbol{x} \in \mathbb{C}^n$

if $n = 1$ **then**
  $y \leftarrow x$
else
  $\bar{\boldsymbol{x}} \leftarrow (x_0, x_2, \ldots, x_{n-2})$
  $\tilde{\boldsymbol{x}} \leftarrow (x_1, x_3, \ldots, x_{n-1})$
  $\bar{\boldsymbol{y}} \leftarrow \mathrm{FFT}(\bar{\boldsymbol{x}})$
  $\tilde{\boldsymbol{y}} \leftarrow \mathrm{FFT}(\tilde{\boldsymbol{x}})$
  $\omega \leftarrow e^{-2\pi i/n}$
  $y_k \leftarrow \bar{y}_k + \omega^{-k}\tilde{y}_k, \quad k = 0, 1, \ldots, n/2 - 1$
  $y_k \leftarrow \bar{y}_{k-n/2} + \omega^{-k}\tilde{y}_{k-n/2}, \quad k = n/2, n/2 + 1, \ldots, n - 1$
end if

Algorithm 1.1: Simple FFT.

In fact the code is even briefer when implemented in Matlab,[2] as shown in the listing below.

```
function y = simplefft(x)
N = length(x);
if N == 1
  y = x;
else
   omega = exp(-2*pi*i/N);
   ybar = simplefft(x(1:2:N-1));
   ytilde = simplefft(x(2:2:N));
   y = [ ybar ; ybar ] + omega.^-(0:N-1)' .* [ ytilde; ytilde ];
end
```

Algorithm 1.2: Simple FFT implemented in Matlab.

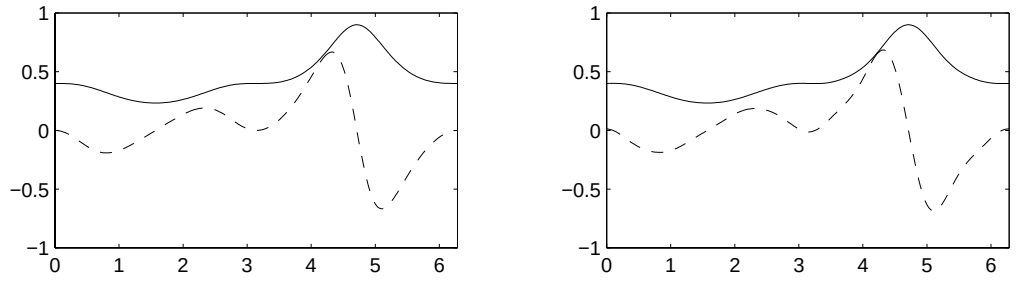
Let us count the amount of work for this algorithm. For simplicity we count only the multiplications and ignore the additions. We also ignore the computation of the powers of ω_N needed in the algorithm. In an efficient implementation, all N powers would be computed once (via N multiplications). If we let m_N be the number of multiplications needed to compute the action of $\mathcal{F}_N$, we have $m_N = 2m_n + N$ where $n = N/2$. Also $m_1 = 0$. This gives $m_{2^k} = k2^k$, i.e., $m_N = N \log_2 N$.

We close with some applications. Figure 1.24 shows a plot of a smooth periodic function and its derivative on the left. The function was sampled at 16 equally spaced points and its

[2]Matlab offers an FFT function, `fft`, which is more sophisticated and also works when N is not a power of 2.

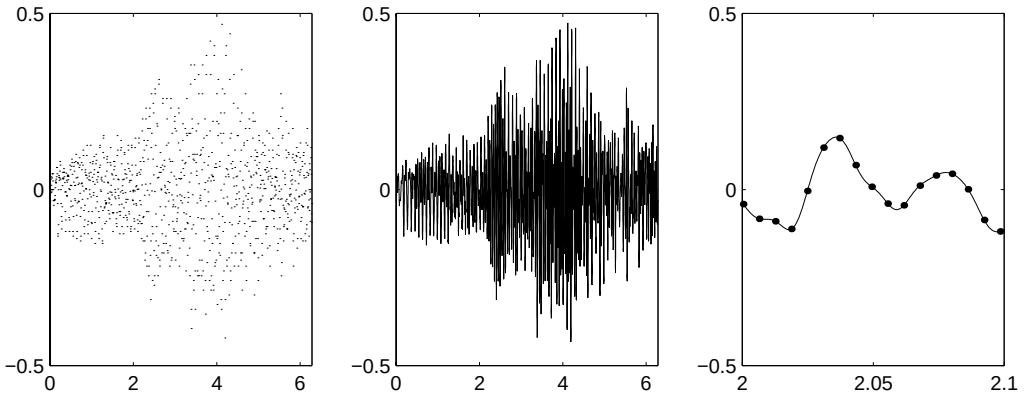
trigonometric polynomial interpolant $\sum_{k=-8}^{8} c_k e^{ikx}$ was computed and plotted on the right. The derivative, computed as $\sum_{k=-8}^{8} ikc_k e^{ikx}$, is also shown. This approach to reconstructing the derivative of a function from its values at a discrete set of points is the basis of *spectral methods* for solving differential equations.

FIGURE 1.24. A trigonometric interpolant and its derivative.



Of course when $N = 16$ the efficiency afforded by the FFT is not important. Figure 1.25 shows an application involving 1,024 interpolation points, for which the efficiency is significant. On the left are 1024 points (taken from a sound sample of people laughing). In the middle is the trigonometric polynomial interpolant of the sample values (computed via the FFT), and on the right is a blow-up of the region $2 \leq x \leq 2.1$ showing that the polynomial does indeed interpolate the given values.

FIGURE 1.25. Trigonometric interpolation at 1,024 points.



EXERCISES

(1) Show that for any $1 \leq p < q \leq \infty$ the L^p and L^q norms on $C(I)$ are not equivalent.

(2) Let $f(x) = e^x$. a) For $p = 1$, 2, and ∞ find the best L^p approximation to f in $\mathcal{P}_0(I)$. b) Same thing but in $\mathcal{P}_1(I)$.

(3) Prove or disprove: if $f \in C([-1,1])$ is odd then a best approximation to f by odd polynomials of degree at most n is a best approximation to f among all polynomials of degree n.

(4) Prove or disprove: if $f \in C([-1,1])$ has mean value zero, then a best approximation to f by polynomials of degree at most n with mean value zero is a best approximation to f among all polynomials of degree n.

(5) State and prove the Jackson theorem in $C^k([a,b])$ paying attention to the dependence of the constant on the interval $[a,b]$.

(6) If $f \in C(\mathbb{R})$ and $\delta > 0$ define $R_\delta f \in C(\mathbb{R})$ by

$$R_\delta f(x) = \frac{1}{\delta} \int_{x-\delta/2}^{x+\delta/2} f(t)\, dt.$$

Note that $R_\delta f \in C^1(\mathbb{R})$. Prove that $\|f - R_\delta f\| \le \omega(\delta)$, where ω denotes the modulus of continuity of f (i.e., $\omega(\delta)$ is the supremum of $|f(x)-f(y)|$ over x, y for which $|x-y| \le \delta$).

(7) Let $f \in C_{2\pi}$ and let ω denote its modulus of continuity. Using the Jackson theorem in $C^1_{2\pi}$ and the regularization operator of the previous problem, prove that

$$\inf_{p \in \mathcal{T}_n} \|f - p\|_\infty \le c\, \omega\left(\frac{1}{n+1}\right).$$

Give an explicit expression for c.

(8) Let $f \in C([-1,1])$ and let ω denote its modulus of continuity. Prove that

$$\inf_{p \in \mathcal{P}_n} \|f - p\|_\infty \le c\, \omega\left(\frac{1}{n+1}\right).$$

Give an explicit expression for c.

(9) Suppose that $f \in C([-1,1])$ satisfies the Holder condition $|f(x) - f(y)| \le M|x-y|^\alpha$ where $M, \alpha > 0$. What can you say about the rate of convergence of the best uniform approximation to f by polynomials of increasing degree?

In the exercises 10–17, which treat divided differences and Newton's formula for the interpolating polynomial, we denote by f a real-valued function on an interval, by $x_0, \ldots, x_n$ $n+1$ distinct points in J and by $p_k \in \mathcal{P}_k$ the Lagrange interpolating polynomial for f at the first $k+1$ points $x_0, \ldots, x_k$.

(10) Prove that $p_n(x) - p_{n-1}(x) = c(x-x_0)\cdots(x-x_{n-1})$ for some constant c. We use the notation $f[x_0, \ldots, x_n]$ to denote this constant and call it the *nth divided difference of f at the x_i*. Use Lagrange's formula for the interpolating polynomial to derive an expression for $f[x_0, \ldots, x_n]$ in terms of x_i and $f(x_i)$.

(11) Prove that $f[x_0, \ldots, x_n]$ is a symmetric function of its $n+1$ arguments.

(12) Prove the recursion relation

$$f[x_0, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0},$$

where, by convention, $f[x] := f(x)$. (This explains the terminology "divided difference".)

(13) Give explicit formulas for $f[a]$, $f[a,b]$, $f[a,b,c]$, and $f[x, x+h, x+2h, \ldots, x+nh]$.

(14) Prove Newton's formula for the interpolating polynomial

$$p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \cdots + f[x_0, \ldots, x_n](x - x_0) \cdots (x - x_{n-1}),$$

and the error formula

$$f(x) - p_n(x) = f[x_0, \ldots, x_n, x](x - x_0) \cdots (x - x_n).$$

(15) Prove that if $f \in C^n(J)$, then there exists a point ξ in the interior of J such that

$$f[x_0, \ldots, x_n] = \frac{1}{n!} f^{(n)}(\xi).$$

(16) Assuming that $f \in C^n(J)$, use the recursion defining the divided differences to establish the Hermite-Gennochi formula

$$f[x_0, \ldots, x_n] = \int_{S_n} f^{(n)}(t_0 x_0 + t_1 x_1 + \cdots + t_n x_n)\, d\boldsymbol{t},$$

where

$$S_N = \left\{ \boldsymbol{t} = (t_1, \ldots, t_n) \in \mathbb{R}^n \,|\, t_i \geq 0, \sum_1^n t_i \leq 1 \right\},$$

and $t_0 = 1 - \sum_1^n t_i$.

(17) The Hermite-Gennochi formula shows that as a function of $n+1$ variables the nth divided difference extends to a function on all of J^{n+1} (the arguments need not be distinct). Find simple closed form expressions for $f[a, a]$, $f[a, a, b]$, and $f[a, a, b, b]$.

For the next 3 problems define $\Pi_n : C([-1, 1]) \to \mathcal{P}_n([-1, 1])$ *by* $\|f - \Pi_n f\|_w = \inf_{p \in \mathcal{P}_n} \|f - p\|_w$, *where* $\|f\|_w^2 = \int_{-1}^1 |f(x)|^2 (1 - x^2)^{-1/2}\, dx$.

(18) Give a formula for $\Pi_n f$.

(19) Prove that

$$\|f - \Pi_n f\|_\infty \leq c\sqrt{n+1} \inf_{p \in \mathcal{P}_n} \|f - p\|_\infty$$

for some constant c. N.B.: This estimate is not sharp. It can be show to hold with $\sqrt{n+1}$ replaced by $1 + \log(2n+1)$, but that is harder to prove.

(20) Prove that if $f \in \mathcal{P}_{n+1}$, then

$$\|f - \Pi_n f\|_\infty = \inf_{p \in \mathcal{P}_n} \|f - p\|_\infty,$$

that is, Π_n coincides with best minimax approximation when applied to polynomials of degree $n + 1$.

(21) In proving the convergence of the conjugate gradient method for solving linear systems, a key step is showing that

$$\min_{\substack{p \in \mathcal{P}_n \\ p(0)=1}} \max_{x \in [a,b]} |p(x)| = \frac{2}{\left(\frac{1+\sqrt{a/b}}{1-\sqrt{a/b}}\right)^n + \left(\frac{1-\sqrt{a/b}}{1+\sqrt{a/b}}\right)^n} \tag{1.16}$$

for $0 < a < b$. In fact, the polynomial for which the minimum is achieved is a scaled Chebyshev polynomial:

$$p(x) = T_n(\hat{x}) / T_n\left(\frac{b+a}{b-a}\right), \text{ where } x = \frac{b+a}{2} - \frac{b-a}{2}\hat{x},$$

and the right-hand side of (1.16) is just $1/T_n\big((b+a)/(b-a)\big)$. Prove all this.

(22) Let $f \in C^1([a,b])$. Prove that the cubic spline interpolant with derivative end conditions minimizes the quantity $\|g''\|_{L^2([a,b])}$ among all C^2 functions on $[a,b]$ which interpolate f at the x_i and f' at a and b. Since the second derivative is a measure of curvature, this says that in a certain sense the cubic spline interpolant is the straightest, or smoothest, function satisfying the interpolation conditions.

(23) Let $x_0 < x_1 < \cdots < x_p$ and suppose that s is a cubic spline defined on all of $\mathbb{R}$ with breakpoints at the x_i (only), and for such that $s(x) \equiv 0$ if $x \le x_0$ or $x \ge x_p$. Prove that if $p \le 3$, then $s(x) \equiv 0$. In other words, there does not exist a nonzero cubic spline supported in just 3 intervals.

(24) With the same notation as the previous problem show that that such a nonzero cubic spline $s(x)$ does exist if $p = 4$. Show that $s(x)$ is determined uniquely up to a constant multiple. With appropriate normalization $s(x)$ is called the cubic B-spline for the knots $x_0, \dots x_4$.

(25) Give the explicit formula for the cubic B-spline $B(x)$ with knots $x_i = i$, $i = 0, \dots, 4$, normalized so that $\sum B(i) = 1$. Draw a plot of this function.

(26) Let $\Pi_N = \{\, (a_i)_{i=-\infty}^{\infty} \,|\, a_i \in \mathbb{C}, a_{i+N} = a_i \,\}$ denote the space of bi-infinite N-periodic complex sequences. If $\boldsymbol{a}$, $\boldsymbol{b} \in \Pi_N$ we define the *convolution* $\boldsymbol{c} = \boldsymbol{a} * \boldsymbol{b}$ by

$$c_k = \sum_{j=0}^{N-1} a_j b_{k-j}.$$

Prove that the discrete Fourier transform converts convolution into multiplication: $(\mathcal{F}_N \boldsymbol{c})_k = (\mathcal{F}_N \boldsymbol{a})_k (\mathcal{F}_N \boldsymbol{b})_k$.

(27) Let p and q be polynomials of degree less than n, where n is a power of 2. Explain how the coefficients of the product pq can be computed from the coefficients of p and q in $O(n \log_2 n)$ operations.

Bibliography

1. J. H. Bramble and S. R. Hilbert, *Estimation of linear functionals on Sobolev spaces with application to Fourier transforms and spline interpolation*, SIAM J. Numer. Anal. **7** (1970), 112–124.
2. Philippe G. Ciarlet, *The finite element method for elliptic problems*, North-Holland, Amsterdam, 1978.
3. J. Deny and J. L. Lions, *Les espaces du type de Beppo Levi*, Ann. Inst. Fourier, Grenoble **5** (1953–54), 305–370 (1955).
4. Todd Dupont and Ridgway Scott, *Polynomial approximation of functions in Sobolev spaces*, Math. Comp. **34** (1980), no. 150, 441–463.
5. Theodore J. Rivlin, *An introduction to the approximation of functions*, Dover Publications Inc., New York, 1981.
6. ———, *Chebyshev polynomials*, John Wiley & Sons, New York, 1990.

CHAPTER 2

Numerical Quadrature

1. Basic quadrature

The problem is to compute $\int_a^b f(x)\,dx$ given a, b, and an integrable function f on $[a,b]$. We shall mainly use quadrature rules of the form

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n} w_i f(x_i),$$

with *points* $x_i \in [a,b]$ and *weights* $w_i \in \mathbb{R}$. More abstractly put, we want to approximate the functional $f \mapsto \int_a^b f$ by a linear combination of point-evaluation functionals.

Typical examples are:

left endpoint rule: $\int_a^b f \approx (b-a)f(a)$

midpoint rule: $\int_a^b f \approx (b-a)f\big((a+b)/2\big)$

trapezoidal rule: $\int_a^b f \approx \frac{b-a}{2}[f(a)+f(b)]$

Simpson's rule: $\int_a^b f \approx \frac{b-a}{6}[f(a)+4f\big((a+b)/2\big)+f(b)]$

2 point Gauss rule:
$\int_a^b f \approx \frac{b-a}{2}[f\big((a+b)/2-(b-a)/(2\sqrt{3})\big)+f\big((a+b)/2+(b-a)/(2\sqrt{3})\big)]$

composite midpoint rule (equal subintervals):
$\int_a^b f = \sum_{i=1}^{n} hf(a-h/2+ih)$, $h=(b-a)/n$

composite midpoint rule ($a=x_0<x_1<\cdots<x_n=b$):
$\int_a^b f = \sum_{i=1}^{n} h_i f\big((x_{i-1}+x_i)/2\big)$, $h_i = x_i - x_{i-1}$

For any choice of distinct points $x_i \in [a,b]$, $i=0,1,\ldots,n$ there is a natural way to assign weights w_i: let $p \in \mathcal{P}_n$ interpolate f at the points x_i and approximate $\int_a^b f$ by $\int_a^b p$. Using Lagrange's formula $p(x) = \sum_i f(x_i)l_i(x)$, with $l_i(x) = \Pi_{j\neq i}(x-x_j)/(x_i-x_j)$, we can write the resulting quadrature rule $\int_a^b f \approx \sum_i w_i f(x_i)$ where $w_i = \int_a^b l_i(x)\,dx$. Such rules are called *interpolatory* quadrature rules. All the rule listed above except the composite midpoint rule are of this form (the composite rules are based on piecewise polynomial interpolation). By construction an interpolatory quadrature with $n+1$ points has *degree of precision* at least n, that is, the rule is exact on all polynomials of degree at most n. For some choices of points a higher degree of precision is achieved. For example, Simpson's rule has degree of precision 3 as does the 2 point Gauss rule.

A well-known class of interpolatory quadrature rules are based on using $n+1$ equally spaced points including the end points. These are called the *closed Newton-Cotes* rules (the word "closed" refers to the fact that the endpoints are included). For $n=1$ this is the trapezoidal rule, for $n=2$ it is Simpson's rule. For $n=3$ we get Simpson's 3/8 rule with weights 1/8, 3/8, 3/8, 1/8 (on an interval of unit length). For $n=4$ we get Boole's rule with

weights 7/90, 32/90, 12/90, 32/90, 7/90. By construction the Newton-Cotes rule with $n+1$ points has degree of precision n. But it is also provides the exact value when applied to any odd power of $x-(a+b)/2$ (namely 0) by symmetry. Thus when n is even the Newton-Cotes rule with $n+1$ points has degree of precision $n+1$.

In view of the large Lebesgue constant of high degree interpolation with equally spaced points, it is not suprising that the Newton-Cotes rules are not very good for n large. Starting with $n=8$ some of the weights are negative, and for larger n the coefficients become very large and oscillatory (they sum to one). As a result the formulas are very sensitive to errors in f and difficult to evaluate in finite precision (cancellation).

For interpolatory quadrature rule we can deduce the error from the error formula for Lagrange interpolation. As an example consider the trapezoidal rule on the unit interval $[0,1]$, which is based on Lagrange interpolation at the points 0 and 1. Denoting by p the interpolant, we have $f(x)-p(x)=f[x,0,1]x(x-1)$, so the error in the trapezoidal rule is

$$\text{err}=\int_0^1 f-\int_0^1 p=\int_0^1 f[x,0,1]x(x-1)\,dx. \tag{2.1}$$

We now recall the *integral mean value theorem*: if $u\in C([a,b])$ and w is an integrable (but not necessarily continuous) function on $[a,b]$ which doesn't change sign, then

$$\int_a^b u(x)w(x)\,dx=u(\eta)\int_a^b w(x)\,dx$$

for some $\eta\in(a,b)$. Applying this to the integral on the right-hand side of (2.1) with $u(x)=f[x,0,1]$ and $w(x)=x(x-1)$ we find that

$$\int_0^1 f-\int_0^1 p=-\frac{1}{6}f[\eta,0,1]$$

for some $\eta\in(0,1)$, and hence, if $f\in C^2([0,1])$,

$$\text{err}=f[\eta,0,1]\int_0^1 x(x-1)\,dx=-\frac{1}{12}f''(\xi)$$

for some $\xi\in(0,1)$.

Next we scale this result to an arbitrary interval $[\alpha,\beta]$. If $f\in C^2([\alpha,\beta])$, we define $\hat f(\hat x)=f(x)$ where $x=\alpha+(\beta-\alpha)\hat x$. Then

$$\begin{aligned}\int_\alpha^\beta f-\frac{\beta-\alpha}{2}[f(\alpha)+f(\beta)]&=(\beta-\alpha)\left\{\int_0^1\hat f-\frac{1}{2}[\hat f(0)+\hat f(1)]\right\}\\&=-\frac{1}{12}(\beta-\alpha)\hat f''(\hat\xi)=-\frac{1}{12}(\beta-\alpha)^3f''(\xi),\end{aligned}$$

for some $\xi\in(\alpha,\beta)$.

Now consider the composite trapezoidal rule using n equal subintervals of size $h=(b-a)/n$. Applying the above result on each subinterval and summing we get

$$\int_a^b f-h[\frac{1}{2}f(a)+\sum_{i=1}^{n-1}f(a+ih)+\frac{1}{2}f(b)]=-\frac{1}{12}h^3\sum_{i=1}^{n}f''(\xi_i)=-\frac{1}{12}(b-a)h^2f''(\xi).$$

Here $\xi_i \in (a+(i-1)h, a+ih)$ and we have used the fact that $\sum_i f''(\xi_i) = nf''(\xi)$ for some $\xi \in (a,b)$ (again a consequence of the intermediate value theorem). This is an exact formula, not just a bound, for the error. It implies the bound $|\text{err}| \le (b-a)h^2/12\|f''\|_{L^\infty}$.

Now consider the composite trapezoidal rule with unequal subintervals determined by a partition $a = x_0 < \dots x_n = b$. With $h_i = x_i - x_{i-1}$, $h = \max_i h_i$, we have

$$\left| \int_a^b f - \sum_{i=0}^{n} \frac{h_i}{2}[f(x_{i-1}) + f(x_i)] \right| = \left| -\frac{1}{12} \sum_{i=1}^{n} h_i^3 f''(\xi_i) \right|$$
$$\le \frac{1}{12} h^2 \sum h_i \|f''\|_{L^\infty} = \frac{1}{12}(b-a)h^2\|f''\|_{L^\infty},$$

and we see that this bound is sharp since equality must hold if f'' is constant.

Note that we get the same bound $|\text{err}| \le (b-a)h^2/12\|f''\|_{L^\infty}$ for an unequal spacing of points as for uniform spacing. Of course this bound doesn't show any particular advantage to choosing a nonuniform spacing. For that we would have to consider more carefully the sum $\sum_i h_i^3 f''(\xi_i)$. If we adjust the spacing so that h_i is smaller where $|f''|$ is larger we can decrease the error.

If we try to apply the same arguments to the midpoint rule we come across one difference. For the midpoint rule on $[0,1]$ the interpolant is simply the constant value $f(1/2)$, and so we get, instead of (2.1),

$$\text{err} = \int_0^1 f - f(1/2) = \int_0^1 f[x, 1/2](x - 1/2)\, dx.$$

However now the kernel $x - 1/2$ changes sign on $[0,1]$ and so we cannot continue using the integral mean value theorem as before.

A simple approach for the midpoint rule is to use Taylor's theorem. Assuming that $f \in C^2([0,1])$ we have

$$f(x) = f(1/2) + f'(1/2)(x - 1/2) + \frac{1}{2} f''(\xi_x)(x - 1/2)^2, \quad x \in (0,1)$$

where $\xi_x \in (0,1)$ depends continuously on x. Integrating over $x \in (0,1)$ we get

$$\int_0^1 f - f(1/2) = \frac{1}{2}\int_0^1 f''(\xi_x)(x - 1/2)^2\, dx = \frac{1}{24} f''(\xi), \tag{2.2}$$

for some $\xi \in (0,1)$. Note that term involving $x - 1/2$ integrated to zero and we were able to use the integral mean value theorem in the last term because $(x - 1/2)^2$ is everywhere non-negative. Once we have the expression (2.2) for the error in the simple midpoint rule on the unit interval, we can scale to an arbitary interval and add up over subintervals just as for the trapezoidal rule. We find that the error is bounded by $(1/24)(b-a)h^2\|f''\|_{L^\infty}$, exactly $1/2$ times the bound we got for the trapezoidal rule.

2. The Peano Kernel Theorem

In the last section we derived expressions for the error in the simple trapezoidal and midpoint rules on the unit interval, and then scaled and summed to get bounds for the error in the composite rules. In this section we describe an approach that can be used to give easily manipulated expressions for the error in any quadrature rule.

Define two linear functionals on $C([a,b])$ by the integral and by the quadrature rule: $If = \int_a^b f$, $Jf = \sum_{i=0}^n w_i f(x_i)$. If the degree of the precision of the quadrature rule is d, then the error functional $Ef := If - Jf$ vanishes on $\mathcal{P}_d$. From this fact alone we can derive a very useful expression for the error. The approach is very much like the one we used to derive interpolation error estimates on triangles in Chapter 1.6.2. In deriving the Bramble–Hilbert lemma we represented an arbitrary function in C^k as the sum of a polynomial of degree at most k (the averaged Taylor polynomial), and an integral of the $k+1$st derivatives of the function times suitable kernel functions. In one dimension the story is simpler, because we can use the ordinary Taylor theorem with remainder, without the averaging.

Let $0 \le k \le d$ be an integer and suppose that $f \in C^{k+1}([a,b])$. Then, by Taylor's theorem,

$$f(x) = p(x) + \frac{1}{k!}\int_a^x f^{(k+1)}(t)(x-t)^k\,dt,$$

with $p \in \mathcal{P}_k$ the Taylor polynomial for f about $x = a$. Now let us write $x_+^n = x^n$ for $x > 0$, 0 otherwise. We can then express the remainder in Taylor's theorem as $r(x) = \frac{1}{k!}\int_a^b f^{(k+1)}(t)(x-t)_+^k\,dt$. Since $f = p + r$ and $Ep = 0$, we have $Ef = Er$. This gives the Peano Kernel representation of the error:

$$Ef = \int_a^b f^{(k+1)}(t)K(t)\,dt$$

where $K(t) = E_x[(x-t)_+^k]/k!$. The key point is that when the linear functional E vanishes on $\mathcal{P}_k$, we can express Ef as the integral of $f^{(k+1)}(t)$ times an explicit function $K(t)$. The function $K(t)$ is called the Peano kernel of the error. To establish this result we required only that E be a linear functional which vanishes on polynomials of degree k and be of a form that commutes with integration. So it can be used for bounding errors in other contexts than numerical integration. Note also that if E vanishes on $\mathcal{P}_k$, it also vanishes on $\mathcal{P}_{k-1}$, $\mathcal{P}_{k-2}$, etc. So we have k different Peano kernel representations of the error, which express it as integral involving f', f'', …, $f^{(k+1)}$. We refer to the corresponding Peano kernels as the first derivative Peano kernel, the second derivative Peano kernel, etc.

As an example of the application of the Peano kernel error representation, we reanalyze the midpoint rule from using it. Again, we consider first the unit interval and so define

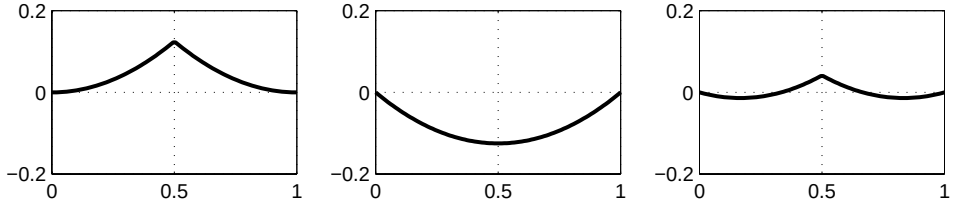
$$Ef = \int_0^1 f - f(1/2).$$

The degree of precision is 1, so we have two error representations. If $f \in C^2$, we have $Ef = \int_0^1 f''(t)K(t)\,dt$ with

$$K(t) = \int_0^1 (x-t)_+\,dx - (1/2-t)_+ = \begin{cases} t^2/2, & 0 \le t \le 1/2, \\ (1-t)^2/2, & 1/2 \le t \le 1 \end{cases}$$

This is the second derivative Peano kernel for the midpoint rule on $[0,1]$, and is plotted on the left of Figure 2.1, which also shows the second derivative Peano kernel for the trapezoidal rule and Simpson's rule. Note that the midpoint rule Peano kernel satisfies $K \ge 0$ on $[0,1]$, $\|K\|_{L^\infty} = 1/8$, $\|K\|_{L^1} = 1/24$.

FIGURE 2.1. The second derivative Peano kernels for the midpoint rule, the trapezoidal rule, and Simpson's rule. on $[0,1]$.



Next scale to an arbitrary interval $[\alpha,\beta]$ to find the Peano kernel in that case. To distinguish between the Peano kernel for the midpoint rule on $[0,1]$ and the Peano kernel for the midpoint rule on $[\alpha,\beta]$, at this point we will write $K_{[0,1]}$ and $K_{[\alpha,\beta]}$ for the latter. With $\hat{t} = (t-\alpha)/(\beta-\alpha)$ we have

$$\int_\alpha^\beta f(x)\,dx - (\beta-\alpha) f\big(\frac{\alpha+\beta}{2}\big) = (\beta-\alpha)\left[\int_0^1 \hat{f}(\hat{x})\,d\hat{x} - \hat{f}(1/2)\right] = (\beta-\alpha)\int_0^1 \hat{f}''(\hat{t}) K_{[0,1]}(\hat{t})\,d\hat{t}$$
$$= (\beta-\alpha)^3 \int_0^1 f''(t) K_{[0,1]}(\hat{t})\,d\hat{t} = (\beta-\alpha)^2 \int_\alpha^\beta f''(t) K_{[0,1]}(\hat{t})\,dt = \int_\alpha^\beta f''(t) K_{[\alpha,\beta]}(t)\,dt,$$

where

$$K_{[\alpha,\beta]}(t) = (\beta-\alpha)^2 K_{[0,1]}(\hat{t}) = \begin{cases} (t-\alpha)^2/2, & \alpha \le t \le \frac{\alpha+\beta}{2}, \\ (\beta-t)^2/2, & \frac{\alpha+\beta}{2} \le t \le \beta. \end{cases}$$

For the Peano kernel on $[\alpha,\beta]$ we have $K_{[\alpha,\beta]} \ge 0$ and $\|K_{[\alpha,\beta]}\|_{L^1} = (\beta-\alpha)^3/24$.

As an immediate consequences we obtain:

$$Ef = \frac{(\beta-\alpha)^3}{24} f''(\xi) \text{ for some } \xi \in (\alpha,\beta), \quad |Ef| \le \frac{(\beta-\alpha)^3}{24} \|f''\|_{L^\infty},$$

Note that the first result is an exact expression for the error and used the fact that $K_{[\alpha,\beta]}$ is of one sign, while the only property of $K_{[\alpha,\beta]}$ entering the error bound are its L^1 norm.

Now consider the composite midpoint rule arising from a partition $a = x_0 < \cdots < x_n = b$. Once again we have a Peano kernel error representation,

$$\int_a^b f - \sum_{i=1}^n h_i f\big(\frac{x_{i-1}+x_i}{2}\big) = \int_a^b f''(t) K(t)\,dt, \tag{2.3}$$

where now

$$K(t) = \begin{cases} (t-x_{i-1})^2/2, & x_{i-1} \le t \le (x_{i-1}+x_i)/2, \\ (x_i - t)^2/2, & (x_{i-1}+x_i)/2 \le t \le x_i, \end{cases} \tag{2.4}$$

$i = 1,\dots,n$. Note that again $K \ge 0$ and now $\|K\|_{L^\infty} = \sup_i h_i^2/8 = h^2/8$. We thus obtain

$$Ef = \frac{1}{24}\big(\sum_i h_i^3\big) f''(\xi) \text{ for some } \xi \in (a,b), \quad |Ef| \le \frac{(b-a)h^2}{24} \|f''\|_{L^\infty}. \tag{2.5}$$

Thus if $f \in C^2([a,b])$, the composite midpoint rule converges as the second power of the maximal subinterval size.

We can also use the Peano kernel theorem to represent the error as the integral of the first derivative of the integrand times the kernel $K_0(t) = E[(\cdot - t)+^0]$. In fact, rather than compute this, let us derive the final (composite rule) representation by integrating by parts in (2.3). Writing K_1 for the kernel given by (2.4) we have

$$\int_a^b f - \sum_{i=1}^n h_i f\big(\frac{x_{i-1}+x_i}{2}\big) = \int_a^b f''(t)K_1(t)\,dt = -\int_a^b f'(t)K_1'(t)\,dt,$$

so

$$K_0(t) = -K_1'(t) = \begin{cases} x_{i-1} - t, & x_{i-1} \le t \le (x_{i-1}+x_i)/2, \\ x_i - t, & (x_{i-1}+x_i)/2 \le t \le x_i. \end{cases}$$

Note that in this case the kernel does not have constant sign. The L^1 norm is easily bounded: $\|K_0\|_{L^1} = \sum h_i^2/4 \le (b-a)h/4$. We thus get another estimate for the composite midpoint rule, valuable especially for f which is not in $C^2([a,b])$:

$$|Ef| \le \frac{(b-a)h}{4}\|f'\|_{L^\infty}.$$

REMARK. We have bounded the Peano kernels in L^1 in order to obtain an error bound involving the L^∞ norm of a derivative of the solution. It is also possible to use an L^p bound for the Peano kernel ($p > 1$) in order to obtain a bound in terms of the L^q ($q = p/(p-1) < \infty$) norm of a derivative of the solution. This is preferable in cases were the derivative is singular or nearly so.

3. Richardson Extrapolation

Let J_1 be a quadrature rule on the unit interval with degree of precision p. That is $E_1 f := If - J_1 f = 0$ for $f \in \mathcal{P}_p$, but not, in general for $f \in \mathcal{P}_{p+1}$. We can then write $E_1 f = \int_0^1 f^{p+1}(t)K_1(t)\,dt$. Let $c_1 = \int_0^1 K_1$. Note that $c_1 \neq 0$, since that would imply degree of precision at least $p+1$.

Now let J_2 be another quadrature of the same degree of precision, and let c_2 denote the corresponding constant. Assume that $c_2 \neq c_1$. Then it is possible to find a linear combination $J = \alpha_1 J_1 + \alpha_2 J_2$ of the rules, which is itself a quadrature rule with degree of precision greater than p. Indeed, define the α_i by the equations $\alpha_1 + \alpha_2 = 1$ and $\alpha_1 c_1 + \alpha_2 c_2 = 0$. With these values, suppose $f \in \mathcal{P}_{p+1}$ so $f^{(p+1)}$ is constant. Then

$$If - (\alpha_1 J_1 f + \alpha_2 J_2 f) = \alpha_1(If - J_1 f) + \alpha_2(If - J_2 f) = \alpha_1 c_1 f^{(p+1)} + \alpha_2 c_2 f^{(p+1)} = 0.$$

That is, $\bar{J}f := \alpha_1 J_1 f + \alpha_2 J_2 f$ is a new quadrature rule of higher precision.

As an example, let $J_1 f = [f(0) + f(1)]/2$ be the trapezoidal rule and $J_2 f = f(1/2)$ the midpoint rule. Then we have $c_1 = -1/12$, $c_2 = 1/24$, so $\alpha_1 = 1/3$, $\alpha_2 = 2/3$, and $\bar{J}f = f(0)/6 + 2/3 f(1/2) + f(1)/6$, which is precisely Simpson's rule. Note that the new rule has degree of precision at least 2 by construction, but actually 3, since, as usual, a symmetric rule integrates odd powers exactly.

One common way to apply Richardson's extrapolation is with a quadrature rule and the same rule applied in composite form with two subintervals of equal length. Thus, for example, we could combine the trapezoidal rule $J_1 f$ with $J_2 f := [f(0) + 2f(1/2) + f(1)]/4$. For this purpose it is not even necessary to know the error constant c_1 for J_1, but only the fact that the degree of precision of the rule is 1. When we apply the composite form

this means we bring in a factor of the second power of the subinterval size, i.e., a factor of $1/4$, so $c_2 = c_1/4$. Thus $\alpha_1 = -1/3$ and $\alpha_2 = 4/3$, and the new rule turns out to be Simpson's rule again. The general pattern is that if J_1 is a rule with degree of precision p, and J_2 is the same rule applied with two subintervals, then the constants c_i are in the ratio $2^{p+1} : 1$ so $\alpha_1 = -1/(2^{p+1} - 1)$, $\alpha_2 = 2^{p+1}/(2^{p+1} - 1)$. As a second example, suppose that $J_1 f = [f(0) + 3f(1/3) + 3f(2/3) + f(1)]/8$ is Simpson's 3/8 rule, so $p = 3$. Then

$$\begin{aligned}\bar{J}f &= -\frac{1}{15}[f(0) + 3f(1/3) + 3f(2/3) + f(1)]/8 \\ &\quad + \frac{16}{15}[f(0) + 3f(1/6) + 3f(1/3) + 2f(1/2) + 3f(2/3) + 3f(6/6) + f(1)]/16 \\ &= \frac{1}{120}[7f(0) + 24f(1/6) + 21f(1/3) + 16f(1/2) + 21f(2/3) + 24f(5/6) + 7f(1)].\end{aligned}$$

The result 7 point rule has degree of precision 5 (so does not coincide with the 7 point closed Newton-Cotes rule, which has degree of precision 7).

4. Asymptotic error expansions

Consider again the error in the trapezoidal rule

$$Ef = If - Jf = \int_0^1 f - \frac{f(0) + f(1)}{2}.$$

We have $Ef = \int_0^1 Kf''$ where $\int_0^1 K = -1/12$. Now if $f \in \mathcal{P}_2$ then f'' is constant, so $\int_0^1 (K + 1/12)f'' = 0$. Thus

$$Ef = \int_0^1 Kf'' = -\frac{1}{12}\int_0^1 f'' + \int_0^1 (K + 1/12)f'' = \frac{1}{12}[f'(0) - f'(1)] + 0.$$

In other words, if the quadrature rule

$$\tilde{J}f = \frac{f(0) + f(1)}{2} + \frac{f'(0) - f'(1)}{12},$$

(known as the trapezoidal rule with endpoint corrections), has degree of precision at least 2. By symmetry, again, we see it has degree of precision 3. The use of derivative values in the quadrature rule is unusual, but we still get a linear functional to which we can apply the Peano kernel theorem, and hence this rule will be fourth order accurate when applied in composite form, that is, the error will be bounded by a multiple of $h^4 \|f^{(4)}\|_{L^\infty}$ where h is the maximum subinterval size. The fact that the derivative values enter only as the difference (not the sum!) at the endpoints leads to a big simplification for the composite rule with *equal* subintervals. If $h = (b - a)/n$, $x_i = a + ih$, and $J_h f$ denotes the usual composite trapezoidal rule, then the composite trapezoidal rule with endpoint corrections is simply

$$\int_a^b f \approx \tilde{J}_h f = J_h f + h^2 \frac{f'(a) - f'(b)}{12}.$$

Since this rule is fourth order accurate, we have identified the leading term of an asymptotic expansion of the error is the (ordinary) composite trapezoidal rule as a function of h. We

knew before that $E_h = \int_a^b f - J_h f = O(h^2)$. We know now that

$$E_h f = c_2 h^2 + O(h^4),$$

where c_2 is precisely $[f'(b) - f'(a)]/12$.

With careful analysis we can determine the entire asymptotic expansion of $E_h f$. To do this, we need to introduce the *Bernoulli polynomials* $B_n(x)$. There are many ways to define the Bernoulli polynomials. We use:

- $B_0(x) = 1$
- $B_1(x) = x - 1/2$
- $B_n'(x) = nB_{n-1}(x)$, $n = 2, 3, \ldots$
- $B_n(0) = B_n(1) = 0$ for $n = 3, 5, 7, \ldots$

This clearly uniquely determines all the odd-indexed polynomials as the solution of the boundary value problem

$$B_n''(x) = n(n-1)B_{n-2}(x), \quad B_n(0) = B_n(1) = 0, \quad n = 3, 5, \ldots,$$

and then the equation $nB_{n-1}(x) = B_n'(x)$ determines the even-indexed Bernoulli polynomials. Note that $B_n(x)$ is a monic polynomial of degree n and is an odd or even function of $x - 1/2$ according to whether n is odd or even.

$$B_0(x) = 1, \tag{2.6}$$

$$B_1(x) = x - \frac{1}{2}, \tag{2.7}$$

$$B_2(x) = (x - \frac{1}{2})^2 - \frac{1}{12} = x^2 - x + \frac{1}{6}, \tag{2.8}$$

$$B_3(x) = (x - \frac{1}{2})^3 - \frac{1}{4}(x - \frac{1}{2}) = x^3 - \frac{3}{2}x + \frac{1}{2}x, \tag{2.9}$$

$$B_4(x) = (x - \frac{1}{2})^4 - \frac{1}{2}(x - \frac{1}{2})^2 + \frac{7}{48}(x - \frac{1}{2}) = x^4 - 2x^3 + x^2 - \frac{1}{30}. \tag{2.10}$$

The *Bernoulli numbers* are defined by $B_k = B_k(0)$. Thus $B_1 = -1/2$ and $B_k = 0$ for $k = 3, 5, 7 \ldots$. $B_0 = 1$, $B_2 = 1/6$, $B_4 = -1/30$, $B_6 = 1/42$.

REMARK. Euler gave a *generating function* for the Bernoulli polynomials:

$$\frac{te^{tx}}{e^t - 1} = \sum_{n=0}^{\infty} B_n(x)\frac{t^n}{n!},$$

and this is often used to define them. Setting $x = 0$ we get a generating function for the Bernoulli numbers:

$$\frac{t}{e^t - 1} = \sum_{n=0}^{\infty} B_n \frac{t^n}{n!}.$$

Now we apply the Bernoulli polynomials to expand $\int_0^1 f$ in terms of the values of f and its derivatives at 0 and 1. This will give the trapezoidal rule, the endpoint corrections, and

then higher endpoint corrections. The derivation is simply repeated integration by parts:

$$\begin{aligned}\int_0^1 f(x)\,dx &= \int_0^1 f(x)B_0(x)\,dx = \int_0^1 f(x)B_1'(x)\,dx\\ &= \frac{1}{2}[f(0)+f(1)] - \int_0^1 f'(x)B_1(x)\,dx\\ &= \frac{1}{2}[f(0)+f(1)] - \frac{1}{2}\int_0^1 f'(x)B_2'(x)\,dx\\ &= \frac{1}{2}[f(0)+f(1)] + \frac{B_2}{2}[f'(0)-f'(1)] + \frac{1}{2}\int_0^1 f''(x)B_2(x)\,dx.\end{aligned}$$

Now

$$\frac{1}{2}\int_0^1 f''(x)B_2(x)\,dx = \frac{1}{4!}\int_0^1 f''(x)B_4''(x)\,dx,$$

so integrating by parts two more times gets us to

$$\begin{aligned}&\int_0^1 f(x)\,dx\\ &\quad= \frac{1}{2}[f(0)+f(1)] + \frac{B_2}{2}[f'(0)-f'(1)] + \frac{B_4}{4!}[f'''(0)-f'''(1)] + \frac{1}{4!}\int_0^1 f^{(4)}(x)B_4(x)\,dx.\end{aligned}$$

Continuing this argument (formally, using induction) we prove:

THEOREM 2.1. *Let m be a positive integer, $f \in C^{2m}([0,1])$. Then*

$$\begin{aligned}&\int_0^1 f(x)\,dx\\ &\quad= \frac{1}{2}[f(0)+f(1)] + \sum_{k=1}^{m}\frac{B_{2k}}{(2k)!}[f^{(2k-1)}(0)-f^{(2k-1)}(1)] + \frac{1}{(2m)!}\int_0^1 f^{(2m)}(x)B_{2m}(x)\,dx.\end{aligned}$$

This theorem gives the formula for the trapezoidal rule with m endpoint corrections and shows that it has degree of precision at least $2m-1$, and exhibits the Peano kernel. However $\int_0^1 B_{2m}(x)\,dx = 2m[B_{2m-1}(1) - B_{2m-1}(0)] = 0$, so the degree of precision is at least $2m$, and by parity, actually $2m+1$. To derive the Peano kernel error representation in terms of the $2m+2$ derivative, replace m by $m+1$ in the expansion and combine the final term of the sum with the integral to get

$$\int_0^1 f(x)\,dx = \frac{1}{2}[f(0)+f(1)] + \sum_{k=1}^{m}\frac{B_{2k}}{(2k)!}[f^{(2k-1)}(0)-f^{(2k-1)}(1)] + R_m,$$

where

$$\begin{aligned}R_m &= \frac{B_{2m+2}}{(2m+2)!}[f^{(2m+1)}(0)-f^{(2m+1)}(1)] + \frac{1}{(2m+2)!}\int_0^1 f^{(2m+2)}(x)B_{2m+2}(x)\,dx\\ &= \frac{1}{(2m+2)!}\int_0^1 f^{(2m+2)}(x)[B_{2m+2}(x) - B_{2m+2}]\,dx.\end{aligned}$$

Thus we see that the trapezoidal rule with m end corrections has degree of precision $2m+1$ and the Peano kernel for the $2m+2$nd derivative is $[B_{2m+2}(x) - B_{2m+2}(0)]/(2m+2)!$. Note that this kernel does not change sign (since the even indexed Bernoulli polynomials are monotonic on $[0, 1/2]$ and even), and that its integral is $-B_{2m+2}/(2m+2)!$ (since the even indexed Bernoulli polynomials have mean value zero). Thus we have the following theorem.

THEOREM 2.2. *Let m be a positive integer, $f \in C^{2m+2}([0,1])$. Then*

$$\int_0^1 f(x)\,dx = \frac{1}{2}[f(0)+f(1)] + \sum_{k=1}^{m} \frac{B_{2k}}{(2k)!}[f^{(2k-1)}(0) - f^{(2k-1)}(1)] + R_m$$

where

$$R_m = \frac{1}{(2m+2)!}\int_0^1 f^{(2m+2)}(x)[B_{2m+2}(x) - B_{2m+2}]\,dx = -\frac{B_{2m+2}}{(2m+2)!} f^{(2m+2)}(\xi)$$

for some $\xi \in (0,1)$.

From this we easily get a result for the composite trapezoidal rule.

COROLLARY 2.3. *Let m and n be positive integers and $f \in C^{2m+2}([a,b])$ for some $a < b$. Set $h = (b-a)/n$, $x_i = a + ih$. Then*

$$\int_a^b f(x)\,dx = \frac{h}{2}[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)] + \sum_{k=1}^{m} \frac{B_{2k}}{(2k)!} h^{2k}[f^{(2k-1)}(a) - f^{(2k-1)}(b)] + R_m$$

where

$$R_m = -\frac{B_{2m+2}}{(2m+2)!}(b-a)h^{2m+2} f^{(2m+2)}(\xi),$$

for some $\xi \in (a,b)$.

For smooth periodic functions integrated over a period, the endpoint corrections disappear and the ordinary trapezoidal rule (with constant subintervals) is $O(h^p)$ for any p:

COROLLARY 2.4. *Let m and n be positive integers and $f \in C^{2m+2}(\mathbb{R})$ is periodic with period T. Let $a \in \mathbb{R}$, $b = a + T$, $h = T/n$, $x_i = a + ih$. Then*

$$\int_a^b f(x)\,dx = h\sum_{i=0}^{n-1} f(x_i) + R_m$$

where

$$R_m = -\frac{B_{2m+2}}{(2m+2)!}(b-a)h^{2m+2} f^{(2m+2)}(\xi),$$

for some $\xi \in (a,b)$.

In fact, if f is real-analytic and periodic, then the convergence of the trapezoidal rule is exponential. We can deduce this from our previous analysis of the approximation of periodic analytic functions by trigonometric polynomials (and so we don't need the Euler-Maclaurin expansion for this). Assume for simplicity that the period is 2π and recall that we proved that for such f there exist positive constants C and δ for which $\inf_{q\in\mathcal{T}_n}\|f-q\|_{L^\infty} \le Ce^{-\delta n}$. Now if $q(x) = e^{imx}$, $m \neq 0$, then $\int_0^{2\pi} q = 0$ and $\sum_{k=0}^{n-1} q(k/n) = 0$ for all n which are not divisors on m. In particular, if we write E_n for the error operator for the trapezoidal rule

with n equal subintervals, $E_n q = 0$ for all $q \in \mathcal{T}_{n-1}$. Note also that $|E_n g| \le 2\|g\|_{L^\infty}$ for any continuous g. Thus

$$|E_n f| = \inf_{q\in\mathcal{T}_{n-1}} |E_n(f-q)| \le 2 \inf_{q\in\mathcal{T}_{n-1}} \|f-q\|_{L^\infty} \le 2Ce^{-\delta(n-1)} = C'e^{-\delta n}.$$

THEOREM 2.5. *Let f be real-analytic and T-periodic, $b = a+T$. Then there exist positive constants δ and C such that*

$$\left| \int_a^b f - \frac{T}{n} \sum_{k=0}^{n-1} f(a + kT/n) \right| \le Ce^{-\delta n}.$$

5. Romberg Integration

Suppose we compute the trapezoidal rule approximation to $I = \int_a^b f$ using equal subintervals of size h, and compute it again using twice as many subintervals of size $h/2$. Call the resulting approximations $T_h f$ and $T_{h/2} f$, respectively. Then, assuming f is smooth, we have

$$I = T_h f + c_1 h^2 + c_2 h^4 + \cdots,$$
$$I = T_{h/2} f + \frac{1}{4} c_1 h^2 + \frac{1}{16} c_2 h^4 + \cdots.$$

Here the c_i are independent of h ($c_1 = [f'(a) - f'(b)]/12, \ldots$). We may then use Richardson extrapolation to obtain an $O(h^4)$ approximation to I: with $T^1_{h/2} = (4T_{h/2}f - T_h)/3$ we have

$$I = T^1_{h/2} + c^1_2 h^4 + c^1_3 h^6 + \cdots$$

for some numbers c^2_1 independent of h. Of course we know that the fourth order rule T^1 is just Simpson's rule.

If we also compute $T_{h/4} f$ as well, that we can combine similarly it with $T_{h/2} f$ to obtain $T^1_{h/4} f$, and for which the leading term of the error will be $(c^1_2/16)h^4$. We can then combine $T^1_{h/2} f$ and $T^1_{h/4}$ by Richardson extrapolation to obtain a sixth order rule: $T^2_{h/4} = (16T^1_{h/4} - T^1_{h/2})/15$. In fact, using the same set of functional evaluations we need to compute the 2nd order rule $T_{h/2^m} f$ we can obtain a $2m$th order rule $T^m_{h/2^m} f$. The following diagram shows the order of computation.

$$\begin{array}{ccccccc}
T_h f & & & & & & \\
 & \searrow & & & & & \\
T_{h/2} f & \to & T^1_{h/2} f & & & & \\
 & \searrow & & \searrow & & & \\
T_{h/4} f & \to & T^1_{h/4} & \to & T^2_{h/4} f & & \\
\vdots & & \vdots & & \vdots & \ddots & \\
T_{h/2^m} f & \to & T^1_{h/2^m} f & \to & T^2_{h/2^m} f & \cdots & T^m_{h/2^m} f
\end{array}$$

This systematic use of Richardson extrapolation to obtain the highest possible order from the given point evaluations is called *Romberg Integration*. The computation of the final approximation $T^m_{h/2^m} f$ is very cheap once the first column has been generated (namely once f has been evaluated at all the necessary points), but it often gives a drastic improvement in accuracy.

This diagram is reminiscent of a divided difference table for computing an interpolating polynomial. In fact, there is a direct connection. Let us denote by $T(h)$ the trapezoidal rule approximation to I using equal subintervals of size h. (Of course $T(h)$ is only defined when $h = (b-a)/n$ for some positive integer n.) The $\lim_{h\to 0} T(h) = I$. If we have computed $T(h)$ for $m+1$ distinct positive values of h, $h_0, \dots, h_m$, then a natural way to estimate $I = T(0)$ is to compute the Lagrange interpolating polynomial determined by these $m+1$ values of h and $T(h)$ and estimate I by the value of the interpolating polynomial at 0. Actually, since we know from the Euler-Maclaurin expansion that $T(h)$ has an asymptotic expansion in powers of h^2, we shall use a polynomial in h^2: $P(h) = Q(h^2) = \sum_{k=0}^{m} a_k h^{2k}$. The polynomial $Q \in \mathcal{P}_m$ is determined by the conditions $Q(h_i^2) = T_{h_i} f$. For example, if we have computed $T_{h_0} f$ and $T_{h_1} f$, then

$$Q(x) = T_{h_0} f + \frac{T_{h_1} f - T_{h_0} f}{h_1^2 - h_0^2}(x - h_0^2),$$

so

$$I \approx Q(0) = \frac{h_0^2 T_{h_1} f - h_1^2 T_{h_0} f}{h_1^2 - h_0^2},$$

In particular, if $h_1 = h_0/2$, then

$$I \approx Q(0) = \frac{4T_{h_1} f - T_{h_0} f}{3},$$

and we see that this procedure reduces to Richardson extrapolation (this in fact explains the use of the word "extrapolation"). In general, if we compute $T_{h_i} f$ for $i = 0, \dots, m$ with $h_i = h_0/2^i$, and then compute the polynomial $Q \in \mathcal{P}_m$ determined by $Q(h_i^2) = T_{h_i} f$, it can be checked that $Q(0)$ is exactly $T^m_{h_0/2^m} f$, the result of Romberg integration. For this reason, Romberg integration is sometimes called *extrapolation to the limit*. This analysis also shows that it is not necessary that the $h_i = h_0/2^i$. We could use another sequence of values of h_i as well. Of course the sequence $h_0/2^i$ is convenient, because it means that all the function evaluation needed to compute $T_{h_m} f$ are sufficient to compute all the $T_{h_i} f$.

Remark. Note that the idea of extrapolation to the limit applies not only to error in quadrature, but whenever we know that the error has an asymptotic expansion in powers of a parameter h. We have assumed the expansion is a sum of even powers, but this can be generalized to allow an expansion in an arbitrary sequence of (known) powers.

6. Gaussian Quadrature

Consider an ordinary quadrature rule with n points $x_1 < x_2 < \cdots < x_n$ and weights $w_1, w_2, \dots, w_n$:

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i).$$

What is the maximal possible degree of precision? For example, it's obvious that if $n = 1$, the maximal possible degree of precision is 1, which is achieved by the midpoint rule. For any number of points, an upper bound is immediate:

Theorem 2.6. *The degree of precision of an n-point rule is $< 2n$.*

PROOF. The function $f(x) = \prod_{i=1}^{n}(x - x_i)^2$ is a polynomial of degree $2n$. Clearly $\int f > 0 = \sum w_i f(x_i)$. □

We shall show that the maximum possible degree of precision $2n - 1$ is achieved by a unique n-point rule. Without loss of generality, we may restrict to the interval $[a, b] = [-1, 1]$. Our chief tool will be the Legendre polynomials $P_n(x)$. Recall that $\deg P_n = n$ and P_n is $L^2([-1, 1])$-orthogonal to $\mathcal{P}_{n-1}$. Let $x_1 < \cdots < x_n$ denote the roots of P_n, which we know to be distinct and to belong to $[-1, 1]$. These are called the n Gauss points on $[-1, 1]$.

We define a quadrature rule by $\int_{-1}^{1} f \approx \int_{-1}^{1} I_n f$ where $I_n f \in \mathcal{P}_{n-1}$ is the Lagrange interpolant to f at the n Gauss points. This is a standard interpolatory quadrature rule:

$$\int_{-1}^{1} I_n f = \sum_{i=1}^{n} w_i f(x_i), \qquad w_i = \int_a^b \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} dx.$$

This rule is called the n-point Gauss rule.

THEOREM 2.7. *The n-point Gauss rule has degree of precision $= 2n - 1$.*

PROOF. Given $f \in \mathcal{P}_{2n-1}$, we can write $f(x) = q(x)P_n(x) + r(x)$, $q \in \mathcal{P}_{n-1}$, $r \in \mathcal{P}_{n-1}$. Then

$$\sum w_i f(x_i) = \sum w_i r(x_i) = \int_{-1}^{1} I_n r(x)\, dx = \int_{-1}^{1} r(x)\, dx,$$

since $I_n r = r$. Also

$$\int_{-1}^{1} f(x)\, dx = \int_{-1}^{1} q(x)P_n(x)\, dx + \int_{-1}^{1} r(x)\, dx = \int_{-1}^{1} r(x)\, dx.$$

□

The weights of the n-point Gauss rule are positive for all n. To see this, let $l_j \in \mathcal{P}_{n-1}$ be the function that is 1 at x_j and 0 at the other x_i. Then $l_j^2 \in \mathcal{P}_{2n-2}$, so

$$w_j = \sum_i w_i l_j^2(x_i) = \int_{-1}^{1} l_j^2(x)\, dx > 0.$$

Note that the n-point Gauss rule is the only n-point quadrature rule with degree of precision $2n - 1$. For if $\sum_{i=1}^{n} w_i f(x_i) = \int_{-1}^{1} f$, then we find that

$$\int_{-1}^{1} q(x)\Pi_{i=1}^{n}(x - x_i)\, dx = 0, \quad q \in \mathcal{P}_{n-1}.$$

This implies that $\Pi_{i=1}^{n}(x - x_i)$ is a multiple of P_n, and hence the x_i are the n Gauss points.

Example: Since $P_2(x) = (3x^2 - 1)/2$, the only 2-point rule with degree of precision 3 is the 2-point Gauss rule

$$\int_{-1}^{1} f(x)\, dx \approx f(a) + f(-a),$$

where $a = \sqrt{3}/3 \approx 0.57735$.

6.1. Weighted Gaussian quadrature. The error in numerical integration depends on the smoothness of the integrand. Frequently an integrand can be written as the product of a relatively simple, but unsmooth function, times a smoother function. This leads to the subject of product integration, the determination of quadrature rules for integrands of the form $f(x)\omega(x)$ where $f(x)$ is smooth and $\omega(x)$ is in some sense simple or standard.

If $\omega(x)$ is any non-negative integrable weight function on $[a, b]$, we can generalize the notion of Gaussian quadrature to computing the product $\int_a^b f(x)\omega(x)\,dx$. We summarize the result as:

THEOREM 2.8. *Let $p_0, p_1, \ldots$ be the orthogonal polynomials on $[a, b]$ with respect to the innerproduct $(f, g) = \int_a^b f(x)g(x)\omega(x)\,dx$. Then p_n has n distinct roots in (a, b), $x_1, \ldots, x_n$. If weight w_i are defined by*

$$w_i = \int_a^b \prod_{\substack{j=1\\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}\omega(x)\,dx.$$

then

$$\int_a^b f(x)\omega(x)\,dx = \sum w_i f(x_i) \quad \textit{for all } f \in \mathcal{P}_{2n-1},$$

and no other choice of n points and weights realizes this. The weights w_i are all positive.

The reader should supply the proof.

For weight $\equiv 1$ on $[-1, 1]$, we get ordinary Gaussian integration, with the points at the roots of the Legendre polynomials.

The zeros of the Chebyshev polynomials are the best points to use for approximating product integrals of the form

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1 - x^2}}\,dx.$$

The Hermite polynomials, which are orthogonal polynomials on $\mathbb{R}$ with the weight e^{-x^2}, give the best points for estimating

$$\int_{-\infty}^{\infty} f(x)e^{-x^2}dx.$$

6.2. The error in Gaussian quadrature. Let x_i and w_i be the Gaussian quadrature weights and points for approximating $\int_{-1}^{1} f(x)\omega(x)\,dx$ (we allow a general non-negative weight, because it adds no additional difficulty). Then the error functional

$$Ef := \int_{-1}^{1} f(x)\omega(x)\,dx - \sum_{i=1}^{n} w_i f(x_i)$$

vanishes for $f \in \mathcal{P}_{2n-1}$, so the Peano kernel theorem implies that

$$Ef = \int_{-1}^{1} K(x)f^{(2n)}(x)\,dx, \quad \text{for } f \in C^{(2n)},$$

so

$$|Ef| \leq C_n \|f^{(2n)}\|_\infty.$$

We can use Hermite interpolation to get an explicit expression for the constant C_n.

Let $h \in \mathcal{P}_{2n-1}$ be the Hermite interpolant of f:

$$h(x_i) = f(x_i), \quad h'(x_i) = f'(x_i), \quad i = 1, \ldots, n.$$

Then

$$\int_{-1}^{1} h(x)\omega(x)\,dx = \sum_{i=1}^{n} w_i h(x_i) = \sum_{i=1}^{n} w_i f(x_i),$$

so

$$Ef = \int_{-1}^{1} [f(x) - h(x)]\omega(x)\,dx.$$

Now by the error formula for interpolation,

$$\begin{aligned} f(x) - h(x) &= f[x_1, x_1, x_2, x_2, \ldots, x_n, x_n, x](x - x_1)^2(x - x_2)^2 \cdots (x - x_n)^2 \\ &= f[x_1, x_1, x_2, x_2, \ldots, x_n, x_n, x]|p_n(x)|^2, \end{aligned}$$

(where for simplicity we normalize p_n to be monic). Thus

$$Ef = \int_{-1}^{1} f[x_1, x_1, x_2, x_2, \ldots, x_n, x_n, x]|p_n(x)|^2\omega(x)\,dx.$$

Now $|p_n(x)|^2\omega(x)$ is positive, and $f[x_1, x_1, x_2, x_2, \ldots, x_n, x_n, x]$ is a continuous function of x (by the Hermite-Gennochi Theorem), so the integral mean value theorem applies giving

$$\begin{aligned} Ef &= f[x_1, x_1, x_2, x_2, \ldots, x_n, x_n, \eta] \int_{-1}^{1} |p_n(x)|^2\omega(x)\,dx \\ &= \frac{f^{(2n)}(\xi)}{(2n)!} \int_{-1}^{1} |p_n(x)|^2\omega(x)\,dx = \|p_n\|_\omega^2 \frac{f^{(2n)}(\xi)}{(2n)!} \end{aligned}$$

for some $\eta, \xi \in (-1, 1)$ (where $\|\cdot\|_\omega$ is the L^2-norm with weight ω). Thus we have shown:

THEOREM 2.9. *If $f \in C^{(2n)}([-1, 1])$, then the error in the Gaussian quadrature rule satisfies*

$$\int_{-1}^{1} f(x)\omega(x)\,dx - \sum_{i=1}^{n} w_i f(x_i) = \|p_n\|_\omega^2 \frac{f^{(2n)}(\xi)}{(2n)!}$$

for some $\xi \in (-1, 1)$.

If we restrict to the case $\omega = 1$, we can scale this result from the unit interval to an interval of length h, and add up on subintervals to obtain the usual sort of error estimate when the Gauss rule is used as a composite quadrature rule. (If $f \in C^{2n}([a, b])$, then the error in the composite n-point Gauss rule tends to zero like h^{2n} where h is the largest subinterval size.

Now we want to analyze the behavior of the error $E_n f$ in the simple n-point Gauss rule as $n \to \infty$. For the Newton-Cotes rules, we don't have convergence as the number of points tends to infinity, even for f analytic on $[-1, 1]$ (this is quite believable, in view of the Runge example). However the situation for the Gauss rules is altogether different. First we consider the Lebesgue constant of the n-point Gauss rule:

$$L_n = \sup_{\|f\|_{L^\infty} \le 1} |\sum w_i f(x_i)| = \sum |w_i|.$$

Since the w_i are all positive and sum to $\|\omega\|_{L^1} = \int_a^b \omega(x)\,dx$, we have $L_n = \|\omega\|_{L^1}$. Note that we also have $|\int_a^b f\omega| \le \|\omega\|_{L^1}\|f\|_{L^\infty}$, so $|E_n f| \le 2\|\omega\|_{L^1}\|f\|_{L^\infty}$. We then use the standard argument:

$$|E_n f| = \inf_{q\in\mathcal{P}_{2n-1}} |E_n(f-q)| \le 2\|\omega\|_{L^1} \inf_{q\in\mathcal{P}_{2n-1}} \|f-q\|_{L^\infty}.$$

We can thus bound the error using what we know about polynomial approximation. For example using the Weierstrass theorem we get:

THEOREM 2.10. *For any $f \in C([a,b])$ error $E_n f$ in the n-point (weighted) Gauss rule tends to zero as n tends to infinity.*

Similarly, using the Jackson theorems, we get:

THEOREM 2.11. *For each positive integer m, there exists a constant C such that $|E_n f| \le Cn^{-m}\|f^{(m)}\|_{L^\infty}$ for all $f \in C^m([a,b])$.*

Finally we get exponential convergence for f analytic.

THEOREM 2.12. *If f is analytic on $[a,b]$, then there exist positive constant C and δ such that $|E_n f| \le Ce^{-\delta n}$.*

Thus to improve the error in a Gaussian quadrature calculation we have two choices: we may increase n, or we may use the rule as a composite rule and increase the number of subinterval size.

A disadvantage of the Gaussian rules, is that, whichever of these two choices we make, we won't be able to use the functional evaluations from the previous computation for the new one.

7. Adaptive quadrature

The idea of adaptive quadrature is to use a composite rule with (unequal) subintervals chosen automatically so as to make the error small with as few function evaluations as possible. To explain the ideas we will first discuss the composite Simpson's rule.

Let $a = x_0 < \ldots < x_n = b$, $h_i = x_i - x_{i-1}$. Let $S_{[x_{i-1},x_i]}f = h_i[f(x_{i-1}) + 4f((x_{i-1}+x_i)/2) + f(x_i)]/6$ denote the Simpson's rule on the ith subinterval and $E_i = E_{[x_{i-1},x_i]}f = \int_{x_{i-1}}^{x_i} f - S_{[x_{i-1},x_i]}f$ the error. We know that $E_i = ch_i^5 f^{(4)}(\xi_i)$ for some $\xi_i \in (x_{i-1}, x_i)$ and some absolute constant c. (For Simpson's rule, $c = -1/2880$, but we don't need to know the particular value in what follows.) We will assume that h_i is small enough that $f^{(4)}$ is roughly constant on (x_{i-1}, x_i), so $E_i f = ch_i^5 f_i^{(4)}$ where $f_i^{(4)}$ is this constant value (e.g., the value of $f^{(4)}$ at the midpoint). The error in the composite Simpson's rule is given by $|E_1 + \cdots + E_n|$. Since we don't want to count on the errors from some subintervals cancelling those from other subintervals, we shall try to minimize $|E_1| + \cdots + |E_n|$. If we vary the subinterval sizes h_i, we vary this quantity. We claim that when an optimal choice of the h_i is made, then the *error per unit subinterval size*, $|E_i/h_i| = ch_i^4|f_i^{(4)}|$, will be roughly the same on each subinterval (i.e., independent of i). To see this, consider how the error on the ith subinterval changes when we increase the interval size by a small amount δ. The change is

$$\delta\frac{d|E_i|}{dh_i} = 5c\delta h_i^4|f_i^{(4)}|.$$

So if we decrease h_i by δ and increase h_j by δ, the total change to the error will be about $5c\delta(h_j^4|f_j^{(4)}| - h_i^4|f_i^{(4)}|)$. If we are at an optimum, this quantity must be 0 (since if it were negative it would be advantageous to make this change, and if it were positive it would be advantageous to change in the opposite way). Thus for the optimal mesh distribution, $h_i^4|f_i^{(4)}|$ is indeed roughly constant.

This suggest the basic structure of an adaptive quadrature algorithm. Start with a coarse partition (e.g., one or a few equal subintervals), and a tolerance ϵ for $|E_i|/h_i$, i.e., the solution will be considered acceptable when $|E_i|/h_i \le \epsilon$ for all subintervals. This will ensure that $|E_1| + \cdots + |E_n| \le \epsilon(h_1 + \cdots + h_n) = \epsilon(b-a)$ and the user should supply the tolerance accordingly. Now check each of the subintervals for the condition $|E_i|/h_i \le \epsilon$, and bisect (or otherwise refine) any one which does not pass.

In order to be able to implement a scheme along these lines we need a way to estimate $E_{[\alpha,\beta]}f$ for $h = \beta - \alpha$ small. Since we don't know the exact value $\int_\alpha^\beta f$ we can't use the definition of the error as the difference between the exact and computed values. Since we don't know $f^{(4)}$ we can't use our asymptotic formula for the error either. Instead we shall use Richardson extrapolation. Set $\gamma = (\alpha+\beta)/2$ and let $\tilde{S}_{[\alpha,\beta]} = S_{[\alpha,\gamma]}f + S_{[\gamma,\beta]}f$, the double Simpson's rule (composite rule with two equal subintervals). Then

$$\int_\alpha^\beta f = S_{[\alpha,\beta]}f + Ch^5 + O(h^7),$$
$$\int_\alpha^\beta f = \tilde{S}_{[\alpha,\beta]}f + \frac{1}{16}Ch^5 + O(h^7),$$

where the constant $C = cf^{(4)}(\gamma)$. Combining these we have

$$\int_\alpha^\beta f = \frac{1}{15}(16\tilde{S}_{[\alpha,\beta]}f - S_{[\alpha,\beta]}f) + O(h^7),$$
$$\int_\alpha^\beta f - S_{[\alpha,\beta]}f = \frac{16}{15}(\tilde{S}_{[\alpha,\beta]}f - S_{[\alpha,\beta]}f) + O(h^7),$$
$$\int_\alpha^\beta f - \tilde{S}_{[\alpha,\beta]}f = \frac{1}{15}(\tilde{S}_{[\alpha,\beta]}f - S_{[\alpha,\beta]}f) + O(h^5).$$

Thus if we compute both the simple and the double Simpson's rule, we may estimate the error in either by a multiple of their difference. We can then test whether this error passes our tolerance test $|E|/h < \epsilon$. Since we have to compute the double Simpson's rule to estimate the errors, and this is almost surely more accurate than the simple rule (with about 1/16th the error), it is sensible to use the double rule as the value we test and eventually accept. That is, we check if $(1/15)[\tilde{S}_{[\alpha,\beta]}f - S_{[\alpha,\beta]}f]|/(\beta-\alpha) \le \epsilon$, and, if so, we use $\tilde{S}_{[\alpha,\beta]}f$ to estimate $\int_\alpha^\beta f$.

REMARK. Actually, it is reasonable to use the Richardson extrapolated value $(16\tilde{S}_{[\alpha,\beta]}f - S_{[\alpha,\beta]}f)/15$ to estimate $\int_\alpha^\beta f$, since this is expected to have a much smaller error. However it is not possible, without doing further Richardson extrapolation, to estimate the error in this rule. Thus the code writer has a design choice to make: (1) use $\tilde{S}$ because the estimates apply to this, or (2) use $(16\tilde{S} - S)/15$ because it is probably better and will therefore probably

supply the user with a even less error than requested. In either case, the error would be estimated by $(\tilde{S} - S)/15$.

If we are willing to use recursion, the implementation of such an algorithm, it is quite simple. Here is metacode that illustrates the basic flow:

$\boldsymbol{I} = \text{adapt}(f, a, b, \epsilon)$
input: a, b, endpoints of interval of integration
$f \in C([a, b])$, integrand
ϵ error per unit subinterval tolerance
output: $I \approx \int_a^b f(x)\,dx$

---

compute $S_{[a,b]}f$, $\tilde{S}_{[a,b]}f$, and $E = (\tilde{S}_{[a,b]}f - S_{[a,b]}f)/15$
if $|E| \le \epsilon(b - a)$ **then**
$I = \tilde{S}_{[a,b]}f$
else
$I = \text{adapt}(f, a, (a + b)/2, \epsilon) + \text{adapt}(f, (a + b)/2, b, \epsilon)$
end if

Algorithm 2.1: Basic flow for an adaptive quadrature routine.

The following Matlab function, `adaptsimp.m` implements this flow. It includes some basic practicalities. First, it takes care not to evaluate f at the same point twice. Second, in order to avoid very long or possibly non-terminating computations for very bad integrands it puts a limit of 10 on the number of times a subinterval can be bisected. It accomplishes both of these by using two different calling syntaxes. When called by the user, the syntax is `adaptsmp(f,a,b,tol)` as above. But when the routine calls itself recursively it uses the call `adaptsmp(f,a,b,tol,lev,fa,fm,fb)`. The extra parameter `lev` simply keeps track of the recursion level so that an exit can be effected if it exceeds 10. The parameters `fa`, `fm`, `fb` are the values of f at a, $(a + b)/2$, and b, since these have already been computed and will be needed again. Finally, from among the various ways to pass a function (f) as the argument to another function (`adaptsimp`) in Matlab, we have chosen to make the argument `f` a string containing the name of another Matlab function, which should be of the form `function y=f(x)`, and then we use, e.g., `feval(f,a)` to evaluate this function at a.

```
function int = adaptsmp(f,a,b,tol,lev,fa,fm,fb)
%ADAPTSMP Adaptive Simpson's rule quadrature
%
%  Call as ADAPTSMP('f',a,b,tol) to approximate the integral of f(x)
%  over the interval a < x < b, attempting to achieve a absolute error
%  of at most tol(b-a).  The first argument should be a string containing
%  the name of a function of one variable.  The return value is the
%  approximate integral.
%
%  ADAPTSMP calls itself recursively with the argument list
```

```
%  ADAPTSMP('f',a,b,tol,lev,fa,fm,fb). The variable lev gives the
%  recursion level (which is used to terminate the program if too many
%  levels are used), and fa, fb, and fm are the values of the integrand
%  at a, b, and (a+b)/2, respectively, (which are used to avoid
%  unnecessary function evaluations).

% initialization, first call only
if nargin == 4
  lev = 1;
  fa = feval(f,a);
  fm = feval(f,(a+b)/2);
  fb = feval(f,b);
end

% recursive calls start here

% start by checking for too many levels of recursion; if so
% don't do any more function evaluations, just use the already
% evaluated points and return
if lev > 10
    disp('10 levels of recursion reached.  Giving up on this interval.')
    int = (b-a)*(fa+4*fm+fb)/6;
else
% Divide the interval in half and apply Simpson's rule on each half.
% As an error estimate for this double Simpson's rule we use 1/15 times
% the difference between it and the simple Simpson's rule (which is
% an asymptotically exact error estimate).
    h = b - a;
    flm = feval(f,a+h/4);
    frm = feval(f,b-h/4);
    simpl = h*(fa + 4*flm + fm)/12;
    simpr = h*(fm + 4*frm + fb)/12;
    int = simpl + simpr;
    simp = h*(fa+4*fm+fb)/6;
    err = (int-simp)/15;

    % if tolerance is not satisfied, recursively refine approximation
    if abs(err) > tol*h
       m = (a + b)/2;
       int = adaptsmp(f,a,m,tol,lev+1,fa,flm,fm) ...
           + adaptsmp(f,m,b,tol,lev+1,fm,frm,fb);
    end
end
```

Algorithm 2.2: Matlab routine for adaptive Simpson's rule quadrature.

FIGURE 2.2. Evaluation points for the adaptive quadrature routine `adaptsmp`.

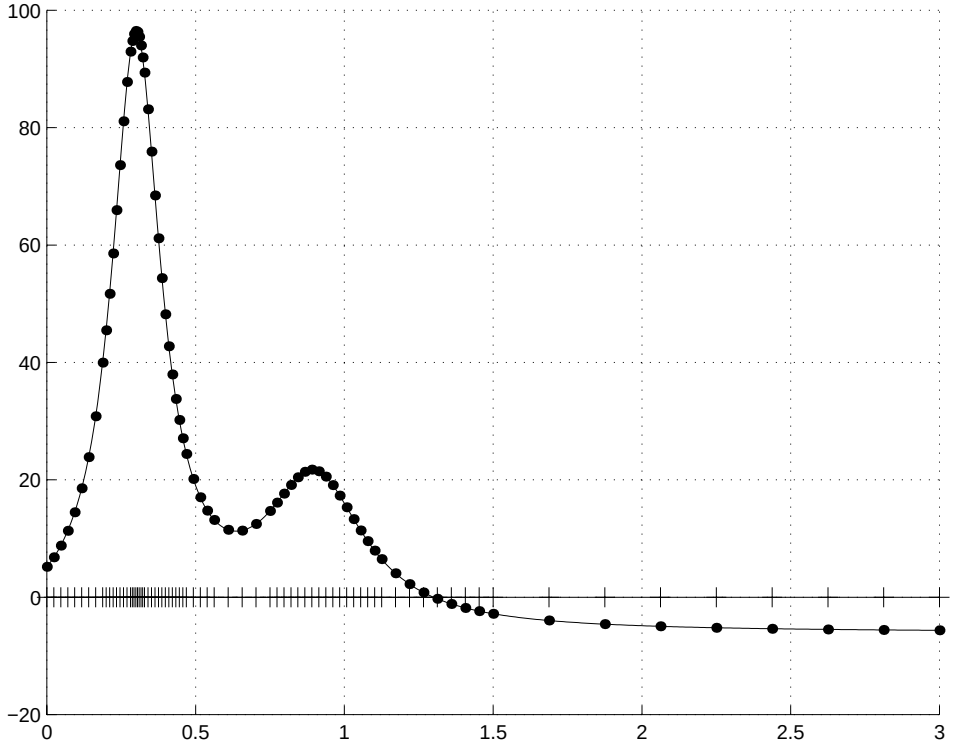


Figure 2.2 shows the performance of this routine in computing

$$\int_0^3 \left[\frac{1}{(x-0.3)^2+0.01} + \frac{1}{(x-0.9)^2+0.04} - 6 \right] dx.$$

We supplied a tolerance of 0.001 and the routine evaluated the integrand 77 times before terminating. The evaluation points were spaced by $3/512 \approx 0.006$ in a small region near the sharp peak of the integrand but only by $3/16 \approx 0.2$ on the entire right half of the interval of integration. The computed solution is 23.9693, which differs from the exact answer of $23.96807984\ldots$ by about 0.0012 or 0.00005%. Thus the error per unit subinterval is 0.0004 which is well below our tolerance of 0.001. This is to be expected since whenever a subinterval exceed tolerance, even if only by a little, it is bisected, reducing the error per unit subinterval substantially (by about a factor of 16).

EXERCISES

(1) Consider the function $f_\alpha(x) = x^\alpha$ on $(0,1)$. For $\alpha > -1$ this function is integrable. Using the computer investigate the rate of convergence of the composite midpoint rule with

equal subintervals for computing $\int_0^1 f_\alpha$ for various values of α. Let r_α denote the rate of convergence, i.e., the largest real number so that the error can be bounded by Ch_α^r for some constant C independent of h. Based on your experiments conjecture an expression for r_α in terms of α valid for all $\alpha > -1$. State your result precisely and prove it that it is indeed true.

(2) Make the same study and analysis for the composite Simpson's rule, except restrict to $\alpha \geq 0$ (since Simpson's rule is not defined if $f(0)$ is not defined).

(3) Give a thorough analysis of Simpson's rule using the Peano kernel theorem. More specifically, there are four separate Peano kernel representations for the error in Simpson's rule depending on the degree of smoothness we assume of the integrand. Give all four in the case of the simple rule on the interval $[-1, 1]$. Give explicit expressions for all four kernels and plot them (indicate the scale on the y axis). Apply this to analyze the error for the composite Simpson's rule on an arbitrary interval using equal subintervals under the assumptions that $f^{(i)}$ is bounded or just integrable for $i = 1, 2, 3$, or 4. For the case $f \in C^4$ also give the result for the composite rule without assuming equal subintervals.

(4) Find the simple quadrature rule of highest degree of precision for estimating $\int_{-1}^1 f(x)\,dx$ in terms of the value of f at $-1/2$, 0, and $1/2$. Give a complete convergence analysis for the corresponding composite quadrature rule using an arbitary subdivision of the interval of integration into subintervals.

(5) Suppose that J_h is an approximation of a desired quantity I for which the asymptotic expansion $I \sim J(h) + c_1 h^{r_1} + c_2 h^{r_2} + \cdots$ holds as $h \to 0$. Here $0 < r_1 < r_2 < \cdots$ and the c_i are independent of h. Imagine that we have computed $J(h)$, $J(h/2)$, $J(h/4)$. Show how Richardson extrapolation can be used to the maximum extent to combine these values to get a higher order approximation to I. What is the order of this approximation?

(6) Find the 1- and 2-point Gaussian quadrature rules for the weight function $\log(1/x)$ on $[0, 1]$. Find expression for the errors.

(7) The n-point Gauss-Lobatto quadrature rule ($n > 1$) is the rule $\int_{-1}^1 f \approx \sum_{i=1}^n w_i f(x_i)$ where the $x_1 = -1$, $x_n = 1$, and the other nodes and the weights are chosen so that the degree of precision is as high as possible. Determine the rule for $n = 2$, 3, and 4. Explain how, for general n, the points relate to the orthogonal polynomials with weight $1 - x^2$. Give a formula for the weights.

(8) One way to define a non-uniform partition of an interval is through a *grading function*. A grading function for the interval $[a, b]$ is a monotone increasing function f mapping $[0, 1]$ one-to-one and onto $[a, b]$. We can then define a partition of $[a, b]$ into n subintervals via the points $x_i = \phi(i/n)$, $i = 0, \ldots, n$. Consider using this partition to compute the integral $\int_0^1 f(x)\,dx$ with the trapezoidal rule. Justify heuristically but convincingly that the optimal choice of grading function should satisfy $f''\big(\phi(t)\big)[\phi'(t)]^2 = \text{const}$. For the function $f(x) = x^\alpha$, $0 < \alpha < 1$, find a grading function satisfying this equation (hint: try $\phi(t) = t^\beta$). For $\alpha = 0.2$, and $n = 1, 2, \ldots, 1024$, compute the trapezoidal rule approximation to $\int_0^1 x^\alpha\,dx$ using the optimal grading function, and verify numerically that the error behaves as $O(1/n^2)$. Thus, using the appropriate grading function, we obtain the same rate of convergence for this singular integrand as for a smooth integrand.

(9) Prove that the rate of convergence for the appropriately graded trapezoidal rule approximation to $\int_0^1 x^\alpha\,dx$ is indeed $O(1/n^2)$ by making a change of variable in the integral so

that the trapezoidal rule with the graded partition for the original integral corresponds to the trapezoidal rule with a uniform partition for the new integral.

CHAPTER 3

Direct Methods of Numerical Linear Algebra

1. Introduction

To fix ideas and notations we start with a very simple algorithm: multiplication of an n-vector by an $m \times n$ matrix: $b = Ax$, $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, so $b \in \mathbb{R}^m$. An obvious algorithm is

for $i = 1 : m$
  $b_i = \sum_{j=1}^{n} a_{ij}x_j$
end

or, written out in full,

for $i = 1 : m$
  $b_i \leftarrow 0$
  for $j = 1 : n$
    $b_i \leftarrow b_i + a_{ij}x_j$
  end
end

Thus the algorithm involves nm additions and nm multiplications. The number of operations is proportional to the number of input data, which is optimal.

Note that the operation $b_i = \sum_{j=1}^{n} a_{ij}x_j$ may be viewed as the computation of a dot product of the vector x with the ith row of A. Since many linear algebra algorithms involve dot products, there are optimized routines to compute dot products on almost all computers, and an efficient implementation can be built on these. This is one example of a BLAS (basic linear algebra subroutine).

Note that our algorithm accesses the matrix A by row order. If the matrix is stored by columns in the computer memory, it is more efficient to use a column oriented algorithm (especially if the matrix is so large that it does not fit in main memory and must be paged to disk; similar considerations apply to cache memory). The computer languages Fortran and Matlab store matrices by column. The computer language C stores by rows.

In fact the matrix-vector multiplication algorithm can be reordered to become column-oriented. Namely, we think of b as a linear combination of the columns of a: $b = \sum_j x_j a_j$, where a_j denotes the jth column of A. Written out fully in terms of scalar operations the algorithm is now

for $i = 1 : m$
  $b_i \leftarrow 0$
end
for $j = 1 : n$
  for $i = 1 : m$
    $b_i \leftarrow b_i + a_{ij}x_j$
  end
end

The inner loop computes a SAXPY operation. A SAXPY is a BLAS which takes a scalar s and two vectors x and y and computes $sx + y$. Thus we may implement matrix-vector multiplication as a row-oriented algorithm consisting chiefly of m dot products of size n or a column-oriented algorithm consisting chiefly of n SAXPYs of size m.

2. Triangular systems

We recall the forward elimination algorithm to solve $Lx = b$ where L is lower triangular. We have $x_i = (b_i - \sum_{j=1}^{i-1} l_{ij}x_j)/l_{ii}$. We may overwrite the b_i with the x_i to get the algorithm:

$b_1 \leftarrow b_1/l_{11}$
for $i = 2 : n$
  for $j = 1 : (i - 1)$
    $b_i \leftarrow b_i - l_{ij}b_j$
  end
  $b_i \leftarrow b_i/l_{ii}$
end

With the usual convention for empty loops, we can write this more briefly as

for $i = 1 : n$
  for $j = 1 : (i - 1)$
    $b_i \leftarrow b_i - l_{ij}b_j$
  end
  $b_i \leftarrow b_i/l_{ii}$
end

or just

for $i = 1 : n$
  $b_i \leftarrow (b_i - \sum_{j=1}^{i-1} l_{ij}b_j)/l_{ii}$
end

Note that this algorithm accesses L by rows and can be implemented in terms of dot products. To get a column-oriented algorithm, we can partition L as

$$\begin{pmatrix} l_{11} & 0 \\ \tilde{l} & \tilde{L} \end{pmatrix} \begin{pmatrix} x_1 \\ \tilde{x} \end{pmatrix} = \begin{pmatrix} b_1 \\ \tilde{b} \end{pmatrix}.$$

This shows that, after computing $x_1 = b_1/l_{11}$ we can reduce to the $(n-1) \times (n-1)$ lower triangular system $\tilde{L}\tilde{x} = \tilde{b} - x_1\tilde{l}$. This idea leads to the algorithm

for $j = 1 : n$
  $b_j \leftarrow b_j/l_{jj}$
  for $i = (j+1) : n$
    $b_i \leftarrow b_i - l_{ij}b_j$
  end
end

Note that this algorithm can be implemented in terms of SAXPY operations.

Of course we may also solve upper triangular matrix systems. We just start with the last equation. This is called back substitution.

3. Gaussian elimination and LU decomposition

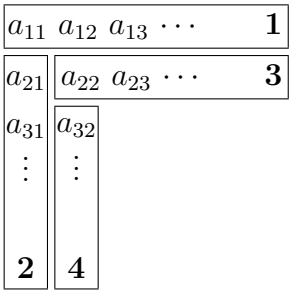
Recall the classical Gaussian elimination algorithm, which begins with a matrix $A = A^{(1)}$ and in $n-1$ steps transforms it to an upper triangular matrix $A^{(n-1)}$.

for $k = 1 : (n-1)$
  for $i = (k+1) : n$
    $m_{ik} = a_{ik}^{(k)}/a_{kk}^{(k)}$
    for $j = (k+1) : n$
      $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}$
    end
  end
end

The multipliers m_{ik} determine a unit lower triangular matrix L with the property that, setting $U = A^{(n-1)}$, we have $A = LU$. That is, Gaussian elimination computes the *Doolittle decomposition* of a matrix as a product of a unit lower triangular matrix times an upper triangular matrix.

We may store the $a_{ij}^{(k)}$ over the initial a_{ij} and the multiplier m_{ik} over a_{ik} (since after m_{ik} is computed, a_{ik} is set equal to zero). This leads to:

FIGURE 3.1. The bold numbers indicate the order in which the equations are considered.



for $k = 1 : (n-1)$
  for $i = (k+1) : n$
    $a_{ik} \leftarrow a_{ik}/a_{kk}$
    for $j = (k+1) : n$
      $a_{ij} \leftarrow a_{ij} - a_{ik}a_{kj}$
    end
  end
end

Note that Gaussian elimination may break-down, namely one of the diagonal elements $a_{kk}^{(k)}$ may be zero. The most obvious situation in which this happens is when a_{11} happens to be zero. To investigate when this happens, let us consider a more direct algorithm for computing the Doolittle LU decomposition. Suppose $A = LU$ with L unit lower triangular, U upper triangular. Then

$$a_{ij} = \sum_{k=1}^{n} l_{ik}u_{kj} = \begin{cases} \sum_{k=1}^{j} l_{ik}u_{kj}, & j < i, \\ \sum_{k=1}^{i} l_{ik}u_{kj}, & j \geq i \end{cases}$$

Using a Matlab-like notation for rows and columns we may write:

$$\begin{aligned} a(1, 1:n) &= u(1, 1:n), \\ a(2:n, 1) &= l(2:n, 1)u(1, 1), \\ a(2, 2:n) &= l(2, 1)u(1, 2:n) + u(2, 2:n), \\ a(3:n, 2) &= l(3:n, 1)u(1, 2) + l(3:n, 2)u(2, 2), \\ &\vdots \end{aligned}$$

The first equation uniquely determines the first row of u. Supposing $u_{11} \neq 0$ the second equation then uniquely determines the first column of L. Then the third equation uniquely determines the second row of U. Supposing that $u_{22} \neq 0$, the next equation uniquely determines the second column of L, etc.

Now let $A^k = a(1:k, 1:k)$ denote the kth principal minor of A. Clearly $u_{11} \neq 0$ if and only if A^1 is nonsingular ($u_{11} = a_{11} = A^1$). In that case the second principle minors L^2 of L and U^2 of U are uniquely determined and $L^2U^2 = A^2$. Thus U^2 is nonsingular if and

only if A^2 is nonsingular and, since we know that $u_{11} \neq 0$, this holds if and only if u_{22} is nonsingular. Continuing in this way, we conclude

THEOREM 3.1. *An $n \times n$ nonsingular matrix A admits a decomposition LU with L unit lower triangular and U upper triangular if and only if all its principal minors are nonsingular.*

We have seen two algorithms for computing the Doolittle LU factorization. They involve basically the same operations in different order. In particular the number of additions and multiplications required are $n^3/3$ in each case. The classical Gaussian elimination is easily implemented with the inner loop consisting of row-oriented SAXPY operations. It is also straightforward to devise a column-oriented version.

If the nonsingular matrix admits a Doolittle decomposition, we may write the upper triangular part as DU with D diagonal and U unit upper triangular. Thus A has a unique decomposition as LDU with D diagonal and L and U unit lower and upper triangular, respectively. In addition to the Doolittle decomposition $L(DU)$, there is the Crout decomposition $(LD)U$ into a lower triangular matrix times a unit upper triangular. Algorithms for this may be constructed as for the Doolittle decomposition.

If A is symmetric, then in the LDU decomposition $U = L^T$. If, in addition, the elements of D are positive, then we can use the symmetric decomposition $LD^{1/2}D^{1/2}L^T$. Writing L for $LD^{1/2}$, so now L is a lower triangular matrix with positive diagonal elements, we have $A = LL^T$. This is the *Cholesky decomposition* of A. If A admits a Cholesky decomposition, it is SPD (symmetric positive-definite). Conversely, reasoning much as we did for the Doolittle decomposition, we can show that every SPD matrix admits a unique Cholesky decomposition (left as an exercise to the reader). Here is an algorithm. In view of the symmetry of A, it never refers to the elements of A lying above the diagonal:

for $k = 1 : n$
  for $i = 1 : (k-1)$
    $l_{ki} = (a_{ki} - \sum_{m=1}^{i-1} l_{im}l_{km})/l_{ii}$
  end
  $l_{kk} = \sqrt{a_{kk} - \sum_{m=1}^{k-1} l_{km}^2}$
end

We may overwrite the lower triangular part of A with L:

for $k = 1 : n$
  for $i = 1 : (k-1)$
    $a_{ki} \leftarrow (a_{ki} - \sum_{m=1}^{i-1} a_{im}a_{km})/a_{ii}$
  end
  $a_{kk} \leftarrow \sqrt{a_{kk} - \sum_{m=1}^{k-1} a_{km}^2}$
end

Note the Cholesky algorithm costs $n^3/6$ multiplications asymptotically.

4. Pivoting

Gaussian elimination breaks down at the first step if $a_{11} = 0$. In this case we may switch the first row with any row that has a nonzero element in the first column. Such a row must exist, if the matrix is nonsingular. Similarly, at some later stage of the procedure one of the updated diagonal elements a_{kk} (the so-called *pivots*) may vanish and the algorithm will break-down. Even if the pivot is not zero, it may be very small, and this increases the possibility there will be large round-off errors introduced. Indeed if a very small pivot is used, the corresponding multiplier will be very large, and if the solution is not large, there must be some cancellation occuring when we compute it. To understand this in more detail, let's consider the 2×2 case.

For the system

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Gaussian elimination reads:

1: $m \leftarrow a_{21}/a_{11}$
2: $a_{22} \leftarrow a_{22} - m \cdot a_{12}$
3: $b_2 \leftarrow b_2 - m \cdot b_1$
4: $x_2 \leftarrow b_2/a_{22}$
5: $x_1 \leftarrow (b_1 - a_{12} \cdot x_2)/a_{11}$

Precision can be lost, possibly, in steps 2, 3, and 5, since only these involve addition or subtraction.

Now consider the specific system

$$\begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

where ϵ is very small (near unit round-off). If $\epsilon = 0$, the solution is $x_1 = x_2 = 1$, and if ϵ is small, the solution is $x_1 = 1/(1-\epsilon) = 1 + O(\epsilon)$, $x_2 = (1-2\epsilon)/(1-\epsilon) = 1 + O(\epsilon)$. The multiplier m is ϵ^{-1}, so the subtractions in steps 2 and 3 are $1 - \epsilon^{-1}$ and $2 - \epsilon^{-1}$ which will not entail much loss of precision. However, the subtraction in step 5 is $1 - x_2$ with x_2 very near 1, and so will result in catastrophic cancellation. Hence we would expect that for ϵ very small x_2 will be computed accurately, but not x_1. In fact, a Fortran single precision program with $\epsilon = 10^{-6}$ gives $x_1 = 1.013279$, $x_2 = 0.999999$. The correct answers to 7 places are 1.000001 and 0.999999, and the relative errors are 1.3×10^{-2} for x_1, 1.3×10^{-8} for x_2.

If we consider this system with $\epsilon = 0$, the solution is obvious: interchange the rows. The same is advisable for ϵ small. In this case the multiplier is ϵ, and the subtractions are $1 - \epsilon$, $1 - 2\epsilon$, and $2 - x_2$ where x_2 is close to 1. Thus there are no serious cancellations.

In general, if facing a matrix in which the $(1,1)$ element is very small, a reasonable idea is to switch the first row with the row containing an element which is not small. A reasonable choice, which is usually made, is to switch with the row which has the largest element (in absolute value) in the first column. This results in multipliers which are no greater than 1, which prevents the creation of large magnitude element during this step of the elimination, which tends to decrease cancellation in later stages.

Similarly, before performing the second step of the elimination we switch the second row with whatever row below the first has the largest magnitude element in the second column. Etc. This procedure is known as partial pivoting. Actually, in practice one does not switch the rows, but rather changes the order of the rows used in elimination. That is, rather than using the order $1, 2, \ldots, n$, one uses some permutation $p_1, p_2, \ldots, p_n$, which is determined in the course of the algorithm (as described above). Thus the basic Gaussian elimination routine which reads

for $k = 1, 2, \ldots, n-1$
  for $i = k+1, k+2, \ldots, n$
    $a_{ik} \leftarrow a_{ik}/a_{kk}$
    $a_{ij} \leftarrow a_{ij} - a_{ik}a_{kj}, \quad j = k+1, k+2, \ldots, n$
  end for
end for

becomes

for $p_i = i$, $i = 1, 2, \ldots, n$
for $k = 1, 2, \ldots, n-1$
  choose $i \geq k$ and interchange p_k and p_i
  for $k = 1, 2, \ldots, n-1$
  for $i = k+1, k+2, \ldots, n$
    $m \leftarrow a_{p_i k}/a_{p_k k}$
    $a_{p_i j} \leftarrow a_{p_i j} - m a_{p_k j}, \quad j = k+1, k+2, \ldots, n$
  end for
end for

The values of the p_i are determined in the course of the algorithm (before they are used of course!). It can be seen that Gaussian elimination with partial pivoting is equivalent to factoring the matrix PA as LU where PA is the matrix obtained from A by switching the order of the rows from the outset. This is actually the matrix product PA where P is a permutation matrix, i.e., a matrix with the same columns as the identity matrix (but in a different order). For any nonsingular matrix, partial pivoting determines a permutation such that PA admits an LU decomposition. In other words, in exact arithmetic Gaussian elimination with partial pivoting always works. (In the next section we shall discuss the effect of partial pivoting on the propagation of round-off error when floating point arithmetic is used.)

There are also several situations in which it can be shown that pivoting is not necessary. The two most important cases are positive definite matrices and diagonally dominant matrices (matrices with $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$.

5. Backward error analysis

Now we consider the effects of floating point arithmetic on Gaussian elimination. That is we assume that every real number x is represented in the computer by a floating point number $\mathrm{fl}(x)$ satisfying $\mathrm{fl}(x) = x(1+\delta)$ for some δ bounded in magntitude by $\mathbf{u}$, the unit round-off error for the floating point number system under consideration. For 32 bit IEEE arithmetic $\mathbf{u} = 2^{-24} \approx 6 \times 10^{-8}$, and for 64 bit IEEE arithmetic $\mathbf{u} = 2^{-53} \approx 10^{-16}$. Moreover, if x and y are two floating point numbers (i.e., $x = \mathrm{fl}(x)$, $y = \mathrm{fl}(y)$, then we assume that the result of computing $x + y$, $x - y$, $x \times y$, and x/y on the computer is $\mathrm{fl}(x+y)$, $\mathrm{fl}(x-y)$, etc.

Now we consider the effect of computing the LU decomposition of an $n \times n$ matrix A via classical Gaussian elimination using floating point arithmetic at every step of the algorithm. For simplicity we assume that A is a matrix of floating point numbers (the approximation of the elements of A can be considered separately).

We use the following notation: if A is a matrix with entries a_{ij}, $|A|$ denotes the matrix whose (i, j) entry is $|a_{ij}|$. If A and B are matrices of the same size, then $A \leq B$ means $a_{ij} \leq b_{ij}$ for each i, j. Thus, for example, if A and B are matrices of floating point numbers of the same size and C is the computed sum of A and B using floating point arithmetic ($c_{ij} = \mathrm{fl}(a_{ij} + b_{ij})$), then $C = A + B + E$ where the matrix E satisfies $|E| \leq (|A| + |B|)\mathbf{u}$.

THEOREM 3.2. *Let A be an $n \times n$ matrix of floating point numbers and let $\hat{L}$ and $\hat{U}$ be computed by Gaussian elimination with floating point arithmetic assuming no zero pivots are encountered. Then*

$$\hat{L}\hat{U} = A + E,$$

where $|E| \leq 2(n-1)(|A| + |\hat{L}||\hat{U}|)\mathbf{u} + O(\mathbf{u}^2)$.

PROOF. Induction on n, the case $n = 1$ being obvious. Partition A as

$$A = \begin{pmatrix} a_{11} & u^T \\ v & B \end{pmatrix},$$

with $u, v \in \mathbb{R}^{n-1}$, $B \in \mathbb{R}^{(n-1)\times(n-1)}$. With exact arithmetic Gaussian elimination yields $A = LU$ with

$$L = \begin{pmatrix} 1 & 0 \\ l & L_1 \end{pmatrix}, \quad U = \begin{pmatrix} a_{11} & u^T \\ 0 & U_1 \end{pmatrix}.$$

Here $l \in \mathbb{R}^{n-1}$ is the given by $l = v/a_{11}$ and L_1 and U_1 are triangular matrices of size $n-1$, coming from Gaussian elimination applied to the matrix $A^{(1)} = B - lu^T$. Now, if we use floating point arithmetic we get instead

$$\hat{L} = \begin{pmatrix} 1 & 0 \\ \hat{l} & \hat{L}_1 \end{pmatrix}, \quad \hat{U} = \begin{pmatrix} a_{11} & u^T \\ 0 & \hat{U}_1 \end{pmatrix},$$

where $\hat{l} = \mathrm{fl}(a/a_{11})$ (the fl operator is applied componentwise), and $\hat{L}_1$ and $\hat{U}_1$ are obtained by Gaussian elimination with floating point arithmetic applied to $\hat{A}^{(1)} = \mathrm{fl}(B - \mathrm{fl}(\hat{l}u^T))$. Thus

$$\hat{L}\hat{U} = \begin{pmatrix} a_{11} & u^T \\ a_{11}\hat{l} & \hat{L}_1\hat{U}_1 + \hat{l}u^T \end{pmatrix} =: A + \begin{pmatrix} 0 & 0 \\ f & F \end{pmatrix}, \quad |\hat{L}||\hat{U}| = \begin{pmatrix} |a_{11}| & |u|^T \\ |a_{11}||\hat{l}| & |\hat{L}_1||\hat{U}_1| + |\hat{l}||u|^T \end{pmatrix}.$$

Thus we need to show that

$$|f| \leq 2(n-1)(|v| + |a_{11}||\hat{l}|)\mathbf{u} + O(\mathbf{u}^2), \tag{3.1}$$

$$|F| \leq 2(n-1)(|B| + |\hat{L}_1||\hat{U}_1| + |\hat{l}||u|^T)\mathbf{u} + O(\mathbf{u}^2). \tag{3.2}$$

Now

$$|f| = |a_{11}\hat{l} - a_{11}l| = |a_{11}||\hat{l} - l| \leq |a_{11}||l|\mathbf{u} = |v|\mathbf{u},$$

so (3.1) is trivially satisfied and the proof will be complete if we can establish (3.2).

Now

$$F = \hat{L}_1\hat{U}_1 + \hat{l}u^T - B = (\hat{L}_1\hat{U}_1 - \hat{A}^{(1)}) + (\hat{A}^{(1)} + \hat{l}u^T - B). \tag{3.3}$$

First consider the second term on the right-hand side. From the definition of $\hat{A}^{(1)}$, we have $\hat{A}^{(1)} = [B - (\hat{l}u^T + G)] + H$ with $|G| \le |\hat{l}||u|^T\mathbf{u}$, $|H| \le (|B| + |\hat{l}||u|^T)\mathbf{u} + O(\mathbf{u}^2)$, so

$$|\hat{A}^{(1)} + \hat{l}u^T - B| = |-G + H| \le 2(|B| + |\hat{l}||u|^T)\mathbf{u} + O(\mathbf{u}^2). \tag{3.4}$$

We also deduce from this estimate that

$$|\hat{A}^{(1)}| \le (1 + 2\mathbf{u})(|B| + |\hat{l}||u|^T) + O(\mathbf{u}^2). \tag{3.5}$$

Turning to the first term on the right-hand side of (3.3), we invoke the inductive hypothesis and then use (3.5) to get

$$\begin{aligned} |\hat{L}_1\hat{U}_1 - \hat{A}^{(1)}| &\le 2(n-2)(|\hat{A}^{(1)}| + |\hat{L}_1||\hat{U}_1|)\mathbf{u} + O(\mathbf{u}^2) \\ &\le 2(n-2)(|B| + |\hat{l}||u|^T + |\hat{L}_1||\hat{U}_1|)\mathbf{u} + O(\mathbf{u}^2). \end{aligned} \tag{3.6}$$

Combining (3.3), (3.4), and (3.6), we easily obtain (3.2), and so conclude the proof. □

It is important to interpret the result of this theorem. First of all, it is a backward error analysis. That is, we are not bounding the errors $L - \hat{L}$ and $U - \hat{U}$ (which is harder to do), but rather the residual $E = \hat{L}\hat{U} - A$. Second, in the bound derived for $|E|$, the term $2(n-1)$ should be considered of little importance. Since in practice $\mathbf{u}$ is many orders of magnitude smaller than $1/n$, we can think of this factor as contributing little. (In fact, at the expense of a fussier proof and less transparent final statement, we could sharpen this factor, but there is little point to doing so, just as there is little point in giving an explicit bound for the $O(\mathbf{u}^2)$ term although it would be possible to do.) So psychologically the bound on $|E|$ should be read as $O\big((|A| + |\hat{L}||\hat{U}|)\mathbf{u}\big)$. Now if the bound were simply $O(|A|\mathbf{u})$, this would be a very satisfactory result: it would say that $\hat{L}\hat{U}$ is the LU decomposition of a matrix whose elements approximate those of A with the same order of accuracy as the closest floating point approximation. Put in another way, it would say that the errors due to floating point arithmetic are of the same order as the errors that occur when the matrix (not assumed to have exact floating point entries) is rounded to machine numbers. Now if the matrices $\hat{L}$ and $\hat{U}$ are not much larger than A, then we still have a very satisfactory result. However, if we use Gaussian elimination without pivoting, we have seen that even in the 2×2 case, that small pivots may arise, and then it will usually happen that $\hat{L}$ and $\hat{U}$ (or just L and U) have much larger entries than A. In this case the bound suggests (correctly) that the error matrix E may be much larger than one would get by just approximating the original matrix. One nice aspect, is that one can monitor this potential problem. After computing the LU decomposition one can simply check whether $\hat{L}$ and $\hat{U}$ are very large. If this is not the case, the relative residual $\hat{L}\hat{U} - A$ will not be large.

A similar backward error analysis can be given for the solution to triangular systems. If we combine these results with the theorem above, we get a backward error bound for the solution of linear systems. A proof is given in Golub and Van Loan, *Matrix Computations*.

THEOREM 3.3. *Let A be an $n \times n$ matrix of floating point numbers and b an n-vector of floating point numbers. Let $\hat{L}$ and $\hat{U}$ be computed by Gaussian elimination with floating point arithmetic assuming no zero pivots are encountered and let $\hat{x}$ be computed by solving*

$\hat{L}\hat{U}\hat{x} = b$ using forward elimination and back substitution with floating point arithmetic. Then there exists a matrix E satisfying

$$|E| \leq 5n(|A| + |\hat{L}||\hat{U}|)\mathbf{u} + O(\mathbf{u}^2),$$

such that $(A + E)\hat{x} = b$.

Thus, if $\hat{L}$ and $\hat{U}$ are not very large, the computed solution is the exact solution to the problem where the matrix has been replaced by a nearby matrix. Again, if we don't pivot it may happen that $\hat{L}$ and/or $\hat{U}$ *is* very large.

Let us now consider what happens if we perform Gaussian elimination with pivoting. We may think of this as applying Gaussian elimination to a matrix which has been reordered so that at every step the pivot element exceeds in magnitude all elements below it in the column. In other words, the elements of the matrix L are all bounded by 1 in magnitude. The question then becomes, how much larger the eliminated matrix U can be than the original matrix A. In fact, it can be much larger. If we let

$$A = \begin{pmatrix} 1 & 0 & 0 & \dots & 1 \\ -1 & 1 & 0 & \dots & 1 \\ -1 & -1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \dots & 1 \end{pmatrix},$$

then it is easy to carry out the elimination by hand. All the multipliers are -1 (so partial pivoting won't occur) and the final matrix U is

$$U = \begin{pmatrix} 1 & 0 & 0 & \dots & 1 \\ 0 & 1 & 0 & \dots & 2 \\ 0 & 0 & 1 & \dots & 4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 2^{n-1} \end{pmatrix}.$$

Thus, although the largest magnitude of an element of A is 1, the eliminated matrix U has an element equal to 2^{n-1}.

Fortunately cases like the above are rare, and the received wisdom is that in practice Gaussian elimination with partial pivoting rarely fails.

For complete pivoting the worst possible growth of matrix size due to elimination is much smaller. Let us define the growth factor as $\max_{ij} |u_{ij}| / \max_{ij} |a_{ij}|$. (Actually the growth factor is generally defined as $g = \max_{i,j,k} |a_{ij}^{(k)}| / \max_{i,j} |a_{ij}|$ where the $a_{ij}^{(k)}$ are the elements of the intermediate matrices which come up during the elimination process, however for complete pivoting the maximum in the numerator is achieved by one of the pivots, and hence are present (on the diagonal) in the final matrix, so the two ratios are equal.) No one has proved a sharp bound on the growth factor for complete pivoting. The chief theoretical bound seems to be Wilkinson's from 1961: the ratio of the largest element of U to that of A with complete pivoting does not exceed $n^{1/2} \prod_{j=2}^{n} j^{1/[2(j-1)]}$. This gives a value of about 19 for $n = 10$; 570 for $n = 50$; 3,500 for $n = 100$; 9,000,000 for $n = 1000$; and 10^{17} for $n = 100,000$. However based on numerical experience Wilkinson conjectured that the ratio is actually bounded by n. This turned out to be false: a counterexample was given (by Nick

Gould, corrected by Alan Edelman) in 1991, with a matrix of size $n = 13$ with factor a bit above 13.02. Similarly a matrix of size 25 was computed with a growth factor of 32.99. In any case, as mentioned above, the received wisdom is that the growth due to partial pivoting is almost always acceptable in practice, and so the slower growth due to complete pivoting rarely justifies the extra expense. (Partial pivoting requires $n^2/2$ comparisons, and thus invokes a small cost compared to the $O(n^3)$ operations necessary to perform the elimination, but complete pivoting requires $n^3/3$ comparisons, which is not negligeable.)

6. Conditioning

Backward error analysis essentially assures us that if we use Gaussian elimination with pivoting to solve a linear system with floating point arithmetic, then the residual of the resulting solution will be small. It is important to realize that this does not imply that the error will be small.

Thus, suppose $Ax = b$ with A nonsingular. If we perturb b to $\tilde{b}$, this leads to a change in x to $\tilde{x}$ defined by $A\tilde{x} = \tilde{b}$. If the relative error

$$\frac{\|\tilde{b} - b\|}{\|b\|}$$

is small (measured in some vector norm), what about the relative error in x?

$$\frac{\|\tilde{x} - x\|}{\|x\|} = \frac{\|A^{-1}(\tilde{b} - b)\|}{\|b\|} \frac{\|Ax\|}{\|x\|} \leq \kappa(A) \frac{\|\tilde{b} - b\|}{\|b\|}$$

where

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is the condition number of A with respect to the associated matrix norm. (Given any norm on $\mathbb{R}^n$, there is an associated norm on $\mathbb{R}^{n\times n}$ defined by $\|A\| = \sup_{0\neq x\in\mathbb{R}^n} \|Ax\|/\|x\|$. Such a norm is sometimes called an *operator norm*, since it is the natural norm in the space of linear operators from the normed vector space $\mathbb{R}^n$ to itself. The associated operator matrix norm is *compatible* with the given vector norm in the sense that $\|Ax\| \leq \|A\|\|x\|$.)

Example (due to R. Wilson):

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \qquad b = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}, \qquad x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

$$\tilde{b} = \begin{pmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{pmatrix}, \qquad \tilde{x} = \begin{pmatrix} 9.2 \\ -12.6 \\ 4.5 \\ -1.1 \end{pmatrix}.$$

$$\frac{\|\tilde{b} - b\|}{\|b\|} = \frac{.4}{119} = .0034, \qquad \frac{\|\tilde{x} - x\|_1}{\|x\|_1} = \frac{27.4}{4} = 6.85,$$

$$A^{-1} = \begin{pmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{pmatrix}, \qquad \kappa_1(A) = 4,488.$$

The condition number also measures the sensitivity of the solution to perturbations in A. Suppose that $Ax = b$ with A nonsingular, and let E be a matrix such that $\|A^{-1}E\| < 1$. Then $A + E$ is nonsingular and

$$\|(A+E)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}E\|}.$$

Let $\tilde{x}$ be the solution of the perturbed system: $(A+E)\tilde{x} = b$. Then $x - \tilde{x} = A^{-1}E\tilde{x}$, so

$$\|x - \tilde{x}\| \leq \|A^{-1}\|\|E\|\|\tilde{x}\|,$$

or

$$\frac{\|x - \tilde{x}\|}{\|\tilde{x}\|} \leq \kappa(A)\frac{\|E\|}{\|A\|}.$$

Finally, we may use the condition number to get an a posteriori bound on the error in the solution. If we have somehow computed an approximate solution $\tilde{x}$ to $Ax = b$, although we don't know $\|x - \tilde{x}\|$, we can always compute the residual $A\tilde{x} - b$. Now $x - \tilde{x} = A^{-1}(b - A\tilde{x})$, so $\|x - \tilde{x}\| \leq \|A^{-1}\|\|A\tilde{x} - b\|$. We also clearly have $\|x\| \geq \|b\|/\|A\|$. If we divide these two inequalities we get

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \kappa(A)\frac{\|A\tilde{x} - b\|}{\|b\|}.$$

Thus the condition number relates the relative error in the computed solution to the relative residual it engenders.

Although the condition number is not easy to compute exactly, it can be cheaply (if not always accurately) estimated from the LU decomposition. There is no way to accurately solve a system of linear equations when the matrix is ill-conditioned, and one should always check the condition number.

Exercises

(1) Prove that if A is symmetric positive definite, then $A = LL^T$ for a unique lower triangular matrix L. State an algorithm for computing L, and show that it does not break down (no divisions by zero or square-roots of negative numbers).

(2) Write a column oriented algorithm to compute the Doolittle LU decomposition. That is, given a matrix A with nonsingular principal minors, your algorithm should overwrite the elements on or above the diagonal of A with the corresponding elements of an upper triangular matrix U and overwrite the below-diagonal elements of A with the corresponding elements of a unit lower triangular matrix L such that $LU = A$, and your algorithm should access the elements of A by columns. In addition to writing the algorithm, submit a direct Matlab translation along with a verification that it works using a random 4×4 matrix. Discuss the implementation of your algorithm using BLAS.

CHAPTER 4

Numerical solution of nonlinear systems and optimization

1. Introduction and Preliminaries

In this chapter we consider the solution of systems of n nonlinear equations in n unknowns. That is, with Ω an open subset of $\mathbb{R}^n$ and $F : \Omega \to \mathbb{R}^n$ a continuous function we wish to find $x_* \in \Omega$ such that $F(x_*) = 0$.

For nonlinear systems there is rarely a direct method of solution (an algorithm which terminates at the exact solution), so we must use iterative methods which produce a sequence of approximate solutions $x_0, x_1, \ldots$ in Ω for which, hopefully, $\lim x_i$ exists and equals a root x_* of F.

First some definitions relating to the speed of convergence of sequences in $\mathbb{R}^n$. Let x_i be a sequence in $\mathbb{R}^n$ which converges to 0. For $p > 1$ we say that the sequence converges to 0 with order p if there exists a constant C and a number N so that $\|x_{i+1}\| \leq C\|x_i\|^p$ for all $i \geq N$. This definition doesn't depend on the particular norm: if a sequence converges to 0 with order p in one norm, it converges with order p in all norms. Of course we extend this definition to sequences that converge to an arbitrary x_* by saying that x_i converges to x_* with order p if and only if $x_i - x_*$ converges to 0 with order p.

For $p = 1$ it is common to use the same definition except with the requirement that the constant be less than unity: a sequence would then be said to converge linearly to 0, if there exists $r < 1$ and N such that $\|x_{i+1}\| \leq r\|x_i\|$ for all $i \geq N$. However, this notion *is* norm-dependent. According to this definition, the sequence in $\mathbb{R}^2$

$$(1,1), (1,0), (1/4,1/4), (1/4,0), (1/16,1/16), (1/16,0), \ldots$$

converges linearly to 0 in the 1-norm, but does not converge linearly to 0 with respect to the ∞-norm. To avoid the norm-dependence, we note that the above definition implies that there exists a constant C such that $\|x_i\| \leq Cr^i$ for all i. (Proof: $\|x_i\| \leq r^{i-N}\|x_N\|$ for all $i \geq N$. Equivalently, $\|x_i\| \leq C_0 r^i$ for $i \geq N$ where $C_0 = r^{-N}\|x_N\|$. Setting $C = \max(C_0, \max_{0\leq i<N}\|x_i\|/r^i)$, we obtain the result.) We take this inequality as our definition of linear convergence: x_i converges to 0 linearly if there exists a constant C and a number $r < 1$ such that $\|x_i\| \leq Cr^i$. This notion is independent of norm (if it holds for one norm, then it holds for another with the same value of r, but possibly a different value of C). Note also that if this definition of linear convergence holds for some $r < 1$, then it also holds for all larger r. The infimum of all such r is called the *rate* of the linear convergence. If the infimum is 0, we speak of *superlinear* convergence.

Note that if $1 < p_1 < p_2$ and $0 < r_1 < r_2 < 1$, then

$$\text{convergence with order } p_2 \implies \text{convergence with order } p_1 \implies \text{superlinear convergence}$$
$$\implies \text{linear convergence with rate } r_1 \implies \text{linear convergence with rate } r_2$$

2. One-point iteration

For many iterative methods, x_{i+1} depends only on x_i via some formula that doesn't depend on i: $x_{i+1} = G(x_i)$. Such a method is called a (stationary) one-point iteration. Before considering specific iterations to solve $F(x) = 0$, we consider one-point iterations in general.

Assuming the iteration function G is continuous, we obviously have that if the iterates $x_{i+1} = G(x_i)$ converge to some limit x_*, then $x_* = G(x_*)$, i.e., x_* is a *fixed point* of G.

A basic result is the contraction mapping theorem. Recall that a map $G : B \to \mathbb{R}^n$ ($B \subset \mathbb{R}^n$) is called a contraction (with respect to some norm on $\mathbb{R}^n$) if G is Lipschitz with Lipschitz constant strictly less than 1.

THEOREM 4.1. *Suppose G maps a closed subset B of $\mathbb{R}^n$ to itself, and suppose that G is a contraction (with respect to some norm). Then G has a unique fixed point x_* in B. Moreover, if $x_0 \in B$ is any point, then the iteration $x_{i+1} = G(x_i)$ converges to x_*.*

If $G \in C^1$ a practical way to check whether G is a contraction (with respect to some norm on $\mathbb{R}^n$) is to consider $\|G'(x)\|$ (in the associated matrix norm). If $\|G'(x)\| \le \lambda < 1$ on some convex set Ω (e.g., some ball), then G is a contraction there. In one dimension this is an immediate consequence of the mean value theorem. In n dimensions we don't have the mean value theorem, but we can use the fundamental theorem of calculus to the same end. Given $x, y \in \Omega$ we let $g(t) = G(x + t(y - x))$, so $g'(t) = G'(x + t(y - x))(y - x)$. From the fundamental theorem of calculus we get $g(1) - g(0) = \int_0^1 g'(t)\,dt$, or

$$G(y) - G(x) = \left[\int_0^1 G'(x + t(y - x))\,dt\right](y - x),$$

whence

$$\|G(y) - G(x)\| \le \sup_{0\le t\le 1} \|G'(x + t(y - x))\|\|y - x\| \le \lambda\|y - x\|,$$

and so G is a contraction.

If we assume that x_* is a fixed point of G, $G \in C^1$, and $r = \|G'(x_*)\| < 1$, then we can conclude that the iteration $x_{i+1} = G(x_i)$ converges for any starting iterate x_0 sufficiently close to x_*. This is called a *locally convergent* iteration. The above argument also shows that convergence is (at least) linear with rate r.

In this connection, the following theorem, which connects $\|A\|$ to $\rho(A)$ (the spectral radius of A, i.e., the maximum modulus of its eigenvalues), is very useful.

THEOREM 4.2. *Let $A \in \mathbb{R}^{n\times n}$. Then*

1. *For any operator matrix norm, $\|A\| \ge \rho(A)$.*
2. *If A is symmetric, then $\|A\|_2 = \rho(A)$.*
3. *If A is diagonalizable, then there exists an operator norm so that $\|A\| = \rho(A)$.*
4. *For any A and any $\epsilon > 0$, there exists an operator norm so that $\rho(A) \le \|A\| \le \rho(A) + \epsilon$.*

PROOF. 1. If $Ax = \lambda x$ where $x \ne 0$ and $|\lambda| = \rho(A)$, then from $\|Ax\| = |\lambda|\|x\|$ we see that $\|A\| \ge \rho(A)$.

2. $\|A\|_2 = \sqrt{\rho(A^T A)} = \sqrt{\rho(A^2)} = \rho(A)$.

3. First note that if $S \in \mathbb{R}^{n\times n}$ is nonsingular and $\|\cdot\|_0$ any vector norm, then $\|x\| := \|Sx\|_0$ is another vector norm, and the associated matrix norms satisfy $\|A\| = \|SAS^{-1}\|_0$. Now if A is diagonalizable, then there exists S nonsingular so that SAS^{-1} is a diagonal matrix with the eigenvalues of A on the diagonal (the columns of S^{-1} are the eigenvectors of A). Hence if we apply the above relation beginning with the ∞-norm for $\|\cdot\|_0$, we get $\|A\| = \rho(A)$.

4. The proof is similar in this case, but we use the Jordan canonical form to write $SAS^{-1} = J$ where J has the eigenvalues of A on the diagonal, 0's and ϵ's above the diagonal, and 0's everywhere else. (The usual Jordan canonical form is the case $\epsilon = 1$, but if we conjugate a Jordan block by the matrix $\mathrm{diag}(1, \epsilon, \epsilon^2, \ldots)$ the 1's above the diagonal are changed to ϵ.) Thus for the matrix norm associated to $\|x\| := \|Sx\|_\infty$, we have $\|A\| = \|J\|_\infty \le \rho(A) + \epsilon$. □

COROLLARY 4.3. *If G is C^1 in a neighborhood of a fixed point x_* and $r = \rho(G'(x_*)) < 1$, the one point iteration with iteration function G is locally convergent to x_* with rate r.*

Although we don't need immediately it, we note another useful corollary of the proceeding theorem.

COROLLARY 4.4. *Let $A \in \mathbb{R}^{n\times n}$. Then $\lim_{n\to\infty} A^n = 0$ if and only if $\rho(A) < 1$, and in this case the convergence is linear with rate $\rho(A)$.*

PROOF. $\|A^n\| \ge \rho(A^n) = \rho(A)^n$, so if $\rho(A) \ge 1$, then A^n does not converge to 0.

Conversely, if $\rho(A) < 1$, then for any $\bar\rho \in (\rho(A), 1)$ we can find an operator norm so that $\|A\| \le \bar\rho$, and then $\|A^n\| \le \|A\|^n = \bar\rho^n \to 0$. □

Finally, let us consider the case $G'(x_*) = 0$. Then clearly the iteration is superlinearly convergent. If G is C^2, or, less, if G' is Lipschitz, then we can show that the convergence is in fact quadratic. First note that for any C^1 function G,

$$G(y) - G(x) - G'(x)(y-x) = \int_0^1 [G'(x + t(y-x)) - G'(x)]\, dt(y-x).$$

Hence, if G' is Lipschitz,

$$\|G(y) - G(x) - G'(x)(y-x)\| \le \frac{C}{2}\|y-x\|^2,$$

where C is the Lipschitz constant. Applying this with $x = x_*$ and $y = x_i$ and using the fact that $G(x_*) = x_*$, $G'(x_*) = 0$, we get

$$\|x_{i+1} - x_*\| \le \frac{C}{2}\|x_i - x_*\|^2,$$

which is quadratic convergence. In the same way we can treat the case of G with several vanishing derivatives.

THEOREM 4.5. *Suppose that G maps a neighborhood of x_* in $\mathbb{R}^n$ into $\mathbb{R}^n$ and that x_* is a fixed point of G. Suppose also that all the derivatives of G of order up to p exist, are Lipschitz continuous, and vanish at x_*. Then the iteration $x_{i+1} = G(x_i)$ is locally convergent to x_* with order $p+1$.*

3. Newton's method

An important example of a one-point iteration is Newton's method for root-finding. Let $F : \Omega \to \mathbb{R}^n$ be C^1 with $\Omega \subset \mathbb{R}^n$. We wish to find a root x_* of F in Ω. If $x_0 \in \Omega$ is an initial guess of the root, we approximate F by the linear part of its Taylor series near x_0:

$$F(x) \approx F(x_0) + F'(x_0)(x - x_0).$$

The left-hand side vanishes when x is a root, so setting the right-hand side equal to zero gives us an equation for a new approximate root, which we take to be x_1. Thus x_1 is determined by the equation

$$F(x_0) + F'(x_1)(x_1 - x_0) = 0,$$

or, equivalently,

$$x_1 = x_0 - F'(x_0)^{-1}F(x_0).$$

Continuing in this way we get Newton's method:

$$x_{i+1} = x_i - F'(x_i)^{-1}F(x_i), \quad i = 0, 1, \ldots.$$

(Of course it could happen that some $x_i \notin \Omega$ or that some $F'(x_i)$ is singular, in which case Newton's method breaks down. We shall see that under appropriate conditions this doesn't occur.)

Note that Newton's method is simply iteration of the function

$$G(x) = x - F'(x)^{-1}F(x).$$

Now if x_* is a root of F and $F'(x_*)$ is nonsingular (i.e., if x_* is a simple root), then G is continuous in a neighborhood of x_*, and clearly x_* is a fixed point of G. We have that

$$G'(x) = I - K(x)F(x) - F'(x)^{-1}F'(x) = -K(x)F(x)$$

where K is the derivative of the function $x \mapsto F'(x)^{-1}$ (this function maps a neighborhood of x_* in $\mathbb{R}^n$ into $\mathbb{R}^{n\times n}$). It is an easy (and worthwhile) exercise to derive the formula for $K(x)$ in terms of $F'(x)$ and $F''(x)$, but we don't need it here. It suffices to note that K exists and is Lipschitz continuous if F' and F'' are. In any case, we have that $G'(x_*) = 0$. Thus, assuming that F is C^2 with F'' Lipschitz (e.g., if F is C^3), we have all the hypotheses necessary for local quadratic convergence. Thus we have proved:

THEOREM 4.6. *Suppose that $F : \Omega \to \mathbb{R}^n$, $\Omega \subset \mathbb{R}^n$ is C^2 with F'' Lipschitz continuous, and that $F(x_*) = 0$ and $F'(x_*)$ is nonsingular for some $x_* \in \Omega$. Then if $x_0 \in \Omega$ is sufficiently close to x_*, the sequence of points defined by Newton's method is well-defined and converges quadratically to x_*.*

The hypothesis that the root be simple is necessary for the quadratic convergence of Newton's method, as can easily be seen by a 1-dimensional example. However, the smoothness assumption can be weakened. The following theorem requires only that F' (rather than F'') be Lipschitz continuous (which holds if F is C^2). In the statement of the theorem any vector norm and corresponding operator matrix norm can be used.

THEOREM 4.7. *Suppose that $F(x_*) = 0$ and that F' is Lipschitz continuous with Lipschitz constant γ in a ball of radius r around x_*. Also suppose that $F'(x_*)$ is nonsingular with*

$\|F'(x_*)^{-1}\| \leq \beta$. If $\|x_0 - x_*\| \leq \min[r, 1/(2\beta\gamma)]$, then the sequence determined by Newton's method is well-defined, converges to x_*, and satisfies

$$\|x_{i+1} - x_*\| \leq \beta\gamma\|x_i - x_*\|^2.$$

PROOF. First we show that $F'(x_0)$ is nonsingular. Indeed,

$$\|F'(x_*)^{-1}[F'(x_0) - F'(x_*)]\| \leq \beta\gamma\|x_0 - x_*\| \leq 1/2,$$

from which follows the nonsingularity and the estimate

$$\|F'(x_0)^{-1}\| \leq \|F'(x_*)^{-1}\|\frac{1}{1-1/2} \leq 2\beta.$$

Thus x_1 is well-defined and

$$\begin{aligned} x_1 - x_* &= x_0 - x_* - F'(x_0)^{-1}F(x_0) \\ &= x_0 - x_* - F'(x_0)^{-1}[F(x_0) - F(x_*)] \\ &= F'(x_0)^{-1}[F(x_*) - F(x_0) - F'(x_0)(x_* - x_0)]. \end{aligned}$$

We have previously bounded the norm of the bracketed quantity by $\gamma\|x_* - x_0\|^2/2$ and $\|F'(x_0)^{-1}\| \leq 2\beta$, so

$$\|x_1 - x_*\| \leq \beta\gamma\|x_0 - x_*\|^2.$$

This is the kind of quadratic bound we need, but first we need to show that the x_i are indeed converging to x_*. Using again that $\|x_0 - x_*\| \leq 1/(2\beta\gamma)$, we have the linear estimate

$$\|x_1 - x_*\| \leq \|x_* - x_0\|/2.$$

Thus x_1 also satisfies, $\|x_1 - x_*\| \leq \min[r, 1/(2\beta\gamma)]$, and the identical argument shows

$$\|x_2 - x_*\| \leq \beta\gamma\|x_1 - x_*\|^2, \text{ and } \|x_2 - x_*\| \leq \|x_1 - x_*\|/2.$$

Continuing in this way we get the theorem. □

The theorem gives a precise sufficient condition on how close the initial iterate x_0 must be to x_* to insure convergence. Of course it is not a condition that one can apply practically, since one cannot check if x_0 satisfies it without knowing x_0. There are several variant results which weaken the hypotheses necessary to show quadratic convergence for Newton's method. A well-known, but rather complicated one is Kantorovich's theorem (1948). Unlike the above theorems, it does not assume the existence of a root x_* of F, but rather states that if an initial point x_0 satisfies certain conditions, then there is a root, and Newton's method will converge to it. Basically it states: if F' is Lipschitz near x_0 and nonsingular at x_0, and if the value of $F(x_0)$ is sufficiently small (how small depending on the Lipschitz constant for F', and the norm of $F'(x_0)^{-1}$), then Newton's method beginning from x_0 is well-defined and converges quadratically to a root x_*. The exact statement is rather complicated, so I'll omit it. In principle, one could pick a starting iterate x_0, and then compute the norms of $F(x_0)$ and $F'(x_0)$, and check to see if they fulfil the conditions of Kantorovich's theorem (if one knew a bound for the Lipschitz constant of F' in a neighborhood of x_0), and thus tell in advance whether Newton's method would converge. In practice this is difficult to do and would rule out many acceptable choices of initial guess, so it is rarely used.

4. Quasi-Newton methods

Each iteration of Newton's method requires the following operations: evaluate the function F at the current approximation, evaluate the derivative F' at the current approximation, solve a system of equations with the latter as matrix and the former as right-hand side, and update the approximation. The evaluation of F' and the linear solve are often the most expensive parts. In some applications, no formula for F' is available, and exact evaluation of F' is not possible. There are many variations of Newton's method that attempt to maintain good local convergence properties while avoiding the evaluation of F', and/or simplifying the linear solve step. We shall refer to all of these as quasi-Newton methods, although some authors restrict that term to specific types of modification to Newton's method.

Consider the iteration

$$x_{i+1} = x_i - B_i^{-1} F(x_i),$$

where for each i, B_i is a nonsingular matrix to specified. If $B_i = F'(x_i)$, this is Newton's method. The following theorem states that if B_i is sufficiently close to $F'(x_i)$ then this method is still locally convergent. With a stronger hypothesis on the closeness of B_i to $F'(x_i)$ the convergence is quadratic. Under a somewhat weaker hypothesis, the method still converges superlinearly.

THEOREM 4.8. *Suppose F' is Lipschitz continuous in a neighborhood of a root x_* and that $F'(x_*)$ is nonsingular.*

1. *Then there exists $\delta > 0$ such that if $\|B_i - F'(x_i)\| \le \delta$ and $\|x_0 - x_*\| \le \delta$, then the generalized Newton iterates are well-defined by the above formula, and converge to x_*.*
2. *If further $\|B_i - F'(x_i)\| \to 0$, then the convergence is superlinear.*
3. *If there is a constant c such that $\|B_i - F'(x_i)\| \le c\|F(x_i)\|$, then the convergence is quadratic.*

PROOF. Set $\beta = \|F'(x_*)^{-1}\| < \infty$. Choosing δ small enough, we can easily achieve

$$\|x - x_*\| \le \delta,\ \|B - F'(x)\| \le \delta \implies \|B^{-1}\| \le 2\beta.$$

Let γ be a Lipschitz constant for F'. Decreasing δ if necessary we can further achieve $2\beta(\gamma/2 + 1)\delta \le 1/2$.

Now let x_0 and B_0 be chosen in accordance with this δ. Then

$$\begin{aligned} x_1 - x_* &= x_0 - x_* - B_0^{-1} F(x_0) = B_0^{-1}[F(x_*) - F(x_0) - B_0(x_* - x_0)] \\ &= B_0^{-1} \int_0^1 [F'((1-t)x_0 + tx_*) - B_0]dt\,(x_* - x_0). \end{aligned}$$

Now $\|F'((1-t)x_0 + tx_*) - B_0\| \le \gamma t\|x_0 - x_*\| + \delta$, by the triangle inequality, the Lipschitz condition, and the condition on B_0. Thus

$$\|x_1 - x_*\| \le 2\beta(\gamma\|x_0 - x_*\|/2 + \delta)\|x_0 - x_*\| \le 2\beta(\gamma/2 + 1)\delta\|x_0 - x_*\| \le \|x_0 - x_*\|/2.$$

In particular $\|x_1 - x_*\| \le \delta$, so this process may be repeated. (1) follows easily. Note that we obtained linear convergence with rate $1/2$, but by choosing δ sufficiently small we could obtain the linear convergence with any desired rate $r \in (0, 1)$.

From the above reasoning we get

$$\|x_{i+1} - x_*\| \leq 2\beta(\gamma\|x_i - x_*\|/2 + \|B_i - F'(x_i)\|)\|x_i - x_*\|,$$

which gives superlinearity under the additional hypothesis of (2). From the additional hypothesis of (3), we get

$$\|B_i - F'(x_i)\| \leq c\|F(x_i) - F(x_*)\| \leq c'\|x_i - x_*\|,$$

which gives the quadratic convergence. □

Some examples of quasi-Newton methods:

- Replace $\partial F^i(x)/\partial x_j$ by $[F^i(x + he_j) - F^i(x)]/h$ for small h or by a similar difference quotient. From the theorem, convergence is guaranteed if h is small enough.
- Use a single evaluation of F' for several iterations. (Then one can factor F' one time, and back solve for the other iterations.) Other methods, including Broyden's method which we consider next, use some sort of procedure to "update" a previous Jacobian.
- Use $B_i = \theta_i^{-1}F'(x_i)$, where θ_i is a *relaxation* parameter. Generally θ_i is chosen in $(0, 1]$, so that $x_{i+1} = x_i - \theta_i F'(x_i)^{-1}F(x_i)$ is a more "conservative" step than a true Newton step. This is used to stabilize Newton iterations when not sufficiently near the root. From the theorem, convergence is guaranteed for θ_i sufficiently near 1, and is superlinear if $\theta_i \to 1$. If $|\theta_i - 1| \leq c\|F(x_i)\|$ for all i sufficiently large, convergence is quadratic.
- Another possibility, which we shall study when we consider the minimization methods, is $B_i = \theta_i F'(x_i) + (1 - \theta_i)I$. Convergence statements similar to those for the relaxed method hold.

5. Broyden's method

Broyden's method (published by C. G. Broyden in 1965) is an important example of a quasi-Newton method. It is one possible generalization to n-dimensions of the secant method. For a single nonlinear equation, the secant method replaces $f'(x_i)$ in Newton's method with the approximation $[f(x_i) - f(x_{i-1})]/(x_i - x_{i-1})$, to obtain the iteration

$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i).$$

Of course, we cannot directly generalize this idea to $\mathbb{R}^n$, since we can't divide by the vector $x_i - x_{i-1}$. Instead, we can consider the equation

$$B_i(x_i - x_{i-1}) = F(x_i) - F(x_{i-1}).$$

However, this does not determine the matrix B_i, only its action on multiples of $x_i - x_{i-1}$. To complete the specification of B_i, Broyden's method sets the action on vectors orthogonal to $x_i - x_{i-1}$ to be the same as B_{i-1}. Broyden's method is an *update method* in the sense that B_i is determined as a modification of B_{i-1}.

In order to implement Broyden's method, we note:

THEOREM 4.9. *Given vectors $s \neq 0$, v in $\mathbb{R}^n$, $C \in \mathbb{R}^{n\times n}$, there is a unique matrix $B \in \mathbb{R}^{n\times n}$ such that*

$$Bs = v,$$
$$Bz = Cz, \text{ for all } z \text{ such that } s^T z = 0.$$

To see this, we note that there is certainly at most one such B. To see that such a B exists, we give the formula:

$$B = C + \frac{1}{s^T s}(v - Cs)s^T.$$

It is important to note that B is derived from C by the addition of a matrix of rank 1. In a certain sense C is the closest matrix to B which takes s to v (see the exercises).

We are now ready to give Broyden's method:

Choose $x_0 \in \mathbb{R}^n$, $B_0 \in \mathbb{R}^{n\times n}$
for $i = 0, 1, \ldots$
  $x_{i+1} = x_i - B_i^{-1}F(x_i)$
  $s_i = x_{i+1} - x_i$
  $v_i = F(x_{i+1}) - F(x_i)$
  $B_{i+1} = B_i + \dfrac{1}{s_i^T s_i}(v_i - B_i s_i)s_i^T$
end

REMARK. For example, B_0 can be taken to be $F'(x_0)$. In one dimension, Broyden's method reduces to the secant method.

Key to the effectiveness of Broyden's method is that the matrix B_{i+1} differs from B_i only by a matrix of rank 1. But, as shown in the following theorem, once one can compute the action of the inverse of a matrix on a vector efficiently (e.g., by by forward and back substitution once the matrix has been factored into triangular matrices), then once can compute the action of the inverse of any rank 1 perturbation of the matrix.

THEOREM 4.10 (Sherman-Morrison-Woodbury formula). *Let $B \in \mathbb{R}^{n\times n}$, $y, v \in \mathbb{R}^n$, and suppose that both B and $\tilde{B} := B + vy^T$ are nonsingular. Then $1 + y^T B^{-1} v \neq 0$ and*

$$\tilde{B}^{-1} = B^{-1} - \frac{1}{1 + y^T B^{-1} v} B^{-1} v y^T B^{-1}.$$

PROOF. Given any $u \in \mathbb{R}^n$, let $x = \tilde{B}^{-1}u$, so

$$Bx + (y^T x)v = u. \tag{4.1}$$

Multiplying on the left by $y^T B^{-1}$ gives $(y^T x)(1 + y^T B^{-1} v) = y^T B^{-1} u$. In particular, if we take $u = By$, then the right-hand side is $y^T y$, and so $1 + y^T B^{-1} v \neq 0$ and we obtain

$$y^T x = \frac{y^T B^{-1} u}{1 + y^T B^{-1} v}.$$

Combining this expression and (4.1) we see that

$$Bx = u - \frac{y^T B^{-1} u}{1 + y^T B^{-1} v} v.$$

Multiplying by B^{-1} and recalling that $x = \tilde{B}^{-1} u$ we obtain the Sherman–Morrison–Woodbury formula. □

Thus to compute the action of $\tilde{B}^{-1}$ on a vector, we just need to know the action of B^{-1} on that vector and on v and y, and to compute some inner products and simple expressions.

A good way to implement this formula for Broyden's method is to store $H_i := B_i^{-1}$ rather than B_i. The algorithm then becomes:

Choose $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n \times n}$
for $i = 0, 1, \ldots$
  $x_{i+1} = x_i - H_i F(x_i)$
  $s_i = x_{i+1} - x_i$
  $v_i = F(x_{i+1}) - F(x_i)$
  $H_{i+1} = H_i + \dfrac{1}{s_i^T H_i v_i}(s_i - H_i v_i) s_i^T H_i$
end

Note that if H_0 is B_0^{-1} this algorithm is mathematically equivalent to the basic Broyden algorithm.

5.1. Convergence of Broyden's method. Denote by x_* the solution of $F(x_*) = 0$, and let x_i and B_i denote the sequences of vectors and matrices produced by Broyden's method. Set

$$e_i = x_i - x_*, \quad M_i = B_i - F'(x_*).$$

Roughly speaking, the key to the convergence of Broyden's method are the facts that (1) e_{i+1} will be small compared to e_i if M_i is not large, and (2) M_{i+1} will not be much larger than M_i if the e_i's are small. Precise results will be based on the following identities, which follow directly from the definitions of x_i and B_i,

$$e_{i+1} = -B_i^{-1}[F(x_i) - F(x_*) - F'(x_*)(x_i - x_*)] + B_i^{-1} M_i e_i, \tag{4.2}$$

$$M_{i+1} = M_i \left(I - \frac{1}{s_i^T s_i} s_i s_i^T \right) + \frac{1}{s_i^T s_i}(v_i - F'(x_*) s_i) s_i^T. \tag{4.3}$$

Our first result gives the local convergence of Broyden's method, with a rate of convergence that is at least linear. The norms are all the 2-norm.

THEOREM 4.11. *Let F be differentiable in a ball Ω about a root $x_* \in \mathbb{R}^n$ whose derivative has Lipschitz constant γ on the ball. Suppose that $F'(x_*)$ is invertible, with $\|F'(x_*)^{-1}\| \le \beta$. Let $x_0 \in \Omega$ and $B_0 \in \mathbb{R}^{n \times n}$ be given satisfying*

$$\|M_0\| + 2\gamma \|e_0\| \le \frac{1}{8\beta}.$$

Then the iterates x_i, B_i given by Broyden's method are well defined, and the errors satisfy $\|e_{i+1}\| \le \|e_i\|/2$, for $i = 0, 1, \cdots$.

PROOF. Claim 1: If x_i and B_i are well-defined and $\|M_i\| \le 1/(2\beta)$, then B_i is invertible (so x_{i+1} is well-defined), and

$$\|B_i^{-1}\| \le 2\beta, \qquad \|e_{i+1}\| \le (\gamma\beta\|e_i\| + 2\beta\|M_i\|)\|e_i\|.$$

Indeed, $F'(x_*)^{-1}B_i = I + F'(x_*)^{-1}M_i$, and since $\|M_i\| \le 1/(2\beta)$, $\|F'(x_*)^{-1}M_i\| \le 1/2$, so B_i is invertible with $\|B_i^{-1}\| \le 2\beta$. Therefore, x_{i+1} is well-defined. The estimate on $\|e_{i+1}\|$ follows easily from the bound on B_i^{-1} and (4.2).

Note that from claim 1 and the hypotheses of the theorem we know that x_1 is well-defined and $\|e_1\| \le \|e_0\|/2$.

Claim 2: If $B_0, \dots, B_i$ are defined and invertible, then

$$\|M_{i+1}\| \le \|M_i\| + \gamma \max(\|e_i\|, \|e_{i+1}\|).$$

To prove this, we use (4.3). The first term on the right-hand side is the product of M_i with the orthogonal projection onto the orthogonal complement of s_i, so its 2-norm is bounded by $\|M_i\|$. For the second term, note that

$$v_i - F'(x_*)s_i = \int_0^1 [F'((1-t)x_{i+1} + tx_i) - F'(x_*)]\,dt\, s_i.$$

Since $\|F'(1-t)x_{i+1} + tx_i) - F'(x_*)\| \le \gamma \max(\|e_i\|, \|e_{i+1}\|)$,

$$\|v_i - F'(x_*)s_i\| \le \gamma \max(\|e_i\|, \|e_{i+1}\|)\|s_i\|,$$

and the second term on the right-hand side of (4.3) is bounded in norm by

$$\gamma \max(\|e_i\|, \|e_{i+1}\|),$$

which establishes the claim.

We are now ready to prove the theorem. We shall show, by induction on i, that $x_0, \dots, x_{i+1}$ are well-defined and

$$\|e_i\| \le \frac{1}{8\gamma\beta}, \quad \|M_i\| \le \frac{1}{8\beta}, \quad \|e_{i+1}\| \le \|e_i\|/2.$$

This is clearly true for $i = 0$. Assuming it true for i and all smaller indices, we immediately get the first inequality with i replaced by $i+1$. Using claim 2 repeatedly (and noting that $\|e_{i+1}\| \le \|e_i\| \le \cdots$, we have

$$\begin{aligned}\|M_{i+1}\| &\le \|M_0\| + \gamma(\|e_0\| + \|e_1\| + \cdots + \|e_{i+1}\|) \\ &\le \|M_0\| + \gamma\|e_0\|(1 + 1/2 + \cdots) = \|M_0\| + 2\gamma\|e_0\| \le 1/(8\beta),\end{aligned}$$

which establishes the second inequality, and then applying claim 1 gives the third inequality. □

Notice that the constant 1/2 in the linear convergence estimate arose from the proof rather than anything inherent to Broyden's method. Rearranging the proof, one could change this constant to any positive number. Thus the convergence is actually superlinear. It would be natural to try to prove this as an application of Theorem 4.8, but this is not possible, because it can be shown by example, that B_i need not converge to $F'(x_*)$. The superlinear convergence of Broyden's method was first proved in 1973 by Broyden, Dennis, and Moré. They proved slightly more, namely that $\|x_{i+1} - x_*\| \le r_i\|x_i - x_*\|$ where $r_i \to 0$.

6. Unconstrained minimization

We now turn to the problem of minimizing a real-valued function F defined on $\mathbb{R}^n$. (The problem of minimizing F over a subset of $\mathbb{R}^n$, e.g., a subspace or submanifold, is known as constrained minimization and is an important subject, which, however, we will not consider in this course.) We shall sometimes refer to F as the cost function. Usually we will have to content ourselves with finding a local minimum of the cost function since most methods cannot distinguish local from global minima. Note that the word "local" comes up in two distinct senses when describing the behavior of minimization methods: methods are often only locally convergent (they converge only for initial iterate x_0 sufficiently near x_*), and often the limit x_* is only a local minimum of the cost function.

If $F : \mathbb{R}^n \to \mathbb{R}$ is smooth, then at each x its gradient $F'(x)$ is a row vector and its Hessian $F''(x)$ is a symmetric matrix. If F achieves a local minimum at x_*, then $F'(x_*) = 0$ and $F''(x_*)$ is positive semidefinite. Moreover, if $F'(x_*) = 0$ and $F''(x_*)$ is positive definite, then F definitely achieves a local minimum at x_*.

There is a close connection with the problem of minimizing a smooth real-valued function of n variables and that of finding a root of an n-vector-valued function of n variables. Namely if x_* is a minimizer of $F : \mathbb{R}^n \to \mathbb{R}$, then x_* is a root of $F' : \mathbb{R}^n \to \mathbb{R}^n$. Another connection is that a point is a root of the function $K : \mathbb{R}^n \to \mathbb{R}^n$ if and only if it is a minimizer of $F(x) = \|K(x)\|^2$ (we usually use the 2-norm or a weighted 2-norm for this purpose, since then $F(x)$ is smooth if K is).

7. Newton's method

The idea of Newton's method for minimization problems is to approximate $F(x)$ by its quadratic Taylor polynomial, and minimize that. Thus

$$F(x) \approx F(x_i) + F'(x_i)(x - x_i) + \frac{1}{2}(x - x_i)^T F''(x_i)(x - x_i).$$

The quadratic on the right-hand side achieves a unique minimum value if and only if the matrix $F''(x_i)$ is positive definite, and in that case the minimum is given by the solution to the equation

$$F''(x_i)(x - x_i) + F'(x_i)^T = 0.$$

Thus we are lead to the iteration

$$x_{i+1} = x_i - F''(x_i)^{-1} F'(x_i)^T.$$

Note that this is exactly the same as Newton's method for solving the equation $F'(x) = 0$. Thus we know that this method is locally quadratically convergent (to a root of F', which might be only a local minima of F).

Newton's method for minimization requires the construction and "inversion" of the entire Hessian matrix. Thus, as for systems, there is motivation for using quasi-Newton methods in which the Hessian is only approximated. In addition, there is the fact that Newton's method is only locally convergent. We shall return to both of these points below.

8. Line search methods

Line search methods, of which there are many, take the form

choose initial iterate x_0
for $i = 0, 1, \ldots$
  choose search direction vector $s_i \in \mathbb{R}^n$
  choose step length $\lambda_i \in \mathbb{R}$
  $x_{i+1} = x_i + \lambda_i s_i$
end

There is a great deal of freedom in choosing the direction and the step length. The major criterion for the search direction is that a lower value of F than $F(x_i)$ exist nearby on the line $x_i + \lambda s_i$. We may as well assume that the step λ_i is positive in which case this criteria is that s_i is a *descent direction*, i.e., that $F(x_i + \lambda s_i)$ decreases as λ increases from 0. In terms of derivatives this condition is that $F'(x_i)^T s_i < 0$. Geometrically, this means that s_i should make an acute angle with the negative gradient vector $-F'(x_i)^T$. An obvious choice is $s_i = -F'(x_i)^T$ (or $-F'(x_i)^T/\|F'(x_i)\|$ if we normalize), the direction of steepest descents.

For the choice of step length, one possibility is *exact line search*. This means that λ_i is chosen to minimize $F(x_i + \lambda s_i)$ as a function of λ. In combination with the steepest descent direction we get the *method of steepest descents*:

choose $\lambda_i > 0$ minimizing $F(x_i - \lambda F'(x_i)^T)$ for $\lambda > 0$ set $x_{i+1} = x_i - \lambda_i F'(x_i)^T$

This method can be shown to be globally convergent to a local minimizer under fairly general circumstances. However, it may not be fast. To understand the situation better consider the minimization of a quadratic functional $F(x) = x^T Ax/2 - x^T b$ where $A \in \mathbb{R}^{n\times n}$ is symmetric postive definite and $b \in \mathbb{R}^n$. The unique minimizer of F is then the solution x_* to $Ax = b$. In this case, the descent direction at any point x is simply $-F'(x)^T = b - Ax$, the residual. Moreover, for any search direction s, the step length λ minimizing $F(x + \lambda s)$ (exact line search) can be computed analytically in this case:

$$F(x + \lambda s) = \frac{\lambda^2}{2} s^T As + \lambda(s^T Ax - s^T b) + \frac{1}{2} x^T Ax - x^T b,$$

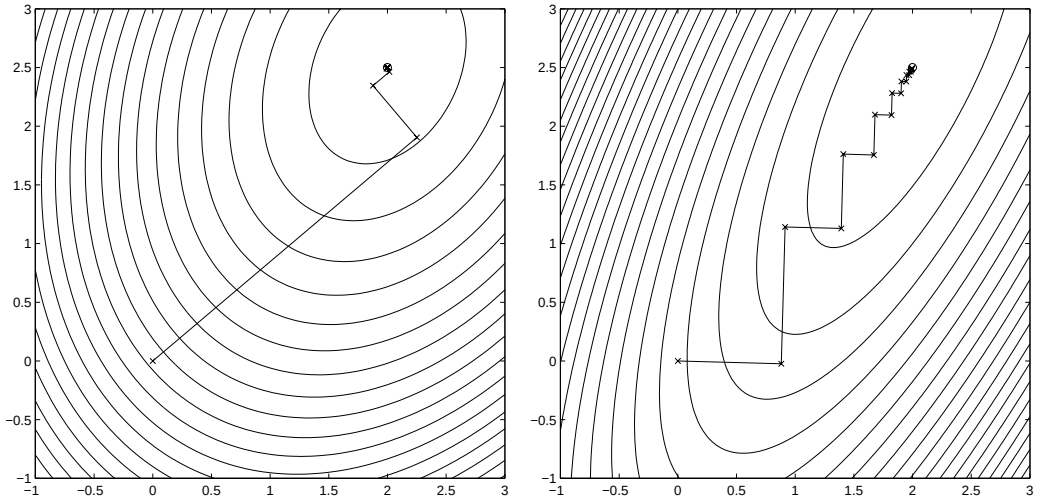
$$\frac{d}{d\lambda} F(x + \lambda s) = \lambda s^T As + s^T(Ax - b),$$

so that at the minimum $\lambda = s^T(b - Ax)/s^T As$, and if $s = b - Ax$, the direction of steepest descent, $\lambda = s^T s/s^T As$. Thus the steepest descent algorithm for minimizing $x^T Ax/2 - x^T b$, i.e., for solving $Ax = b$ is

choose initial iterate x_0
for $i = 0, 1, \ldots$
  $s_i = b - Ax$
  $\lambda_i = \frac{s_i^T s_i}{s_i^T A s_i}$
  $x_{i+1} = x_i + \lambda_i s_i$
end

It can be shown that this algorithm is globally convergent to the unique solution x_* as long as the matrix A is positive definite. However the convergence order is only linear and the rate is $(\kappa - 1)/(\kappa + 1)$ where $\kappa(A)$ is the 2-norm condition number of A, i.e., the ratio of the largest to the smallest eigenvalues of A. Thus the convergence will be very slow if A is not well-conditioned.

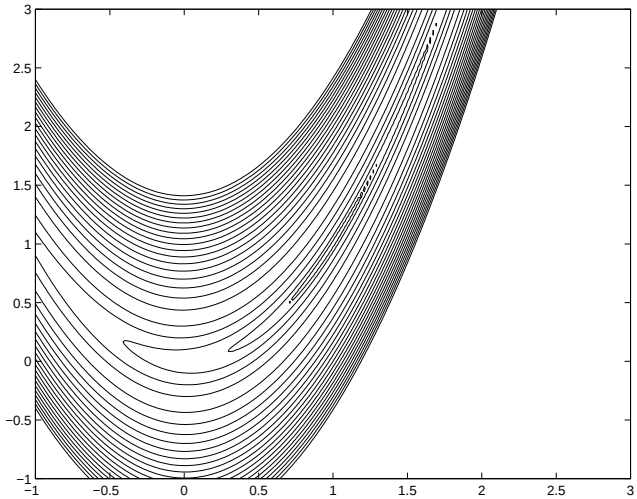
FIGURE 4.1. Convergence of steepest descents with a quadratic cost function. Left: condition number 2; right: condition number: 10.



This highlights a weakness of the steepest descent direction. It will be even more pronounced for a difficult non-quadratic cost function, such as Rosenbrock's example in $\mathbb{R}^2$

$$F(x) = (y - x^2)^2 + .01(1 - x)^2.$$

FIGURE 4.2. Some contours of the Rosenbrock function. Minimum is at $(1, 1)$.



While exact line search is possible for a quadratic cost functions, in general it is a scalar minimization problem which can be expensive or impossible to solve. Moreover, as illustrated

by the performance of steepest descents above, since the minimum may not be very near the search line, it is often not worth the effort to search too carefully. Thus many methods incorporate more or less sophisticated approximate line search algorithms. As we shall see, it is possible to devise an approximate line search method which, when used in conjunction with a reasonable choice of search direction, is globably convergent.

We begin our analysis with a simple calculus lemma.

LEMMA 4.12. *Let $f : \mathbb{R} \to \mathbb{R}$ be C^1 and bounded below and suppose that $f'(0) < 0$. For any $0 < \alpha < 1$ there exists a non-empty open interval $J \subset (0, \infty)$ such that*

$$f(x) < f(0) + \alpha x f'(0), \qquad f'(x) > \alpha f'(0), \tag{4.4}$$

for all $x \in J$.

PROOF. Since $f'(0) < 0$ and $0 < \alpha < 1$, we have $0 > \alpha f'(0) > f'(0)$. Thus the line $y = f(0) + \alpha f'(0)x$ lies above the curve $y = f(x)$ for sufficiently small positive x. But, since f is bounded below, the line lies below the curve for x sufficiently large. Thus $x_1 := \inf\{\, x > 0 \,|\, f(x) \geq f(0) + \alpha f'(0)x \,\} > 0$. Choose any $0 < x_0 < x_1$. By the mean value theorem there exists x between x_0 and x_1 such that

$$f'(x) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

For this point x we clearly have (4.4), and by continuity they must hold on an open interval around the point. □

Add a figure.

Now suppose we use a line search method subject to the following restrictions on the search directions s_i and the step lengths λ_i. We suppose that there exist positive constants η, α, β, such that for all i:

(H1) there exists $\eta \in (0, 1]$ such that $-F'(x_i)s_i \geq \eta \|F'(x_i)\| \|s_i\|$
(H2) there exists $\alpha \in (0, 1)$ such that $F(x_i + \lambda_i s_i) \leq F(x_i) + \alpha \lambda_i F'(x_i)s_i$
(H3) there exists $\beta \in (0, 1)$ such that $F'(x_i + \lambda_i s_i)s_i \geq \beta F'(x_i)s_i$

We shall show below that any line-search method meeting these conditions is, essentially, globally convergent. Before doing so, let us discuss the three conditions. The first condition concerns the choice of search direction. If $\eta = 0$ were permitted it would say that the search direction is a direction of non-ascent. By insisting on η positive we insure that the search direction is a direction of descent ($F(x_i + \lambda s_i)$ is a decreasing function of λ at $\lambda = 0$). However the condition also enforces a uniformity with respect to i. Specifically, it says that the angle between s_i and the steepest descent direction $-F'(x_i)^T$ is bounded above by $\arccos \eta < \pi/2$. The steepest descent direction satisfies (H1) for all $\eta \leq 1$ and so if $\eta < 1$ there is a open set of directions satisfying this condition. If the Hessian is positive definite, then the Newton direction $-F''(x_i)^{-1}F'(x_i)^T$ satisfies (H1) for $\eta \leq 1/\kappa_2(F''(x_i))$, with κ_2 the condition number with respect to the 2-norm, i.e., the ratio of the largest to smallest eigenvalues (verify!). One possible strategy to obtain the fast local convergence of Newton's method without sacrificing global convergence is to use the Newton direction for s_i whenever it satisfies (H1) (so whenever $F''(x_i)$ is positive definite and not too badly conditioned), otherwise to use steepest descents. A better approach when the Newton direction fails (H1) may be to use a convex combination of the Newton direction and the steepest descent

direction: $s_i = -[\theta F''(x_i)^{-1}F'(x_i)^T + (1-\theta)F'(x_i)^T]$ which will satisfy (H1) if $\theta > 0$ is small enough. Or similarly, one can take $s_i = [F''(x_i)^{-1} + \nu I]^{-1}F'(x_i)^T$ with ν large enough to insure that the bracketed matrix is positive definite. This is the Levenberg–Marquardt search direction.

Conditions (H2) and (H3) concern the step length. Roughly, (H2) insures that it is not too large, and in particular insures that $F(x_{i+1}) < F(x_i)$. It is certainly satisfied if λ_i is sufficiently small. On the other hand (H3) ensures that the step is not too small, since it is not satisfied for $\lambda_i = 0$. It is however satisfied at a minimizing λ_i if one exists. The lemma insures us that if $0 < \alpha < \beta < 1$, then there is an open interval of values of λ_i satisfying (H2) and (H3), and hence it is possible to design line-search algorithms which find a suitable λ_i in a finite number of steps. See, e.g., R. Fletcher, *Practical Methods of Optimization* or J. Dennis & R. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*. Fletcher also discusses typical choices for α and β. Typically β is fixed somewhere between 0.9 and 0.1, the former resulting in a faster line search while the latter in a more exact line search. Fletcher says that α is generally taken to be quite small, e.g., 0.01, but that the value of of α is not important in most cases, since it is usually the value of β which determines point acceptability.

We now state the global convergence theorem.

THEOREM 4.13. *Suppose that $F : \mathbb{R}^n \to \mathbb{R}$ is C^1 and bounded below, that $x_0 \in \mathbb{R}^n$ is such that $\{\, x \in \mathbb{R}^n : F(x) \le F(x_0) \,\}$ is bounded, and that $x_1, x_2, \ldots$ is defined by a line search method with descent search directions and positive step lengths satisfying the three conditions above. Then $\lim_{i\to\infty} F(x_i)$ exists and $\lim_{i\to\infty} F'(x_i) = 0$.*

The next comment should be filled out. Perhaps the theorem should be stated in the case of a single minimum and then the full state given as a corollary to the proof.

REMARK. If there is only one critical point x_* of F in the region $\{\, x \in \mathbb{R}^n : F(x) \le F(x_0) \,\}$, then the theorem guarantees that $\lim x_i = x_*$. In general the theorem does not quite guarantee that the x_i converge to anything, but by compactness the x_i must have one or more accumulation points, and these must be critical points.

PROOF. Since the sequence $F(x_i)$ is decreasing and bounded below, it converges. Hence $F(x_i) - F(x_{i+1}) \to 0$. By (H2),

$$F(x_i) - F(x_{i+1}) \ge -\alpha\lambda_i F'(x_i)s_i \ge 0,$$

so $\lambda_i F'(x_i)s_i \to 0$. By the (H1), this implies $\|F'(x_i)\|\lambda_i\|s_i\| \to 0$. There are now two possibilities: either $\|F'(x_i)\| \to 0$, in which case we are done, or else there exists $\epsilon > 0$ and a subsequence $\mathcal{S}$ with $\|F'(x_i)\| \ge \epsilon$, $i \in \mathcal{S}$. In view of the previous inequality, $\lambda_i s_i \to 0$, i.e., $x_i - x_{i+1} \to 0$, for $i \in \mathcal{S}$. Since all the iterates belong to the compact set $\{\, x \in \mathbb{R}^n : F(x) \le F(x_0) \,\}$, we may invoke uniform continuity of F' to conclude that $F'(x_{i+1}) - F'(x_i) \to 0$ as $i \to \infty$, $i \in \mathcal{S}$. We shall show that this is a contradiction.

Using (H3) and (H1), we have for all i

$$\begin{aligned}\|F'(x_{i+1}) - F'(x_i)\|\|s_i\| \\ \ge [F'(x_{i+1}) - F'(x_i)]s_i \ge (1-\beta)[-F'(x_i)s_i] \ge \eta(1-\beta)\|F'(x_i)\|\|s_i\|.\end{aligned}$$

Hence for $i \in \mathcal{S}$,

$$\|F'(x_{i+1}) - F'(x_i)\| \geq \eta(1-\beta)\epsilon > 0,$$

which gives the contradiction. □

The next theorem shows that if the point x_i is sufficiently close to a minimum, then choosing s_i to be the Newton direction and $\lambda_i = 1$ satisfies (H1)–(H3). This means that it is possible to construct algorithms which are globally convergent, but which are also quadratically convergent, since they eventually coincide with Newton's method.

THEOREM 4.14. *Suppose that F is smooth, x_* is a local minimum of F, and $F''(x_*)$ is positive definite. Let $0 < \alpha < 1/2$, $\alpha < \beta < 1$. Then there exists $\epsilon > 0$ such that if $\|x_i - x_*\| \leq \epsilon$, $s_i = -F''(x_i)^{-1}F'(x_i)^T$, and $\lambda_i = 1$, then (H1)–(H3) are satisfied with $\eta = 1/\{4\kappa_2[F''(x_*)]\}$.*

PROOF. Let D denote the ball about x_* of radius ϵ, where $\epsilon > 0$ will be chosen below. From our analysis of Newton's method we know that by taking ϵ sufficiently small, $x_i \in D \implies x_i + s_i \in D$. By continuity of F'', we may also arrange that whenever $x \in D$, $F''(x)$ is positive definite, $\|F''(x)\| \leq 2\|F''(x_*)\|$, and $\|F''(x)^{-1}\| \leq 2\|F''(x_*)^{-1}\|$.

Then

$$\begin{aligned} -F'(x_i)s_i &= F'(x_i)F''(x_i)^{-1}F'(x_i)^T \\ &\geq \lambda_{min}[F''(x_i)^{-1}]\|F'(x_i)\|^2 = \frac{1}{\|F''(x_i)\|}\|F'(x_i)\|^2 \geq \frac{1}{2\|F''(x_*)\|}\|F'(x_i)\|^2. \end{aligned}$$

Now

$$\|F'(x_i)\| \geq \frac{1}{\|F''(x_i)^{-1}\|}\|s_i\| \geq \frac{1}{2\|F''(x_*)^{-1}\|}\|s_i\|,$$

and (H1) follows from the last two estimates.

By Taylor's theorem,

$$F(x_i + s_i) - F(x_i) = F'(x_i)s_i + \frac{1}{2}s_i^T F''(\bar{x})s_i,$$

for some $\bar{x} \in D$. Thus

$$F(x_i + s_i) - F(x_i) = \frac{1}{2}F'(x_i)s_i + \frac{1}{2}s_i^T[F'(x_i)^T + F''(x_i)s_i] + \frac{1}{2}s_i^T[F''(\bar{x}) - F''(x_i)]s_i.$$

Now the second term on the right-hand side vanishes by the choice of s_i, and the third term can be bounded by a Lipschitz condition on F'', so

$$F(x_i + s_i) - F(x_i) \leq \frac{1}{2}F'(x_i)s_i + \frac{\gamma\epsilon}{2}\|s_i\|^2.$$

Since we have already established (H1), we have

$$\|s_i\|^2 \leq 2\|F''(x_*)^{-1}\|\|F'(x_i)\|\|s_i\| \leq -\frac{2}{\eta}\|F''(x_*)^{-1}\|F'(x_i)s_i. \tag{4.5}$$

Combining the last two estimates and choosing ϵ sufficiently small gives (H2) with any desired $\alpha < 1/2$.

For (H3), we note that

$$F'(x_i + s_i) = F'(x_i) + s_i^T F''(x_i) + s_i^T[F''(\tilde{x}) - F''(x_i)] = s_i^T[F''(\bar{x}) - F''(x_i)],$$

for some $\tilde{x} \in D$. Using the Lipschitz condition and (4.5) we get

$$F'(x_i + s_i)s_i \geq -\gamma\epsilon\|s_i\|^2 \geq \gamma\epsilon\frac{2}{\eta}\|F''(x_*)^{-1}\|F'(x_i)s_i,$$

and the desired estimate holds for ϵ sufficiently small. □

9. Conjugate gradients

Now we return to the case of minimization of a positive definite quadratic function $F(x) = x^TAx/2 - x^Tb$ with $A \in \mathbb{R}^{n\times n}$ symmetric positive definite and $b \in \mathbb{R}^n$. So the unique minimizer x_* is the solution to the linear system $Ax = b$. Consider now a line search method with exact line search:

choose initial iterate x_0
for $i = 0, 1, \ldots$
  choose search direction s_i
  $\lambda_i = \dfrac{s_i^T(b - Ax_i)}{s_i^TAs_i}$
  $x_{i+1} = x_i + \lambda_i s_i$
end

Thus $x_1 = x_0 + \lambda_0 s_0$ minimizes F over the 1-dimensional affine space $x_0 + \text{span}[s_0]$, and then $x_2 = x_0+\lambda_0 s_0+\lambda_1 s_1$ minimizes F over the affine space 1-dimensional $x_0+\lambda_0 s_0+\text{span}[s_1]$. However x_2 does not minimize F over the 2-dimensional affine space $x_0+\text{span}[s_0, s_1]$. If that were the case, then for 2-dimensional problems we would have $x_2 = x_*$ and we saw that that was not the case for steepest descents.

However, it turns out that there is a simple condition on the search directions s_i that ensures that x_2 *is* the minimizer of F over $x_0+\text{span}[s_0, s_1]$, and more generally that x_i is the minimizer of F over $x_0 + \text{span}[s_0, \ldots, s_{i-1}]$. In particular (as long as the search directions are linearly independent), this implies that $x_n = x_*$.

THEOREM 4.15. *Suppose that x_i are defined by exact line search using search directions which are A-orthogonal: $s_i^TAs_j = 0$ for $i \neq j$. Then*

$$F(x_i) = \min\{\, F(x) \,|\, x \in x_0 + \text{span}[s_0, \ldots, s_{i-1}] \,\}.$$

PROOF. By induction on i, the case $i = 1$ being clear. Write W_i for $\text{span}[s_0, \ldots, s_{i-1}]$. Now

$$\min_{x_0+W_{i+1}} F = \min_{y\in x_0+W_i}\ \min_{\lambda\in\mathbb{R}} F(y + \lambda s_i).$$

But

$$F(y + \lambda s_i) = \frac{1}{2}y^TAy + \lambda s_i^TAy + \frac{\lambda^2}{2}s_i^TAs_i - y^Tb - \lambda s_i^Tb.$$

The second term on the right-hand side appears to couple the minimizations with respect to y and λ, but in fact this is not so. Indeed, $x_i \in x_0 + W_i$, so for $y \in x_0 + W_i$, $y - x_i \in W_i$ and so is A-orthogonal to s_i. That is, $s_i^TAy = s_i^TAx_i$, whence

$$F(y + \lambda s_i) = [\frac{1}{2}y^TAy - y^Tb] + [\frac{\lambda^2}{2}s_i^TAs_i + \lambda s_i^T(Ax_i - b)],$$

and the minimization problem decouples. By induction the minimum of the first term in brackets over $x_0 + W_i$ is achieved by $y = x_i$, and clearly the second term is minimized by $\lambda = s_i^T(b - Ax_i)/s_i^T As_i$, i.e., the exact line search. Thus $x_{i+1} = x_i + \lambda_i s_i$ minimizes F over $x_0 + W_{i+1}$. □

Any method which uses A-orthogonal (also called "conjugate") search directions has the nice property of the theorem. However it is not so easy to construct such directions. By far the most useful method is the method of conjugate gradients, or the CG method, which defines the search directions by A-orthogonalizing the residuals $r_i = b - Ax_i$:

- $s_0 = r_0$
- $s_i = r_i - \sum_{j=0}^{i-1} \frac{s_j^T Ar_i}{s_j^T As_j} s_j.$

The last formula (which is just the Gram-Schmidt procedure) appears to be quite expensive to implement, but fortunately we shall see that it may be greatly simplified.

LEMMA 4.16. (1) $W_i = \text{span}[s_0, \ldots, s_{i-1}] = \text{span}[r_0, \ldots, r_{i-1}]$.
(2) *The residuals are l_2-orthogonal: $r_i^T r_j = 0$ for $i \neq j$.*
(3) *There exists $m \leq n$ such that $W_1 \subsetneq W_2 \subsetneq \cdots \subsetneq W_m = W_{m+1} = \cdots$ and $x_0 \neq x_1 \neq \cdots \neq x_m = x_{m+1} = \cdots = x_*$.*
(4) *For $i \leq m$, $\{ s_0, \ldots, s_{i-1} \}$ is an A-orthogonal basis for W_i and $\{ r_0, \ldots, r_{i-1} \}$ is an l_2-orthogonal basis for W_i.*
(5) *$s_i^T r_j = r_i^T r_i$ for $0 \leq j \leq i$.*

PROOF. The first statement comes directly from the definitions. To verify the second statement, note that, for $0 \leq j < i$, $F(x_i + tr_j)$ is minimal when $t = 0$, which gives $r_j^T(Ax_i - b) = 0$, which is the desired orthogonality. For the third statement, certainly there is a least integer $m \in [1, n]$ so that $W_m = W_{m+1}$. Then $r_m = 0$ since it both belongs to W_m and is orthogonal to W_m. This implies that $x_m = x_*$ and that $s_m = 0$. Since $s_m = 0$, $W_{m+1} = W_m$ and $x_{m+1} = x_m = x_*$. Therefore $r_{m+1} = 0$, which implies that $s_{m+1} = 0$, therefore $W_{m+2} = W_{m+1}$, $x_{m+2} = x_*$, etc.

The fourth statement is an immediate consequence of the preceding ones. For the last statement, we use the orthogonality of the residuals to see that $s_i^T r_i = r_i^T r_i$. But, if $0 \leq j \leq i$,then

$$s_i^T r_j - s_i^T r_0 = s_i^T A(x_0 - x_j) = 0,$$

since $x_0 - x_j \in W_i$. □

Since $s_i \in W_{i+1}$ and the r_j, $j \leq i$ are an orthogonal basis for that space for $i < m$, we have

$$s_i = \sum_{j=0}^{i} \frac{s_i^T r_j}{r_j^T r_j} r_j.$$

In view of part 5 of the lemma, we can simplify

$$s_i = r_i^T r_i \sum_{j=0}^{i} \frac{r_j}{r_j^T r_j} = r_i + r_i^T r_i \sum_{j=0}^{i-1} \frac{r_j}{r_j^T r_j},$$

whence

$$s_i = r_i + \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}} s_{i-1}.$$

This is the formula which is used to compute the search direction. In implementing this formula it is useful to compute the residual from the formula $r_{i+1} = r_i - \lambda_i A s_i$ (since $x_{i+1} = x_i + \lambda_i s_i$). Putting things together we obtain the following implementation of CG:

choose initial iterate x_0, set $s_0 = r_0 = b - Ax_0$
for $i = 0, 1, \ldots$
$\lambda_i = \dfrac{r_i^T r_i}{s_i^T A s_i}$
$x_{i+1} = x_i + \lambda_i s_i$
$r_{i+1} = r_i - \lambda_i A s_i$
$s_{i+1} = r_{i+1} + \dfrac{r_{i+1}^T r_{i+1}}{r_i^T r_i} s_i$
end

At each step we have to perform one multiplication of a vector by A, two dot-products, and three SAXPYs. When A is sparse, so that multiplication by A is inexpensive, the conjugate gradient method is most useful. Here is the algorithm written out in full in pseudocode:

choose initial iterate x
$r \leftarrow b - Ax$
$r2 \leftarrow r^T r$
$s \leftarrow r$
for $i = 0, 1, \ldots$
$t \leftarrow As$ (matrix multiplication)
$s2 \leftarrow s^T t$ (dot product)
$\lambda \leftarrow r2/s2$
$x \leftarrow x + \lambda s$ (SAXPY)
$r2old \leftarrow r2$
$r \leftarrow r - \lambda t$ (SAXPY)
$r2 \leftarrow r^T r$ (dot product)
$s \leftarrow r + (r2/r2old)s$ (SAXPY)
end

The conjugate gradient method gives the exact solution in n iterations, but it is most commonly terminated with far fewer operations. A typical stopping criterion would be to test if $r2$ is below a given tolerance. To justify this, we shall show that the method is linearly convergence and we shall establish the rate of convergence. For analytical purposes, it is most convenient to use the vector norm $\|x\|_A := (x^T A x)^{1/2}$, and its associated matrix norm.

LEMMA 4.17. *$W_i = \text{span}[r_0, Ar_0, \ldots, A^{i-1} r_0]$ for $i = 1, 2, \ldots, m$.*

PROOF. Since $\dim W_i = i$, it is enough to show that $W_i \subset \operatorname{span}[r_0, Ar_0, \ldots, A^{i-1}r_0]$, which we do by induction. This is certainly true for $i = 1$. Assume it holds for some i. Then, since $x_i \in x_0 + W_i$, $r_i = b - Ax_i \in r_0 + AW_i \in \operatorname{span}[r_0, Ar_0, \ldots, A^i r_0]$, and therefore W_{i+1}, which is spanned by W_i and r_i belongs to $\operatorname{span}[r_0, Ar_0, \ldots, A^i r_0]$, which completes the induction. □

The space $\operatorname{span}[r_0, Ar_0, \ldots, A^{i-1}r_0]$ is called the *Krylov space* generated by the matrix A and the vector r_0. Note that we have as well

$$W_i = \operatorname{span}[r_0, Ar_0, \ldots, A^{i-1}r_0] = \{\, p(A)r_0 \,|\, p \in \mathcal{P}_{i-1} \,\} = \{\, q(A)(x_* - x_0) \,|\, q \in \mathcal{P}_i, q(0) = 0 \,\}.$$

Since r_i is l_2-orthogonal to W_i, $x_* - x_i$ is A-orthogonal to W_i so

$$\|x_* - x_i\|_A = \inf_{w \in W_i} \|x_* - x_i + w\|_A.$$

Since $x_i - x_0 \in W_i$,

$$\inf_{w \in W_i} \|x_* - x_i + w\|_A = \inf_{w \in W_i} \|x_* - x_0 + w\|_A.$$

Combining the last three equations, we get

$$\|x_* - x_i\|_A = \inf_{\substack{q \in \mathcal{P}_i \\ q(0)=0}} \|x_* - x_0 + q(A)(x_* - x_0)\|_A = \inf_{\substack{p \in \mathcal{P}_i \\ p(0)=1}} \|p(A)(x_* - x_0)\|_A.$$

Applying the obvious bound $\|p(A)(x_* - x_0)\|_A \le \|p(A)\|_A \|x_* - x_0\|_A$ we see that we can obtain an error estimate for the conjugate gradient method by estimating

$$C = \inf_{\substack{p \in \mathcal{P}_i \\ p(0)=1}} \|p(A)\|_A.$$

Now if $0 < \rho_1 < \cdots < \rho_n$ are the eigenvalues of A, then the eigenvalues of $p(A)$ are $p(\rho_j)$, $j = 1, \ldots, n$, and $\|p(A)\|_A = \max_j |p(\rho_j)|$ (this is left as exercise 6). Thus[1]

$$C = \inf_{\substack{p \in \mathcal{P}_i \\ p(0)=1}} \max_j |p(\rho_j)| \le \inf_{\substack{p \in \mathcal{P}_i \\ p(0)=1}} \max_{\rho_1 \le \rho \le \rho_n} |p(\rho)|.$$

The final infimum can be calculated explicitly using the Chebyshev polynomials, see Figure 4.3 and (1.16). The minimum value is precisely

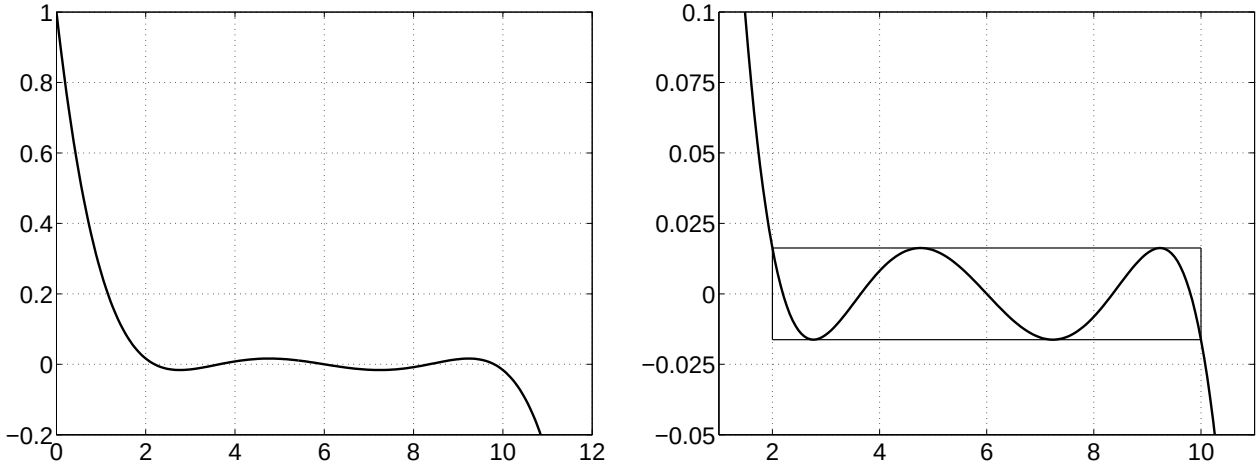
$$\frac{2}{\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i} \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i,$$

where $\kappa = \rho_n/\rho_1$ is the condition number of A. (To get the right-hand side, we suppressed the second term in the denominator of the left-hand side, which is less than 1 and tends to zero with i, and kept only the first term, which is greater than 1 and tends to infinity with i.) We have thus proven that

$$\|x_i - x_*\|_A \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i \|x_0 - x_*\|_A,$$

[1]Here we bound $\max_j |p(\rho_j)|$ by $\max_{\rho_1 \le \rho \le \rho_n} |p(\rho)|$ simply because we can minimize the latter quantity explicitly. However this does not necessarily lead to the best possible estimate, and the conjugate gradient method is often observed to converge faster than the result derived here. Better bounds can sometimes be obtained by taking into account the distribution of the spectrum of A, rather than just its minimum and maximum.

FIGURE 4.3. The quintic polynomial equal to 1 at 0 with the smallest L^∞ norm on $[2, 10]$. This is a scaled Chebyshev polynomial, and so the norm can be computed exactly.



which is linear convergence with rate

$$r = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}.$$

Note that $r \sim 1 - 2/\sqrt{\kappa}$ for large κ. So the convergence deteriorates when the condition number is large.

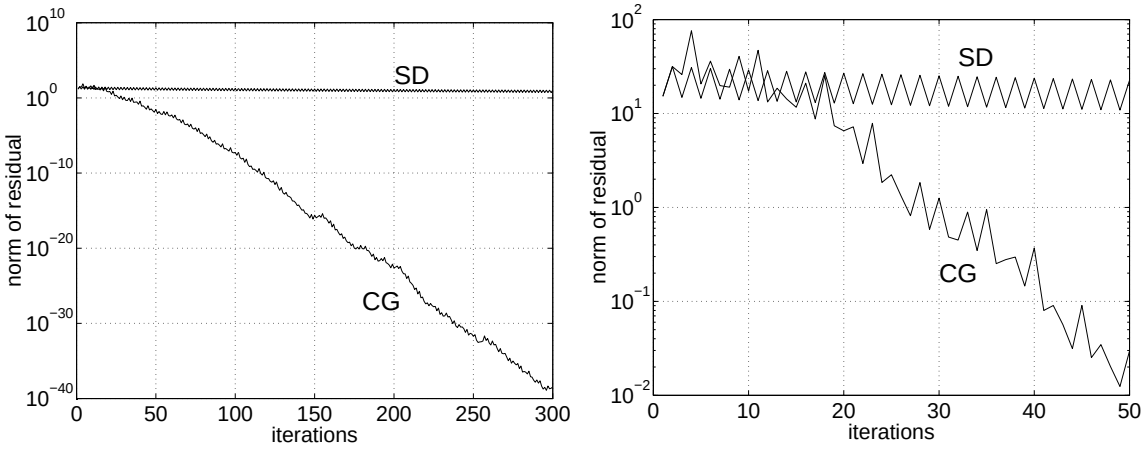
Let us compare this convergence estimate with the analogous one for the method of steepest descents. To derive an estimate for steepest descents, we use the fact that the first step of conjugate gradients coincides with steepest descents, and so

$$\|x_* - x_1\|_A \leq \frac{2}{\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}} \|x_* - x_0\|_A = \frac{\kappa - 1}{\kappa + 1} \|x_* - x_0\|_A.$$

Of course, the same result holds if we replace x_0 by x_i and x_1 by x_{i+1}. Thus steepest descents converges linearly, with rate $(\kappa - 1)/(\kappa + 1)$. Notice that the estimates indicate that a large value of κ will slow the convergence of both steepest descents and conjugate gradients, but, since the dependence is on $\sqrt{\kappa}$ rather than κ, the convergence of conjugate gradients will usually be much faster.

The figure shows a plot of the norm of the residual versus the number of iterations for the conjugate gradient method and the method of steepest descents applied to a matrix of size 233 arising from a finite element simulation. The matrix is irregular, but sparse (averaging about 6 nonzero elements per row), and has a condition number of about $1,400$. A logarithmic scale is used on the y-axis so the near linearity of the graph reflects linear convergence behavior. For conjugate gradients, the observed rate of linear convergence is about .8, and it takes 80 iterations to reduce the initial residual by a factor of about 10^6. The convergence of steepest descents is too slow to be useful: in 400 iterations the residual is not even reduced by a factor of 2.

FIGURE 4.4. Convergence of conjugate gradients for solving a finite element system of size 233. On the left 300 iterations are shown, on the right the first 50. Steepest descents is shown for comparison.



REMARK. 1. The conjugate gradient algorithm can be generalized to apply to the minimization of general (non-quadratic) functionals. The Fletcher–Reeves method is such a generalization. However in the non-quadratic case the method is significantly more complicated, both to implement and to analyze.

2. There are a variety of conjugate-gradient-like iterative methods that apply to matrix problems $Ax = b$ where A is either indefinite, non-symmetric, or both. Many share the idea of approximation of the solution in a Krylov space.

9.1. Preconditioning. The idea is we choose a matrix $M \approx A$ such that the system $Mz = c$ is relatively easy to solve. We then consider the *preconditioned system* $M^{-1}Ax = M^{-1}b$. The new matrix $M^{-1}A$ is SPD with respect to the M innerproduct, and we solve the preconditioned system using conjugate gradients but using the M-inner product in place of the l_2-inner product. Thus to obtain the preconditioned conjugate gradient algorithm, or PCG, we substitute $M^{-1}A$ for A everywhere and change expressions of the form $x^T y$ into $x^T My$. Note that the A-inner product $x^T Ay$ remains invariant under these two changes. Thus we obtain the algorithm:

choose initial iterate x_0, set $s_0 = \bar{r}_0 = M^{-1}b - M^{-1}Ax_0$
for $i = 0, 1, \ldots$
$\lambda_i = \dfrac{\bar{r}_i^T M \bar{r}_i}{s_i^T A s_i}$
$x_{i+1} = x_i + \lambda_i s_i$
$\bar{r}_{i+1} = \bar{r}_i - \lambda_i M^{-1} A s_i$
$s_{i+1} = \bar{r}_{i+1} + \dfrac{\bar{r}_{i+1}^T M \bar{r}_{i+1}}{\bar{r}_i^T M \bar{r}_i} s_i$
end

Note that term $s_i^T A s_i$ arises as the M-inner product of s_i with $M^{-1}As_i$. The quantity $\bar{r}_i$ is the residual in the preconditioned equation, which is related to the regular residual, $r_i = b - Ax_i$ by $r_i = M\bar{r}_i$. Writing PCG in terms of r_i rather than $\bar{r}_i$ we get

choose initial iterate x_0, set $r_0 = b - Ax_0$, $s_0 = M^{-1}r_0$
for $i = 0, 1, \ldots$
$\lambda_i = \dfrac{r_i^T M^{-1} r_i}{s_i^T A s_i}$
$x_{i+1} = x_i + \lambda_i s_i$
$r_{i+1} = r_i - \lambda_i A s_i$
$s_{i+1} = M^{-1} r_{i+1} + \dfrac{r_{i+1}^T M^{-1} r_{i+1}}{r_i^T M^{-1} r_i} s_i$
end

Thus we need to compute $M^{-1}r_i$ at each iteration. Otherwise the work is essentially the same as for ordinary conjugate gradients. Since the algorithm is just conjugate gradients for the preconditioned equation we immediately have an error estimate:

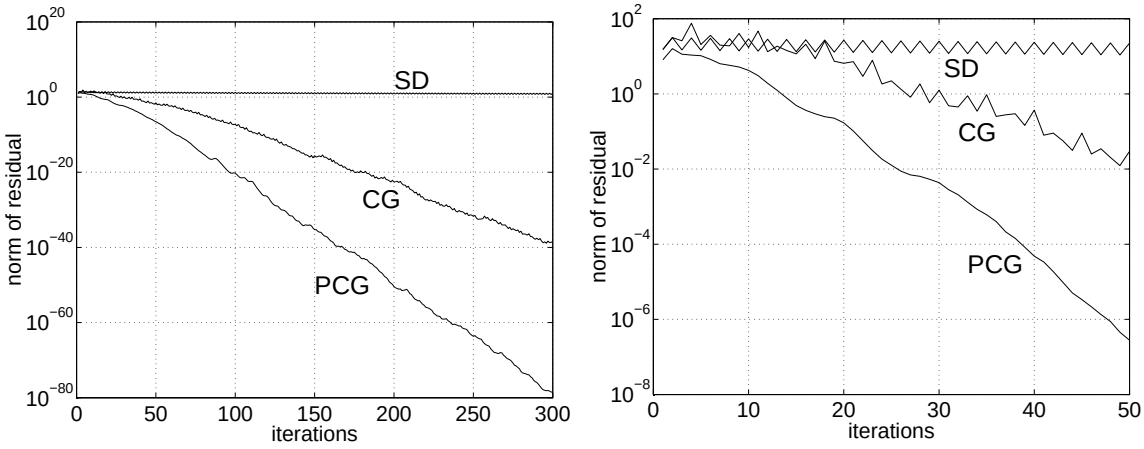
$$\|x_i - x_*\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i \|x_0 - x_*\|_A,$$

where κ now is the ratio of the largest to the least eigenvalue of $M^{-1}A$. To the extent that M approximates A, this ratio will be close to 1 and so the algorithm will converge quickly.

The matrix M is called the *preconditioner*. A good preconditioner should have two properties. First, it must be substantially easier to solve systems with the matrix M than with the original matrix A, since we will have to solve such a system at each step of the preconditioned conjugate gradient algorithm. Second, the matrix $M^{-1}A$ should be substantially better conditioned than A, so that PCG converges faster than ordinary CG. In short, M should be near A, but much easier to invert. One simple possibility is to take M to be the diagonal matrix with the same diagonal entries as A. This certainly fulfils the first criterion (easy invertibility), and for some matrices A, the second criterion is met as well. A similar possibility is to take M to be a tridiagonal matrix with its nonzero entries taken from A. A third possibility which is often applied when A is sparse is to determine M via the *incomplete Cholesky factorization*. This means that a triangular matrix L is computed by the Cholesky algorithm applied to A, except that no fill-in is allowed: only the non-zero elements of A are altered, and the zero elements left untouched. One then takes $M = LL^T$, and, so M^{-1} is easy to apply. Other preconditioners take into account the source of the matrix problem. For example, if a matrix arises from the discretization of a complex partial differential equation, we might precondition it by the discretization matrix for a simpler related differential equation (if that lead to a linear systems which is easier to solve). In fact the derivation of good preconditioners for important classes of linear systems remain a very active research area.

We close with numerical results for the simplest preconditioner: the diagonal preconditioner. The following figure reproduces the results shown in Figure 4.4, together with the norm of the residual for PCG. An error reduction of 10^{-6} occurs with 44 iterations of PCG, as opposed to 80 of CG.

FIGURE 4.5. Convergence of preconditioned conjugate gradients for solving a finite element system of size 233. On the left 300 iterations are shown, on the right the first 50. Unpreconditioned CG and Steepest descents are shown for comparison.



EXERCISES

(1) Let $f : \mathbb{R} \to \mathbb{R}$ be a C^2 function with a root x_* such that neither f' nor f'' has a root. Prove that Newton's method converges to x_* for *any* initial guess $x_0 \in \mathbb{R}$.

(2) Consider the 2×2 system of nonlinear equations

$$f(x, y) = 0, \quad g(x, y) = 0, \qquad x, y \in \mathbb{R}.$$

The *Jacobi iteration* for solving this system beginning from an initial guess x_0, y_0 is Thus

for $i = 0, 1, 2, \ldots$
  solve $f(x_{i+1}, y_i) = 0$ for x_{i+1}
  solve $g(x_i, y_{i+1}) = 0$ for y_{i+1}
end

each step of the iteration requires the solution of 2 *scalar* nonlinear equations. (N.B.: Of course the method extends to systems of n equations in n unknowns.) If we combine the Jacobi iteration with Newton's method to solve the scalar equations, we get the *Newton–Jacobi* iteration:

choose initial guess x_0, y_0
for $i = 1, 2, \ldots$

$$x_{i+1} = x_i - \frac{\partial f}{\partial x}(x_i, y_i)^{-1} f(x_i, y_i)$$

$$y_{i+1} = y_i - \frac{\partial g}{\partial y}(x_i, y_i)^{-1} g(x_i, y_i)$$

end

Determine under what conditions this algorithm is locally convergent.

(3) The Gauss–Seidel iteration for a 2×2 system of nonlinear equations differs from the Jacobi iteration in that the equation determining y_{i+1} is $g(x_{i+1}, y_{i+1}) = 0$. Formulate the Newton–Gauss–Seidel iteration, determine conditions under which it is locally convergent, and compare the conditions to those for the Newton–Jacobi iteration.

(4) Recall that in Broyden's method we update a matrix B to obtain a matrix $\tilde{B}$ which satisfies $\tilde{B}s = v$ for given vectors $s \neq 0$, v. Show that $\tilde{B}$ the closest matrix to B which satisfies this equation, that is that $\|\tilde{B} - B\| \leq \|\bar{B} - B\|$ for any matrix $\bar{B}$ satisfying $\bar{B}s = v$ where the norm is the matrix 2-norm. Show that the same result holds if the norm is the Frobenius norm, which is defined by $\|A\| = (\sum_{i,j} a_{ij}^2)^{1/2}$, and that in this case $\tilde{B}$ is the *unique* nearest matrix to B satisfying the desired equation.

(5) Consider a system of n equations in n unknowns consisting of m linear equations and $n - m$ nonlinear equations

$$Ax - b = 0, \qquad g(x) = 0, \qquad A \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^m, \quad g : \mathbb{R}^n \to \mathbb{R}^{n-m}.$$

Let $x_0, x_1, \ldots$ be the sequence of iterates produced by Newton's method. Show that all the iterates after the initial guess satisfy the linear equations exactly. Show the same result is true when the x_i are determined by Broyden's method with B_0 chosen to be $F'(x_0)$.

(6) Prove that if A is a symmetric positive-definite matrix with eigenvalues $\rho_1, \ldots, \rho_n$, and p is a polynomial, then $\|p(A)\|_A = \max\limits_{1 \leq j \leq n} |p(\rho_j)|$.

(7) Prove that for the conjugate gradient method the search directions s_i and the errors $e_i := x_* - x_i$ satisfy $s_i^T e_{i+1} \leq 0$ (in fact $s_i^T e_j \leq 0$ for all i, j). Use this to show that the l_2-norm of the error $\|e_i\|$ is a non-increasing function of i.

(8) We analyzed preconditioned conjugate gradients, with a symmetric positive definite preconditioner M, as ordinary conjugate gradients applied to the problem $M^{-1}Ax = M^{-1}b$ but with the M-inner product rather than the l_2-inner product in $\mathbb{R}^n$. An alternative approach which doesn't require switching inner products in $\mathbb{R}^n$ is to consider the ordinary conjugate gradient method applied to the symmetric positive definite problem $(M^{-1/2}AM^{-1/2})z = M^{-1/2}b$ for which the solution is $z = M^{1/2}x$. Show that this approach leads to exactly the same preconditioned conjugate gradient algorithm.

(9) The Matlab command `A=delsq(numgrid('L',n))` is a quick way to generate a symmetric positive definite sparse test matrix: it is the matrix arising from the 5-point finite difference approximation to the Laplacian on an L-shaped domain using an $n \times n$ grid (e.g., if $n = 40$, A will be $1,083 \times 1,083$ sparse matrix with $5,263$ nonzero elements and a condition number of about 325. Implement the conjugate gradient algorithm for the system $Ax = b$ for this A (and an arbitrary vector b, e.g., all 1's). Diagonal preconditioning does no good for this problem. (Why?) Try two other possibilities: tridiagonal preconditioning and incomplete Cholesky preconditioning (Matlab comes equipped with an incomplete Cholesky routine, so you don't have to write your own). Study and report on the convergence in each case.

CHAPTER 5

Numerical Solution of Ordinary Differential Equations

1. Introduction

In this chapter we are concerned with the numerical solution of the initial value problem (IVP)

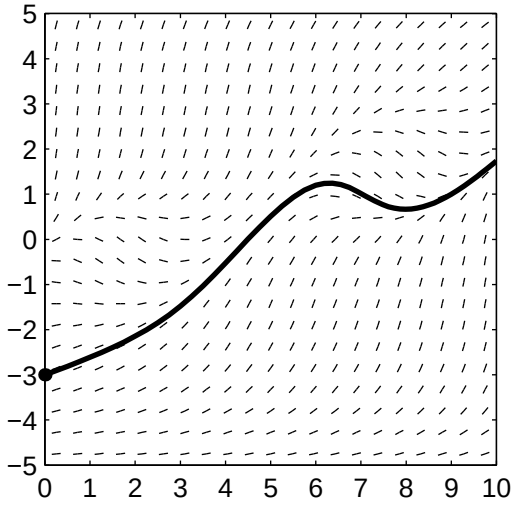
$$y' = f(t, y), \quad y(t_0) = y_0. \tag{5.1}$$

More precisely, we suppose we are given a function $f(t, y)$ defined in some domain $D \subset \mathbb{R}^2$ and a point $(t_0, y_0) \in D$. We wish to find an interval I containing t_0 and a function $y : I \to \mathbb{R}$ satisfying

$$y'(t) = f\big(t, y(t)\big), \quad t \in I,$$

as well as the initial condition. The function f specifies a slope field on the domain D, which we can visualize graphically as a small line segment passing through each point of D, and the IVP says that the graph of y passes through (t_0, y_0) and at each point is tangent to the line segment at that point.

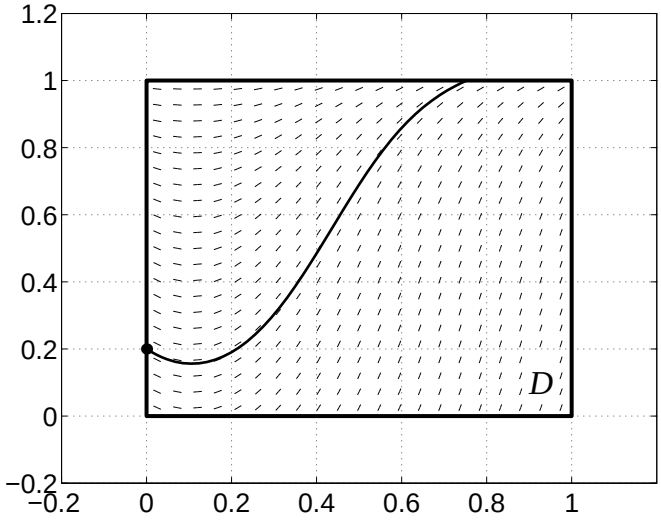
FIGURE 5.1. The slopefield for $f(t, y) = [(y + 1 - t/3)^2 - \sin t]e^{-.15(y-1)^2}$ and the solution to $y'(t) = f\big(t, y(t)\big)$, $y(0) = -3$.



THEOREM 5.1 (Local Existence Theorem). *Let f be a continuous function on a domain $D \subset \mathbb{R}^2$ and let (t_0, y_0) be an interior point of D. Then there exists an interval I containing t_0 in its interior and a function $y : I \to \mathbb{R}$ solving* (5.1).

The question of existence of solutions on a given interval I containing t_0 is more subtle. It can certainly happen that the solution curve $(t, y(t))$ leaves the domain D before t traverses the entire given interval. See Figures 5.2

FIGURE 5.2. The solution to $y' = f(t, y)$, $y(0) = .2$. In this example the domain D of f is the unit square, and the solution curve leaves D at $t = .75$, so there is no solution to the initial value problem defined on the whole interval $I = [0, 1]$.



Even if $D \supset I \times \mathbb{R}$, it may happen that the solution tends to infinity as t tends to some interior point of I, and so the solution can again not be continued to the end of the interval I. For example, the solution to the IVP

$$y' = y^2, \quad y(1) = 1,$$

is $y(t) = 1/(2 - t)$, which blows up as t approaches 2. Thus there is no solution to this simple-looking IVP on the interval $[1, 3]$. See Figure 5.3.

Another issue which must be faced before we can study numerical methods and their convergence is *uniqueness of solutions*. If we only assume that f is continuous, then we cannot assert uniqueness. For example, the function $f(x, y) = \sqrt{|1 - y^2|}$ is continuous on $\mathbb{R}^2$, and the IVP

$$y' = \sqrt{|1 - y^2|}, \quad y(-\pi/2) = -1,$$

has infinitely many solutions, among them $y(t) = \sin t$ and $y(t) = -1$.

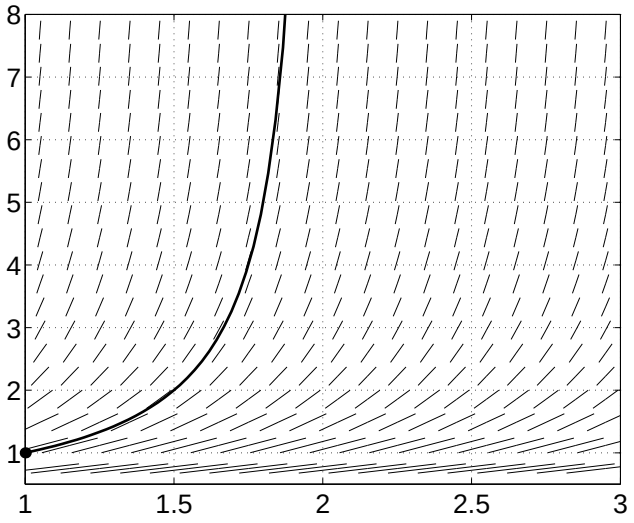
Fortunately there is a simple condition which implies both global existence and uniqueness.

DEFINITION. Let f be a function defined on $I \times \mathbb{R}$ where I is an interval. Then f satisfies a *uniform Lipschitz condition with respect to its second variable* if

$$|f(t, y) - f(t, z)| \leq K|y - z|, \quad t \in I, \quad y, z \in \mathbb{R}.$$

Note that if $f \in C^1(I \times \mathbb{R})$ and $\partial f/\partial y$ is bounded, then f satisfies a uniform Lipschitz condition with respect to y.

FIGURE 5.3. The solution to $y' = y^2$, $y(1) = 1$ cannot be continued to $t \geq 2$.



THEOREM 5.2 (Global existence–uniqueness theorem). *Let f be a continuous function on $I \times \mathbb{R}$ which satisfies a uniform Lipschitz condition with respect to its second variable. Then, for any $(t_0, y_0) \in I \times \mathbb{R}$, there exists a unique solution $y : I \to \mathbb{R}$ to the IVP*

$$y' = f(t, y), \quad y(t_0) = y_0.$$

Notice that many simple, smooth functions f on $I \times \mathbb{R}$, such as $f(t, y) = y^2$ or $f(t, y) = ty^2$ fail to satisfy a uniform Lipschitz condition with respect to its second variable, because the partial derivative $\partial f / \partial y$ is unbounded. For such functions, we cannot assert global existence in general, but on an interval on which a solution exists, it is unique. Indeed, suppose that $y' = f(t, y)$ and $z' = f(t, z)$ on some finite closed interval I, and that $y(t_0) = z(t_0)$ for some $t_0 \in I$. Then $y \equiv z$ on I. To see this, let J be a finite interval which contains the ranges $y(I)$ and $z(I)$ (both of which are bounded, since y and z are continuous and I is closed and bounded). Let ϕ be a smooth function which is identically equally to unity on J but identically equal to zero outside some large bounded closed interval K, and set $F(t, y) = \phi(y) f(t, y)$. Then

Here we need material on uniqueness, existence and uniqueness if we assume a Lipschitz condition with respect to y, and sensitivity of solutions with respect to initial data (stability theorem to be labelled th:odestab).

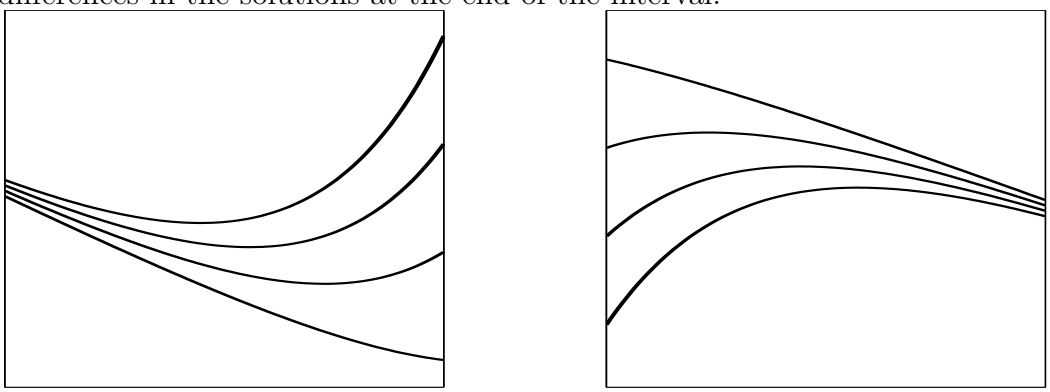
THEOREM 5.3. ...

Make reference to Figure 5.4.

2. Euler's Method

2.1. Derivation. If we recall the interpretation of a solution to the IVP as a function whose graph passes through (t_0, y_0) as is tangent to the slope field determined by f, this suggests a graphical approach to the approximate solution of the IVP. We start to draw its graph at the initial point, extending it in the direction of, say, increasing t, along the line

FIGURE 5.4. The left figure shows several solutions of an ODE for which the solution is very sensitive to the initial data. The right figures shows the opposite situation: even large differences in the initial data cause only small differences in the solutions at the end of the interval.



through that point with slope $f(t_0, y_0)$. This determines an approximate solution on a short time interval $[t_0, t_0 + h]$ as

$$y^h(t) = y_0 + tf(t_0, y_0), \quad t_0 \le t \le t_0 + h.$$

If h is sufficiently small this should not differ much from the true solution $y(t)$ (since a curve does not differ much from its tangent in a small interval). We may then repeat the process starting from $t_1 := t_0 + h$ and using the slope at (t_1, y_1) where $y_1 = y^h(t_1) = y_0 + hf(t_0, y_0)$, and so forth. Defining $t_n = t_0 + nh$, we thus get approximations $y_n = y^h(t_n) \approx y(t_n)$ satisfying

$$y_{n+1} = y_n + hf(t_n, y_n), \quad n = 0, 1, \ldots$$

This is Euler's method for solving the IVP. For most purposes it is sufficient to think of the approximate solution y^h as defined only at the discrete points t_n and thus given by the values y_n. For others it is useful to consider the approximate solution as the piecewise linear function with break points at the t_n and which interpolates y_n at t_n.

The graphical derivation just given does not easily generalize to give other numerical methods, but here are three other derivations of Euler's method which do.

2.1.1. *Taylor series.* The exact solution satisfies

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + O(h^2).$$

Neglecting the $O(h^2)$ term we get

$$y(t_{n+1}) \approx y(t_n) + hy'(t_n) = y(t_n) + hf(t_n, y(t_n)).$$

This suggests the method

$$y^h(t_{n+1}) = y^h(t_n) + hf(t_n, y^h(t_n)) \tag{5.2}$$

or

$$y_{n+1} = y_n + hf(t_n, y_n),$$

which is Euler's method.

2.1.2. *Numerical differentiation.* Approximating the derivative $y'(t_n)$ by the forward difference $[y(t_{n+1}) - y(t_n)]/h$ gives

$$\frac{y(t_{n+1}) - y(t_n)}{h} \approx f\big(t_n, y(t_n)\big),$$

or

$$y(t_{n+1}) \approx y(t_n) + hy'(t_n) = y(t_n) + hf\big(t_n, y(t_n)\big).$$

which again suggests Euler's method.

2.1.3. *Numerical integration.* The exact solution satisfies the integral condition

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f\big(t, y(t)\big)\, dt.$$

Approximating the integral by the left endpoint rule we get, once again,

$$y(t_{n+1}) \approx y(t_n) + hf\big(t_n, y(t_n)\big).$$

2.2. Convergence. In this section we analyze the convergence of Euler's method. For simplicity we assume that $f \in C(I \times \mathbb{R})$, $I = [t_0, t^*]$, and satisfies a uniform Lipschitz condition with respect to its second variable (so there is a unique solution defined on all of I. For any $h > 0$ define $N = N^h = \lfloor (t^* - t_0)/h \rfloor$ (so that t_N is the largest break point in I), and define $y_n = y^h(t_n)$ for $0 \le n \le N$ by Euler's method. The error is $e^h = y^h - y$. We shall measure the error in the *discrete max norm*

$$\|e^h\|_{\infty,h} = \max_{0 \le n \le N^h} |e^h(t_n)|,$$

and determine the behavior of this quantity as $h \downarrow 0$. It would be a small matter, but not now worth the effort, to use the max norm on the whole interval I.

THEOREM 5.4 (Convergence of Euler's method). $\lim_{h \downarrow 0} \|y^h - y\|_{\infty,h} = 0$.

PROOF. Define the *local truncation error* on the $(n+1)$st step by the equation

$$y(t_{n+1}) = y(t_n) + hf\big(t_n, y(t_n)\big) + T_n. \tag{5.3}$$

Thus T_n is the amount by which the exact solution fails to satisfy Euler's method. It is local to the $n+1$st step in that we compare the exact solution at the end of the step to what we would have obtained using Euler's method over the step starting with the exact solution at the beginning of the step, and ignoring all the accumulated errors up to that point.

By the mean value theorem,

$$y(t_{n+1}) = y(t_n) + hy'(\xi_n) \text{ for some } \xi_n \in (t_n, t_{n+1}),$$

while the differential equation gives $f\big(t_n, y(t_n)\big) = y'(t_n)$. Thus

$$T_n = h[y'(\xi_n) - y'(t_n)]. \tag{5.4}$$

Subtracting (5.3) from (5.2) gives

$$e_{n+1} = e_n + h[f\big(t_n, y^h(t_n)\big) - f\big(t_n, y(t_n)\big)] - T_n.$$

Setting $T = \max_{0 \le n \le N-1} |T_n|$ and using the Lipschitz condition we get

$$|e_{n+1}| \le (1 + hL)|e_n| + T, \quad 0 \le n \le N - 1,$$

and, since we start with the exact initial value, $e_0 = 0$. We now apply a simple lemma (which is easily established by induction):

LEMMA 5.5. *Let A, B, $\eta_0, \eta_1, \dots, \eta_N$ be non-negative numbers satisfying*

$$\eta_{n+1} \le A\eta_n + B, \quad n = 0, 1, \dots, N-1.$$

Then

$$\eta_n \le A^n \eta_0 + \Big(\sum_{i=0}^{n-1} A^i\Big) B, \quad n = 0, 1, \dots, N.$$

For $A \neq 1$ the quantity in parenthesis is equal to $(A^n - 1)/(A - 1)$.

Applying the lemma, we get

$$|e_n| \le \frac{(1 + hL)^n - 1}{hL} T, \quad n = 0, 1, \dots, N.$$

Since $1 + x \le e^x$ for all x, this gives

$$|e_n| \le \frac{e^{L|t^* - t_0|} - 1}{L} \frac{T}{h}, \tag{5.5}$$

and we have reduced the theorem to showing that $\lim_{h \to 0} T/h = 0$.

Since f is continuous, the solution y' is uniformly continuous on the closed interval I. Therefore, give $\epsilon > 0$ there exists $h_0 > 0$ such that $|y'(t) - y'(u)| \le \epsilon$ if $t, u \in I$ satisfy $|t - u| \le h_0$. In view of (5.4) we have $T/h \le \epsilon$ whenever $h \le h_0$. □

If we require a little extra regularity, we can get an estimate with a *rate of convergence*.

THEOREM 5.6 (Rate of convergence for Euler's method). *If $y \in C^2(I)$ then*

$$\|y^h - y\|_{\infty,h} \le Ch,$$

where

$$C = \|y''\|_\infty \frac{e^{L|t^* - t_0|} - 1}{2L}. \tag{5.6}$$

PROOF. By Taylor's theorem

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2} y''(\xi_n),$$

for some ξ_n. In view of the definition (5.3), $T_n = (h^2/2)y''(\xi_n)$, and so $|T/h| \le (h/2)\|y''\|_\infty$. The theorem therefore follows from (5.5). □

The constant C asserted in the preceeding theorem, is often very pessimistic compared to the outcome of actual computation, but the first order convergence is not. The next theorem shows (assuming a bit more smoothness), that the error at any time does indeed behave like ch, up to higher order, for some c.

THEOREM 5.7 (Asymptotic error estimate for Euler's method). *Suppose that $y \in C^3(I)$ and that $\partial f/\partial y$ and $\partial^2 f/\partial y^2$ are both continuous. Then there exists a function $\delta : I \to \mathbb{R}$ independent of h such that*

$$y^h(t_n) - y(t_n) = \delta(t_n)h + O(h^2), \quad n = 0, 1, \dots, N(h).$$

The function δ is the solution of the linear initial value problem

$$\delta'(t) = \frac{\partial f}{\partial y}\big(t, y(t)\big)\delta(t) - \frac{1}{2}y''(t), \quad \delta(t_0) = 0. \tag{5.7}$$

PROOF. Using Taylor's theorem and the definition of Euler's method we get

$$e_{n+1} = e_n + h[f\big(t_n, y^h(t_n)\big) - f\big(t_n, y(t_n)\big)] - \frac{h^2}{2}y''(t_n) - \frac{h^3}{6}y'''(\xi_n),$$

for some ξ_n. Applying Taylor's theorem to f as a function of y we get

$$f\big(t_n, y^h(t_n)\big) = f\big(t_n, y(t_n)\big) + \frac{\partial f}{\partial y}\big(t_n, y(t_n)\big)e_n + \frac{1}{2}\frac{\partial^2 f}{\partial y^2}(t_n, \zeta_n)e_n^2$$

for some ζ_n between $y(t_n)$ and $y^h(t_n)$. Combining these two expansions we get

$$e_{n+1} = \left[1 + h\frac{\partial f}{\partial y}\big(t_n, y(t_n)\big)\right] e_n - \frac{h^2}{2}y''(t_n) + R_n \tag{5.8}$$

where

$$R_n = -\frac{h^3}{6}y'''(\xi_n) - \frac{1}{2}\frac{\partial^2 f}{\partial y^2}(t_n, \zeta_n)e_n^2 h.$$

By the first order convergence of Euler's method we know that ζ_n stays bounded for all n and h and that $e_n = O(h)$. Thus $R_n = O(h^3)$.

Now we define g_n by replacing e_n by g_n in (5.8) and dropping the term R_n:

$$g_{n+1} = \left[1 + h\frac{\partial f}{\partial y}\big(t_n, y(t_n)\big)\right] g_n - \frac{h^2}{2}y''(t_n), \quad g_0 = 0. \tag{5.9}$$

To complete the proof we will show that

$$g_n = \delta(t_n)h + O(h^2), \tag{5.10}$$

$$e_n = g_n + O(h^2). \tag{5.11}$$

Now (5.10) is equivalent to

$$d_n = \delta(t_n) + O(h),$$

where $d_n = g_n/h$. In terms of d_n, (5.9) becomes

$$d_{n+1} = d_n + h\left[\frac{\partial f}{\partial y}\big(t_n, y(t_n)\big)d_n - \frac{1}{2}y''(t_n)\right], \quad g_0 = 0.$$

which is precisely Euler's method for the initial value problem (5.7). Thus (5.10) follows from Theorem 5.6.

To prove (5.11), let $k_n = e_n - g_n$. Subtract (5.9) from (5.8) to get

$$k_{n+1} = [1 + h\frac{\partial f}{\partial y}\big(t_n, y(t_n)\big)]k_n + R_n.$$

Therefore

$$|k_{n+1}| \le (1 + Kh)|k_n| + \max_n |R_n| \text{ with } K = \max_{t_0 \le t \le t^*}\left|\frac{\partial f}{\partial y}\big(t_n, y(t_n)\big)\right|.$$

Noting that $k_0 = 0$ and applying Lemma 5.5 we get

$$|k_n| \le \frac{(1+Kh)^n - 1}{Kh} \max |R_n|.$$

Bounding $(1+Kh)^n$ by $\exp(K|t^* - t_0|)$ and recalling that $R_n = O(h^3)$ we obtain the theorem. □

REMARK. The initial value problem (5.7) is linear so its solution can be written in closed terms:

$$\delta(t) = -\frac{1}{2}\rho(t) \int_{t_0}^{t} \frac{y''(s)}{\rho(s)}\, ds, \quad \rho(s) = \exp[\int_{t_0}^{t} \frac{\partial f}{\partial y}\big(s, y(s)\big)\, ds].$$

This is not very useful in practice, since we don't know y, much less y''. So the significance of the theorem is mainly the assertion that the error behaves like δh, not the particular form of δ. This is useful for many purpose, e.g., for Richardson extrapolation.

2.3. Variable step size. There is no reason why the same value of h has to be used in each step of Euler's method. We can instead vary h, determining it in advance or as we go along in some adaptive way (adaptive step size selection will be treated in § 5 of this chapter). Euler's method with variable step size is thus

$y^h(t_0) = y_0$
for $n = 0, 1, \ldots$
  choose $h_n > 0$
  $t_{n+1} = t_n + h_n$
  if $t_{n+1} \le t^*$ **then**
    $y^h(t_{n+1}) = y^h(t_n) + h_n f\big(t_n, y^h(t_n)\big)$
  else
    stop
  end if
end

Let $e = y^h - y$ denote the error. We again use the discrete maximum norm:

$$\|e\| = \max |e(t_n)|,$$

where the maximum is taken over the particular mesh points used. We also set $H = \max h_n$, the largest step size. We then again have convergence (we continue to assume that f satisfies a uniform Lipschitz condition with respect to y), and, if the solution is C^2, the convergence is first order in H:

THEOREM 5.8 (Convergence of Euler's method with variable step). *For Euler's method with variable step size*

$$\lim_{H \to 0} \|e\| = 0,$$

i.e., for all $\epsilon > 0$ there exists $H_0 > 0$ such that for any choice of steps h_n with $\max h_n \le H_0$ there holds $\max |e(t_n)| \le \epsilon$.

Moreover, if the solution $y \in C^2(I)$, then $\|e\| \le CH$, where C is again defined by (5.6).

The proof of this theorem follows very much along the lines of the proofs of Theorems 5.4 and 5.6. The bounds on the local truncation error are the same. Lemma 5.5 which is used to bound the accumulated contributions of the local truncation errors on all the steps, must be generalized as follows.

LEMMA 5.9. *Let $A_n > 1$ and $B_n \geq 0$ for $n = 0, 1, \ldots, N-1$ and let $\eta_0, \eta_1, \ldots, \eta_N \geq 0$. Suppose that*

$$\eta_{n+1} \leq A_n \eta_n + B_n, \quad n = 0, 1, \ldots, N-1.$$

Then

$$\eta_n \leq \Big(\prod_{i=0}^{n-1} A_i\Big)\eta_0 + \Big(\prod_{i=0}^{n-1} A_i - 1\Big) \sup_{0 \leq i \leq n-1} \frac{B_i}{A_i - 1}, \quad n = 1, 2, \ldots, N.$$

The proof of the lemma—which is a straightforward induction—and that of Theorem 5.8, is left to the reader.

REMARK. Without some assumption on the way the step sizes are chosen we cannot prove (or even sensibly state) an asymptotic error estimate. One possibility is to assume that the step sizes are determined using a step size parameter $h > 0$ and a continuous step modification function $\Theta : I \to \mathbb{R}_+$ by the formula $h_n = \Theta(t_n)h$. That is, as the mesh is refined the ratio of step sizes in one part of the interval to those in another is determined by Θ. In practice, Θ would reflect the nature of the solution (larger where the solution is very smooth and smaller where the solution varies rapidly). In this case it is possible to prove an asymptotic error estimate very similar to Theorem 5.7.

3. Linear multistep methods

Euler's method is an example of a one-step method: the numerical solution y_{n+1} at t_{n+1} is determined from the numerical solution at the single preceding point y_n. More generally, we can consider methods take a constant step size h and determine y_{n+1} using the values from several preceding steps:

$$y_{n+1} = \Phi(f, t_n, y_{n+1}, y_n, y_{n-1}, \ldots, y_{n-k}, h).$$

Here y_{n+1} depends on $k+1$ previous values, so this is called a $k+1$-*step method*. Notice that we have allowed y_{n+1} to appear in the right-hand side of the equation as well as the left. When this happens, we speak of an *implicit* method, and we need to solve a nonlinear equation to determine y_{n+1}. In any case, we need to determine the first $k+1$ values $y_0, y_1, \ldots y_k$ by some other method, such as a single step method.

If Φ does not depend on y_{n+1} we speak of an *explicit* method. Thus Euler's method is an explicit one-step method. The *backward Euler method*

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}),$$

and the *trapezoidal method*

$$y_{n+1} = y_n + \frac{h}{2}[f(t_{n+1}, y_{n+1}) + f(t_n, y_n)],$$

are examples of implicit one-step methods.

Notice that in each of these cases, the Φ is a linear function of y_n, y_{n+1}, $f(t_n, y_n)$, and $f(t_{n+1}, y_{n+1})$. By contrast, for the *improved Euler method*,

$$y_{n+1} = y_n + \frac{h}{2}[f(t_n, y_n) + f\big(t_{n+1}, y_n + hf(t_n, y_n)\big)],$$

which is an explicit one-step method, and the *implicit midpoint method*,

$$y_{n+1} = y_n + hf(\frac{t_n + t_{n+1}}{2}, \frac{y_n + y_{n+1}}{2}),$$

which is an implicit one-step method, the dependence of Φ on y_n and y_{n+1} is more complicated. In the next section, we shall consider such nonlinear one-step methods, while in this section we study multistep methods, but assume linear dependence. That is, we consider *linear multistep methods* with constant step size, which, by definition, are methods of the form

$$y_{n+1} = -a_0 y_n - a_1 y_{n-1} + \cdots - a_k y_{n-k} + h[b_{-1} f_{n+1} + b_0 f_n + \cdots + b_k f_{n-k}]. \tag{5.12}$$

Here we have written f_n for $f(t_n, y_n)$ (for brevity). The a_i and b_i are constants which must be given and determine the specific method. For an explicit linear multistep method $b_{-1} = 0$. It is also convenient to define $a_{-1} = 1$, so that the method can be written

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f_{n-j}.$$

One obvious question concerning implicit methods is whether the formula for the method determines y_{n+1} (whether the equation has a solution and whether it is unique). The answer is yes, at least for h sufficiently small.

THEOREM 5.10. *Let $h_0 = (|b_{-1}|L)^{-1}$ where L is the Lipschitz constant for f. Then for any $h < h_0$ and any $y_n, y_{n-1}, \ldots, y_{n-k}$ there is a unique y_{n+1} such that*

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j}). \tag{5.13}$$

PROOF. Define

$$F(z) = -\sum_{j=0}^{k} a_j y_{n-j} + hb_{-1} f(t_{n+1}, z) + h \sum_{j=0}^{k} b_j f(t_{n-j}, y_{n-j}) = hb_{-1} f(t_{n+1}, z) + \cdots,$$

where the dots represent terms not depending on z. Then the equation is simply the fixed point equation $y_{n+1} = F(y_{n+1})$. Now F is a Lipschitz with Lipschitz constant $\leq h|b_{-1}|L$. By hypothesis the Lipschitz constant is stritly less than 1, i.e., F is a contraction. The contraction mapping theorem then guarantees a unique fixed point. □

REMARK. The contraction mapping theorem also implies that the solution can be computed by fixed point iteration, and this is often done in practice. Of course only finitely many iterations are made (often quite few), introducing an additional source of error.

Several examples of linear multistep methods are listed in Table 5.1. All the methods in the table except the last can be derived from the integral relation $y(t_{n+1}) = y(t_{n-k}) + \int_{t_{n-i}}^{t_{n+1}} f\big(t, y(t)\big)\, dt$ using an appropriate interpolatory quadrature rule. Note that for the two

TABLE 5.1. Examples of linear multistep methods.

Euler's method (1-step, explicit)	$y_{n+1} = y_n + hf_n$
backward Euler method (1-step, implicit)	$y_{n+1} = y_n + hf_{n+1}$
trapezoidal method (1-step, implicit)	$y_{n+1} = y_n + \frac{h}{2}(f_{n+1} + f_n)$
midpoint method (2-step, explicit)	$y_{n+1} = y_{n-1} + 2hf_n$
Milne–Simpson method (2-step, implicit)	$y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1})$
Adams–Bashford 2-step method (explicit)	$y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1})$
Adams–Moulton 2-step method (implicit)	$y_{n+1} = y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1})$
3-step backward differentiation formula (implicit)	$y_{n+1} - 19y_n + 9y_{n-1} - 2y_{n-2} = 6f_{n+1}$

Adams methods (the Adams–Bashford and Adams–Moulton families of methods will be discussed in detail in § 3.3 below) the quadrature rule is open, i.e., contains quadrature points outside the interval of integration. The final method of the table is one of the backward differentiation formula, or BDF, family of methods, which can be derived by replacing $y'(t_{n+1})$ in the equation $y'(t_{n+1}) = f\big(t_{n+1}, y(t_{n+1})\big)$ with a backward difference approximation obtained by differentiating an interpolating polynomial.

3.1. Consistency and order. Clearly the coefficients a_j and b_j in (5.12) must be chosen carefully if the multistep method is to have a chance of being convergent. Specifically, we should have

$$\sum_{j=-1}^{k} a_j y(t_{n-j}) \approx h \sum_{j=-1}^{k} b_j y'(t_{n-j})$$

for the exact solution y. Since this has to hold for any solution to any ODE, it should hold for all reasonably smooth functions y. Thus we define

$$\mathcal{L}[y, h, t] = \sum_{j=-1}^{k} a_j y(t - jh) - h \sum_{j=-1}^{k} b_j y'(t - jh) \tag{5.14}$$

for any $y \in C^1$, $h > 0$ and $t \in \mathbb{R}$. Note that if y is the exact solution, then $\mathcal{L}[y, h, t_n]$ is simply the local truncation error at the $(n+1)st$ step.

DEFINITION. The linear multistep is *consistent* if

$$\lim_{h \downarrow 0} \max_{k \le n < N} \left| \frac{\mathcal{L}[y, h, t_n]}{h} \right| = 0$$

for all $y \in C^1(I)$. The method has *order* p (at least) if for all $y \in C^{p+1}(I)$ there exists constants $C, h_0 > 0$ such that

$$\max_{k \le n < N} \left| \frac{\mathcal{L}[y, h, t_n]}{h} \right| \le C h^p$$

whenever $h < h_0$.

Warning: it is *not* true that every method of order p converges with order p, or even converges at all!

Using Taylor's theorem, we can derive simple algebraic criteria for consistency and order.

THEOREM 5.11. *A linear multistep method is consistent if and only if*

$$\sum_{j=-1}^{k} a_j = 0, \quad \sum_{j=-1}^{k} j a_j + \sum_{j=-1}^{k} b_j = 0. \tag{5.15}$$

The method is of order p if and only if

$$\sum_{j=-1}^{k} (-j)^m a_j - m \sum_{j=-1}^{k} (-j)^{m-1} b_j = 0, \quad m = 0, 1, \ldots, p. \tag{5.16}$$

The algebraic conditions (5.15) are called the *consistency conditions* and the conditions (5.16) are called the *order conditions*. Before giving the proof we notice an immediate corollary: a method is consistent if and only if it is of order at least 1.

PROOF. The proof is just Taylor's theorem. We have $y(t_n - jh) = y(t_n) - jhy'(\xi_j)$, for some $\xi_j \in [t_n - kh, t_n + h]$, $j = -1, 0, \ldots, k$. Note that each $\xi_j \to t$ as $h \to 0$. Plugging into (5.14)

$$\begin{aligned} \mathcal{L}[y, h, t_n] &= \sum_{j=-1}^{k} a_j [y(t_n) - jhy'(\xi_j)] - h \sum_{j=-1}^{k} b_j y'(t_n - jh) \\ &= y(t_n) C_0 + hy'(t_n) C_1 + hR, \end{aligned}$$

where

$$C_0 = \sum_{j=-1}^{k} a_j, \quad C_1 = -\sum_{j=-1}^{k} j a_j - \sum_{j=-1}^{k} b_j,$$

and

$$R = -h \sum_{j=0}^{k} j a_j [y'(\xi_j) - y'(t_n)] - h \sum_{j=-1}^{k} b_j [y'(t_n - jh) - y'(t_n)].$$

By the uniform continuity of y', we see that $R/h \to 0$ as $h \to 0$. Therefore $\mathcal{L}[y, h, t_n]/h \to 0$ if and only if $C_0 = C_1 = 0$, i.e., the consistency conditions are satisfied.

Similarly, if $y \in C^{p+1}$ we have

$$y(t_n - jh) = \sum_{m=0}^{p} \frac{(-j)^m}{m!} h^m y^{(m)}(t_n) + \frac{(-j)^{p+1}}{(p+1)!} h^{p+1} y^{(p+1)}(\xi_j),$$

$$y'(t_n - jh) = \sum_{m=1}^{p} \frac{(-j)^{m-1}}{(m-1)!} h^{m-1} y^{(m)}(t_n) + \frac{(-j)^p}{p!} h^p y^{(p+1)}(\zeta_j),$$

for some $\xi_j, \zeta_j \in [t_n - kh, t_n + h]$. This gives

$$\mathcal{L}[y, h, t_n] = \sum_{m=0}^{p} h^m y^{(m)}(t_n) C_m + R,$$

where

$$C_m = \frac{1}{m!}\Big[\sum_{j=-1}^{k} (-j)^m a_j - m \sum_{j=-1}^{k} (-j)^{m-1} b_j\Big], \tag{5.17}$$

and

$$R = h^{p+1} \sum_{j=-1}^{k} [a_j \frac{(-j)^{p+1}}{(p+1)!} y^{(p+1)}(\xi_j) - b_j \frac{(-j)^p}{p!} y^{(p+1)}(\zeta_j)].$$

Since $R = O(h^{p+1})$, $\mathcal{L}/h = O(h^p)$ if and only if all the C_m vanish. □

Remark. This theorem furnishes an example of how a complicated analytic condition may sometimes be reduced to a simple algebraic criterion. Many such criteria for multistep methods can be expressed in terms of the *characteristic polynomials* of the method:

$$\rho(z) = \sum_{j=-1}^{k} a_j z^{k-j}, \quad \sigma(z) = \sum_{j=-1}^{k} b_j z^{k-j}.$$

For example, the consistency conditions are simply $\rho(1) = 0$ and $\rho'(1) = \sigma(1)$.

As an example of the use of the order conditions, we will use the method of undetermined coefficients to find the 2-step method of highest order. For a 2-step method there are five undetermined coefficients: a_0, a_1, b_{-1}, b_0, and b_1. The first five order conditions are

$$1 + a_0 + a_1 = 0, \quad 1 - a_1 - b_{-1} - b_0 - b_1 = 0, \quad 1 + a_1 - 2b_{-1} + 2b_1 = 0,$$
$$1 - a_1 - 3b_{-1} - 3b_1 = 0, \quad 1 + a_1 - 4b_{-1} + 4b_1 = 0.$$

This system of linear equations has a unique solution:

$$a_0 = 0, \quad a_1 = -1, \quad b_{-1} = \frac{1}{3}, \quad b_0 = \frac{4}{3}, \quad b_1 = \frac{1}{3},$$

which are precisely the coefficients of the Milne-Simpson method. Thus the Milne-Simpson method is the unique fourth order 2-step method.

If we consider instead *explicit* 2-step methods we no longer have the coefficient b_{-1} at our disposal. Setting b_{-1} to zero in the first four linear equations above we get

$$1 + a_0 + a_1 = 0, \quad 1 - a_1 - b_0 - b_1 = 0, \quad 1 + a_1 + 2b_1 = 0, \quad 1 - a_1 - 3b_1 = 0,$$

which gives $a_0 = 4$, $a_1 = -5$, $b_0 = 4$, $b_1 = 2$. Thus the unique explict 2-step method of order 3 is

$$y_{n+1} + 4y_n - 5y_{n-1} = h[4f_n + 2f_{n-1}].$$

Warning: we shall see below that this is not a good method. Order is not everything!

3.2. Stability and convergence. In this section we study the convergence of linear multistep methods. Recalling that the initial $k+1$ values of y_n^h must be determined by some other method, we define convergence as follows.

DEFINITION. A linear multistep method is *convergent* if whenever the initial values y_n^h are chosen so that $\max_{0\le j\le k} |e_j^h| \to 0$ as $h \to 0$, then $\max_{0\le j\le N^h} |e_j^h| \to 0$.

There is a rather complete convergence theory available for linear multistep methods with constant step size, due largely to G. Dahlquist. It is somewhat technical, and we will only sketch it below. However, for many linear multistep methods a simple proof of convergence can be given along the lines of the proof we used for Euler's method. For example, consider the midpoint method

$$y_{n+1} = y_{n-1} + 2hf(t_n, y_n).$$

Checking the order conditions we find that this method is second order, i.e., if y is a smooth function, then

$$y(t_{n+1}) = y(t_{n-1}) + 2hy'(t_n) + \mathcal{L}[y, h, t_n]$$

where $\mathcal{L}[y, h, t_n] = O(h^3)$. Now let $e_n = y_n - y(t_n)$. Then

$$e_{n+1} = e_{n-1} + 2h[f(t_n, y_n) - f\big(t_n, y(t_n)\big)] - \mathcal{L}[y, h, t_n],$$

so

$$|e_{n+1}| \le |e_{n-1}| + 2hL|e_n| + T, \quad n = 1, 2, \ldots, N-1,$$

where $T = \max |\mathcal{L}[y, h, t_n]| = O(h^3)$. This immediately implies that

$$\max(|e_{n+1}|, |e_n|) \le (1 + 2hL) \max(|e_n|, |e_{n-1}|) + T, \quad n = 0, 1, 2, \ldots, N-1,$$

whence, by Lemma 5.5,

$$\max(|e_{n+1}|, |e_n|) \le \frac{e^{2L|t^*-t_0|} - 1}{2L} \frac{T}{h} + e^{2L|t^*-t_0|} \max(|e_1|, |e_0|), \quad n = 0, 1, 2, \ldots, N-1,$$

i.e., $\|e\|_{\infty,h} \le C_1 T/h + C_2 \max(|e_1|, |e_0|)$ with C_1, C_2 independent of h. If the initial values are choosen so that $\max(|e_1|, |e_0|) \to 0$ as $h \to 0$, then we have that $\|e\|_{\infty,h} \to 0$. This is precisely the definition of convergence. For the midpoint method we see as well, that if the initial error is $O(h^2)$ the error is globally $O(h^2)$. Note the one new idea of the proof: instead of considering the propagation of the error $(e_{n-1}, e_n) \mapsto e_{n+1}$, we instead consider the propagation of the pair of values, $(e_{n-1}, e_n) \mapsto (e_n, e_{n+1})$.

Next we consider the two step method of highest order

$$y_{n+1} = -4y_n + 5y_{n-1} + h(4f_n + 2f_{n-1}).$$

We shall show that this method is not convergent even for the most trivial of initial value problems:

$$y'(t) = 0, \quad 0 \le t \le 1, \quad y(0) = 0.$$

The multistep method is then $y_{n+1} = -4y_n + 5y_{n-1}$. If we take as initial values $y_0 = 0$, $y_1 = \epsilon_h$, then the multistep method is easily seen to give

$$y_n = [1 - (-5)^n]\epsilon_h/6,$$

and so, if $h = 1/N$, $y^h(1) = y_N = [1 - (-5)^{1/h}]\epsilon_h/6$. It is clearly not sufficient that $\epsilon_h \to 0$ in order to have convergence, we need $5^{1/h}\epsilon_h \to 0$, i.e., exponentially accurate initial values.

Note that if we take exact starting values $y_0 = y_1 = 0$, then $y_n = 0$ for all n. Thus a perturbation of size ϵ in the starting values leads to a difference of size roughly $5^{1/h}\epsilon$ in the discrete solution. Thus the method is not *stable*:

DEFINITION. A linear $k+1$-step method is *stable* if for any admissable initial value problem (satisfying a Lipschitz condition) and for all $\epsilon > 0$ there exists $\delta, h_0 > 0$ such that if $h \le h_0$ and two choices of starting values y_j and $\bar{y}_j$ are chosen satisfying

$$\max_{0\le j\le k} |y_j - \bar{y}_j| \le \delta,$$

then the corresponding approximate solutions satisfy

$$\max_{0\le j\le N} |y_j - \bar{y}_j| \le \epsilon.$$

If, as above, we consider the trivial differential equation $y' = 0$, then the general linear multistep method becomes

$$y_{n+1} + \sum_{j=0}^{k} a_j y_{n-j} = 0, \quad n = k, k+1, \ldots.$$

This is an example of a *homogeneous linear difference equation of order $k+1$ with constant coefficients*. For such an equation there is a simple approach to finding the general solution in closed form, which we shall now briefly present. We shall allow complex solutions. Then there is clearly a unique solution $(y_n)_{n=0}^{\infty}$ for any choice of initial values $(y_n)_{n=0}^{k} \in \mathbb{C}^{k+1}$. Thus the space of solutions is a complex vectorspace of dimension $k+1$.

To find the general solution, we first try for a solution of the form $(\lambda^n)_{n=0}^{\infty}$. Plugging this into difference equation, we see that it is a solution if and only if λ is a root of the characteristic polynomial $\rho(t) = t^{k+1} + \sum_{j=0}^{k} a_j t^{k-j}$. If there are $k+1$ distinct roots λ_i, $i = 0, \ldots, k$, we obtain a full basis of $k+1$ linearly independent solutions in this way. To see linear independence, notice that the matrix of initial values $(\lambda_i^n)_{0\le i,n\le k}$ is a Vandermonde matrix, and so nonsingular. In the case of multiple roots this does not give a complete set of solutions. In the case where λ is a double root, we obtain, in addition to the solution $(\lambda^n)_{n=0}^{\infty}$ the solution $(n\lambda^n)_{n=0}^{\infty}$. For a triple root we find that $(n^2\lambda^n)_{n=0}^{\infty}$ is a solution as well, and so forth. In this case we obtain a complete set of solutions in terms of the roots of the characteristic polynomial. We illustrate with a simple example unrelated to linear multistep methods. The difference equation $y_{n+1} = y_n + y_{n-1}$ together with the initial conditions $y_0 = 0$, $y_1 = 1$ defines the Fibonacci sequence. The characteristic polynomial of the difference equation is $\rho(t) = t^2 - t - 1$ with roots $(1 \pm \sqrt{5})/2$. Thus the general solution is $y_n = c_0[(1 - \sqrt{5})/2]^n + c_1[(1 + \sqrt{5})/2]^n$. Imposing the initial conditions $y_0 = y_1 = 1$ we

find Binet's formula for the nth Fibonacci number:

$$y_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right].$$

Returning to linear multistep methods, consider again the trivial differential equation $y' = 0$, so the method reduces to a homogeneous linear difference equation. If the first characteristic polynomial has a root of modulus greater than 1, then the difference equation will have exponentially growing solutions, and so $y^h(1)$ which equals y_N when $h = 1/N$, will grow exponentially unless the initial values are chosen to *exactly* suppress the coefficient of this solution. Thus the method will not be stable. The same is true if the characteristic polynomial has a multiple root of modulus 1, although the instability will be much less severe because the growth will be algebraic rather than exponential. These considerations lead to the following definition and theorem, another example in which a complex analytic condition for the behavior of a linear multistep condition is reduced to a simple algebraic condition.

DEFINITION. A linear multistep method satisfies the *root condition* if

(1) every root of its first characteristic polynomial has modulus ≤ 1,
(2) all roots of modulus 1 are simple.

THEOREM 5.12. *A stable linear multistep method satisfies the root condition. If a consistent linear multistep method satisfies the root conditions, it is stable.*

The considerations above lead to the proof that the root condition is necessary for stability. To prove the reverse implication (which obviously lies deeper, since it concerns an arbitrary differential equation, and not just the trivial equation $y' = 0$), will be discussed below.

We now state the fundamental convergence theorem for linear multistep methods. In view of Theorems 5.11 and 5.12, this theorem states that it can be determined whether or not a linear multistep method is convergent simply by checking some algebraic conditions concerning its characteristic polynomials.

THEOREM 5.13. *A linear multistep method is convergent if and only if it is consistent and stable.*

We only sketch the proof. The statement consists of three parts:

- convergence implies consistency,
- convergence implies stability,
- consistency and stability imply convergence.

The first statement is easy. For the trivial initial value problem $y' = 0$, $y(0) = 1$,

The first two statements are straightforward (assuming Theorem 5.12). They can be proven considering only the simple differential equations $y' = 0$, for the root condition and the first consistency condition, and $y' = 1$, for the second consistency condition. The difficult part is the third part (and the second statement in Theorem 5.12 is proven in the same way). To show the main idea, we prove it in the case of an explicit two step method (most of the ideas of the full proof arise already in this case).

We write the two step method

$$y_{n+1} = -a_0 y_n - a_1 y_{n-1} + h(b_0 f_n + b_1 f_{n-1})$$

in matrix form:

$$\begin{pmatrix} y_n \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_1 & -a_0 \end{pmatrix} \begin{pmatrix} y_{n-1} \\ y_n \end{pmatrix} + \begin{pmatrix} 0 \\ h(b_0 f_n + b_1 f_{n-1}) \end{pmatrix}.$$

The exact solution satisfies a similar equation with the addition of the local truncation error:

$$\begin{pmatrix} y(t_n) \\ y(t_{n+1}) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_1 & -a_0 \end{pmatrix} \begin{pmatrix} y_{n-1} \\ y_n \end{pmatrix} + \begin{pmatrix} 0 \\ h[b_0 f\big(t_n, y(t_n)\big) + b_1 f\big(t_{n-1}, y(t_{n-1})\big)] + \mathcal{L}[y, h, t_n] \end{pmatrix}.$$

Subtracting gives an equation for the error. Let

$$e_n = y_n - y(t_n), \quad E^n = \begin{pmatrix} e_{n-1} \\ e_n \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ -a_1 & -a_0 \end{pmatrix},$$

$$Q^n = \begin{pmatrix} 0 \\ h[b_0 f_n - f\big(t_n, y(t_n)\big) + b_1 f_{n-1} - f\big(t_{n-1}, y(t_{n-1})\big)] - \mathcal{L}[y, h, t_n] \end{pmatrix}.$$

Then

$$E^{n+1} = AE^n + Q^n, \quad n = 1, 2, \ldots \tag{5.18}$$

Note that $\|Q^n\| \le Ch\|E^n\| + T$ where $T = \max_n |\mathcal{L}[y, h, t_n]| = O(h^{p+1})$ where p is the order of the method (and we use the l_∞ norm for the vectors). Iterating (5.18) gives

$$E^n = A^{n-1}E^1 + \sum_{j=1}^{n-1} A^{n-1-j}Q^j, \quad n = 1, \ldots, N.$$

Now we use the root condition. The characteristic polynomial of the matrix A is $\rho(t) = t^2 + a_0 t + a_1$, i.e., the first characteristic polynomial of the linear multistep method. Thus the eigenvalues of A all have modulus ≤ 1 with only simple eigenvalues of modulus 1. This is precisely the condition for the powers of A to remain bounded: $\sup_{n\ge 0}\|A^n\| < \infty$. Thus we have

$$\|E^n\| \le C(\|E^1\| + \sum_{j=1}^{n-1}\|Q^j\|) \le C'(h\sum_{j=1}^{n-1}\|E^j\| + \|E^1\| + \frac{T}{h}), \quad n = 1, \ldots, N,$$

for some constants C and C'. Now from this relation it follows (by a sort of discrete Gronwall lemma[1] that

$$\max_{1\le n\le N}\|E^n\| \le K(\|E^1\| + \frac{T}{h}),$$

for a suitable constant K. For a consistent method $T/h \to 0$ as $h \to 0$, so convergence follows.

We end this section with a statement of the attainable order of multistep methods. Counting coefficients and order conditions one would guess (correctly) that the highest order attainable by a k step method is $2k$. However, such a method is not stable for any $k > 1$. The next theorem states that only about half this order is attainable by a stable method.

---

[1]Specifically, if $\xi_m \le \alpha \sum_{j=1}^{m-1} \xi_j + \beta$, $m = 1, 2, \ldots$, where the ξ_m, α, and β are non-negative, then $\xi_m \le \beta(1+\alpha)^{m-1}$.

THEOREM 5.14 (First Dahlquist barrier). *The highest order of a stable k step method is $k+1$ if k is odd and $k+2$ if k is even.*

3.3. Adams methods. The Adams methods are linear multistep methods with the best possible stability properties. Namely the first characteristic polynomial has all its roots at the origin except for the mandatory root at 1. That is, the first characteristic polynomial of a $k+1$ step Adams method is $\rho(t) = t^{k+1} - t^k$. There are two Adams methods for each k, an explicit method, called an Adams–Bashford method, and an implicit method, called an Adams–Moulton method. They can be derived by quadrature as follows. We start with the equation

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f\big(t, y(t)\big)\, dt.$$

Now, assuming that y_j is known for $j \le n$, let $P(t) \in \mathcal{P}_k$ denote the Lagrange interpolating polynomial satisfying $P(t_j) = f_j := f(t_j, y_j)$, $j = n, n-1, \dots, n-k$. We then define

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} P(t)\, dt,$$

which is the $k+1$ step Adams–Bashford method. For $k = 0$, 1, and 2, the formulas are

$$\begin{aligned}
y_{n+1} &= y_n + hf_n, \quad \text{(Euler's method)} \\
y_{n+1} &= y_n + h(\frac{3}{2}f_n - \frac{1}{2}f_{n-1}), \\
y_{n+1} &= y_n + h(\frac{23}{12}f_n - \frac{4}{3}f_{n-1} + \frac{5}{12}f_{n-2}).
\end{aligned}$$

Using Lagrange's formula for the interpolating polynomial, we can easily find the general formula

$$y_{n+1} = y_n + h\sum_{j=0}^{k} b_j f_{n-j},$$

where

$$b_j = \frac{1}{h}\int_{t_n}^{t_{n+1}} \prod_{\substack{0\le m\le k\\ m\ne j}} \frac{t - t_{n-m}}{t_{n-j} - t_{n-m}} dt = \int_0^1 \prod_{\substack{0\le m\le k\\ m\ne j}} \frac{m+t}{m-j} dt.$$

(Note that the first expression can be used also in the case of non-constant step size.)

The Adams–Moulton methods are constructed similarly, except that $P \in \mathcal{P}_{k+1}$ interpolates f_{n-j} at t_{n-j} for $j = -1, \dots, k$. The first few formulas are

$$\begin{aligned}
y_{n+1} &= y_n + h(\frac{1}{2}f_{n+1} + \frac{1}{2}f_n), \quad \text{(trapezoidal method)} \\
y_{n+1} &= y_n + h(\frac{5}{12}f_{n+1} + \frac{2}{3}f_n - \frac{1}{12}f_{n-1}), \\
y_{n+1} &= y_n + h(\frac{3}{8}f_{n+1} + \frac{19}{24}f_n - \frac{5}{24}f_{n-1} + \frac{1}{24}f_{n-2}).
\end{aligned}$$

We may also think of the backward Euler method as an Adams–Moulton method with $k = -1$.

It is easy to check the order of the Adams methods. E.g., for the $k+1$ step Adams–Bashford method

$$\mathcal{L}[y,h,t_n] = \int_{t_n}^{t_{n+1}} [y'(t) - P(t)]\,dt,$$

where $P \in \mathcal{P}_k$ interpolates y' at $t_n, \ldots, t_{n-k}$. By the Newton error formula for Lagrange interpolation,

$$y'(t) - P(t) = y'[t_n, \ldots, t_{n-k}, t](t-t_n)\cdots(t-t_{n-k}),$$

so, using the integral mean value theorem,

$$\begin{aligned}\mathcal{L}[y,h,t_n] &= y'[t_n, \ldots, t_{n-k}, \xi] \int_{t_n}^{t_{n+1}} (t-t_n)\cdots(t-t_{n-k})\,dt \\ &= \frac{1}{(k+1)!} y^{(k+2)}(\eta) \int_{t_n}^{t_{n+1}} (t-t_n)\cdots(t-t_{n-k})\,dt.\end{aligned}$$

for some ξ. The integral is clearly order $O(h^{k+2})$, so the method is of order $k+1$. In fact it equals $\gamma_{k+1}h^{k+2}$ where $\gamma_1 = 1/2$, $\gamma_2 = 5/12$, $\gamma_3 = 3/8$, ... Thus for the Adams–Bashford methods, the order is equal to the number of steps.

Similarly, for k step Adams–Moulton method also has order $k+1$ and the local truncation error satisfies

$$\mathcal{L}[y,h,t_n] = \frac{1}{(k+1)!} y^{(k+1)}(\eta)\gamma^*_{k+1}h^{k+2}.$$

The first few γ^*_k are $\gamma^*_0 = -1/2$ (for backward Euler), $\gamma^*_1 = -1/12$ (for trapezoidal), $\gamma^*_2 = -1/24$, ... Thus to achieve the same order $k+1$ we can use a $k+1$ step Adams–Bashford method or a k step Adams–Moulton method. The coefficient of h^{k+2} in the local truncation error is significantly smaller for the Adams–Moulton method. When we study the notion of absolute stability later, we shall find other advantages of the Adams–Moulton methods over the Adams–Bashford methods.

3.4. Predictor-corrector schemes. To implement an Adams–Moulton method, or any implicit method, we need a way to solve, at least approximately, the nonlinear equation arising at each time step. The most common method is to solve this equation approximately using a small number of fixed point iterations starting from an initial approximation obtained by an explicit method. Thus the scheme takes the form:

1. predict:	$p_{n+1} = E(y_n, y_{n-1}, \ldots, f_n, f_{n-1}, \ldots)$
2. evaluate:	$f^p_{n+1} = f(t_{n+1}, p_{n+1})$
3. correct:	$y^{(1)}_{n+1} = I(y_n, y_{n-1}, \ldots, f^p_{n+1}, f_n, f_{n-1}, \ldots)$
4. evaluate:	$f^{(1)}_{n+1} = f(t_{n+1}, y^{(1)}_{n+1})$
5. correct:	$y^{(2)}_{n+1} = I(y_n, y_{n-1}, \ldots, f^{(1)}_{n+1}, f_n, f_{n-1}, \ldots)$
6. evaluate:	$f^{(2)}_{n+1} = f(t_{n+1}, y^{(2)}_{n+1})$
	$\vdots$

Here $E(y_n, y_{n-1}, \ldots, f_n, f_{n-1}, \ldots)$ refers to some explicit scheme, e.g., an Adams–Bashford scheme, and $I(y_n, y_{n-1}, \ldots, f_{n+1}, f_n, \ldots)$ to some implicit, e.g., Adams–Moulton, scheme. At

some point we stop and declare $y_{n+1} = y_{n+1}^{(m)}$. We could stop in response to some stopping criterion, but more commonly, a fixed, usually small, number of iterations are made. For example, we may stop the iteration after step 4, and accept $y_{n+1}^{(1)}$. This would then be referred to as a PECE scheme. Other possibilities are PECECE, P(EC)^{3}E, etc. As a simple example, consider the PECE method with a 2 step Adams–Bashford predictor and a 2 step Adams–Moulton corrector. This gives

$$p_{n+1} = y_n + h[3f(t_n, y_n) - f(t_{n-1}, y_{n-1})]/2,$$
$$y_{n+1} = y_n + h[5f(t_{n+1}, p_{n+1}) + 8f(t_n, y_n) - f(t_{n-1}, y_{n-1})]/12.$$

Thus

$$y_{n+1} = y_n + h\big[5f\big(t_{n+1}, y_n + h[3f(t_n, y_n) - f(t_{n-1}, y_{n-1})]/2\big) + 8f(t_n, y_n) - f(t_{n-1}, y_{n-1})\big]/12.$$

Thus the resulting method is a *nonlinear* 2 step method.

The analysis of the error of such a scheme is relatively straightforward. We just briefly sketch the main ideas for a PECE scheme. There are two contributions to the local truncation error, one arising from predictor formula and one arising from the corrector formula. If the predictor formula has order p and the corrector formula has order q, so their local truncation errors are $O(h^{p+1})$ and $O(h^{q+1})$ respectively, then the local truncation error for the PECE scheme will be $O(h^{p+2}) + O(h^{q+1}) = O(h^{\min(p+2,q+1)})$. The extra order in predictor contribution comes from the fact that the term $f(t_{n+1}, p_{n+1})$ involving the predictor is multiplied by h in the formula for y_{n+1}. Thus if $p \geq q - 1$, the local truncation error will be $O(h^{q+1})$ just as if the predictor equation were solved exactly. In fact, if we take $p \geq q$, the local truncation error for the PECE scheme will be asymptotically equal to that for the full implicit scheme (i.e., they will agree up to terms of higher order). Thus the most common choices are either $p = q - 1$ or $p = q$. For a PECE scheme based on Adams methods, we might use the k step Adams–Bashford for predictor (order k) and the k step Adams–Moulton for corrector (order $k + 1$), and thus achieve a method of order $k + 1$. Or we might take a $k + 1$ step Adams–Bashford predictor, which would still achieve the same order, but which we would expect to have an asymptotically smaller error. Yet another possibility is to use a PECECE scheme with k step Adams methods for both predictor and corrector. This would again produce asymptotically the same error as the fully implicit k step Adams–Moulton method.

4. One step methods

We have seen three linear one step methods thus far:

$$y_{n+1} = y_n + hf_n, \quad \text{(Euler's method)}$$
$$y_{n+1} = y_n + hf_{n+1}, \quad \text{(backward Euler method)}$$
$$y_{n+1} = y_n + h(\frac{1}{2}f_{n+1} + \frac{1}{2}f_n). \quad \text{(trapezoidal method)}$$

Actually these are all special cases of the θ-method

$$y_{n+1} = y_n + h[(1 - \theta)f_{n+1} + \theta f_n],$$

which is a consistent stable linear 1 step method for any real θ (of course, all consistent 1 step methods are stable). Except for Euler's method, all these methods are implicit.

We may also obtain an explicit, but nonlinear, single step method using Euler as a predictor and one of the other methods as a corrector in a PECE (or PECECE, ...) scheme. E.g., with the trapezoidal method as corrector, we get

$$y_{n+1} = y_n + h[f(t_n, y_n) + f\big(t_{n+1}, y_n + hf(t_n, y_n)\big)]/2,$$

which is known as the improved Euler method or Heun's method.

An example of an implicit, nonlinear single step method is a variant of the theta method:

$$y_{n+1} = y_n + hf\big((1-\theta)y_{n+1} + \theta y_n\big).$$

All single step methods may be written in the form

$$y_{n+1} = y_n + h\Phi(f; t_n, y_n, h),$$

where Φ is called the relative increment function. For an explicit method, $\Phi(f; t_n, y_n, h)$ is an explicit function of t_n and y_n involving f and h. For implicit methods, $\Phi(f; t_n, y_n, h)$ is given in terms of the solution to an algebraic equation. In fact we can even consider the exact relative increment function

$$\Delta(f; t_n, y_n, h) = \frac{y(t_{n+1}) - y(t_n)}{h},$$

where u is the exact solution to the differential equation $u'(t) = f\big(t, y(t)\big)$, $y(t_n) = y_n$. This one step method gives the exact solution, but is not implementable. A single step method with increment function Φ is of order p (its local truncation error is $O(h^{p+1})$) if and only if

$$\Phi(f; t_n, y_n, h) = \Delta(f; t_n, y_n, h) + O(h^p).$$

4.1. Taylor series methods. In a sense, there is a canonical pth order one step method. Taylor's theorem tells us that

$$y(t+h) = y(t) + hy'(t) + \cdots + \frac{h^p}{p!}y^{(p)}(t) + O(h^{(p+1)}).$$

Now the differential equation tells us that $y'(t) = f\big(t, y(t)\big)$. But we can also differentiate the differential equation to get

$$y''(t) = \frac{\partial f}{\partial t}\big(t, y(t)\big) + f\big(t, y(t)\big)\frac{\partial f}{\partial y}\big(t, y(t)\big),$$

or, briefly, $y'' = f_t + ff_y =: Df$, the *total derivative* of f. Differentiating again gives

$$y''' = f_{tt} + 2ff_{ty} + f_tf_y + ff_y^2 + f^2f_{yy} =: D^2f,$$

$y^{(4)} = D^3f$, etc. (The expressions for the total derivatives get very complicated, very quickly.) The p-term Taylor series method is the single step method

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2}Df_n + \cdots + \frac{h^p}{p!}D^{p-1}f_n,$$

which clearly has order p. This method can be implemented in some cases, but it requires evaluation of the partial derivatives of f, and is not commonly used.

4.2. Runge–Kutta methods. Now let us return to Heun's method, which we derived as a PECE scheme with Euler's method as predictor and the trapezoidal method as corrector. The relative increment function is

$$\Phi = [f(t_n, y_n) + f\big(t_{n+1}, y_n + hf(t_n, y_n)\big)]/2.$$

Expanding the second term in brackets in a Taylor polynomial around (t_n, y_n) we get

$$\Phi = f + \frac{h}{2}(f_t + ff_y) + O(h^2),$$

where the f and its derivatives are evaluated at (t_n, y_n). Comparing with the Taylor expansion of the exact relative increment we see that $\Phi = \Delta + O(h^2)$, and hence Heun's method is second order. (If we expand Φ out to terms of order h^2, we get

$$\Phi = f + \frac{h}{2}(f_t + ff_y) + \frac{h^2}{4}(f_{tt} + 2ff_{ty} + f^2 f_{yy}) + O(h^3).$$

The coefficient of h^2 does *not* agree with $D^2 f/3!$, so Heun's method is definitely not of higher than second order.

Heun's method is an explicit single step method which requires 2 evaluations of f per step (but does not require knowledge of the derivatives of f). This is an example of a Runge–Kutta (RK) method. The general form of an explicit RK method is

$$y_{n+1} = y_n + h(b_1\phi_1 + \cdots + b_q\phi_q),$$

where $\phi_i = f(t_n + c_i h, \eta_i)$ and

$$\begin{aligned}
\eta_1 &= y_n, \\
\eta_2 &= y_n + ha_{2,1}\phi_1, \\
\eta_3 &= y_n + h(a_{3,1}\phi_1 + a_{3,2}\phi_2), \\
&\vdots \\
\eta_q &= y_n + h(a_{q,1}\phi_1 + a_{q,2}\phi_2 + \cdots + a_{q,q-1}\phi_{q-1}).
\end{aligned}$$

To specify a particular method of this form we must give the *number of stages* $q \geq 1$, the coefficients b_i, c_i, $1 \leq i \leq q$, and a_{ij}, $1 \leq i \leq q$, $1 \leq j < i$. The b_i are called the *weights*, the c_i (or the points $t_n + c_i h$) the *nodes*, and the η_i, or, sometimes, the ϕ_i, are called the *stages* of the RK method. Thus, a RK method is specified by weight and node vectors, $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^q$, and a strictly lower triangular matrix $A \in \mathbb{R}^{q\times q}$. These are often recorded in a *RK tableau* thus:

$\boldsymbol{c}$	A
	$\boldsymbol{b}^T$

The tableau for Heun's method is

0	
1	1
	$\frac{1}{2}$ $\frac{1}{2}$

where we don't bother to write the zeros in the upper triangle of A. Other well-known RK methods are the modified Euler method, Heun's 3 stage method, and the Runge–Kutta–Simpson 4-stage method (often referred to as *the* Runge–Kutta method), whose tableaux are

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array} \qquad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

respectively. For any of these methods, it is an elementary, but tedious computation (easily carried out by a computer algebra program) to determine their order by an expansion in Taylor series. One finds that the modified Euler method is of order 2, Heun's 3-stage method is of order 3, and the Runge–Kutta-Simpson method is of order 4. These are the highest possible orders achievable with the given number of stages. That is, it can be verified, by similarly tedious computation, that the highest order achievable by a Runge–Kutta method with q stages is q for each q from 1 to 4. However for q from 5 to 7 order $q-1$ is the best achievable, and for $q = 8$, order 6 is the best achievable. Butcher has developed graph theoretic techniques for constructing and analyzing RK methods which are necessary when working with more than a modest number of stages.

It is also possible to use implicit Runge–Kutta methods. These are defined by the same equations, except that the coefficient matrix A is no longer required to be lower triangular. This requires the solution of a coupled system of q nonlinear equations to determine the η_i at each step (and if, as is usually the case, we are solving a system of, say, d, ODEs, each η_i is itself a vector with d components, and we have to solve a system of qd coupled nonlinear algebraic equations). For this reason one rarely uses implicit RK methods with more than a few stages.

4.3. Convergence of single step methods. We now consider the convergence of the general single-step method

$$y_{n+1} = y_n + h\Phi(f; t_n, y_n, h).$$

We assume as usual that $f(t, y)$ belongs to $C([t_0, t^*] \times \mathbb{R})$ and satisfies a uniform Lipschitz condition with respect to y. We assume that the relative increment function $\Phi(f; t, y, h)$, which is defined for $t \in [t_0, t^*]$, $y \in \mathbb{R}$, $h \in [0, t^* - t]$, is continuous there. Moreover we assume the uniform Lipschitz condition

$$|\Phi(f; t, y, h) - \Phi(f; t, \bar{y}, h)| \le K|y - \bar{y}|,$$

whenever (t, y, h) and $(\bar{t}, y, h)$ belong to the domain of Φ. For Taylor series methods it is easy to deduce the continuity and Lipschitz condition for Φ for smooth f. For Runge–Kutta methods continuity is evident and the Lipschitz condition is not hard to obtain using the same condition for f.

A single step method with relative increment function Φ is *consistent* if

$$\lim_{h \downarrow 0} [\Phi(f; t, y, h) - \Delta(f; t, y, h)] = 0.$$

In view of continuity we can state this simply as $\Phi(f;t,y,0) = f(t,y)$. The method has order p if $|\Phi(f;t,y,h) - \Delta(f;t,y,h)| \le Ch^p$ (whenever f is smooth).

THEOREM 5.15. *A single step method is convergent if and only if it is consistent.*

PROOF. Define y^h by the single step method:

$$y^h(t_{n+1}) = y^h(t_n) + h\Phi\big(f;t_n,y^h(t_n),h\big),$$

starting from $y^h(t_0) = y_0$. We claim that, whether or not the method is consistent, as $h \to 0$ y^h converges to the solution of the boundary value problem

$$z'(t) = g\big(t,z(t)\big), \quad z(t_0) = y_0,$$

where $g(t,y) = \Phi(f;t,y,0)$. Since, by definition, the method is consistent if and only if $f = g$, this will prove the theorem. (If $f(\bar t,\bar y) \ne g(\bar t,\bar y)$ for some $(\bar t,\bar y)$, then we let $y(t)$ be the solution to $y' = f(t,y)$ passing through $\bar y$ at $\bar t$. Then either $y(\bar t) \ne z(\bar t)$ or $y'(\bar t) = f(\bar t,\bar y) \ne g(\bar t,\bar y) = z'(\bar t)$, so, in either case, $z \not\equiv y$.)

To prove the claim, note that

$$z(t_{n+1}) = z(t_n) + hz'(\xi_n) = z(t_n) + hg\big(\xi_n,z(\xi_n)\big),$$

for some $\xi_n \in (t_n,t_{n+1})$. Putting $e_n = z(t_n) - y^h(t_n)$ and subtracting we get

$$e_{n+1} = e_n + h[g\big(\xi_n,z(\xi_n)\big) - \Phi\big(f;t,y^h(t_n),h\big)].$$

Now the term in brackets may we decomposed as

$$\begin{aligned}&[g\big(\xi_n,z(\xi_n)\big) - g\big(t_n,z(t_n)\big)]\\ +&[\Phi(f;t_n,z(t_n),0) - \Phi\big(f;t_n,z(t_n),h\big)]\\ +&[\Phi\big(f;t_n,z(t_n),h\big) - \Phi\big(f;t_n,y^h(t_n),h\big)].\end{aligned}$$

The first two terms tend to 0 with h by uniform continuity of g and Φ, and the last term can be bounded by $K|e_n|$ using the Lipschitz condition. Thus we have

$$|e_{n+1}| \le (1+Kh)|e_n| + \omega(h),$$

where $\lim_{h\to0}\omega(h) = 0$. Since we also have $e_0 = 0$, it follows, in the usual way, that the sequence e_n tends to 0. □

5. Error estimation and adaptivity

Just as for numerical quadrature, a code for the numerical solution of ordinary differential equations can be much more efficient if the step size is adjusted adaptively to the solution. Very roughly speaking, in parts of its domain where the solution is rapidly varying, and so has large derivatives, a smaller step size will be needed than in places where the solution is more slowly varying. In this section we will study the main ideas behind some modern adaptive ODE solvers.

The first step in designing an adaptive solver is to establish the goal of the solver. In many cases the user may wish to control some global error measure, like the discrete maximum norm error

$$\max_{0\le n\le N} |y^h(t_n) - y(t)|.$$

Thus we could ask the user to input a tolerance ϵ and take as the goal of the code to select step sizes which achieve $\max |y^h(t_n) - y(t)| \leq \epsilon$ as inexpensively (i.e., with as few steps) as possible. Unfortunately, the global error is difficult to estimate and to control. Indeed, as we have seen, small error committed near the beginning of the interval of solution, may, or may not, lead to large errors later in the interval. This cannot be known a code at least until the solution process is complete, and thus there is no way to select the step size near the beginning of the computation to control the final global error. This being the case, most adaptive ODE solvers try to control a different error quantity known as the *local error*.

DEFINITION. Let $w_{n-1}(t)$ be the solution to the differential equation $w'_{n-1} = f(t, w_{n-1})$ satisfying $w_{n-1}(t_{n-1}) = y^h(t_{n-1})$. The local error at the nth step is defined to be

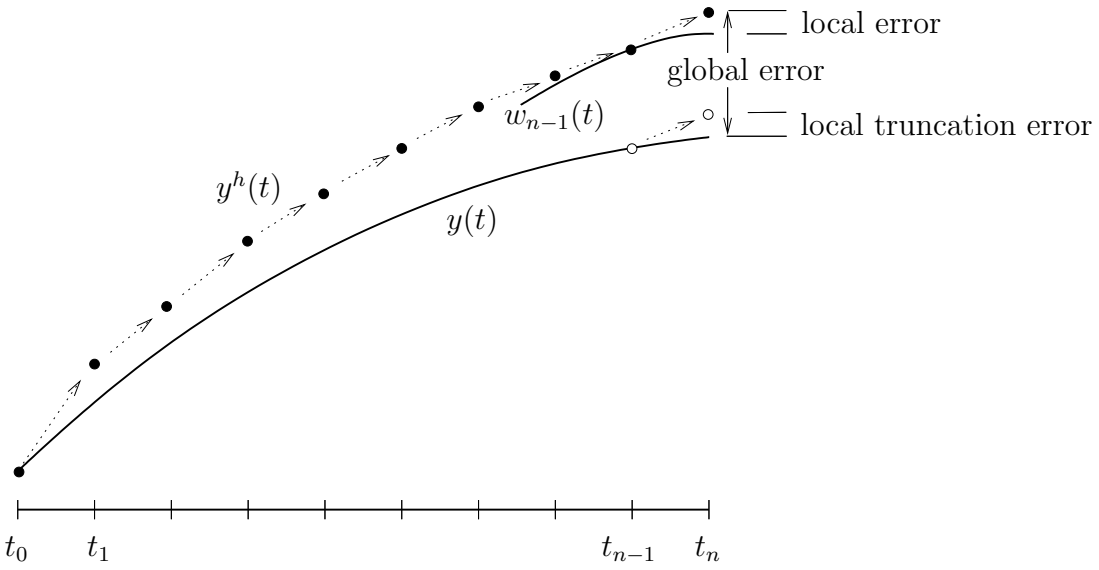
$$l_n = y^h(t_n) - w_{n-1}(t_n).$$

Thus, for a single step method with relative increment function Φ

$$l_n = h[\Phi(f; t_{n-1}, y^h(t_{n-1}), h) - \Delta(f; t_{n-1}, y^h(t_{n-1}), h)].$$

Figure 5.5 shows the local error, the local truncation error, and the global error.

FIGURE 5.5. Local error, local truncation error, and global error.



The role of the local error is clarified by the equation

$$y^h(t_n) - y(t_n) = [y^h(t_n) - w_{n-1}(t_n)] + [w_{n-1}(t_n) - y(t_n)].$$

The first bracketed quantity is the local error at the nth step. The second bracketed quantity is the difference between two *exact* solutions of the ODE, one starting at $y^h(t_{n-1})$ at the beginning of the step and the other starting at $y(t_n)$. Thus this quantity represents the error at the end of the nth step due to the accumulated global error at the start of the step propagated forward by the differential equation, and the equation says that the error at the end of the step is the sum of the local error on the step plus the accumulated errors made previously and propagated forward.

In a similar way, we may write

$$y^h(t_n) - y(t_n) = w_n(t_n) - w_0(t_n) = \sum_{i=1}^{n} [w_i(t_n) - w_{i-1}(t_n)].$$

Here w_i and w_{i-1} are both exact solutions of the ODE. They differ by $w_i(t_i) - w_{i-1}(t_i) = y^h(t_i) - w_{i-1}(t_i) = l_i$ at t_i, and so the term $w_i(t_n) - w_{i-1}(t_n)$ represents the error at t_n obtained by propagating the forward the local error on the ith step to the end of the interval. From the stability theorem, Theorem 5.3, we have for $1 \le i \le n \le N$,

$$|w_i(t_n) - w_{i-1}(t_n)| \le C|l_i|$$

where $C = \exp(L|t_N - t_0|)|l_i|$. In this way we see that the global error is bounded by a multiple of the sum of the local errors, with the multiplier depending on the stability of the differential equation.

In light of these considerations, a useful goal for an adaptive solver is to choose the number of steps and the step sizes to achieve as economically as possible that $\sum_{n=1}^{N} |l_i| \le \epsilon$, where ϵ is a user-supplied tolerance. The basic stucture of such solver is summarized, at a very high level, in the following algorithm:

Initialization. $n \leftarrow 0$, $h \leftarrow$ initial value
Step computation. $t_{n+1} \leftarrow t_n + h$, $y_{n+1} \leftarrow y_n + h\Phi(f; t_n, y_n, h)$
Error estimation. est $\leftarrow$ estimate of l_{n+1}
if est is small enough **then**
  Step acceptance. $n \leftarrow n + 1$, $h \leftarrow$ trial value for next step
  return to step computation.
else
  Step rejection. $h \leftarrow$ new trial value for current step
  return to step computation.
end if

5.1. Error estimation and step size selection. To fill out this algorithm, we need to answer several questions:

(1) How should the initial step size be chosen?
(2) How can we estimate the local error?
(3) When is the local error "small enough" to accept the step?
(4) After an acceptable step, how should the step size for the next step be predicted?
(5) After rejecting a step, how should the step size be adjusted for the next trial?

5.1.1. *The local error per unit step criterion.* Just as when we discussed adaptive quadrature we arrived at the error per unit step criterion, a good criterion for step acceptance for adaptive ODE solvers is the *local error per unit step* criterion. This means that given a desired bound ϵ for the sum of the local errors, we accept a step of size h_n with a local error of size l_n if

$$|l_n| \le \epsilon \frac{h_n}{T - t_0}.$$

If this criterion is met at each step, then we have

$$\sum_{n=1}^{N} |l_n| \le \frac{\epsilon}{T - t_0} \sum_{n=1}^{N} h_n = \epsilon,$$

as desired. If we set $\text{tol} = \epsilon/(T - t_0)$, our step acceptance criterion is just $|l_n|/h_n \le \text{tol}$.

5.1.2. *Step size design.* Next we turn to answering questions 4 and 5. Our goal is to find the largest step size h_n so that the local error per unit step $|l_n|/h_n$ does not exceed *tol*. Now

$$\frac{l_n}{h_n} = \Phi(f; t_{n-1}, y^h(t_{n-1}), h_n) - \Delta(f; t_{n-1}, y^h(t_{n-1}), h_n) = Ch_n^p + O(h_n^{p+1}),$$

where p is the order of the single step method and the constant C depends on the differential equation and the step. Now suppose we have somehow computed est, a good approximation for $|l_n|/h_n$, and found that it exceeds tol, and so we have rejected the step.

STOPPED HERE

5.1.3. *Estimation of the local error.*

5.2. Numerical example.

6. Stiffness

Consider the very simple linear system with 2 components

$$\boldsymbol{y}' = A\boldsymbol{y}, \quad \boldsymbol{y}(0) = \boldsymbol{y}_0, \tag{5.19}$$

where

$$A = \begin{pmatrix} -33.4 & 66.6 \\ 33.3 & -66.7 \end{pmatrix}, \quad \boldsymbol{y}_0 = \begin{pmatrix} 3 \\ 0 \end{pmatrix}.$$

The matrix A has eigenvalues -100 and $-1/10$ and corresponding eigenvalues $\boldsymbol{x}_1 = (1, -1)^T$, $\boldsymbol{x}_2 = (2, 1)^T$. Thus the functions $e^{-100t}\boldsymbol{x}_1$ and $e^{-t/10}\boldsymbol{x}_2$ are both exact solutions to the given differential equation. To find the solution with the given initial value, we note that $\boldsymbol{y}_0 = \alpha_1\boldsymbol{x}_1 + \alpha_2\boldsymbol{x}_2$ where $\alpha_1 = 1$, $\alpha_2 = 2$, so

$$\boldsymbol{y}(t) = \alpha_1 e^{-100t}\boldsymbol{x}_1 + \alpha_2 e^{-t/10}\boldsymbol{x}_2 = \begin{pmatrix} e^{-100t} + 2e^{-t/10} \\ -e^{-100t} + e^{-t/10} \end{pmatrix}. \tag{5.20}$$

The exact solution on the interval $[0, 3]$ is plotted in Figure 5.6. Note that the solution is very smooth and slowly varying, after an initial transient period, related to the large negative eigenvalue -100.

In the first five plots in Figure 5.7 we show the approximate solution to this system obtained by using Euler's method with step size $h = 1/10$, $1/20$, $1/40$, $1/80$, and $1/160$. Notice the wild scale in the first three plots. Although both components of the exact solution are bounded between 0 and 3, the approximate solution is on the order of 10^{28} when $h = 1/10$ and on the order of 10^{36} when $h = 1/20$. For $h = 1/80$ the solution appears qualitatively correct except for a short time near 0, and for $h = 1/160$ the numerical solution is visually indistinguishable from the exact solution.

Since our computation shows that we can integrate this system accurately with Euler's method if we take $h = 1/160$, and since the solution is very smooth for $t \ge 1/2$, a natural idea is to use a step size of $1/160$ to solve on $[0, 1/2]$, and then to increase the step size to,

FIGURE 5.6. The exact solution of the moderately stiff 2×2 linear system (5.19). The solid line shows the first component, the dashed line the second.

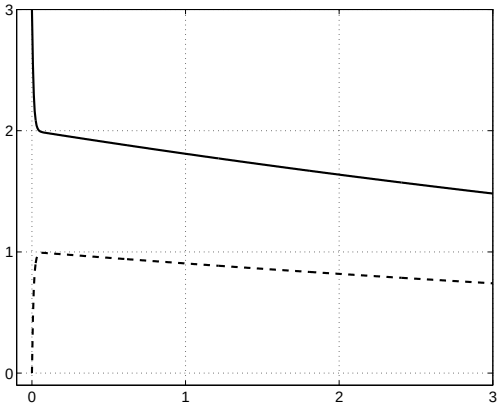
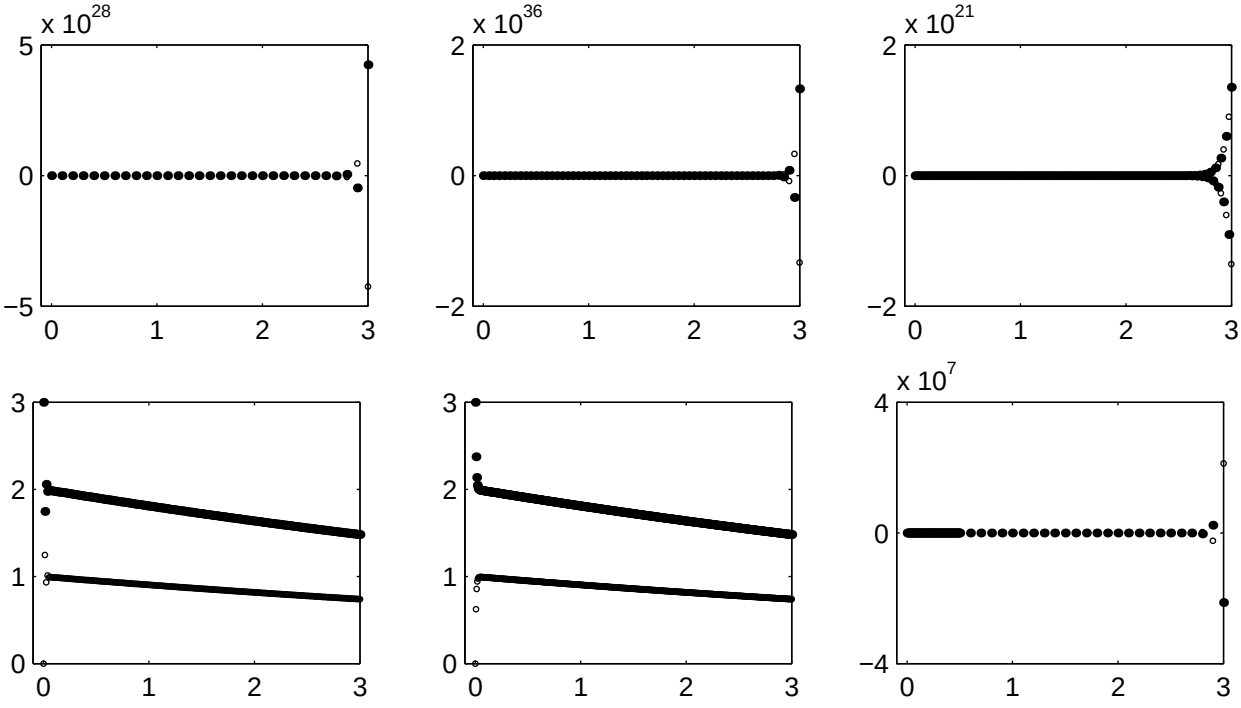


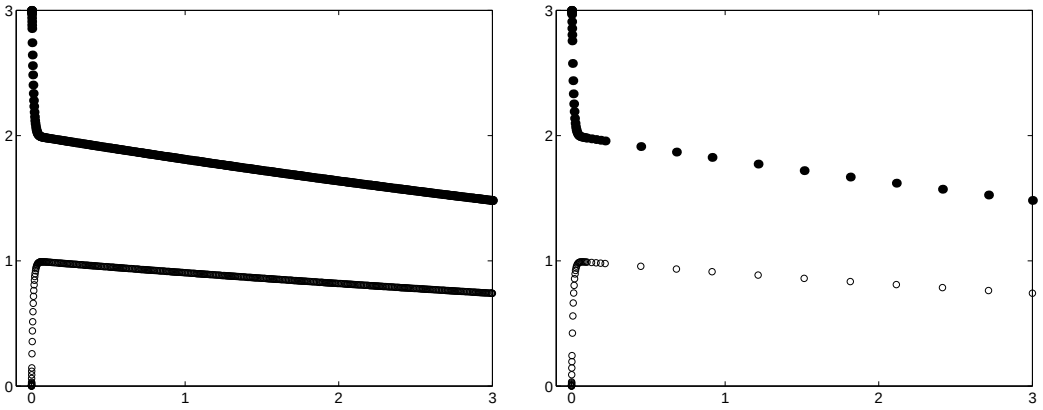
FIGURE 5.7. Euler's method applied to a stiff 2×2 linear system. step sizes are $h = 1/10$, $1/20$, $1/40$, $1/80$, and $1/160$. The final plot uses 80 steps of size $1/160$ followed by steps of size $1/10$.



say, 1/10. The result is shown in the final plot Figure 5.7. We have abject failure: again the solution is off by orders of magnitude.

As a final computational example, we compute the solution using two adaptive single step solvers from Matlab, ode45, which uses an embedded Runge–Kutta pair, and ode15s, which is designed for stiff problems. The results, pictured in Figure 5.8, show clearly that in order to obtain an accurate solution ode45 requires small step size all through the domain,

FIGURE 5.8. The solution using Matlab's ode45 and ode15s.



even where the solution is smooth, while ode15s is able to use large step sizes after the initial transient.

What we are seeing here is the problem of *stiffness*. A stiff problem is characterized by rapidly varying transients which decay quickly, but for some reason require us to take small step sizes even after they have disappeared from the solution. As we see from the ode15s results, there are methods that are able to overcome the problem of stiffness. Stiff problems are important, because they arise (though in more complicated form than this simple example), in a number of applications including chemical reaction modeling, numerical solution of parabolic and hyperbolic PDEs, control theory, and electric circuit modeling. In the rest of this section, we seek to understand what is happening and how stiff problems can be efficiently dealt with.

First let us return to Euler's method with a constant step size. Because of the very simple nature of the ODE, we can give a closed form for the Euler solution. Euler's method for this ODE gives $\boldsymbol{y}_{n+1} = (I + hA)\boldsymbol{y}_n$. This gives $\boldsymbol{y}_n = (I + hA)^n \boldsymbol{y}_0$ for all n, howsoever the initial value $\boldsymbol{y}_0$ is chosen. In particular, if we take the initial value equal to the eigenvector $\boldsymbol{x}_1$, then at the nth step we have $\boldsymbol{y}_n = (1 - 100h)^n \boldsymbol{x}_1$ (since $\boldsymbol{x}_1$ is also an eigenvector of $I + hA$, with eigenvalue $1 - 100h$). Similarly if the initial data is $\boldsymbol{x}_2$, then $\boldsymbol{y}_n = (1 - h/10)^n \boldsymbol{x}_2$. Now any 2-vector can be written as linear combination of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. In particular, the given initial data $\boldsymbol{y}_0 = (3, 0)^T = \alpha_1 \boldsymbol{x}_1 + \alpha_2 \boldsymbol{x}_2$, with $\alpha_1 = 1$ and $\alpha_2 = 2$. Thus the Euler approximation to (5.19) is given by

$$\boldsymbol{y}_n = \alpha_1 (1 - 100h)^n \boldsymbol{x}_1 + \alpha_2 (1 - h/10)^n \boldsymbol{x}_2. \tag{5.21}$$

Comparing with (5.20), we see that e^{-100t} is being approximated by $(1 - 100h)^n$ when $t = hn$. Now if $100h << 1$ this is a reasonable approximation. Then $e^{-100h} \approx 1 - 100h$ and $e^{-100hn} \approx (1 - 100h)^n$. But if $100h$ is not so small, then $(1 - 100h)^n$ does not behave at all like e^{-100hn}. In fact if $100h > 2$, then $1 - 100h$ is a negative number of magnitude greater than one, and so $(1 - 100h)^n$ is exponentially growing, and alternating sign. In that case the first time on the right hand side on (5.21), instead of becoming negligeable as n increases, will dominate the computation. In fact, in order that this term decay as n increases, a necessary and sufficient condition is that $|1 - 100h| < 1$, or $-2 < -100h < 0$, or $h \in (0, 1/50)$, which matches will our numerical experience.

Notice, that we could have analyzed this situation by looking at the simpler problem $y' = -100y$, $y(0) = 1$. In general, an advantageous property for methods to be used with stiff problems is that when they are used to solve the very simple model problem

$$y' = \lambda y, \quad y(0) = 1, \tag{5.22}$$

where $\lambda < 0$, the numerical solution y_n should decay, not grow, as n increases. Generally, this will imply some restriction on the step size. The more severe that restriction, the less suitable the method is for stiff problems.

The model problem (5.22), while very simple, has relevance for a great many problems. Many ODEs can be approximated, at least locally by a linear ODE. Starting with any linear ODE $\boldsymbol{y}' = A\boldsymbol{y}$, where A is a matrix, we may diagonalize to obtain a set of equations of the form of (5.22) where the values of λ which arise are the eigenvalues of A. Of course this requires that A be diagonalizable, but since any matrix is arbitrarily close to a diagonalizable matrix, this is not a big restriction. However, a diagonalizable matrix may well have complex eigenvalues (even when the matrix is real), and thus to results relevant to a reasonably general situation we should consider (5.22) for $\lambda \in \mathbb{C}$. Since the solution $y = e^{\lambda t}$ decays with increasing t, if and only if $\Re\lambda < 0$, we would like that the numerical solution y_n decays with increasing n for $\Re\lambda < 0$ and $h > 0$.

For Euler's method, $y_n = (1 + \lambda h)^n$, which is decaying if and only $|1 + h\lambda| < 1$. Consider next the backward Euler method. Then $y_{n+1} = y_n + h\lambda y_{n+1}$, whence $y_n = [1/(1 - h\lambda)]^n$. For $\Re\lambda < 0$, this holds for all $h > 0$. Thus we would expect (correctly), that stiffness is not a problem for the backward Euler method.

Similarly, for any single step method and any $\lambda \in \mathbb{C}$, we may consider the set of values of h for which the numerical solution obtained when the method is applied to (5.22) is decaying as $n \to \infty$. (It doesn't matter what nonzero initial value is used, since the problem is linear.) Now for virtually any method (certainly any method we have studied, including all implicit and explicit Runge–Kutta methods), the values of y_n depend only on the product $h\lambda$ (check this!). Therefore we define the *region of absolute stability* of the single step method as

$$\mathcal{S} = \{\, h\lambda \in \mathbb{C} : \lim_{n\to\infty} y_n = 0, \text{ when } y_n \text{ is the numerical solution to (5.22) with step size } h \,\}$$

For example, consider the improved Euler method

$$y_{n+1} = y_n + h[f(t_n, y_n) + f\big(t_{n+1}, y_n + hf(t_n, y_n)\big)]/2.$$

Applied to the equation $y' = \lambda y$ this becomes

$$y_{n+1} = y_n + h[\lambda y_n + \lambda(y_n + h\lambda y_n)]/2 = y_n(1 + \bar{h} + \bar{h}^2/2),$$

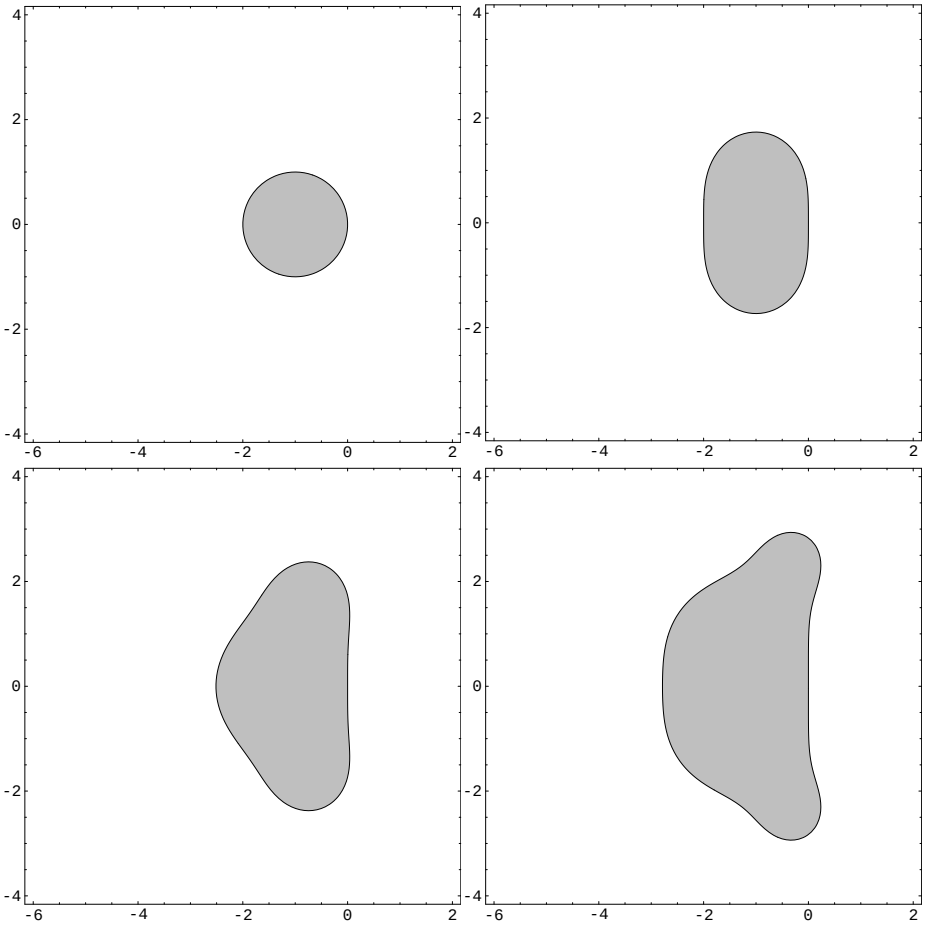
where $\bar{h} = h\lambda$. Thus the region for stability for the improved Euler method is

$$\mathcal{S} = \{\, \bar{h} \in \mathbb{C} \,|\, |1 + \bar{h} + \bar{h}^2/2| < 1 \,\}.$$

This region is plotted in Figure 5.9(b). In fact it is easy to check that the 3 term Taylor series method and any two-stage second order Runge–Kutta method have exactly the same absolute stability region.

The region of absolute stability is defined for linear multistep methods as well. It consists of the set of $\bar{h} = h\lambda \in \mathbb{C}$ such that the numerical solution decays for any choice of initial values. Equivalently, this is the set of $\bar{h}$ such that the *stability polynomial* $\rho(z) - \bar{h}\sigma(z)$ has all its roots in the open unit disk.

FIGURE 5.9. Regions of absolute stability of Runge–Kutta methods. a) Euler's method; b) any 2-stage 2nd order method; c) any 3-stage 3rd order method; d) any 4-stage 4th order method.



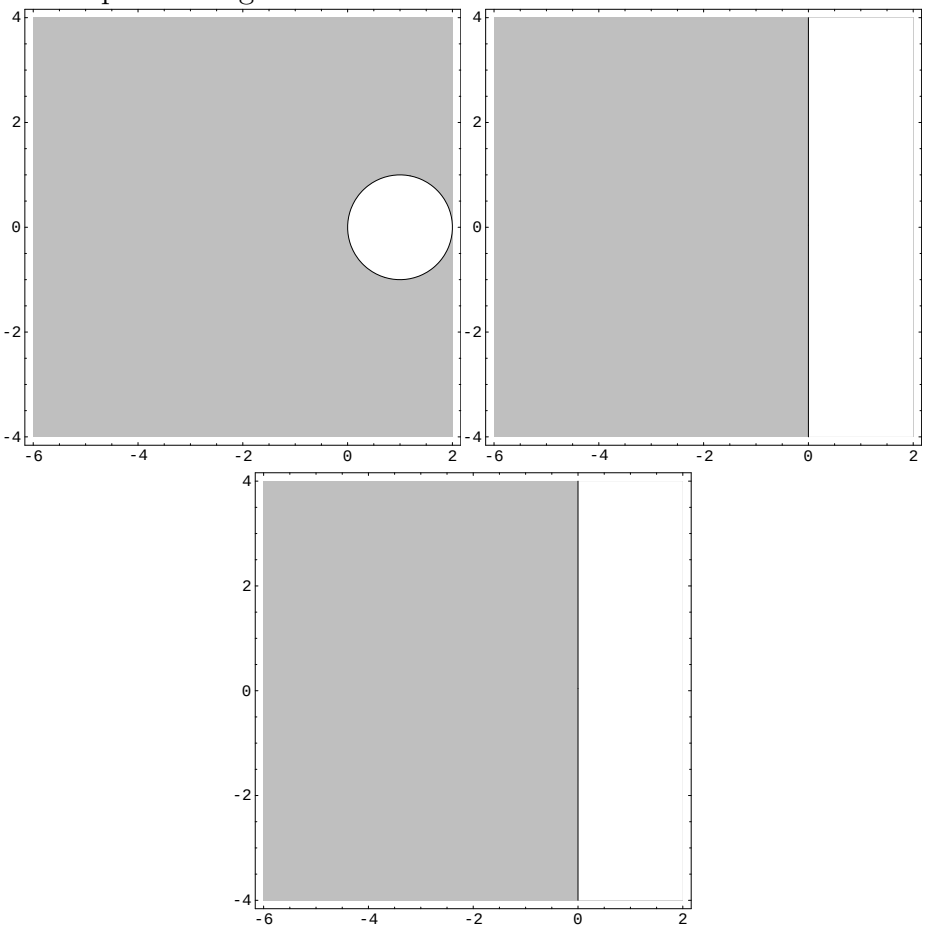
Figures 5.9, 5.10, and 5.11 show the regions of absolute stability for various explicit Runge–Kutta methods, some implicit single step methods, and some Adams methods, respectively

Looking at the Figures 5.9–5.11 we can make several observations.

(1) All the explicit methods shown have bounded regions of absolute stability. In fact, this can be proven to be the case for all explicit methods.
(2) A method is called *A-stable* if its region of absolute stability contains the entire left half plane (so whenever the exact solution decays, so does the numerical solution). Of the methods shown, only the backward Euler method, the trapezoidal method, the Gauss–Legendre 2-stage implicit Runge–Kutta method are A-stable.
(3) The Adams–Moulton methods (which are implicit) have bounded regions of absolute stability, but these are notably larger than for the Adams–Bashford methods.

As just mentioned only implicit methods can be A-stable. However, for linear multistep methods, there is another major limitation as well.

FIGURE 5.10. Regions of absolute stability of some implicit one-step methods. a) Backward Euler method; b) trapezoidal method; c) the 2-stage Gauss–Legendre implicit Runge–Kutta method.

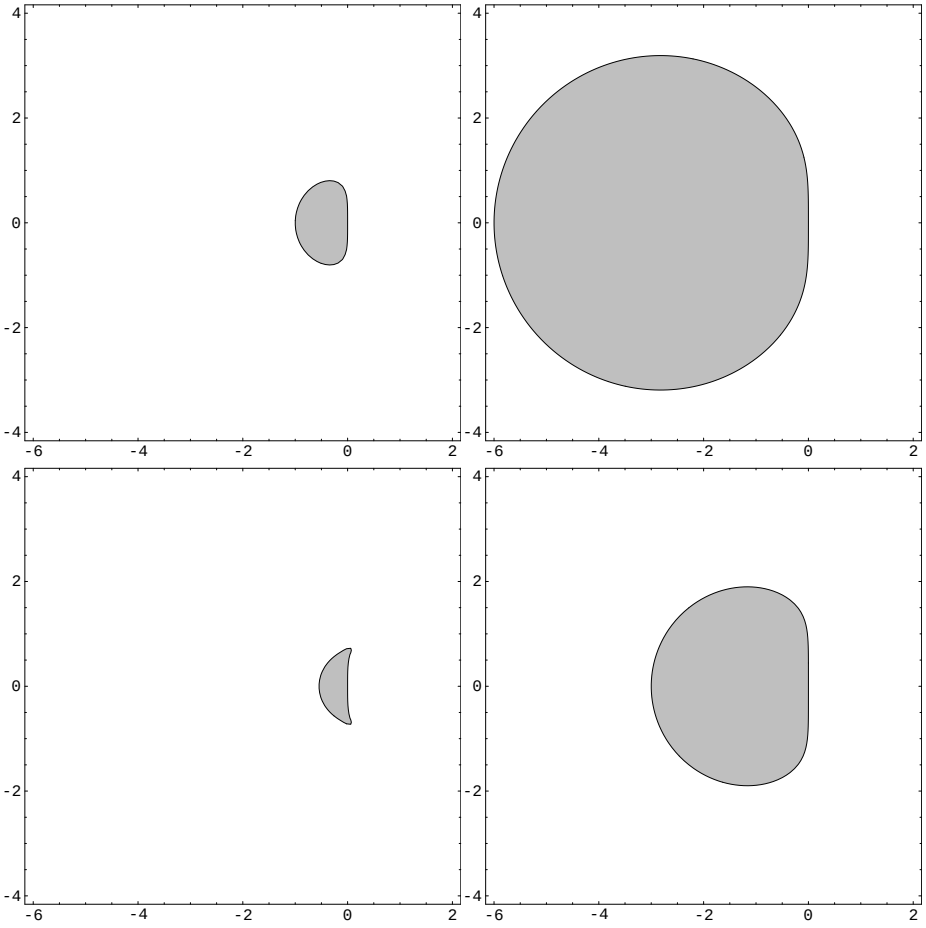


THEOREM 5.16 (Second Dahlquist Barrier). *An A-stable linear multistep method is implicit and of order at most 2.*

Because the Adams methods, and various other linear multistep methods, have rather small regions of absolute stability, other linear multistep methods are preferred for stiff problems. The most popular are the *backward differentiation formula methods* or BDF methods. These are methods of the form

$$\sum_{j=-1}^{k} a_j y_{n-j} = h f_{n+1}.$$

FIGURE 5.11. Regions of absolute stability of some Adams methods. a) Adams–Bashford 2-step; b) Adams–Moulton 2-step; a) Adams–Bashford 3-step; b) Adams–Moulton 3-step.

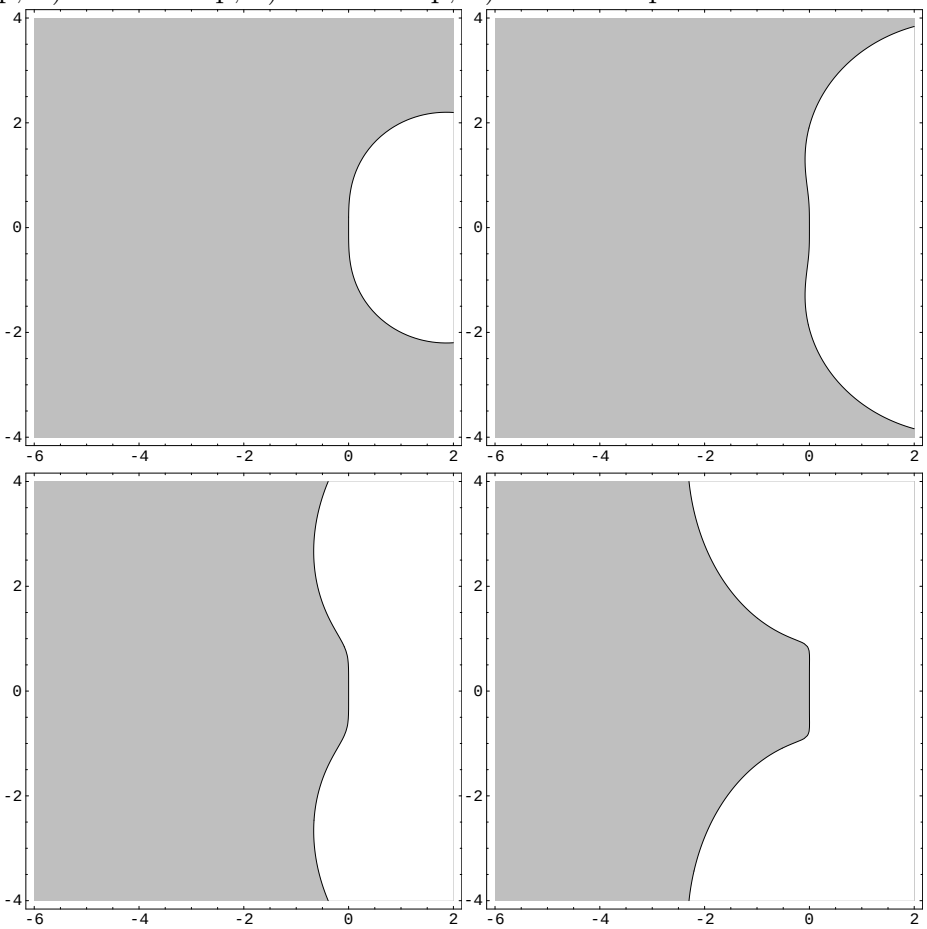


The coefficients are a_j are determined by interpolating y_{n-j} and t_{n-j} $j = -1, \ldots, k$ by a polynomial $p(t)$ of degree k and evaluating $p'(t_{n+1})$. The first few BDF methods are

$$\begin{aligned} y_{n+1} - y_n &= hf_{n+1} && \text{BDF1 (backward Euler)},\\ (3y_{n+1} - 4y_n + y_{n-1})/2 &= hf_{n+1} && \text{BDF2},\\ (11y_{n+1} - 18y_n + 9y_{n-1} - 2y_{n-2})/6 &= hf_{n+1} && \text{BDF3}. \end{aligned}$$

The k-step BDF method is of order k. For $k = 1$ and 2, the method is A-stable. The second Dahlquist barrier implies that this is not true for $k > 2$, but for $k = 3$ the region of absolute stability just misses a tiny piece of the left half plane, so the method is good for most stiff problems. See Figure 5.12. As k increases the region of absolute stability decreases, but it still contains a great deal of the left half plane for $k = 4$ and $k = 5$. For $k > 6$, the BDF formulas are not stable, in the usual sense, and so should not be used..

FIGURE 5.12. Regions of absolute stability of some BDF methods. a) BDF 2-step; b) BDF 3-step; a) BDF 4-step; b) BDF 5-step.



EXERCISES

(1) Prove that Euler's method is stable with respect to perturbations in the initial data y_0 and the function f. That is, prove that if y_n is defined by Euler's method and $\bar{y}_n$ is defined by the perturbed equations:

$$\bar{y}_{n+1} = \bar{y}_n + h\bar{f}(t_n, \bar{y}_n), \quad \bar{y}_0 \text{ given},$$

then $|y_n - \bar{y}_n| \leq C_1|y_0 - \bar{y}_0| + C_2\|f - \bar{f}\|_{L^\infty(I\times\mathbb{R})}$. (State precisely the hypotheses needed and give explicit formulas for C_1 and C_2.)

(2) State precisely and prove an asymptotic error estimate for the trapezoidal method.

(3) Find the most general two stage Runge–Kutta method of order 2.

(4) For solving the equation $y' = f(t, y)$, consider the scheme

$$y_{n+1} = y_n + \frac{h}{2}(y'_n + y'_{n+1}) + \frac{h^2}{12}(y''_n - y''_{n+1}),$$

where $y'_n = f(t_n, y_n)$, and $y''_n = (\partial f/\partial t + f\partial f/\partial y)(t_n, y_n)$. Determine the order of this method. Show that its region of absolute stability contains the entire negative real axis.

(5) Consider a consistent linear multistep methods $y_{n+1} + \sum_{j=0}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f_{n-j}$ for which the coefficients $a_j \leq 0$, $j = 0, \ldots, k$ (there are many such methods). a) Prove that all such consistent methods satisfy the root condition, and so are stable. b) Give an elementary proof that all such methods are convergent (without invoking the Dahlquist theory).

CHAPTER 6

Numerical Solution of Partial Differential Equations

1. BVPs for 2nd order elliptic PDEs

We start with a typical physical application of partial differential equations, the modeling of heat flow. Suppose we have a solid body occupying a region $\Omega \subset \mathbb{R}^3$. The temperature distribution in the body can be given by a function $u : \Omega \times J \to \mathbb{R}$ where J is an interval of time we are interested in and $u(x,t)$ is the temperature at a point $x \in \Omega$ at time $t \in J$. The heat content (the amount of thermal energy) in a subbody $D \subset \Omega$ is given by

$$\text{heat content of } D = \int_D cu\,dx$$

where c is the product of the specific heat of the material and the density of the material. Since the temperature may vary with time, so can the heat content of D. The rate of change of heat in D is given by

$$\text{rate of change of heat in } D = \frac{\partial}{\partial t}\int_D cu\,dx = \int_D \frac{\partial(cu)}{\partial t}(x,t)\,dx.$$

Now any change of heat in D must be accounted for by heat flowing in or out of D through its boundary or by heat entering from external sources (e.g., if the body were in a microwave oven). Fourier's law of heat conduction says that heat flows in the direction opposite the temperature gradient with a rate proportional to the magnitude of the gradient. That is, the heat flow, at any point and any time, is given by

$$\text{heat flow} = -\lambda \operatorname{grad} u,$$

where the positive quantity λ is called the conductivity of the material. (Usually λ is just a scalar, but if the material is thermally anisotropic, i.e., it has preferred directions of heat flow, as might be a fibrous or laminated material, λ can be a 3×3 positive-definite matrix.) Therefore the heat that flows out of D is given by

$$\text{rate of heat flow out of } D = -\int_{\partial D} (\lambda \operatorname{grad} u) \cdot \boldsymbol{n}\,ds.$$

Now the divergence theorem says that for any vectorfield $\boldsymbol{v}$, $\int_{\partial D} \boldsymbol{v} \cdot \boldsymbol{n}\,ds = \int_D \operatorname{div} \boldsymbol{v}\,dx$. Thus

$$\text{rate of heat flow out of } D = -\int_D \operatorname{div}(\lambda \operatorname{grad} u)\,dx.$$

Conservation of energy then gives us

$$\int_D \frac{\partial(cu)}{\partial t}\,dx - \int_D \operatorname{div}(\lambda \operatorname{grad} u)\,dx = \int_D f\,dx,$$

where f is the rate at which heat per unit volume is being added from external sources (if heat is being removed, f is negative). Thus the quantity

$$\frac{\partial(cu)}{\partial t} - \operatorname{div}(\lambda \operatorname{grad} u) - f$$

has vanishing integral on any smoothly bounded subregion D. This happens if and only if this quantity vanishes. Thus we have derived the equation

$$\frac{\partial(cu)}{\partial t} = \operatorname{div}(\lambda \operatorname{grad} u) + f \text{ in } \Omega \times J.$$

The source function f, the material coefficients c and λ and the solution u can all be functions of x and t. If the material is homogeneous (the same everywhere) and not changing with time, then c and λ are constants and the equation simplifies to the *heat equation*,

$$\mu \frac{\partial u}{\partial t} = \Delta u + \tilde{f},$$

where $\mu = c/\lambda$ and we have $\tilde{f} = f/\lambda$. If the material coefficients depend on the temperature u, as may well happen, we get a nonlinear PDE generalizing the heat equation.

The heat equation not only governs heat flow, but all sorts of diffusion processes where some quantity flows from regions of higher to lower concentration. The heat equation is the prototypical *parabolic* differential equation.

Now suppose our body reaches a steady state: the temperature is unchanging. Then the time derivative term drops and we get

$$-\operatorname{div}(\lambda \operatorname{grad} u) = f \text{ in } \Omega, \tag{6.1}$$

where now u and f are functions of f alone. For a homogeneous material, this becomes the Poisson equation

$$-\Delta u = \tilde{f},$$

the prototypical *elliptic* differential equation. For an inhomogeneous material we can leave the steady state heat equation in *divergence form* as in (6.1), or differentiate out to obtain

$$-\lambda \Delta u + \operatorname{grad} \lambda \cdot \operatorname{grad} u = f.$$

To determine the steady state temperature distribution in a body we need to know not only the sources and sinks within the body (given by f), but also what is happening at the boundary $\Gamma := \partial\Omega$. For example a common situation is that the boundary is held at a given temperature

$$u = g \text{ on } \Gamma. \tag{6.2}$$

The PDE (6.1) together with the *Dirichlet boundary condition* (6.2) form an elliptic boundary value problem. Under a wide variety of circumstances this problem can be shown to have a unique solution. The following theorem is one example (although the smoothness requirements can be greatly relaxed).

THEOREM 6.1. *Let Ω be a smoothly bounded domain in $\mathbb{R}^n$, and let $\lambda : \bar{\Omega} \to \mathbb{R}_+$, $f : \bar{\Omega} \to \mathbb{R}$, $g : \Gamma \to \mathbb{R}$ be C^∞ functions. Then there exists a unique function $u \in C^2(\bar{\Omega})$ satisfying the differential equation* (6.1) *and the boundary condition* (6.2)*. Moreover u is C^∞.*

Instead of the Dirichlet boundary condition of imposed temperature, we often see the Neumann boundary condition of imposed heat flux (flow across the boundary):

$$\frac{\partial u}{\partial n} = g \text{ on } \Gamma.$$

For example if $g = 0$, this says that the boundary is insulated. We may also have a Dirichlet condition on part of the boundary and a Neumann condition on another.

2. The five-point discretization of the Laplacian

With the motivation of the previous section, let us consider the numerical solution of the elliptic boundary value problem

$$\Delta u = f \text{ in } \Omega, \quad u = g \text{ on } \Gamma. \tag{6.3}$$

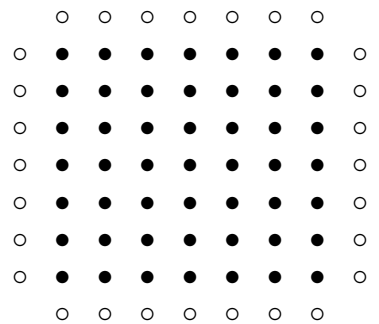
For simplicity we will consider first a very simple domain $\Omega = (0,1) \times (0,1)$, the unit square in $\mathbb{R}^2$. Now this problem is so simplified that we can attack it analytically, e.g., by separation of variables, but it is very useful to use as a *model problem* for studying numerical methods.

Let N be a positive integer and set $h = 1/N$. Consider the *mesh* in $\mathbb{R}^2$

$$\mathbb{R}^2_h := \{\, (mh, nh) \,:\, m, n \in \mathbb{Z} \,\}.$$

Note that each mesh point $x \in \mathbb{R}^2_h$ has four *nearest neighbors* in $\mathbb{R}^2_h$, one each to the left, right, above, and below. We let $\Omega_h = \Omega \cap \mathbb{R}^2_h$, the set of interior mesh points, and we regard this a discretization of the domain Ω. We also define Γ_h as the set of mesh points in $\mathbb{R}^2_h$ which don't belong to Ω_h, but which have a nearest neighbor in Ω_h. We regard Γ_h as a discretization of Γ. We also let $\bar{\Omega}_h := \Omega_h \cup \Gamma_h$

FIGURE 6.1. $\bar{\Omega}_h$ for $h = 1/8$: $\bullet$ – points in Ω_h, $\circ$ – points in Γ_h.



To discretize (6.3) we shall seek a function $u_h : \bar{\Omega}_h \to \mathbb{R}$ satisfying

$$\Delta_h u_h = f \text{ on } \Omega_h, \quad u_h = g \text{ on } \Gamma_h. \tag{6.4}$$

Here Δ_h is an operator, to be defined, which takes functions on $\bar{\Omega}_h$ (*mesh functions*) to functions on Ω_h. It should approximate the true Laplacian in the sense that if v is a smooth function on $\bar{\Omega}$ and $v_h = v|_{\bar{\Omega}_h}$ is the associated mesh function, then we want

$$\Delta_h v_h \approx \Delta v|_{\Omega_h}$$

for h small.

Before defining Δ_h, let us turn to the one-dimensional case. That is, given a function v_h defined at the mesh points nh, $n \in \mathbb{Z}$, we want to define a function $D_h^2 v_h$ on the mesh points, so that $D_h^2 v_h \approx v''|_{\mathbb{Z}h}$ if $v_h = v|_{\mathbb{Z}h}$. One natural procedure is to construct the quadratic polynomial p interpolating v_h at three consecutive mesh points $(n-1)h$, nh, $(n+1)h$, and let $D_h^2 v_h(nh)$ be the constant value of p''. This gives the formula

$$D_h^2 v_h(nh) = 2v_h[(n-1)h, nh, (n+1)h] = \frac{v_h\big((n+1)h\big) - 2v_h(nh) + v_h\big((n-1)h\big)}{h^2}.$$

D_h^2 is known as the 3-point difference approximation to d^2/dx^2. We know that if v is C^2 in a neighborhood of nh, then $\lim_{h\to 0} v[x-h, x, x+h] = v''(x)/2$. In fact, it is easy to show by Taylor expansion (do it!), that

$$D_h^2 v(x) = v''(x) + \frac{h^2}{12} v^{(4)}(\xi), \text{ for some } \xi \in \big(x-h, x+h\big),$$

as long as v is C^4 near x. Thus D_h^2 is a second order approximation to d^2/dx^2.

Remark. Alternatively, we could use the Peano kernel theorem to analyze the error $D_h^2 v(0) - v''(0)$, say when $h = 1$, and then use scaling to get the result for arbitrary h. We leave this as an exercise for the reader.

Now returning to the definition of the $\Delta_h \approx \Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$, we simply use the 3-point approximation to $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$. Writing $v_{m,n}$ for $v(mh, nh)$ we then have

$$\begin{aligned}\Delta_h\, v(mh, nh) &= \frac{v_{m+1,n} - 2v_{m,n} + v_{m-1,n}}{h^2} + \frac{v_{m,n+1} - 2v_{m,n} + v_{m,n-1}}{h^2} \\ &= \frac{v_{m+1,n} + v_{m-1,n} + v_{m,n+1} + v_{m,n-1} - 4v_{m,n}}{h^2}.\end{aligned}$$

From the error estimate in the one-dimensional case we easily get that for $v \in C^4(\bar{\Omega})$,

$$\Delta_h\, v(mh, nh) - \Delta\, v(mh, nh) = \frac{h^2}{12}\left[\frac{\partial^4 v}{\partial x^4}(\xi, nh) + \frac{\partial^4 v}{\partial y^4}(mh, \eta)\right],$$

for some ξ, η. Thus:

Theorem 6.2. *If $v \in C^2(\bar{\Omega})$, then*

$$\lim_{h\to 0} \|\Delta_h\, v - \Delta\, v\|_{L^\infty(\Omega_h)} = 0.$$

If $v \in C^4(\bar{\Omega})$, then

$$\|\Delta_h\, v - \Delta\, v\|_{L^\infty(\Omega_h)} \le \frac{h^2}{6} M_4,$$

where $M_4 = \max(\|\partial^4 v/\partial x^4\|_{L^\infty(\bar{\Omega})}, \|\partial^4 v/\partial y^4\|_{L^\infty(\bar{\Omega})})$.

The discrete PDE $\Delta_h\, u_h = f$ on Ω_h is a system of $(N-1)^2$ linear equations in the unknown values of u_h at the mesh points. Since the values of u_h are given on the boundary mesh points, we may regard (6.4) as a system of $(N-1)^2$ linear equations in $(N-1)^2$

unknowns. For example, in the case $N = 4$ the system is

$$\begin{pmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{pmatrix} = \begin{pmatrix} h^2 f_{1,1} - u_{1,0} - u_{0,1} \\ h^2 f_{2,1} - u_{2,0} \\ h^2 f_{3,1} - u_{3,0} - u_{4,1} \\ h^2 f_{1,2} - u_{0,2} \\ h^2 f_{2,2} \\ h^2 f_{3,2} - u_{4,2} \\ h^2 f_{1,3} - u_{0,3} - u_{1,4} \\ h^2 f_{2,3} - u_{2,4} \\ h^2 f_{3,3} - u_{4,3} - u_{3,4} \end{pmatrix}$$

The matrix may be rewritten as

$$\begin{pmatrix} A & I & O \\ I & A & I \\ O & I & A \end{pmatrix}$$

where I is the 3×3 identity matrix, O is the 3×3 zero matrix, and

$$A = \begin{pmatrix} -4 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 4 \end{pmatrix}.$$

For general N the matrix can be partitioned into $(N-1) \times (N-1)$ blocks, each in $\mathbb{R}^{(N-1)\times(N-1)}$:

$$\begin{pmatrix} A & I & O & \cdots & O & O \\ I & A & I & \cdots & O & O \\ O & I & A & \cdots & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & O & \cdots & I & A \end{pmatrix},$$

where I and O are the identity and zero matrix in $\mathbb{R}^{(N-1)\times(N-1)}$, respectively, and $A \in \mathbb{R}^{(N-1)\times(N-1)}$ is the tridiagonal matrix with -4 on the diagonal and 1 above and below the diagonal. This assumes the unknowns are ordered

$$u_{1,1}, u_{2,1}, \ldots, u_{N-1,1}, u_{1,2}, \ldots, u_{N-1,N-1},$$

and the equations are ordered similarly.

Notice that the matrix has many special properties:

- it is sparse with at most 5 elements per row nonzero
- it is block tridiagonal, with tridiagonal and diagonal blocks
- it is symmetric
- it is diagonally dominant
- its diagonal elements are negative, all others nonnegative
- it is negative definite

2.1. Analysis via a maximum principle. We will now prove that the problem (6.4) has a unique solution and prove an error estimate. The key will be a discrete maximum principle.

THEOREM 6.3 (Discrete Maximum Principle). *Let v be a function on $\bar{\Omega}_h$ satisfying*

$$\Delta_h u \geq 0 \text{ on } \Omega_h.$$

Then $\max_{\Omega_h} v \leq \max_{\Gamma_h} v$. *Equality holds if and only if v is constant.*

PROOF. Suppose $\max_{\Omega_h} v \geq \max_{\Gamma_h} v$. Take $x_0 \in \Omega_h$ where the maximum is achieved. Let x_1, x_2, x_3, and x_4 be the nearest neighbors. Then

$$4v(x_0) = \sum_{i=1}^{4} v(x_i) - h^2 \Delta_h v(x_0) \leq \sum_{i=1}^{4} v(x_i) \leq 4v(x_0),$$

since $v(x_i) \leq v(x_0)$. Thus equality holds throughout and v achieves its maximum at all the nearest neighbors of x_0 as well. Applying the same argument to the neighbors in the interior, and then to their neighbors, etc., we conclude that v is constant. □

REMARKS. 1. The analogous discrete minimum principle, obtained by reversing the inequalities and replacing max by min, holds. 2. This is a discrete analogue of the maximum principle for the Laplace operator.

THEOREM 6.4. *There is a unique solution to the discrete boundary value problem* (6.4).

PROOF. Since we are dealing with a square linear system, it suffices to show nonsingularity, i.e., that if $\Delta_h u_h = 0$ on Ω_h and $u_h = 0$ on Γ_h, then $u_h \equiv 0$. Using the discrete maximum and the discrete minimum principles, we see that in this case u_h is everywhere 0. □

The next result is a statement of maximum norm stability.

THEOREM 6.5. *The solution u_h to* (6.4) *satisfies*

$$\|u_h\|_{L^\infty(\bar{\Omega}_h)} \leq \frac{1}{8}\|f\|_{L^\infty(\Omega_h)} + \|g\|_{L^\infty(\Gamma_h)}. \tag{6.5}$$

This is a stability result in the sense that it states that the mapping $(f, g) \mapsto u_h$ is bounded uniformly with respect to h.

PROOF. We introduce the comparison function $\phi = [(x - 1/2)^2 + (y - 1/2)^2]/4$, which satisfies $\Delta_h \phi = 1$ on Ω_h, and $0 \leq \phi \leq 1/8$ on $\bar{\Omega}_h$. Set $M = \|f\|_{L^\infty(\Omega_h)}$. Then

$$\Delta_h(u_h + M\phi) = \Delta_h u_h + M \geq 0,$$

so

$$\max_{\Omega_h} u_h \leq \max_{\Omega_h}(u_h + M\phi) \leq \max_{\Gamma_h}(u_h + M\phi) \leq \max_{\Gamma_h} g + \frac{1}{8}M.$$

Thus u_h is bounded above by the right-hand side of (6.5). A similar argument applies to $-u_h$ giving the theorem. □

By applying the stability result to the error $u - u_h$ we can bound the error in terms of the *consistency error* $\Delta_h u - \Delta u$.

THEOREM 6.6. *Let u be the solution of the Dirichlet problem* (6.1) *and u_h the solution of the discrete problem* (6.4). *Then*

$$\|u - u_h\|_{L^\infty(\bar{\Omega}_h)} \leq \frac{1}{8}\|\Delta u - \Delta_h u\|_{L^\infty(\bar{\Omega}_h)}.$$

PROOF. Since $\Delta_h u_h = f = \Delta u$ on Ω_h, $\Delta_h(u - u_h) = \Delta_h u - \Delta u$. Also, $u - u_h = 0$ on Γ_h. Apply Theorem 6.5 (with u_h replaced by $u - u_h$), we obtain the theorem. □

Combining with Theorem 6.2, we obtain error estimates.

COROLLARY 6.7. *If $u \in C^2(\bar{\Omega})$, then*

$$\lim_{h\to 0} \|u - u_h\|_{L^\infty(\bar{\Omega}_h)} = 0.$$

If $u \in C^4(\bar{\Omega})$, then

$$\|u - u_h\|_{L^\infty(\bar{\Omega}_h)} \le \frac{h^2}{48} M_4,$$

where $M_4 = \max(\|\partial^4 u/\partial x^4\|_{L^\infty(\bar{\Omega})}, \|\partial^4 u/\partial y^4\|_{L^\infty(\bar{\Omega})})$.

REMARK. The quantity $\|\Delta u - \Delta_h u\|$ is the *consistency error* of the discretization, and the statement that $\lim_{h\to 0}\|\Delta u - \Delta_h u\| = 0$ means that the discretization is *consistent*. An estimate of the form $\|v\| \le C_h \|f\|$ whenever $\Delta_h v = f$ on Ω_h and $v = 0$ on Γ_h, is a *stability estimate*, and if it holds with C_h independent of h, we say the discretization is *stable*. The preceding proof shows that

$$\text{consistency} + \text{stability} \implies \text{convergence}.$$

(Of course all three concepts are defined with respect to specific norms.)

2.2. Fourier analysis. Define $L(\Omega_h)$ to be the set of functions $\Omega_h \to \mathbb{R}$, which is isomorphic to $\mathbb{R}^M$, $M = (N-1)^2$. Sometimes we think of these as functions on $\bar{\Omega}_h$ extended by zero to Γ_h. The discrete Laplacian then defines an isomorphism of $L(\Omega_h)$ onto itself. The stability result from the previous section says simply that $\|\Delta_h^{-1}\| \le 1/8$ where the operator norm is with respect to the L^∞ norm on $L(\Omega_h)$. In this section we use Fourier analysis to establish a similar stability result for a discrete analogue of the L^2 norm.

First consider the one-dimensional case. With $h = 1/N$ let $I_h = \{h, 2h, \ldots, (N-1)h\}$, and let $L(I_h)$ be the space of functions on I_h, which is an $N-1$ dimensional vectorspace. On $L(I_h)$ we define the inner product

$$\langle u, v\rangle_h = h \sum_{k=1}^{N-1} u(kh)v(kh),$$

with the corresponding norm $\|v\|_h$.

The space $L(I_h)$ is a discrete analogue of $L^2(I)$ where I is the unit interval. On this latter space the functions $\sin \pi m x$, $m = 1, 2, \ldots$, form an orthogonal basis consisting of eigenfunctions of the operator d^2/dx^2. The corresponding eigenvalues are $\pi^2, 4\pi^2, 9\pi^2, \ldots$. We now establish the discrete analogue of this result.

Define $\phi_m \in L(I_h)$ by $\phi_m(x) = \sin \pi m x$, $x \in I_h$. It turns out that these mesh functions are precisely the eigenvectors of the operator D_h^2. Indeed

$$D_h^2 \phi_m(x) = \frac{\sin \pi m(x+h) - 2\sin \pi m x + \sin \pi m(x-h)}{h^2} = \frac{2}{h^2}(\cos \pi m h - 1)\sin \pi m x.$$

Thus

$$D_h^2 \phi_m = -\lambda_m \phi_m, \quad \lambda_m = \frac{2}{h^2}(1 - \cos \pi m h) = \frac{4}{h^2}\sin^2 \frac{\pi m h}{2}.$$

Note that

$$0 < \lambda_1 < \lambda_2 < \cdots < \lambda_{N-1} < \frac{4}{h^2}.$$

Note also that for small $m << N$, $\lambda_m \approx \pi^2 m^2$. In particular $\lambda_1 \approx \pi^2$. To get a strict lower bound we note that $\lambda_1 = 8$ for $N = 2$ and λ_1 increases with N.

Since the operator D_h^2 is symmetric with respect to the inner product on $L(I_h)$, and the eigenvalues λ_m are distinct, it follows that the eigenvectors ϕ_m are mutually orthogonal. (This can also be obtained using trigonometric identities, or by expressing the sin functions in terms of complex exponentials and using the discrete Fourier transform studied in Chapter 1.7.) Since there are $N-1$ of them, they form a basis of $L(I_h)$.

THEOREM 6.8. *The functions ϕ_m, $m = 1, 2, \ldots, N-1$ form an orthogonal basis of $L(I_h)$. Consequently, any function $v \in L(I_h)$ can be expanded as $v = \sum_{m=1}^{N-1} a_m \phi_m$ with $a_m = \langle v, \phi_m \rangle_h / \|\phi_m\|_h^2$, and $\|v\|_h^2 = \sum_{m=1}^{N-1} a_m^2 \|\phi_m\|_h^2$.*

From this we obtain immediately a stability result for the one-dimensional Laplacian. If $v \in L(I_h)$ and $D_h^2 v = f$, we expand v in terms of the ϕ_m:

$$v = \sum_{m=1}^{N-1} a_m \phi_m, \quad \|v\|_h^2 = \sum_{m=1}^{N-1} a_m^2 \|\phi_m\|_h^2.$$

Then

$$f = -\sum_{m=1}^{N-1} \lambda_m a_m \phi_m, \quad \|f\|_h^2 = \sum_{m=1}^{N-1} \lambda_m^2 a_m^2 \|\phi_m\|_h^2 \geq 8^2 \|v\|_h^2.$$

Thus $\|v\|_h \leq \|f\|_h / 8$.

The extension to the two-dimensional case is straightforward. We use the basis $\phi_{mn} = \phi_m \otimes \phi_n$, i.e.,

$$\phi_{mn}(x, y) := \phi_m(x)\phi_n(y), \quad m, n = 1, \ldots, N-1,$$

for $L(\Omega_h)$. It is easy to see that these $(N-1)^2$ functions form an orthogonal basis for $L(\Omega_h)$ equipped with the inner product

$$\langle u, v \rangle_h = h^2 \sum_{m=1}^{N-1} \sum_{n=1}^{N-1} u(mh, nh) v(mh, nh)$$

and corresponding norm $\|\cdot\|_h$. Moreover ϕ_{mn} is an eigenvector of Δ_h with eigenvalue $\lambda_{mn} = \lambda_m + \lambda_n \geq 16$. The next theorem follows immediately.

THEOREM 6.9. *The operator Δ_h defines an isomorphism from $L(\Omega_h)$ to itself. Moreover $\|\Delta_h^{-1}\| \leq 1/16$ where the operator norm is with respect to the norm $\|\cdot\|_h$ on $L(\Omega_h)$.*

Since the $\|v\|_h \leq \|v\|_{L^\infty(\Omega_h)}$ we also have consistency with respect to the discrete 2-norm. We leave it to the reader to complete the analysis with a convergence result.

2.3. Analysis via an energy estimate. Let v be a mesh function. Define the backward difference operator

$$\partial_x v(mh, nh) = \frac{v(mh, nh) - v((m-1)h, nh)}{h}, \quad 1 \le m \le N, \quad 0 \le n \le N.$$

In this section we denote

$$\langle v, w \rangle_h = h^2 \sum_{m=1}^{N} \sum_{n=1}^{N} v(mh, nh) w(mh, nh),$$

with the corresponding norm $\| \cdot \|_h$ (this agrees with the notation in the last section for mesh functions which vanish on Γ_h).

LEMMA 6.10. *If $v \in L(\Omega_h)$ (the set of mesh functions vanishing on Γ_h), then*

$$\|v\|_h \le \|\partial_x v\|_h.$$

PROOF. For $1 \le m \le N$, $0 \le n \le N$,

$$\begin{aligned} |v(mh, nh)|^2 &\le \left(\sum_{i=1}^{N} |v(ih, nh) - v((i-1)h, nh)| \right)^2 \\ &= \left(h \sum_{i=1}^{N} |\partial_x v(ih, nh)| \right)^2 \\ &\le \left(h \sum_{i=1}^{N} |\partial_x v(ih, nh)|^2 \right) \left(h \sum_{i=1}^{N} 1^2 \right) \\ &= h \sum_{i=1}^{N} |\partial_x v(ih, nh)|^2. \end{aligned}$$

Therefore

$$h \sum_{m=1}^{N} |v(mh, nh)|^2 \le h \sum_{i=1}^{N} |\partial_x v(ih, nh)|^2$$

and

$$h^2 \sum_{m=1}^{N} \sum_{n=1}^{N} |v(mh, nh)|^2 \le h^2 \sum_{i=1}^{N} \sum_{n=1}^{N} |\partial_x v(ih, nh)|^2.$$

□

This result is a discrete analogue of Poincaré's inequality, which bounds a function in terms of its gradient as long as the function vanishes on a portion of the boundary. The implied constant of 1 in the bound can be improved. The next result is a discrete analogue of Green's Theorem (essentially, integration by parts).

LEMMA 6.11. *If $v, w \in L(\Omega_h)$, then*

$$-\langle \Delta_h v, w \rangle_h = \langle \partial_x v, \partial_x w \rangle_h + \langle \partial_y v, \partial_y w \rangle_h.$$

PROOF. Let $v_0, v_1, \dots, v_N, w_0, w_1, \dots, w_N \in \mathbb{R}$ with $w_0 = w_N = 0$. Then

$$\begin{aligned}\sum_{i=1}^{N}(v_i - v_{i-1})(w_i - w_{i-1}) &= \sum_{i=1}^{N} v_i w_i + \sum_{i=1}^{N} v_{i-1}w_{i-1} - \sum_{i=1}^{N} v_{i-1}w_i - \sum_{i=1}^{N} v_i w_{i-1} \\ &= 2\sum_{i=1}^{N-1} v_i w_i - \sum_{i=1}^{N-1} v_{i-1}w_i - \sum_{i=1}^{N-1} v_{i+1}w_i \\ &= -\sum_{i=1}^{N-1}(v_{i+1} - 2v_i + v_{i-1})w_i.\end{aligned}$$

Hence,

$$\begin{aligned}-h\sum_{i=1}^{N-1}\frac{v((i+1)h, nh) - 2v(ih, nh) + v((i-1)h, nh)}{h^2} w(ih, nh) \\ = h\sum_{i=1}^{N} \partial_x v(ih, nh)\partial_x w(ih, nh),\end{aligned}$$

and thus

$$-\langle D_x^2 v, w\rangle_h = \langle \partial_x v, \partial_x w\rangle_h.$$

Similarly, $-\langle D_y^2 v, w\rangle_h = \langle \partial_y v, \partial_y w\rangle_h$, so the lemma follows. □

Combining the discrete Poincaré inequality with the discrete Green's theorem, we immediately get a stability result. If $v \in L(\Omega_h)$, then

$$\|v\|_h^2 \le \|\partial_x v\|_h^2 \le \|\partial_x v\|_h^2 + \|\partial_y v\|_h^2 = -\langle \Delta_h v, v\rangle_h \le \|\Delta_h v\|_h \|v\|_h.$$

Thus

$$\|v\|_h \le \|\Delta_h v\|_h, \quad v \in L(\Omega_h),$$

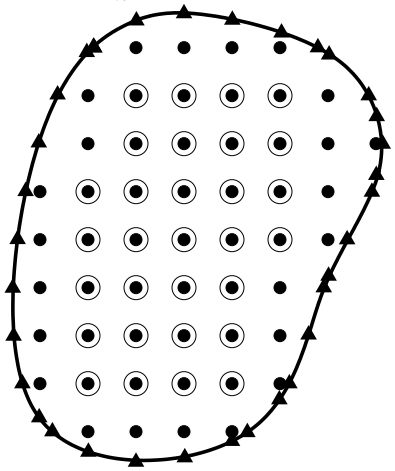
which is a stability result.

2.4. Curved boundaries. Thus far we have studied as a model problem the discretization of Poisson's problem on the square. In this subsection we consider a variant which can be used to discretize Poisson's problem on a fairly general domain.

Let Ω be a smoothly bounded open set in $\mathbb{R}^2$ with boundary Γ. We again consider the Dirichlet problem for Poisson's equation, (6.3), and again set $\Omega_h = \Omega \cap \mathbb{R}_h^2$. If $(x, y) \in \Omega_h$ and the segment $(x + sh, y)$, $0 \le s \le 1$ belongs to Γ, then the point $(x + h, y)$, which belongs to Ω_h, is a *neighbor* of (x, y) to the right. If this segment doesn't belong to Ω we define another sort of neighbor to the right, which belongs to Γ. Namely we define the neighbor to be the point $(x + sh, y)$ where $0 < s \le 1$ is the largest value for which $(x + th, y) \in \Omega$ for all $0 \le t < s$. The points of Γ so constructed (as neighbors to the right or left or above or below points in Ω_h) constitute Γ_h. Thus every point in Ω_h has four nearest neighbors all of which belong to $\bar{\Omega}_h := \Omega_h \cup \Gamma_h$. We also define $\mathring{\Omega}_h$ as those points in Ω_h all four of whose neighbor belong to Ω_h. See Figure 6.2.

In order to discretize the Poisson equation we need to construct a discrete analogue of the Laplacian $\Delta_h v$ for mesh functions v on $\bar{\Omega}_h$. Of course on $\mathring{\Omega}_h$, $\Delta_h v$ is defined as the usual 5-point Laplacian. For $(x, y) \in \Omega_h \setminus \mathring{\Omega}_h$, let $(x + h_1, y)$, $(x, y + h_2)$, $(x - h_3, y)$, and

FIGURE 6.2. The dots are points in Ω_h with the elements of $\mathring{\Omega}_h$ being circled. The triangles are the points of Γ_h.



$(x, y - h_4)$ be the nearest neighbors (with $0 < h_i \le h$), and let v_1, v_2, v_3, and v_4 denote the value of v at these four points. Setting $v_0 = v(x, y)$ as well, we will define $\Delta_h v(x, y)$ as a linear combination of the five values v_i. In order to derive the formula, we first consider approximating $d^2v/dx^2(0)$ by a linear combination of $v(-h_-)$, $v(0)$, and $v(h_+)$, for a function v of one variable. By Taylor's theorem

$$\begin{aligned}\alpha_- v(-h_-) + \alpha_0 v(0) + \alpha_+ v(h_+) = (\alpha_- + \alpha_0 + \alpha_+)v(0) + (\alpha_+ h_+ - \alpha_- h_-)v'(0) \\ + \frac{1}{2}(\alpha_+ h_+^2 + \alpha_- h_-^2)v''(0) + \frac{1}{6}(\alpha_+ h_+^3 - \alpha_- h_-^3)v'''(0) + \cdots .\end{aligned}$$

Thus, to obtain a consistent approximation we must have

$$\alpha_- + \alpha_0 + \alpha_+ = 0, \quad \alpha_+ h_+ - \alpha_- h_- = 0, \quad \frac{1}{2}(\alpha_+ h_+^2 + \alpha_- h_-^2) = 1,$$

which give

$$\alpha_- = \frac{2}{h_-(h_- + h_+)}, \quad \alpha_+ = \frac{2}{h_+(h_- + h_+)}, \quad \alpha_0 = \frac{-2}{h_- h_+}.$$

Note that we have simply recovered the usual divided difference approximation to d^2v/dx^2:

$$\alpha_- v(-h_-) + \alpha_0 v(0) + \alpha_+ v(h_+) = \frac{[v(h_+) - v(0)]/h_+ - [v(0) - v(-h_-)]/h_-}{(h_+ + h_-)/2} = 2v[-h_-, 0, h_+].$$

Returning to the 2-dimensional case, and applying the above considerations to both $\partial^2 v/\partial x^2$ and $\partial^2 v/\partial y^2$ we arrive at the *Shortley–Weller* formula for $\Delta_h v$:

$$\begin{aligned}&\Delta_h v(x, y) \\ &= \frac{2}{h_1(h_1 + h_3)}v_1 + \frac{2}{h_2(h_2 + h_4)}v_2 + \frac{2}{h_3(h_1 + h_3)}v_3 + \frac{2}{h_4(h_2 + h_4)}v_4 - \left(\frac{2}{h_1 h3} + \frac{2}{h_2 h_4}\right)v_0.\end{aligned}$$

Using Taylor's theorem with remainder we easily calculate that for $v \in C^3(\bar{\Omega})$,

$$\|\Delta v - \Delta_h v\|_{L^\infty(\Omega_h)} \le \frac{2M_3}{3}h,$$

where M_3 is the maximum of the L^∞ norms of the third derivatives of v. Of course at the mesh points in $\mathring{\Omega}_h$, the truncation error is actually $O(h^2)$, but for mesh points neighboring the boundary, it is reduced to $O(h)$.

The approximate solution to (6.3) is $u_h : \bar{\Omega}_h \to \mathbb{R}$ determined again by 6.4. This is a system of linear equations with one unknown for each point of Ω_h. In general the matrix won't be symmetric, but it maintains other good properties from the case of the square:

- it is sparse, with at most five elements per row
- it has negative diagonal elements and non-negative off-diagonal elements
- it is diagonally dominant.

Using these properties we can obtain the discrete maximum principle with virtually the same proof as for Theorem 6.3, and then a stability result as in Theorem 6.5 follows as before. In this way we can easily obtain an $O(h)$ convergence result.

In fact this result can be improved. Although the truncation error $\|\Delta\, u - \Delta_h\, u\|_{L^\infty(\Omega_h)}$ is only $O(h)$, it is $O(h^2)$ at all points except those neighboring the boundary, and these account for only $O(h^{-1})$ of the $O(h^{-2})$ points in Ω_h. Moreover these points are within h of the boundary, where the solution is known exactly. For both of these reasons the contribution to the error from these points is smaller than is seen from the simple argument outlined in the previous paragraph. A sharp convergence result was proven by Bramble and Hubbard in a paper in Numerische Mathematik 4 (1962), pp. 313–327.

THEOREM 6.12. *Let u be the solution to* (6.3) *and let u_h be the mesh function satisfying* (6.4). *Then*

$$\|u - u_h\|_{L^\infty(\bar{\Omega}_h)} \le \frac{M_4 d^2}{96} h^2 + \frac{2M_3}{3} h^3,$$

where d is the diameter of the smallest disk containing Ω and $M_k = \max_{i+j=k} \|\partial^k u/\partial x^i \partial y^j\|_\infty$.

Thus the rate of convergence is $O(h^2)$ as in the case of the square, and the points near the boundary contribute only a higher order term (despite the fact that the truncation error is of lower order there).

3. Finite element methods

3.1. The weak formulation of the Dirichlet problem. We start by considering Poisson's equation with homogeneous Dirichlet boundary conditions on a bounded plane domain Ω:

$$\Delta\, u = f \text{ in } \Omega, \quad u = 0 \text{ on } \Gamma. \tag{6.6}$$

Let's assume that f is continuous on $\bar{\Omega}$. Now if we multiply the differential equation by a *test function* v and integrate over Ω, we get that

$$\int_\Omega \Delta\, u\, v\, dx = \int_\Omega f v\, dx,$$

and conversely, if this equation is satisfied for all integrable functions v, then u satisfies Poisson's equation. In fact, it is sufficient that the equation be satisfied for all C^∞ functions

with compact support inside Ω. In particular, if v is a C^1 function on Ω which vanishes on Γ, we may integrate by parts to get

$$\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx = -\int_\Omega f v \, dx. \tag{6.7}$$

Thus it is evident that a C^2 function which vanishes on Γ satisfies Poisson's equation if and only if it satisfies (6.7) for all C^2 functions v which vanish on Γ. The use of C^2 functions is, however, not very natural for the formulation (6.7). A more natural space would be the *Sobolev space*

$$H^1(\Omega) = \{\, v \in L^2(\Omega) \,|\, \nabla v \in L^2(\Omega) \,\}.$$

This is a Hilbert space with inner product

$$\langle u, v \rangle_{H^1(\Omega)} = \int_\Omega (uv + \operatorname{grad} u \cdot \operatorname{grad} v) \, dx,$$

and corresponding norm

$$\|v\|_1 = \|v\|_{H^1(\Omega)} := \sqrt{\|v\|^2_{L^2(\Omega)} + \|\operatorname{grad} v\|^2_{L^2(\Omega)}}.$$

(Note: we are using the same notation $L^2(\Omega)$ both for real-valued and vector-valued square integrable functions.) We also define

$$\mathring{H}^1(\Omega) = \{\, v \in H^1(\Omega) \,|\, v|_\Gamma \equiv 0 \,\}.$$

We then define the *weak formulation* of the Dirichlet problem for Poisson's equation to be:

Find $u \in \mathring{H}^1(\Omega)$ such that

$$\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx = -\int_\Omega f v \, dx \quad \text{for all } v \in \mathring{H}^1(\Omega). \tag{6.8}$$

The weak formulation fits an abstract framework we shall see frequently. We have a Hilbert space V (namely $\mathring{H}^1(\Omega)$), a bilinear form $B : V \times V \to \mathbb{R}$ (given by the left-hand side of (6.8)), a linear functional $F : V \to \mathbb{R}$ (given by the right-hand side of (6.8)), and the weak formulation is

$$\text{Find } u \in V \text{ such that} \quad B(u, v) = F(v) \quad \text{for all } v \in V. \tag{6.9}$$

It is clear that if u is a C^2 function satisfying the classical formulation (6.6) of our boundary value problem, then $u \in \mathring{H}^1(\Omega)$ and u satisfies the weak formulation (6.8). Conversely, if u solves the weak formulation, and if u is C^2, then u is a classical solution to the boundary value problem. However the classical formulation and the weak formulation are not entirely equivalent, because it may happen that there is a solution to the weak formulation which is not C^2. It can be shown that a solution to the weak formulation is automatically smooth if both the forcing function f is smooth and the domain Ω has a smooth boundary, but such *elliptic regularity theorems* are not trivial.

REMARK. In defining the Sobolev spaces $H^1(\Omega)$ and $\mathring{H}^1(\Omega)$ we have glossed over several points. In the definition of the former space, we assumed that it is clear what is meant by $\operatorname{grad} v$ when $v \in L^2(\Omega)$. If the reader is familiar with the theory of distributions, this is no problem: she then knows that $\operatorname{grad} v$ in any case exists as a distribution, which may, or may not, then belong to L^2. Briefly, given an L^2 function v, a vector-valued L^2 function w is

equal to $\operatorname{grad} v$ if and only if $\int_\Omega v \operatorname{div} \phi \, dx = -\int_\Omega w \cdot \phi \, dx$ for all C^∞ vector-valued functions ϕ with compact support in Ω. A more subtle point is that in defining $\mathring{H}^1(\Omega)$, we have assumed that it is clear what is meant by $v|_\Gamma$ for $v \in H^1(\Omega)$. This is not so evident. For example, there is certainly no way to make sense of $v|_\Gamma$ for an arbitrary function $v \in L^2(\Omega)$ (which is defined only almost everywhere, and may not be defined anywhere on Γ). In fact, it can be shown that there is a unique bounded map $\gamma : H^1(\Omega) \to L^2(\Gamma)$ such that $\gamma v = v|_\Gamma$ for all $v \in H^1(\Omega) \cup C^(\bar{\Omega})$. This is an example of a *trace theorem*, and requires some effort to establish. By $v|_\Gamma$ we simply mean γv, for any $v \in H^1(\Omega)$.

The weak formulation is in many ways a very natural formulation of the Dirichlet problem for Poisson's equation. One indication of this is that it is quite simple to establish existence and uniqueness of weak solutions. One first establishes Poincaré's inequality, which states that there exists a constant c depending only on the domain Ω such that $\|u\|_{L^2(\Omega)} \le c\|u\|_{H^1(\Omega)}$ for all $u \in \mathring{H}^1(\Omega)$. This is fairly elementary; a good exercise is to prove it in the one-dimensional case where Ω is a bounded interval (what is the best constant c?). It follows that on the space $\mathring{H}^1(\Omega)$, the quantity $\sqrt{\|\operatorname{grad} u\|_{L^2}}$ is a norm equivalent to the full H^1 norm, and hence the left-hand side of (6.8) defines an inner product on $\mathring{H}^1(\Omega)$ which is equivalent to the H^1 inner product. Existence and uniqueness of a weak solution is then an immediate consequence of the Riesz representation theorem.

Remark. Besides the weak formulation of the boundary value problem (6.6), there is a closely related variational formulation. In this we seek $u \in \mathring{H}^1(\Omega)$ which minimizes the *energy functional*

$$E(w) := \frac{1}{2}\int_\Omega |\operatorname{grad} w|^2 \, dx + \int_\Omega f w \, dx$$

over $w \in \mathring{H}^1(\Omega)$. If u is the solution of the weak formulation and $w \in \mathring{H}^1(\Omega)$ with $w \ne u$ but otherwise arbitrary, we may write $w = u + v$, with $0 \ne v \in \mathring{H}^1(\Omega)$, and then

$$\begin{aligned} E(w) &= \frac{1}{2}\int_\Omega |\operatorname{grad} u|^2 \, dx + \int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx + \frac{1}{2}\int_\Omega |\operatorname{grad} v|^2 \, dx + \int_\Omega f u \, dx + \int_\Omega f v \, dx \\ &= E(u) + \left[\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx + \int_\Omega f v \, dx\right] + \frac{1}{2}\int_\Omega |\operatorname{grad} v|^2 \, dx. \end{aligned}$$

The term in brackets vanishes since u is the weak solution, and the final term is positive if $v \ne 0$. Thus $E(u) < E(w)$, and u is indeed the minimizer. Conversely, if u minimizes E over $\mathring{H}^1(\Omega)$ and $v \in \mathring{H}^1(\Omega)$ is arbitrary, then the quadratic function $G : \mathbb{R} \to \mathbb{R}$ given by

$$G(t) = E(u + tv) = E(u) + t\left[\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx + \int_\Omega f v \, dx\right] + \frac{t^2}{2}\int_\Omega |\operatorname{grad} v|^2 \, dx,$$

has its minimum at $t = 0$, and this immediately shows that u is a weak solution. Thus the notion of a weak solution and a variational solution (a minimizer of the energy functional) are equivalent for this problem. For problems which are not symmetric however (e.g., if the PDE were $\Delta u + \partial u/\partial x = f$), there is no natural variational formulation, while the weak formulation still applies.

3.2. Galerkin's method. In the weak formulation, we seek a function in the *trial space* $\mathring{H}^1(\Omega)$ which satisfies the weak equation (6.8) for all v in the *test space* $\mathring{H}^1(\Omega)$. In Galerkin's method we choose a finite dimensional space $S_h \subset \mathring{H}^1(\Omega)$, and use it in place of $\mathring{H}^1(\Omega)$ as trial and test space. That is, we seek $u_h \in S_h$ satisfying the *discrete weak formulation*

$$\int_\Omega \operatorname{grad} u_h \cdot \operatorname{grad} v \, dx = -\int_\Omega fv \, dx \quad \text{for all } v \in S_h.$$

Let us show that such a function u_h exists and is unique. Let $\{\phi_1, \dots, \phi_N\}$ be any basis of S_h. Then we are seeking $u_h = \sum_{j=1}^N \alpha_j \phi_j$ such that

$$\int_\Omega \operatorname{grad} u_h \cdot \operatorname{grad} \phi_i \, dx = -\int_\Omega f\phi_i \, dx, \quad i = 1, \dots, N.$$

(By linearity, it is enough that the weak equation hold for each basis function.) This means that

$$\sum_{j=1}^N \alpha_j \int_\Omega \operatorname{grad} \phi_j \cdot \operatorname{grad} \phi_i \, dx = -\int_\Omega f\phi_i \, dx.$$

If we define the *stiffness matrix* $M \in \mathbb{R}^{N\times N}$ by

$$M_{ij} = \int_\Omega \operatorname{grad} \phi_j \cdot \operatorname{grad} \phi_i \, dx,$$

and the *load vector* $\boldsymbol{F} \in \mathbb{R}^N$ by

$$F_i = -\int_\Omega f\phi_i \, dx,$$

then the coefficient vector $\boldsymbol{\alpha}$ is determined as the solution of the linear system

$$M\boldsymbol{\alpha} = \boldsymbol{F}.$$

The matrix M is clearly symmetric, and it is positive definite as well, since for any vector $\boldsymbol{\alpha} \in \mathbb{R}^N$,

$$\begin{aligned}\boldsymbol{\alpha}^T M \boldsymbol{\alpha} &= \sum_{ij} \alpha_i \alpha_j \int_\Omega \operatorname{grad} \phi_j \cdot \operatorname{grad} \phi_i \, dx \\ &= \int_\Omega \operatorname{grad} \left(\sum_j \alpha_j \phi_j \right) \cdot \operatorname{grad} \left(\sum_i \alpha_i \phi_i \right) dx = \int_\Omega |\operatorname{grad} v|^2 \, dx,\end{aligned}$$

where $v = \sum_j \alpha_j \phi_j$. Since the latter quantity is a norm of v it is positive unless $v \equiv 0$ which only happens if $\boldsymbol{\alpha} = 0$.

Thus Galerkin's method is implementable. If we choose a space S_h for which we can find a basis which is not too complicated, we can compute the N^2 integrals giving the stiffness matrix and the N integrals giving the load vector, and then solve the resulting $N \times N$ symmetric positive definite linear system to find the coefficients of u_h with respect to the basis.

Remark. Instead of beginning with the weak formulation and restricting the test and trial spaces to a finite dimensional subspace, we may start with the variational formulation and restrict the trial space to S_h. That is, we define $u_h \in S_h$ to be the minimizer of $E(v)$ over

$v \in S_h$. This is called the Ritz method. Following the proof for the continuous case we see that u_h is in fact just the Galerkin solution. This viewpoint, that the approximate solution is determined by restricting the minimization of energy to a finite dimensional space, was the first motivation for the finite element method. However, we obtain greater generality by using the weak formulation and Galerkin's method rather than the variational formulation and the Ritz method.

3.3. A simple finite element method. The finite element method consists of the Galerkin method together with the choice of a particular sort of subspace S_h. Namely we partition Ω into a finite number of disjoint triangles or other simple pieces, and take S_h to be a space of piecewise polynomials with respect to this partition.

More specifically, let us assume, for now, that Ω is a polygon. Given a triangulation $\mathcal{T}_h$ we define $M_0^1(\mathcal{T}_h)$ to be the space of continuous piecewise linear functions subordinate to this triangulation, that is the space of continuous functions on $\bar{\Omega}$ which restrict to polynomials of degree at most 1 on each $T \in \mathcal{T}_h$. The notation is the same as in Chapter 1.5: the superscript 1 refers to the polynomial degree and the subscript 0 to the fact that the continuity of C^0 is enforced. It is easy to check that a piecewise polynomial is in H^1 if and only if it is continuous: the gradient of a continuous piecewise polynomial is the L^2 function obtained by taking the gradient triangle by triangle. Thus for any triangulation, $M_0^1(\mathcal{T}_h)$ is a subspace of H^1. As a subspace of $\mathring{H}^1$, we then take

$$S_h = \mathring{M}_0^1(\mathcal{T}_h) = M_0^1(\mathcal{T}_h) \cap \mathring{H}^1(\Omega).$$

This is the simplest finite element space for the Dirichlet problem. A basis for S_h is given by the hat functions associated with all the interior vertices.

Recall that the hat functions are local basis functions in that each is supported on just a few triangles. Namely, the support of ϕ_j is the union of the triangles sharing the vertex x_j), and there will only be a few other basis functions whose support contains one of these triangles, namely the basis functions associated to vertices of triangles in the support of ϕ_j. If ϕ_i is any other basis function, then the corresponding stiffness matrix entry

$$\int \operatorname{grad} \phi_j \cdot \operatorname{grad} \phi_i \, dx = 0.$$

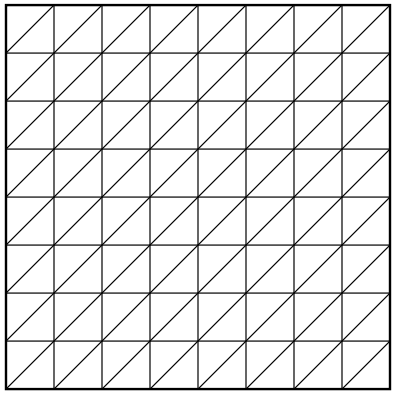
As a consequence we see that with the hat function basis, the stiffness matrix is very sparse (there will be only a few nonzero entries per row, and this number will not increase when we refine the mesh, as long as we don't allow the triangle angles to decrease). Also the stiffness matrix entries which are nonzero are nonetheless easily computed: they are sums of integrals of polynomials over only a few triangles (in fact, except for the diagonal entries of the stiffness matrix, over two triangles). This adds greatly to the efficiency of the implementation of the Galerkin method.

We may now roughly define the finite element method: it is Galerkin's method using a piecewise polynomial trial space with a local basis.

Let us now work out a simple example. We take Ω to be the unit square with a uniform mesh. Namely we divide Ω into $n \times n$ subsquares of size $h = 1/n$ and divide each of these into two triangles using the positively sloped diagonal. See Figure 6.3.

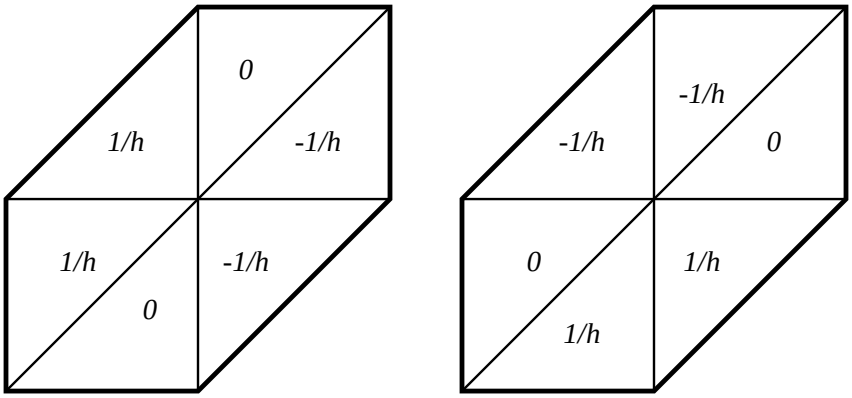
Now if ϕ is a linear function, then $\partial\phi/\partial x$ is constant, and we can find it by evaluating ϕ at any two distinct points on a horizontal line, and taking a difference quotient. Since each

FIGURE 6.3. Uniform mesh of the square.



of our triangles has a horizontal and a vertical edge, and since we know the values of the hat basis functions at the vertices, we can easily compute the partial derivatives of the basis functions. These are shown in the following figure.

FIGURE 6.4. Values of $\partial\phi/\partial x$ and $\partial\phi/\partial y$ for a hat function.



It is then easy to compute the stiffness matrix. Writing ϕ_{ij} for the basis function associated to the vertex $x_{ij} = (ih, jh)$, we find

$$\int \operatorname{grad}\phi_{ij} \cdot \operatorname{grad}\phi_{kl}\, dx = \begin{cases} 4, & i = k, j = l, \\ -1, & i = k \pm 1, j = l \text{ or } i = k, j = l \pm 1, \\ 0, & \text{else.} \end{cases}$$

In other words, *the stiffness matrix for piecewise linear finite elements for the Laplace operator on the unit square using a uniform mesh is exactly the matrix of the five-point Laplacian.* If we set

$$\tilde{f}_{ij} := \frac{1}{h^2} \int f\phi_{ij}.$$

and write the finite element solution as $u_h = \sum_{i,j} U_{ij}\phi_{ij}$, we have

$$\frac{4U_{ij} - U_{i+1,j} - U_{i-1,j} - U_{i,j+1} - U_{i,j-1}}{h^2} + \tilde{f}_{ij} = 0.$$

Note that $\int \phi_{ij}\,dx = h^2$ and if f is at least C^2, then $\tilde{f}_{ij} = f(x_{ij}) + O(h^2)$.

From this fact and our earlier analysis of the five-point Laplacian, it is easy to derive an $O(h^2)$ convergence estimate at the vertices. In fact, as we shall see, one of the great strengths of the finite element method is that there is a very natural approach to the error analysis (which does not involve relating it to finite difference methods). The most natural estimates are obtained in $H^1(\Omega)$ and $L^2(\Omega)$ (error estimates in $L^\infty(\Omega)$ can also be obtained, but are more complicated). Below we shall prove, in more general circumstances, that if the solution $u \in H^2(\Omega)$, then

$$\|u - u_h\|_{H^1(\Omega)} \le Ch\|u\|_{H^2(\Omega)}, \quad \|u - u_h\|_{L^2(\Omega)} \le Ch^2\|u\|_{H^2(\Omega)},$$

for some constant C. Note that unlike our finite difference estimates the norms are norms of functions on Ω, not just at the mesh vertices. Note also that we only require that $u \in H^2$, not even C^2, to obtain $O(h^2)$ convergence. By contrast in the finite difference case we needed $u \in C^4$. For example, if the function f is merely in L^2, then it may not even be defined at the mesh points, and the standard finite difference method is not meaningful, while the finite element method is applicable and will still deliver second order convergence in this case.

Most important, the same error estimates hold on a quite arbitrary domain with an arbitrary triangulation. In this case h is to be interpreted as the diameter of the largest triangle in the triangulation. The constant C in the estimates will depend on the domain, but not on the triangulation (except that we will need to enforce a bound on the smallest angle of a triangle). The derivation of such error estimates will be discussed later. The point to note now is the power and flexibility of the finite element method in providing second order convergent schemes on unstructured meshes. If we were to try to derive such schemes directly, e.g., by Taylor expansions, it would be very messy indeed.

3.4. Application to more general problems. Another great strength of finite element methods is the ease with which they can be adapted to a wide variety of problems.

3.4.1. *More general elliptic PDEs.* For example, suppose we replace Poisson's equation $\Delta u = f$ with a more general second order PDE

$$\sum_{i,j=1}^{2} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right) + \sum_{i=1}^{2} b_i \frac{\partial u}{\partial x_i} + cu = f.$$

Here the coefficients a_{ij}, b_i, and c may be functions of x. Once again, we may multiply by a test function $v \in \mathring{H}^1(\Omega)$ and integrate over Ω by parts to obtain a weak formulation of the Dirichlet problem in the form (6.9). The only difference is that now

$$B(u,v) = \int_\Omega \sum_{i,j} a_{ij}\frac{\partial u}{\partial x_j}\frac{\partial v}{\partial x_i} - \sum_i b_i \frac{\partial u}{\partial x_i} v - cuv\,dx.$$

Restricting the trial and test functions to S_h, we obtain a finite element method as before. If the PDE is elliptic, which means that the matrix $a_{ij}(x)$ is symmetric positive-definite, uniformly in x, then the behavior of the finite element method for this problem will be very similar to that for the Poisson problem. Thus the finite element method is well able to handle variable coefficients, anisotropic equations, and lower order terms.

We can even allow the coefficients to depend on the solution u, and so have a nonlinear PDE. In that case the form $B(u,v)$ will be linear in v but nonlinear in u. Again the finite

element method can be applied, although of course the resulting system of algebraic equations will be nonlinear.

3.4.2. *Neumann boundary conditions.* Yet another great strength of finite element methods is their flexibility in handling different boundary conditions (which can be very tricky for finite difference methods). Consider for example Poisson's equation on a polygon but suppose that some sides are subject to the Dirichlet boundary condition $u = 0$ and some to the Neumann boundary condition, so the problem is to find u satisfying

$$\Delta u = f \text{ in } \Omega, \quad u = 0 \text{ on } \Gamma_D, \quad \frac{\partial u}{\partial n} = 0 \text{ on } \Gamma_N,$$

where Γ_D and Γ_N are disjoint open subsets of Γ such that $\Gamma = \bar{\Gamma}_D \cup \bar{\Gamma}_N$. The first thing you might think to do is to seek u in a subspace of $H^1(\Omega)$ satisfying both boundary conditions, i.e., in

$$\{\, v \in H^1(\Omega) \,|\, v|_{\Gamma_D} \equiv 0,\ \partial v/\partial n|_{\Gamma_N} \equiv 0 \,\}.$$

However, this does not work, because there is no way to define $\partial v/\partial n$ for $v \in H^1$. Such a function has a well defined restriction to Γ (or Γ_D) but its first derivatives, which are merely L^2 functions, do not. To see our way around this problem, let us multiply Poisson's equation by a smooth test function v and integrate over Ω.

Green's formula for integration by parts shows us the way around this difficulty. For any smooth u and v we have

$$\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx = -\int_\Omega \Delta u \, v \, dx + \int_\Gamma \frac{\partial u}{\partial n} v \, ds.$$

If u satisfies Poisson's equation, then

$$\int_\Omega \Delta u \, v \, dx = \int_\Omega f \, v \, dx,$$

and if u satisfies the Neumann boundary condition, then

$$\int_\Gamma \frac{\partial u}{\partial n} v \, ds = \int_{\Gamma_D} \frac{\partial u}{\partial n} v \, ds.$$

Define

$$H^1_D(\Omega) = \{\, v \in H^1(\Omega) \,|\, v|_{\Gamma_D} \equiv 0 \,\}.$$

Note that in this space the Dirichlet boundary condition has been imposed, but the Neumann boundary condition has been ignored (since there is no way to make sense of it in H^1). This leads us to the *weak formulation for the mixed Dirichlet/Neumann boundary value problem*: Find $u \in H^1_D(\Omega)$ such that

$$\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx = -\int_\Omega f \, v \, dx \text{ for all } v \in H^1_D(\Omega).$$

Just as for the pure Dirichlet problem, as an easy consequence of the Riesz representation theorem and the Poincaré inequality, this problem has a unique solution.

In deriving the weak formulation, we have shown that if u is a classical solution to the boundary value problem then it satisfies this weak formulation. Conversely, if u solves the weakly formulated problem, and u is C^2, then we can integrate by parts to find that

$$-\int_\Omega \Delta u \, v \, dx + \int_\Gamma \frac{\partial u}{\partial n} v \, ds = -\int_\Omega f \, v \, dx \text{ for all } v \in H^1_D(\Omega).$$

Taking first v to be smooth and compactly supported in Ω, we conclude that $\Delta u = f$ in Ω. Therefore

$$\int_\Gamma \frac{\partial u}{\partial n} v \, ds = 0$$

for all $v \in H^1_D(\Omega)$. Since such a function can be arbitrary on Γ_N, we conclude that $\partial u/\partial n = 0$ on Γ_N.

To summarize: the weak formulation for the mixed Dirichlet/Neumann boundary value problem has a unique solution. This coincides with the classical solution whenever it is C^2 and whenever a classical solution exists. Note that the Dirichlet and Neumann boundary conditions are treated completely differently in the weak formulation. The Dirichlet condition is imposed a priori by building it into the trial space. The Neumann condition is not built into the trial space, but arises as a consequence of the weak formulation. The terminology used for this is that the Dirichlet condition is an *essential* boundary condition, while the Neumann condition is *natural*.

Once we have the weak formulation we can use Galerkin's method with any subspace S_h of $H^1_D(\Omega)$. This leads to a symmetric positive definite matrix problem. As long as we arrange that each triangle edge in the boundary belongs entirely to either Γ_D or Γ_N we can easily construct a piecewise linear finite element space:

$$M^1_{0D}(\mathcal{T}_h) = M^1_0(\mathcal{T}_h) \cap H^1_D(\Omega).$$

A local basis is given by the hat functions at all the triangulation vertices except those belonging to $\bar{\Gamma}_D$.

We didn't consider the case of pure Neumann conditions because the boundary value problem $\Delta u = f$ in Ω, $\partial u/\partial n = 0$ on Γ is not well-posed. Green's theorem implies that there is no solution unless $\int_\Omega f = 0$, and if there is a solution, we can add any constant to it to get another solution. If we consider instead the differential equation $-\Delta u + u = f$, or, more generally, $-\operatorname{div}(A \operatorname{grad} u) + cu = f$ where A is a symmetric positive definite matrix and $c > 0$, this problem goes away and the considerations above apply equally well to the pure Neumann problem ($\Gamma_N = \Gamma$, $\Gamma_D = \emptyset$).

3.4.3. *Inhomogeneous boundary conditions.* Thus far we have considered homogeneous Dirichlet and Neumann boundary conditions. Now we discuss inhomogeneous boundary conditions. Natural boundary conditions are straightforward. If $\partial u/\partial n = g$ on Γ_N, then for any test function $v \in H^1_D(\Omega)$,

$$\int_\Gamma \frac{\partial u}{\partial n} v \, ds = \int_{\Gamma_N} g v \, ds,$$

and so the weak formulation becomes: Find $u \in H^1_D(\Omega)$ such that

$$\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx = -\int_\Omega f \, v \, dx + \int_{\Gamma_N} g v \, ds \text{ for all } v \in H^1_D(\Omega).$$

This is again of the form $B(u, v) = F(v)$, where now $F(v)$ contains an extra term arising from the Neumann data.

Essential boundary conditions need to be built into the trial space. That is, if the boundary condition is $u = g$ on Γ (for simplicity suppose $\Gamma_N = \emptyset$), we seek

$$u \in H^1_g(\Omega) := \{\, v \in H^1(\Omega) \mid v|_\Gamma \equiv g \,\}.$$

The weak equations are still

$$\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx = -\int_\Omega f\, v \, dx \text{ for all } v \in H^1_D(\Omega),$$

that is, the test space remains homogeneous. To see that there is still a unique solution, choose any function $u_g \in H^1(\Omega)$ such that $u_g|_\Gamma = g$. Then

$$H^1_g(\Omega) = \{\, u_g + v \,|\, v \in \mathring{H}^1(\Omega) \,\},$$

and the weak equations are satisfied by $u = u_g + u_0$ if and only if $u_0 \in \mathring{H}^1(\Omega)$ and

$$B(u_0, v) = F(v) - B(u_g, v) \text{ for all } v \in \mathring{H}^1(\Omega).$$

For the finite element solution we thus can again use $\mathring{M}^1_0(\mathcal{T}_h)$ as test space, but we cannot use as trial space $M^1_0(\mathcal{T}_h) \cap H^1_g(\Omega)$, since, unless g happens to be piecewise linear, this space is empty. Instead we use $M^1_0(\mathcal{T}_h) \cap H^1_{\bar g}(\Omega)$ where $\bar g$ is some piecewise linear approximation to g (with respect to the partition of Γ into edges of triangles of $\mathcal{T}_h$), e.g., its piecewise linear interpolant, or $L^2(\Gamma)$-projection.

3.4.4. *Robin boundary conditions.* As another example, we consider Poisson's problem with *Robin boundary conditions*, which model Newton's law of cooling at the boundary (heat flow through the boundary is proportional to the difference between the body's temperature and the outside temperature). This gives the boundary condition

$$\frac{\partial u}{\partial n} = \alpha(g - u)$$

where g is the outside temperature and α is a positive constant (positive since heat flow out of the domain is positively proportional to $-\partial u/\partial n$ and heat should flow out if u exceeds g. Replacing g with αg we come to a boundary value problem like

$$\Delta\, u = f \text{ in } \Omega, \quad \frac{\partial u}{\partial n} + \alpha u = g \text{ on } \Gamma.$$

Since the boundary condition involves first derivatives, it cannot be imposed in $H^1(\Omega)$. Therefore the Robin boundary condition will prove to be a natural boundary condition. To find the correct weak formulation, we multiply by a test function and integrate by parts to obtain

$$\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx - \int_\Gamma \frac{\partial u}{\partial n} v \, dx = -\int_\Omega f v \, dx.$$

Using the boundary condition we can rewrite this is as

$$\int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx + \int_\Gamma \alpha u v \, dx = -\int_\Omega f v \, dx + \int_\Gamma g v \, dx.$$

The weak formulation is thus again of the form: Find $u \in V$ such that $B(u, v) = F(v)$ for all $v \in V$, where $V = H^1(\Omega)$, as for the pure Neumann problem, but now

$$B(u, v) = \int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx + \int_\Gamma \alpha u v \, dx, \quad F(v) = -\int_\Omega f v \, dx + \int_\Gamma g v \, dx.$$

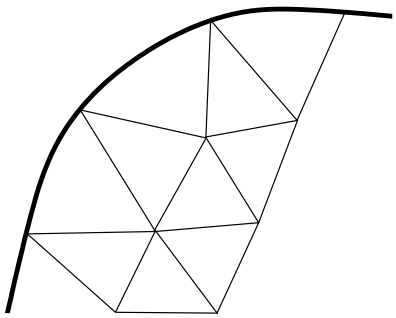
In view of the Poincaré inequality, B is again a bounded symmetric bilinear form on $H^1 \times H^1$ and F is a bounded linear form. Moreover $B(u, u) \geq \|\operatorname{grad} u\|^2_{L^2}$, and $B(u, u) = 0$ only if $u \equiv 0$ (since for such u, $\operatorname{grad} u = 0$ and $u = 0$ on Γ). From this it follows (by a small argument

which we omit) that B is equivalent to the usual inner product in $H^1(\Omega)$. Thus Robin boundary conditions can be incorporated easily into the weak formulation, and therefore into a finite element discretization.

3.4.5. *Curved boundaries.* Finally we consider, briefly, the case where Ω has a curved rather than polygonal boundary. This doesn't affect the weak formulation, but the design of finite element subspaces of $H^1(\Omega)$ or $H^1_D(\Omega)$ or $\mathring{H}^1(\Omega)$ is non-trivial and has engendered many algorithms, codes, and papers. By now curved boundaries are handled routinely in the finite element method, but they do require additional effort.

In the case of natural boundary conditions, there is one obvious possibilitity. We may triangulate a curved domain using ordinary triangles except for a layer of triangles containing one curved edge near the boundary. See the figure.

FIGURE 6.5. A portion of a triangulation using curvilinear triangles near the boundary.



We may then specify a space of piecewise linear functions as before, by determining a function on each triangle, straight or curved, by giving its vertex values. The only difficulty with this approach is that it may not be straightforward to compute the necessary integrals in the stiffness matrix.

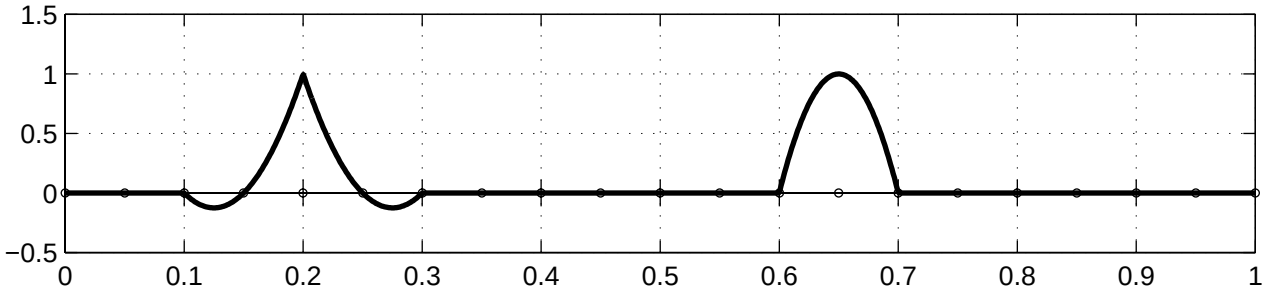
In the case of Dirichlet boundary conditions there is a considerable additional difficulty. Suppose we have a curvilinear triangle with two vertices on the boundary connected by a curve e contained in the boundary. A linear function on the triangle which vanishes at the two vertices will vanish on the entire straight line connecting them, not on the curve e. One way to surmount this problem is not to face it. Instead, simply replace the original domain Ω with a polygonal domain Ω_h obtained by interpolating the vertices on the boundary by a polygonal curve. This of course introduces an additional source of error. In fairly general circumstances it can be shown that this new error doesn't degrade the accuracy of finite element methods based on piecewise linear finite elements, but it does degrade the accuracy of higher order finite elements such as will be discussed below. This can be overcome by simultaneously using higher order polynomial interpolation to the boundary, e.g., approximating the curved edges of triangles by parabolic curves when using quadratic finite elements. Yet another possibility is not to use polynomials trial functions at all on the curved triangles. Instead one can construct a smooth mapping of a straight-edged reference triangle onto a curvilinear triangles and use polynomial functions on the reference triangle. The trial functions used on the curvilinear triangle are then the composition of the inverse mapping to the reference triangle and polynomials on the reference triangle. A strategy which turns out to be relatively easy to implement and to maintain good accuracy is to use

a polynomial map from the reference polynomial to the true polynomial using polynomials as the same degree as the trial functions on the reference triangle. This scheme is known as *isoparametric* finite elements.

3.5. Other finite element spaces. We have thus seen many examples of boundary value problems for which the trial and test spaces are $H^1(\Omega)$ or a subspace of it incorporating essential boundary conditions. For subspaces we have only considered piecewise linear finite elements. One may also use piecewise polynomials of higher degree, and this turns out to give finite element methods with higher rates of convergence.

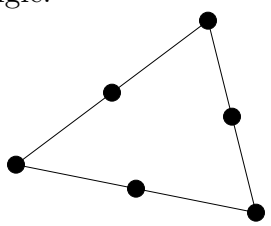
We begin in one dimension: $\Omega = (0,1)$, $\mathcal{T}_h = \{I_1, \ldots, I_n\}$, with $I_n = [x_{n-1}, x_n]$, $0 = x_0 < x_1 < \cdots < x_N = 1$. In this case we studied the piecewise polynomial spaces $M_0^p(\mathcal{T}_h)$ for any degree $p > 0$ in Chapter 1. We may use as a set of degrees of freedom the value of the function at the nodes x_n and at $p-1$ points placed inside each element. This leads us to a local basis. Typical basis elements for $p = 2$ are shown in Figure 6.6.

FIGURE 6.6. Some local basis functions for piecewise quadratic functions using a uniform partition of the unit interval into 10 equal elements. On the axis, vertical lines indicate element boundaries; circles indicate nodes.



In 2 dimensions we can construct higher degree finite element spaces in a similar manner. If T is a triangle, we can uniquely specify a quadratic function on T by giving its value at the three vertices and the three edge midpoints. See Figure 6.7. (Proof: since there are six coefficients and six degrees of freedom it suffices to show that a quadratic vanishing at these six points is identically zero. Since it vanishes at three points on each edge, it must vanish on each edge. If the ith edge is given by $l_i = 0$ where l_i is a linear polynomial, then the quadratic must be divisible by $l_1 l_2 l_3$, which implies that it must be zero.)

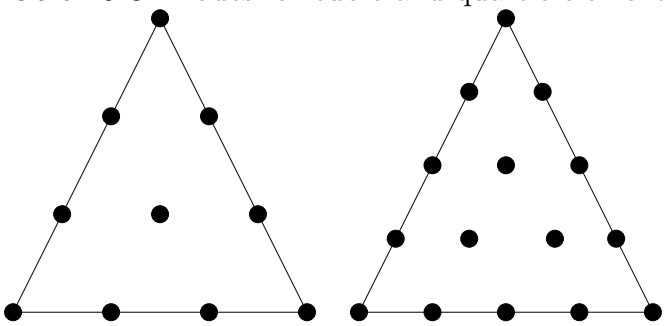
FIGURE 6.7. Evaluation points, or nodes, for the degrees of freedom of a quadratic function on a triangle.



Now suppose we specify a piecewise quadratic function by giving values at all the *nodes*, that is, the vertices and edge midpoints of the triangulation. If two triangles share a common edge, the two quadratic polynomials on the edge obtained by restricting the quadratic polynomials on each of the two triangles to the edge will agree at three points on the edge. Since a quadratic funtion on a line is determined by its value at three points, the restrictions will agree. Thus the piecewise quadratic will be continuous. This shows that the values at the nodes do indeed form a set of degrees of freedom for the space $M_0^2(\mathcal{T}_h)$ of continuous piecewise quadratics with respect to the triangulation $\mathcal{T}_h$. The dimension of this space is thus the sum of the number of vertices and the number of edges, and we have a local basis.

Similar considerations apply to higher degree elements. The figure shows the nodal configuration for continuous piecewise cubics and quartics.

FIGURE 6.8. Nodes for cubic and quartic elements.



3.6. Finite element approximation theory. We have seen that for many 2nd order elliptic boundary value problems, the weak formulation is of the form:

Find $u \in V$ such that

$$B(u, v) = F(v) \text{ for all } v \in V,$$

where V is some closed subspace of $H^1(\Omega)$ incorporating essential boundary conditions, $B : V \times V \to \mathbb{R}$ is a bounded bilinear, and $F : V \to \mathbb{R}$ is a bounded linear form. In addition to the boundedness,

$$|B(v, w)| \le C\|v\|_1\|w\|_1, \quad v, w \in V,$$

for some constant C, we shall also assume coercivity in the sense that

$$B(v, v) \ge \gamma\|v\|_1^2, \quad v \in V,$$

where γ is a positive constant. (This assumption is satisfied in many cases, although it can be weakened in various ways. If B is also symmetric—which we don't assume—then the boundedness and coercivity together mean that B gives an inner product equivalent to the usual one.) The Galerkin solution is defined as $u_h \in S_h$ such that

$$B(u_h, v) = F(v) \text{ for all } v \in S_h.$$

We then have $B(u_h, v) = B(u, v)$ for all $v \in S_h$, so if χ is an arbitrary element of S_h, $B(u_h - \chi, v) = B(u - \chi, v)$ for all $v \in S_h$. Choosing $v = u_h - \chi$ and using the coercivity and

boundedness, we get

$$\begin{aligned}\gamma\|u_h-\chi\|_1^2 &\le B(u_h-\chi,u_h-\chi)\\ &= B(u-\chi,u_h-\chi)\\ \le C\|u-\chi\|_1\|u_h-\chi\|_1.&\end{aligned}$$

Thus $\|u_h-\chi\|_1 \le (C/\gamma)\|u-\chi\|_1$, and $\|u_h-u\|_1 \le (1+C/\gamma)\|u-\chi\|_1$. Since $\chi \in S_h$ is arbitrary, this gives

$$\|u-u_h\|_1 \le c \inf_{\chi\in S_h}\|u-\chi\|_1, \tag{6.10}$$

where $c = 1 + C/\gamma$. This says that when measured in H^1, the error in the Galerkin approximation is no worse than a constant factor times the error in the best approximation to u from S_h (with the constant independent of the particular solution u and the particular subspace S_h). This property of the finite element solution is called *quasioptimality*.

The fundamental quasioptimality result (6.10) reduces the error in a Galerkin approximation to a question of approximation theory. We studied this question in some detail in Chapter 1.6.2. There we obtained the following theorem.

THEOREM 6.13. *Let there be given a family of triangulations $\{\mathcal{T}_h\}$ of a polygonal domain Ω and let $h = \max_{T\in\mathcal{T}_h}\operatorname{diam}(T)$. Let r be a positive integer. For each h let $\Pi_h : C(\Omega) \to M_0^r(\mathcal{T}_h)$ denote the nodal interpolant. Then there is a constant c such that*

$$\|u-\Pi_h u\|_{L^\infty(\Omega)} \le ch^{r+1}\|u^{(r+1)}\|_{L^\infty(\Omega)} \text{ for all } u \in C^{r+1}(\bar\Omega),$$
$$\|u-\Pi_h u\|_{L^2(\Omega)} \le ch^{r+1}\|u^{(r+1)}\|_{L^2(\Omega)} \text{ for all } u \in H^{r+1}(\Omega).$$

If, moreover, the family of triangulations is shape regular, then there is a constant C such that

$$\|\operatorname{grad}(u-\Pi_h u)\|_{L^\infty(\Omega)} \le Ch^{r}\|u^{(r+1)}\|_{L^\infty(\Omega)} \text{ for all } u \in C^{r+1}(\bar\Omega),$$
$$\|\operatorname{grad}(u-\Pi_h u)\|_{L^2(\Omega)} \le Ch^{r}\|u^{(r+1)}\|_{L^2(\Omega)} \text{ for all } u \in H^{r+1}(\Omega).$$

3.7. Error estimates. The combination of the quasioptimality estimate (6.10) and the bounds on interpolation error of the last subsection immediately gives an H^1 error estimate for the finite element method using piecewise linear elements:

$$\|u-u_h\|_1 \le Ch\|D^2u\|_{L^2(\Omega)},$$

or simply

$$\|u-u_h\|_1 \le Ch\|u\|_2,$$

where the norm on the right-hand side is the Sobolev 2 norm. For elements of degree r the analogous result is

$$\|u-u_h\|_1 \le Ch^r\|u\|_{r+1}.$$

In these estimates the constant C doesn't grow as h tends to zero as long as the meshes remain shape regular.

Thus the error in the finite element method with piecewise linear elements converges to 0 with order h in $H^1(\Omega)$ as the mesh is refined. Since the method is closely related to the five point Laplacian when applied to Poisson's equation with a uniform mesh on the square,

we might expect convergence with order h^2. In fact we can obtain such a result by looking at the rate of convergence in L^2 rather than H^1.

To be definite, let us consider the Dirichlet problem for Laplace's equation, so that the bilinear form is

$$B(u,v) = \int_\Omega \operatorname{grad} u \cdot \operatorname{grad} v \, dx,$$

and the finite element subspace is $S_h = \mathring{M}_0^1(\mathcal{T}_h)$.

To obtain an L^2 estimate we define an auxiliary function $\phi \in \mathring{H}^1(\Omega)$ as the solution of the weakly posed boundary value problem

$$\int_\Omega \operatorname{grad} v \cdot \operatorname{grad} \phi \, dx = \int_\Omega v(u - u_h)\, dx \text{ for all } v \in \mathring{H}^1(\Omega). \tag{6.11}$$

This is nothing other than the weak formulation of the boundary value problem

$$-\Delta \phi = u - u_h \text{ in } \Omega, \quad \phi = 0 \text{ on } \Gamma,$$

that is the Dirichlet problem for Poisson's equation where the error $u - u_h$ is taken as the forcing term. At this point we have to invoke an elliptic regularity theorem. Namely if the domain Ω is convex, then it is known that the solution ϕ to the boundary value problem belongs to $H^2(\Omega)$ and there exists a constant c such that $\|\phi\|_2 \le c\|u - u_h\|_0$; that is, the Sobolev 2-norm of the solution is bounded by the L^2 norm of the forcing term. (The same result holds when the boundary of Ω is smooth, but not in general if Ω is a non-convex polygon.)

Setting $v = u - u_h$ in (6.11) we obtain

$$\|u - u_h\|_0^2 = B(u - u_h, \phi) = B(u - u_h, \phi - v) \text{ for all } v \in S_h.$$

The last equality follows from the Galerkin equations. Thus

$$\|u - u_h\|_0^2 \le C\|u - u_h\|_1 \inf_{v \in S_h} \|\phi - v\|_1.$$

To get a bound on the last term we may take $v = \Pi_h \phi$:

$$\inf_{v \in S_h} \|\phi - v\|_1 \le \|\phi - \Pi_h \phi\|_1 \le Ch\|\phi\|_2 \le Ch\|u - u_h\|_0,$$

where we have used the H^1 error estimate for interpolation and the elliptic regularity result, and where we are using the same letter C for various constants. Combining these estimates we have shown that

$$\|u - u_h\|_0 \le Ch\|u - u_h\|_1,$$

that is, we pick up an additional power of h in passing from the H^1 to the L^2 norm of the error. Thus

$$\|u - u_h\|_0 \le Ch^2\|u\|_2$$

for piecewise linear finite elements. Similarly,

$$\|u - u_h\|_0 \le Ch^{r+1}\|u\|_{r+1}$$

for finite elements of degree r.

REMARKS. 1. The bound on the L^2 norm of the error preceded by writing the norm squared in the form $B(u - u_h, \phi)$ for an auxiliary function ϕ defined by a boundary value and then using the Galerkin equation. This approach is known as the Aubin-Nitsche duality argument or sometimes just "Nitsche's trick." The same idea can be used to obtain a variety of different estimates for a variety of different Galerkin methods. 2. The duality argument requires H^2 elliptic regularity, which in turn requires that the polygonal domain be convex. In fact, for a non-convex polygonal domain, it will usually not be true that $\|u-u_h\|_0 = O(h^2)$ even if the solution u is smooth.

4. Difference methods for the heat equation

Consider the heat equation on a spatial domain Ω for a time interval $[0, T]$. The solution is a function $u : \bar{\Omega} \times [0, T]$, such that

$$\frac{\partial u}{\partial t} = c\,\Delta\, u + f \text{ for } x \in \Omega,\ t \in [0, T], \tag{6.12}$$

where the positive constant c depends on the conductivity, specific heat, and density of the material, and f takes into account sources and sinks. To obtain a well-posed problem we need to give boundary conditions such as

$$u = 0 \text{ for } x \in \Gamma,\ t \in [0, T], \tag{6.13}$$

(Neumann or Robin boundary conditions could be used as well), and an initial condition

$$u = u_0 \text{ for } x \in \Omega,\ t = 0. \tag{6.14}$$

Let us suppose that Ω is the unit square in $\mathbb{R}^2$. Then we have a simple discretization of the Laplacian, namely the 5-point Laplacian Δ_h (mapping functions on $\bar{\Omega}_h$ to functions on Ω_h. Thus we seek a function $u_h : \bar{\Omega}_h \times [0, T]$ satisfying

$$\begin{aligned} \frac{\partial u_h}{\partial t} = c\,\Delta_h\, u_h + f \text{ for } x \in \Omega_h,\ t \in [0, T], \\ u_h = 0 \text{ for } x \in \Gamma_h,\ t \in [0, T], \\ u_h = u_0 \text{ for } x \in \Omega_h,\ t = 0. \end{aligned}$$

Since at any time t, u_h is just the finite collection of numbers $u_h(ih, jh, t)$, we may view the above problem as the initial value problem for a system of $(N-1)^2$ ordinary differential equations. The process of reducing the evolutionary PDE to a system of ODEs by using a finite difference approximation of the spatial operator is called *semi-discretization* or the *method of lines*. This is not a full discretization, since we still have to chose a numerical method to solve the ODEs. In principal, any of the methods we studied in Chapter 5 could be used to obtain a full discretization. We shall investigate some of the simplest possibilities.

First we consider the forward Euler method. For simplicity, let us drop down to one space dimension, so $\Omega = (0, 1)$. This is mainly a notational convenience; the analysis in 2D is very similar. Let the spatial mesh size be denoted $h = 1/N$, and the time step $k = T/M$,

and write $U_n^j = u_h(nh, jk)$. Then the fully discrete system is

$$\frac{U_n^{j+1} - U_n^j}{k} = c\frac{U_{n+1}^j - 2U_n^j + U_{n-1}^j}{h^2} + f_n^j, \quad 0 < n < N, \quad j = 0, 1, \dots M-1, \tag{6.15}$$

$$U_0^j = U_N^j = 0, \quad j = 0, 1, \dots, M, \tag{6.16}$$

$$U_n^0 = u_0(nh), \quad 0 < n < N. \tag{6.17}$$

We call this the centered difference/forward difference method for the heat equation. Since the Euler method is *explicit*, we don't have to solve any linear equations to get from $(U_n^j)_n$ to $(U_n^{j+1})_n$. Indeed,

$$U_n^{j+1} = (1-2\lambda)U_n^j + \lambda U_{n+1}^j + \lambda U_{n-1}^j + kf_n^j, \quad 0 < n < N, \quad j = 0, 1, \dots, M-1, \tag{6.18}$$

where $\lambda = ck/h^2$.

Figure 6.9 shows the result of this method applied with $c = 1$, $h = 1/20$, and $k = 1/1200$ for the first plot, $k = 1/600$ for the second. We take 40 time-steps in the first plot, 20 in the second, and so reach $T = 1/30$ in each case. The initial data was taken, rather arbitrarily, to be $u_0(x) = (x - x^2)(x^2 + \sin 2\pi x)$, and the forcing function f was taken to be zero. We see that the first computation gives very reasonable results (and we could have extended it for a much longer time without problem), while the second computation becomes unreasonable after a few time steps. In fact further experimentation shows that this is controlled by the size of the time step in relation to the spatial mesh size. If $\lambda \le 1/2$ (i.e., $k \le h^2/(2c)$), the computation proceeds reasonably, while if $\lambda > 1/2$ the computed solution becomes oscillatory with an exponentially increasing amplitude.

To analyze the situation, consider first the truncation error

$$\tau_n^j = \frac{u_n^{j+1} - u_n^j}{k} - c\frac{u_{n+1}^j - 2u_n^j + u_{n-1}^j}{h^2} - f_n^j,$$

where $u_n^j = u(nh, jk)$. By Taylor's theorem

$$\tau_n^j = \frac{k}{2}\frac{\partial^2 u}{\partial t^2} - c\frac{h^2}{12}\frac{\partial^4 u}{\partial x^4},$$

where the derivatives are evaluated at appropriate points. Now let $e_n^j = U_n^j - u_n^j$. Then

$$e_n^{j+1} = (1-2\lambda)e_n^j + \lambda e_{n+1}^j + \lambda e_{n-1}^j - k\tau_n^j. \tag{6.19}$$

Now suppose that $\lambda \le 1/2$. Then

$$|e_n^{j+1}| = (1-2\lambda)|e_n^j| + \lambda|e_{n+1}^j| + \lambda|e_{n-1}^j| + k|\tau_n^j|.$$

If we let $E^j = \max_n |e_n^j|$ denote the maximum norm of the error at the jth time step, we get

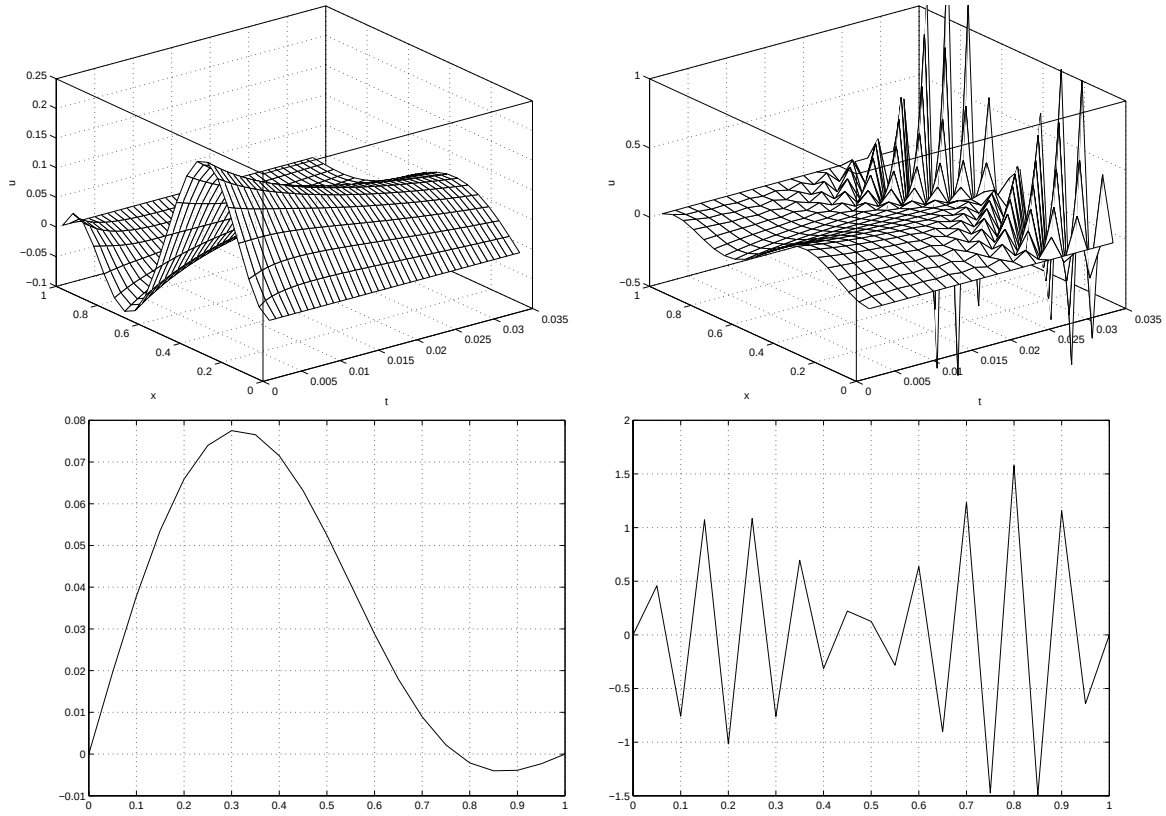
$$E^{j+1} \le E^j + k\tau$$

where $\tau = \max_{n,j}|\tau_n^j|$. It follows that

$$E^j \le E^0 + jk\tau = jk\tau \tag{6.20}$$

for all j. Since we only integrate up to a fixed time T, $jk \le T$, we have $\max_{n,j}|e_n^j| \le T\tau \le C(k + h^2)$. Since $k \le h^2/(2c)$, the error is $O(h^2)$. In particular, we have established convergence $h, k \to 0$ as long as the condition $k \le h^2/(2c)$ is maintained (e.g., we can let

FIGURE 6.9. The heat equation discretized using central differences and forward Euler, with two different time steps ($k = h^2/3$ on the left, $k = 2h^2/3$ on the right). The top plots show the results at all time steps; the bottom figures only at the final time.



$h \to 0$ and set $k = \rho h^2$ for any $\rho \leq 1/(2c)$. Thus we have established *conditional convergence* of the centered difference/forward difference method.

Notice that, just as we derived (6.20) from (6.19) under the assumption that $k \leq h^2/(2c)$, under the same assumption we can deduce from the discrete equations (6.18) that

$$\max_{\substack{0 \leq nh \leq 1 \\ 0 \leq jk \leq T}} |U(nh, jk)| \leq \max_{0 \leq nh} |u_0(nh)|.$$

This is a stability result: it says that the linear map which takes the initial data to the fully discrete solution is bounded if the data is measured in the discrete max norm over the interval and the solution in the discrete space-time max norm, and the bound (which is 1) does not blow-up as the mesh size is decreased. In view of our past experience, we should not be suprised that a method which is stable and consistent (in the sense that the truncation error tends to zero with mesh size), is convergent.

Another very useful way to analyze stability and convergence is to use Fourier analysis, as we did for 5-point Laplacian earlier in this chapter. To get the idea, first recall how the continuous problem (6.12)–(6.14) may be solved in terms of Fourier series. We expand the solution at any given time a Fourier sine series (we only use sines, and not cosines, in view

of the Dirichlet boundary conditions):

$$u(x,t) = \sum_{m=1}^{\infty} a_m(t) \sin m\pi x, \quad x \in I.$$

Assume, for simplicity, that the forcing function f vanishes, and substitute the expansion into the differential equation to get

$$\sum_{m=1}^{\infty} a_m'(t) \sin m\pi x = -c \sum_{m=1}^{\infty} a_m(t) m^2\pi^2 \sin m\pi x,$$

and, using the orthogonality of the sine functions, we conclude that

$$a_m'(t) = -cm^2\pi^2 a_m(t) \text{ so } a_m(t) = a_m(0) e^{-cm^2\pi^2 t}.$$

The values $a_m(0)$ can be determined from the Fourier sine expansion of the initial data:

$$u_0(x) = \sum_{m=1}^{\infty} a_m(0) \sin m\pi x, \quad x \in I.$$

Thus we see that all the modes that are present in the initial data are damped exponentially with increasing time, with the higher frequency modes being damped most quickly. For this reason heat evolution is a smoothing process.

Now let us do the same thing for the semi-discrete problem. Recall the notations we used to introduce the discrete Fourier sine bases. With $I_h = \{\, nh \,|\, 0 \le n \le N \,\}$ and $\bar{I}_h = I_h \cup \{0,1\}$, we let $L(I_h)$ denote the set of real-valued functions on I_h which may be viewed as functions on $\bar{I}_h$ by extension by zero. On $L(I_h)$ we use the inner product $\langle u, v\rangle_h = h \sum_{k=1}^{N-1} u(kh)v(kh)$, and the orthogonal basis $\{\phi_m \,|\, m = 1, \dots, N-1\}$ given by $\phi_m(x) = \sin \pi m x$, $x \in \bar{I}_h$. These are eigenvalues of the 1D discrete Laplacian D_h^2:

$$D_h^2 \phi_m = -\lambda_m \phi_m, \quad \lambda_m = \frac{4}{h^2} \sin^2 \frac{\pi m h}{2}.$$

The eigenvalues satisfy

$$8 \le \lambda_1 < \lambda_2 < \cdots < \lambda_{N-1} < \frac{4}{h^2}.$$

Proceeding as for the continuous solution, at any time t we write the semi-discrete solution $u_h(t, \cdot) = \sum_{m=1}^{N-1} a_m^h(t)\phi_m$. Then

$$\frac{\partial u_h}{\partial t} = \sum_{m=1}^{N-1} \frac{da_m^h}{dt}\, \phi_m, \quad D_h^2 u_h = -\sum_{m=1}^{N-1} a_m^h \lambda_m \phi_m.$$

Thus the semi-discrete equations give

$$\frac{da_m^h}{dt} = -c\lambda_m a_m^h \text{ so } a_m^h(t) = a_m^h(0) e^{-c\lambda_m t},$$

where the numbers $a_m^h(0)$ are the coefficients in the discrete Fourier sine transform of the initial data:

$$u_0(x) = \sum_{m=1}^{N-1} a_m^h(0) \sin \pi m x, \quad x \in \bar{I}_h.$$

Thus the solution of the semi-discrete system may be written as

$$u_h(x,t) = \sum_{m=1}^{N-1} a_m^h(0) e^{-c\lambda_m t} \sin \pi m x, \quad x \in \bar{I}_h.$$

Again all the modes that are present in the discretized initial data are damped exponentially with increasing time.

Finally, consider the fully discrete centered difference/forward difference method. Let U^j be the solution at time $t = jk$ (i.e., $U^j(nh) = U_n^j$). Write

$$U^j(x) = \sum_{m=1}^{N-1} A_m^j \phi_m(x).$$

The difference equations then give

$$\sum_{m=1}^{N-1} A_m^{j+1}\phi_m = \sum_{m=1}^{N-1} A_m^j \phi_m - ck \sum_{m=1}^{N-1} A_m^j \lambda_m \phi_m \text{ i.e., } A_m^{j+1} = (1 - ck\lambda_m) A_m^j.$$

It follows that

$$A_m^j = (1 - ck\lambda_m)^j a_m^h(0), \quad U^j = \sum_{m=1}^{N-1} (1 - ck\lambda_m)^j a_m^h(0) \phi_m.$$

Now $\lambda_m \le 4/h^2$ for all m, so $ck\lambda_m \le 4ck/h^2$, and so, *if we assume that* $ck/h^2 \le 1/2$, we get $ck\lambda_m \le 2$ and $|1 - ck\lambda_m| \le 1$ for all m. Thus in this case (or at least if strict inequality holds), we qualitative behavior of the continuous solution, that all modes of the initial data are damped, is also present in the discrete case as well. On the other hand, if $ck/h^2 > 1/2$, then for h sufficiently small we will have $|1 - ck\lambda_m| > 1$ for larger values of m, and this means that high frequency components of the initial data will increase exponentially with increasing time step. This explains the behavior we saw earlier.

The same ideas can be used to establish a rigorous stability result. Suppose that U_n^j satsifies the fully discrete centered difference/forward difference equations (6.15)–(6.17). Writing U^j, f^j for the restrictions of U and f to $t = jk$, we have

$$U^{j+1} = (I + ckD_h^2)U^j + kf^j, \quad j = 0, 1, \ldots, M-1. \tag{6.21}$$

Now $I + ckD_h^2$ is a symmetric operator on $L(I_h)$, so its operator norm with respect to the discrete L^2 norm on $L(I_h)$ is simply the magnitude of its largest eigenvalue. Now the eigenvectors of the operator $I + ckD_h^2$ are again the ϕ_m, with corresponding eigenvalues $1 - ck\lambda_m$. If $ck/h^2 \le 1/2$, then $\max_m |1 - ck\lambda_m| \le 1$, and hence $\|(I + ckD_h^2)v\|_h \le \|v\|_h$ for any $v \in L(I_h)$. Thus from (6.21), we get

$$\|U^{j+1}\|_h \le \|U^j\|_h + k\|f^j\|, \quad j = 0, 1, \ldots, M-1.$$

Bounding $\|f^j\|$ by $\max_{0\le j\le M-1}\|f^j\|$ and iterating this result (recall that $Mk = T$), we get

$$\max_{0\le j\le M} \|U^j\|_h \le \|U^0\|_h + T \max_{0\le j\le M-1} \|f^j\|_h. \tag{6.22}$$

Thus, under the hypothesis $ck/h^2 \le 1/2$ we have show that the fully discrete solution is bounded (in the norm displayed on the left-hand side of (6.22) by an appropriate norm on the data U^0 and f, with constants that don't depend on h and k. Because of the condition

$ck/h^2 \le 1/2$, which we know is not only sufficient but necessary for stability, the centered difference/forward difference method is called *conditionally stable*.

Once we have stability, we obtain a convergence result in the same way as we did earlier for the 5-point Laplacian. For the error $e_n^j = U_n^j - u_n^j$ we have

$$\frac{e_n^{j+1} - e_n^j}{k} = c\frac{e_{n+1}^j - 2e_n^j + e_{n-1}^j}{h^2} - \tau_n^j, \quad 0 < n < N, \quad j = 0, 1, \dots M-1,$$
$$e_0^j = e_N^j = 0, \quad j = 0, 1, \dots, M,$$
$$e_n^0 = 0, \quad 0 < n < N,$$

where τ_n^j is the local truncation error. It follows from the stability result that

$$\max_{0\le j\le M} \|e^j\|_h \le T \max_{0\le j\le M-1} \|\tau^j\|_h$$

Of course

$$\max_{0\le j\le M-1} \|\tau^j\|_h \le \max_{\substack{0\le j\le M-1\\ 0\le n\le N}} |\tau_n^j| = O(k + h^2),$$

so we have obtained $O(k + h^2)$ convergence of the method. Since we required $k \le h^2/(2c)$ to obtain the result, we may write the error simply as $O(h^2)$.

4.1. The centered difference/backward difference method. We consider now a different time discretization for the heat equation (6.12), namely we consider the backward Euler method rather than the forward Euler method. This leads to the centered difference/backward difference method:

$$\frac{U_n^{j+1} - U_n^j}{k} = c\frac{U_{n+1}^{j+1} - 2U_n^{j+1} + U_{n-1}^{j+1}}{h^2} + f_n^{j+1}, \quad 0 < n < N, \quad j = 0, 1, \dots M-1,$$
$$U_0^j = U_N^j = 0, \quad j = 0, 1, \dots, M,$$
$$U_n^0 = u_0(nh), \quad 0 < n < N.$$

Thus U^{j+1} must be determined by solving the tridiagonal system

$$-\lambda U_{n+1}^{j+1} + (1 + 2\lambda)U_n^{j+1} - \lambda U_{n-1}^{j+1} = U_n^j + kf_n^{j+1}, \quad 0 < j < N, \quad .U_0^{j+1} = U_N^{j+1} = 0.$$

The matrix is strictly diagonally dominant, so there exists a unique and no pivoting is needed if Gaussian elimination is used. The amount of work to solve the system is thus $O(N)$. It is easy to see that the truncation error for the scheme is again $O(k + h^2)$. To determine the stability of this method, we will use Fourier analysis. In operator form the method is

$$\frac{U^{j+1} - U^j}{k} = cD_h^2 U^{j+1} + kf^{j+1},$$

or

$$U^{j+1} = (I - ckD_h^2)^{-1}(U^j + kf^{j+1}).$$

The operator $I - ckD_h^2$ has eigenvalues $1 + ck\lambda_m$ which are all greater than 1, so the norm of $(I - ckD_h^2)^{-1}$ is less than 1. Thus we obtain

$$\|U^{j+1}\|_h \le \|U^j\|_h + k\|f^{j+1}\|, \quad j = 0, 1, \dots, M-1.$$

and hence the stability result

$$\max_{0\le j\le M}\|U^j\|_h \le \|U^0\|_h + T\max_{1\le j\le M}\|f^j\|_h.$$

We did not have to make any assumption on the relation between k and h to obtain this result: the centered difference/forward difference method is *unconditionally stable*. Combining this with the bounds on the truncation error, we find obtain an error estimate

$$\max_{0\le j\le M}\|e^j\|_h = O(k+h^2)$$

for the method.

4.2. The Crank-Nicolson method. If we use the trapezoidal method to discretize in time, we get the Crank-Nicolson method:

$$\frac{U_n^{j+1}-U_n^j}{k} = \frac{c}{2}(D_h^2U^j + D_h^2U^{j+1}) + \frac{1}{2}(f^j+f^{j+1}).$$

Using a Taylor expansion about the point $(nh,(j+1/2)k)$ it is straightforward to show that the truncation error is $O(k^2+h^2)$, so the Crank-Nicolson method is second order in both space and time. In operator terms the method is

$$U^{j+1} = (I-\frac{1}{2}ckD_h^2)^{-1}(I+\frac{1}{2}ckD_h^2)U^j + \frac{k}{2}(I-\frac{1}{2}ckD_h^2)^{-1}(f^j+f^{j+1}).$$

As before the operator norm of $(I-(1/2)ckD_h^2)^{-1}$ is bounded by 1. The eigenvalues of $(I-\frac{1}{2}ckD_h^2)^{-1}(I+\frac{1}{2}ckD_h^2)$ are $(1-ck\lambda_m/2)/(1+ck\lambda_m/2)$ which are all less than 1 since $ck\lambda_m/2>0$. Thus the operator norm of this composition is less than 1 and we can get unconditional stability. The Crank-Nicolson method then converges with $O(k^2+h^2)$. We may choose k proportional to h, rather than to h^2 as in the previous methods, and obtain an error of $O(h^2)$.

Include here a table showing the stencil, implicit/explicit, order, and stability of each of the three methods considered.

5. Difference methods for hyperbolic equations

As the very simplest hyperbolic equation we consider the advection equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0,$$

where c is a constant. We consider an initial value problem, so that the function $u=u(x,t)$ is given when $t=0$ and is to be found for $t>0$. The spatial domain may be an interval, in which case we will need to impose boundary conditions to obtain a well-posed initial value-boundary value problem, or the entire real line (the pure initial-value problem).

Let u be a solution to the pure initial-value problem for the advection equation. Fix some x_0 and let $U(t)=u(x_0+ct,t)$. Then $dU/dt\equiv 0$ so $U(t)=U(0)$ for all t, or $u(x_0+ct,t)=u_0(x_0)$, where u_0 is the initial data. Substituting $x_0=x-ct$ we get

$$u(x,t)=u_0(x-ct), \quad x\in\mathbb{R}, t\ge 0.$$

Thus for the pure initial value problem we have given the solution to the advection equation analytically. The initial data is simply transferred to the right with speed c (for $c>0$). The

solution is constant on the lines $\{(x_0+ct,t\}$ of slope $1/c$, which are called the *characteristics* of the equation.

Now suppose that the spatial domain is an interval, $(0,1)$ say. From the above considerations the solution is determined at the boundary point $x=1$ for $0\le t\le 1/c$ by the initial data: $u(1,t)=u_0(1-ct)$. If the u is given on the left boundary, $u(0,t)=g(t)$, say, then u is determined everywhere:

$$u(x,t)=\begin{cases} u_0(x-ct), & x\ge ct,\\ g(t-x/c), & x<ct.\end{cases}$$

We have thus given the exact solution to the initial value-boundary value problem

$$\begin{gathered}\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}=0,\quad 0<x<1,\quad t>0,\\ u(x,0)=u_0(x),\quad 0\le x\le 1,\\ u(0,t)=g(t),\quad t>0.\end{gathered}$$

The only boundary condition needed is on the left, or *inflow* boundary. It is not necessary or permissible to give a boundary condition on the right or *outflow* boundary.

Although the advection equation is so simple that we can solve it analytically, it admits many generalizations that lead to truly interesting equations, and so is valuable to study as a model problem. Some generalizations are *variable coefficients* $c=c(x,t)$, which result in curves rather than lines as characteristics, *lower order terms*,

$$\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}+du=f,$$

hyperbolic systems, where $u(x,t)$ takes values in $\mathbb{R}^n$ and c is an $n\times n$ matrix, *hyperbolic problems in two or more space dimensions* like

$$\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}+d\frac{\partial u}{\partial y}=0,$$

and *nonlinear hyperbolic equations* such as (inviscid) Burger's equation

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=0,$$

in which the coefficients depend on the solution. *Hyperbolic problems of higher order* (i.e., involving second or higher partial derivatives.)

Suppose we have a system of the form $\partial\boldsymbol{u}/\partial t+C\partial\boldsymbol{u}/\partial x=0$ where $\boldsymbol{u}(x,t)\in\mathbb{R}^n$ and C is an $n\times n$ matrix with real eigenvalues (e.g., a symmetric matrix). Say $C=S^{-1}DS$, with S invertible and D a diagonal matrix. If we change dependent variables by $\boldsymbol{v}=S^{-1}\boldsymbol{u}$, then we get $\partial\boldsymbol{v}/\partial t+D\partial\boldsymbol{v}/\partial x=0$, i.e., $\partial v_i/\partial t+d_i\partial v_i/\partial x=0$, $i=1,\dots,n$. Thus the hyperbolic system decouples into n advection equations whose speeds are the eigenvalues of the original coefficient matrix.

As an example, consider the wave equation $\partial^2 w/\partial t^2-\partial^2 w/\partial x^2=0$. Let $u_1=\partial w/\partial t$, $u_2=\partial w/\partial x$. Then

$$\frac{\partial}{\partial t}\begin{pmatrix}u_1\\u_2\end{pmatrix}+\begin{pmatrix}0&-1\\-1&0\end{pmatrix}\frac{\partial}{\partial x}\begin{pmatrix}u_1\\u_2\end{pmatrix}=0.$$

Thus

$$C = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} = S^{-1}DS, \quad S = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad D = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Thus we can find the solution from $u_1 + u_2$, which is just a wave moving to the left, and $u_1 - u_2$, a wave moving to the right.

5.1. Difference methods for the advection equation. In simple cases, hyperbolic problems can be solved by determining the characteristics (which often involves solving ODEs), and then determining the solution along the characteristics (again often an ODE problem). This *method of characteristics* is a viable numerical methods in some cases. However it range of applicability is too limited for many important hyperbolic problems which arise, for example it cannot be easily applied to hyperbolic systems in several space dimensions. In this section we will study instead difference methods for hyperbolic problems. For simplicity, we will investigate them only for the simple model problem of the advection equation.

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0.$$

We suppose that $c > 0$, so the solution is a wave travelling to the right.

Consider first the most obvious difference methods for the advection equation. We use forward differences to discretize the time derivative, and three different possibilities for the space derivative: forward differences, backward differences, and centered differences. Thus the three methods are:

$$\text{forward-forward} \qquad \frac{U_n^{j+1} - U_n^j}{k} + c\frac{U_{n+1}^j - U_n^j}{h} = 0,$$

$$\text{forward-backward} \qquad \frac{U_n^{j+1} - U_n^j}{k} + c\frac{U_n^j - U_{n-1}^j}{h} = 0,$$

$$\text{forward-centered} \qquad \frac{U_n^{j+1} - U_n^j}{k} + c\frac{U_{n+1}^j - U_{n-1}^j}{2h} = 0.$$

If we set $\lambda = ck/h$, these can be written

$$\text{forward-forward} \qquad U_n^{j+1} = -\lambda U_{n+1}^j + (1+\lambda)U_n^j,$$

$$\text{forward-backward} \qquad U_n^{j+1} = (1-\lambda)U_n^j + \lambda U_{n-1}^j,$$

$$\text{forward-centered} \qquad U_n^{j+1} = -\frac{\lambda}{2}U_{n+1}^j + U_n^j + \frac{\lambda}{2}U_{n-1}^j.$$

The truncation error is clearly $O(k+h)$ for the first two methods and $O(k+h^2)$ for the third.

Include numerical results here.

Numerical experiments suggest that if λ is sufficiently small, the forward-backward method is stable and convergent, but not for larger λ, i.e., that the forward-backward method is conditionally stable. On the other hand the forward-forward method and the forward-centered method appear to be unstable for any positive value of λ.

In fact, the non-convergence of the forward-forward method is easy to establish, even without considering stability. The solution to the advection equation at a point x at time t depends only on the initial data at the point $x - ct$ at time 0—in fact it equals the initial

data there. (Other hyperbolic problems, such as the wave equation, the solution at a point x at time t may depend on the initial data on an interval, not just a point, but the existence of a *bounded domain of dependence* is typical feature of hyperbolic equations. Now consider the forward-forward difference method. It is easy to see that the value at a mesh point $x = nh$ at a time $t = jk$ depends only on the inital data at the mesh points in the interval $[x, x + ct/\lambda]$. Now this interval doesn't contain the point $x - ct$, for any positive value of λ. Hence, if we choose initial data which is equal to 1 at $x - ct$, but identically zero on $[x, \infty)$, the true solution of the advection equation will satisfy $u(x, t) = 1$, but the numerical solution will be zero, and will remain zero as $h, k \to 0$. In short, the numerical domain of dependence fails to include the true domain of dependence, which is necessary for a convergent method. This necessary condition, which fails for the forward-forward difference method, is called the *Courant-Friedrichs-Levy condition*, or CFL condition.

If we apply the CFL condition to the forward-backward method, we see that the method can only be conditionally convergent. Indeed, the numerical domain of dependence is the interval $[x-ct/\lambda, x]$, and this contains the true domain of dependence if and only if $0 < \lambda \leq 1$. Hence $0 < \lambda \leq 1$ is a necessary condition for convergence. In fact, we shall see, that the method is stable and convergent if and only if this condition is met. (Note: we are assuming $c > 0$, so $\lambda > 0$. If $c < 0$, then the forward-backward method never converges, but the forward-forward method converges for $0 > \lambda \geq -1$.)

One should not think, however, that the CFL condition is generally sufficient for convergence. For example, the CFL condition for the forward-centered scheme is $|\lambda| \leq 1$, but the method turns out to be unconditionally unstable.

Conditional stability in the max norm is easy to establish for the forward-backward method. If we consider the method, including a forcing term in the equation, we get

$$U_n^{j+1} = (1 - \lambda)U_n^j + \lambda U_{n-1}^j + k f_n^j.$$

If we assume that $0 \leq \lambda \leq 1$, we easily deduce that

$$\|U^{j+1}\|_\infty \leq \|U^j\|_\infty + k\|f^j\|_\infty,$$

and so

$$\max_{0 \leq j \leq M} \|U^j\|_\infty \leq \|U^0\|_\infty + k \sum_{j=0}^{M-1} \|f^j\|_\infty,$$

i.e., stability with respect to the both the initial data and the forcing function. From this, the usual argument shows that the max norm of the error is bounded by $k \sum_{j=0}^{M-1} \|\tau^j\|_\infty$ where τ is the truncation error, and so we get $O(k + h)$ convergence.

If $\lambda > 1$, however, the magnitudes of the coefficients $1 - \lambda$ and λ sum to more than 1 and this argument fails (as, in fact, does max norm stability). Similarly the sum of the magnitudes of the coefficients for the forward-forward scheme, $1 + \lambda$ and $-\lambda$, exceeds 1 for any positive λ, and the same holds for the coefficients of the forward-centered scheme.

It is important to point out that the forward-forward method is unconditionally unstable for advection to the right, and the forward-backward method is unconditionally unstable for advection to the left. Since the wave equation involves the superposition of both of these, neither scheme is stable for the wave equation.

The *Lax–Friedrichs* method is a variant of the forward-centered method which maintains $O(k + h^2)$ accuracy and which is conditionally stable for advection to the right or left. The

scheme is

$$\frac{U_n^{j+1} - (U_{n+1}^j + U_{n-1}^j)/2}{k} + c\frac{U_{n+1}^j - U_{n-1}^j}{2h} = 0,$$

or

$$U_n^{j+1} = (1/2 - \lambda/2)U_{n+1}^j + (1/2 + \lambda/2)U_{n-1}^j.$$

It isn't hard to see that the method is stable if $|\lambda| \le 1$. The truncation error is $O(k + h^2 + h^2/k)$, which is $O(h)$ if λ is held-fixed.

Note that the Lax–Friedrichs method can be rewritten

$$\frac{U_n^{j+1} - U_n^j}{k} + c\frac{U_{n+1}^j - U_{n-1}^j}{2h} - \frac{ch}{2\lambda}\frac{U_{n+1}^j - 2U_n^j + U_{n-1}^j}{h^2} = 0.$$

Thus the method suggests discretization of the equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} - \frac{ch}{2\lambda}\frac{\partial^2 u}{\partial x^2} = 0.$$

This is an advection–diffusion equation with an $O(h)$ coefficient multiplying the diffusion term. Thus the Lax–Friedrichs method can be viewed as a variant of the forward-centered difference method in which a small amount of *artificial diffusion* has been added to stabilize the numerical method.

5.2. Fourier analysis. We can also use discrete Fourier analysis to study the stability of these methods. This is known as *von Neumann stability analysis*. For simplicity we consider a 1-periodic problem rather than a boundary value problem:

$$\begin{aligned} \frac{\partial u}{\partial t}(x,t) + c\frac{\partial u}{\partial x}(x,t) = 0, \quad x \in \mathbb{R}, t > 0, \\ u(x+1,t) = u(x,t), \quad x \in \mathbb{R}, t > 0, \\ u(x,0) = u_0(x), x \in \mathbb{R}. \end{aligned}$$

We take as spatial mesh points nh, $h = 1/N$, $n \in \mathbb{Z}$, and seek a solution $U_n^j \approx u(nh, jk)$ which satisfies the periodicity condition $U_{n+N}^j = U_n^j$, $n \in \mathbb{Z}$, for each time level j. We will also simplify by considering complex-valued rather than real-valued solutions. Let L_h^{per} denote the space of complex-valued 1-periodic functions on $\mathbb{Z}h$ (functions $U_j = U(jh)$ satisfying $U_{j+N} = U_j$). As a basis for this space we can choose the functions ψ_m, $m = 0, 1, \ldots, N-1$, defined by $\psi_m(x) = \exp(2\pi imx)$, $x \in \mathbb{Z}h$. The ψ_m are orthonormal with respect to the inner product $\langle \phi, \psi \rangle_h = h\sum_{n=0}^{N-1} \phi(nh)\overline{\psi(nh)}$.

Now let τ_h^+ denote the simple forward shift operator on L_h^{per}: $\tau_h^+ U(x) = U(x+h)$ or $\tau_h^+ U_j = U_{j+1}$. Then $\tau_h^+ \psi_m = \exp(2\pi imh)\psi_m$, i.e., ψ_m is an eigenvector of the forward shift operator with eigenvalue $\exp(2\pi imh)$. Similarly ψ_m is an eigenvector of the backward shift operator, and consequently also of the forward difference operator $D_h^+ = (\tau_h^+ - I)/h$, the backward difference operator $D_h^- = (I - \tau_h^-)/h$, and the centered difference operator $D_h = (\tau_h^+ - \tau_h^-)/2h$. For example,

$$D_h^- \psi_m = \frac{1 - e^{-2\pi imh}}{h}\psi_m.$$

Now consider the forward-backward difference equations, which we may write as

$$U^{j+1} = (1 - \lambda)U^j + \lambda\tau_h^- U^j$$

(as usual a forcing term can be added without difficulty). Defining $S : L_h^{\text{per}} \to L_h^{\text{per}}$ by $SV = (1-\lambda)V + \lambda\tau_h^- V$, the method is

$$U^{j+1} = SU^j.$$

Now $S\phi_m = (1 - \lambda + \lambda e^{-2\pi imh})\phi_m$, so

$$\|S\|_{\mathcal{L}(L_h^{\text{per}}, L_h^{\text{per}})} = \max_{m=0,1,\ldots,N-1} |1 - \lambda + \lambda e^{-2\pi imh}|.$$

Since the eigenvalues $1 - \lambda + \lambda e^{-2\pi imh}$ lie on a circle of radius $|\lambda|$ around $1 - \lambda$, we get $\|S\| \le 1$ if and only if $0 \le \lambda \le 1$.

We now give several further examples of von Neumann analysis:

The forward-centered difference method for the advection equation: The eigenvalues are

$$-\frac{\lambda}{2}e^{2\pi imh} + 1 + \frac{\lambda}{2}e^{-2\pi imh} = 1 - i\lambda \sin 2\pi mh.$$

Every eigenvalue has magnitude greater than 1, showing that the method is indeed unconditionally unstable.

Lax-Friedrichs method for the advection equation: The eigenvalues are

$$(\frac{1}{2} - \frac{\lambda}{2})e^{2\pi imh} + (\frac{1}{2} + \frac{\lambda}{2})e^{-2\pi imh} = \cos 2\pi mh - \lambda i \sin 2\pi mh.$$

The method is stable if $|\lambda| \le 1$.

Backward-centered difference method for the advection equation: The method is

$$\frac{U_n^{j+1} - U_n^j}{k} + c\frac{U_{n+1}^{j+1} - U_{n-1}^{j+1}}{2h} = 0,$$

so

$$SU^{j+1} - U^j = 0 \text{ or } U^{j+1} = S^{-1}U^j,$$

where S has as eigenvalues

$$1 + \lambda\frac{e^{2\pi imh} - e^{-2\pi imh}}{2} = 1 - \lambda i \sin 2\pi mh.$$

Thus all the eigenvalues of S have modulus greater than 1, and so the norm of S^{-1} is less than 1. The method is unconditionally stable.

Von Neumann stability analysis applies to a wide variety of evolution equations and difference methods, in fact to virtually any equation with constant coefficients and any difference method on a uniform mesh. In the case of parabolic problems it is very close to the Fourier analysis we considered earlier, except that it assumes periodic rather than Dirichlet boundary conditions. As a final example, we analyze the centered in space, forward in time difference method for the heat equation, this time with periodic boundary conditions. The method is

$$U_n^{j+1} = U_n^j + \lambda(U_{n+1}^j - 2U_n^j + U_{n-1}^j),$$

$\lambda = ck/h^2$. The von Neumann eigenvalues are

$$1 + \lambda(e^{2\pi imh} + e^{-2\pi imh} - 2) = 1 + 2\lambda(\cos 2\pi mh - 1).$$

Since $0 \ge \cos 2\pi mh - 1 \ge -2$ (with both equalities possible), the eigenvalues range from 1 to $1 - 4\lambda$, and to have stability, we need $1 - 4\lambda \ge -1$ or $\lambda \le 1/2$. We have thus recovered the stability condition.

6. Hyperbolic conservation laws

This section, when written will give a very brief introduction to (scalar, 1D, nonlinear) hyperbolic conservation laws ending with a brief presentation of Godunov's method and Glimm's method.

Consider some quantity spread through space, with a density $u(x,t)$ that varies in space and time. We consider here only the case of one space dimension, so $x \in \mathbb{R}$. A good example to have in mind is a long pipe filled with gas, with $u(x,t)$ representing the density of gas a distance x along the pipe at time t. Then the total quantity of gas in some interval $[a,b]$ at some time t is $\int_a^b u(x)\,dx$.

The *flux* F at the point x and time t is, by definition, the rate per unit time at which the quantity flows past the point x. Thus, if $[a,b]$ is an interval, the rate at which the quantity flows into the interval from the left is $F(a,t)$ (this is negative if the quantity is flowing out of the interval at a) and the rate at which it flows in from the right is $-F(b,t)$. The total amount of material that flows in over some time interval $[t_1,t_2]$ is then $\int_{t_1}^{t_2} F(a,t)\,dt - \int_{t_1}^{t_2} F(b,t)\,dt$.

Now suppose that the quantity is *conserved*. That is, the difference between the quantity in some interval at some time t_1 and the quantity in the same interval at a later time t_2 is entirely accounted for by the amount flowing in and out the end points of the interval. We may express this by the equation

$$\int_a^b u(x,t_2)\,dx - \int_a^b u(x,t_1)\,dx = \int_{t_1}^{t_2} F(a,t)\,dt - \int_{t_1}^{t_2} F(b,t)\,dt. \tag{6.23}$$

This equation is to hold for all $a < b$ and all $t_1 < t_2$. This is an integral form of a conservation law.

Now suppose that the both u and F are smooth functions. Then we may use the fundamental theorem of calculus to write (6.23) as

$$\int_{t_1}^{t_2}\int_a^b \left[\frac{\partial u}{\partial t}(x,t) + \frac{\partial F}{\partial x}(x,t)\right] dx dt.$$

This will hold for all choices of intervals if and only if

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0.$$

This is the differential form of the conservation law.

STOPPED HERE

Consider the nonlinear first order hyperbolic equation

$$\frac{\partial u}{\partial t} + c(u)\frac{\partial u}{\partial x} = 0, \quad x \in \mathbb{R}, \quad t > 0, \tag{6.24}$$

which describes the transport of some quantity with a velocity that depends on the amount of the quantity present. We suppose that the inital data $u(x,0) = u_0(x)$ is given. We may solve this problem by the method of characteristics. Assume that $u(x,t)$ is a smooth solution, and let x_0 be any point. A characteristic curve $X(t)$ is defined by the ODE initial value problem

$$\frac{dX}{dt}(t) = c\big(u(X(t),t)\big), \quad X(0) = x_0. \tag{6.25}$$

Then u is constant on the characteristic, because

$$\frac{d}{dt}u\big(X(t),t\big) = \left(\frac{\partial u}{\partial t} + c(u)\frac{\partial u}{\partial x}\right)\big(X(t),t\big) = 0.$$

Thus $u\big(X(t),t\big) = u\big(X(0),0\big) = u_0(x_0)$ for all t. Therefore the ODE (6.25) is trivial. It just says that dX/dt equals the constant value $c\big(u_0(x_0)\big)$, so $X(t)$ is the linear function $X(t) = x_0 + c\big(u_0(x_0)\big)t$.

Thus *the characteristics for the nonlinear advection equation* (6.24) *are straight lines*. The solution u is constant on each characteristic line, with the slope of the line equal to $1/c(u)$.

The simplest nonlinear case is given by $c(u) = u$ so the differential equation is

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0. \tag{6.26}$$

This equation is known as the inviscid Burger's equation. It describes a quantity advecting with a speed proportional to its density.

Suppose first that the initial data u_0 is continuous and monotone increasing. In this case the characteristics sweep out the whole upper half-plane as shown on the left of Figure 6.10 and thus the solution u is uniquely determined everywhere. On the other hand, if the initial data is decreasing, then the characteristics will cross as on the right of Figure 6.10. This constitutes a proof that the initial value problem for (6.26) does not admit a smooth solution for all $t > 0$ when the initial data is increasing. (If a solution did exist, we would obtain a contradiction by following two crossing characteristics to obtain two different values for the solution at the same point.)

This section needs to be finished. We need a bunch of diagrams including the one referred to in the last paragraph.

FIGURE 6.10. ...

EXERCISES

(1) Consider a nine-point difference approximation to the Laplacian of the form

$$\Delta_h^* v_{m,n} = \frac{1}{h^2}[\alpha(v_{m-2,n} + v_{m+2,n} + v_{m,n-2} + v_{m,n+2}) + \beta(v_{m-1,n} + v_{m+1,n} + v_{m,n-1} + v_{m,n+1}) + \gamma v_{m,n}] = f_{m,n}.$$

Show how to choose the constants α, β, and γ so that the scheme $\Delta_h^* v = f$ is consistent to fourth order with the equation $\Delta u = f$.

(2) Next consider a nine-point approximation of the form

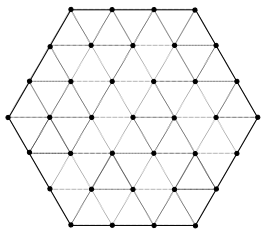
$$\Delta_h^* v_{m,n} = \frac{1}{h^2}[\alpha(v_{m-1,n-1} + v_{m-1,n+1} + v_{m+1,n-1} + v_{m+1,n+1}) + \beta(v_{m-1,n} + v_{m+1,n} + v_{m,n-1} + v_{m,n+1}) + \gamma v_{m,n}].$$

Show that there is no choice of constants α, β, and γ so that the scheme is fourth order accurate. However show that the coefficients can be chosen to give a fourth order scheme of the form $\Delta_h^* v = Rf$ where

$$Rf_{m,n} = (f_{m-1,n} + f_{m+1,n} + f_{m,n-1} + f_{m,n+1} + 8f_{m,n})/12.$$

(3) Show that with the same choice of coefficients as in the last problem the scheme $\Delta^* v = 0$ is a *sixth* order accurate approximation of the homogeneous equation $\Delta u = 0$.

(4) Consider the solution of the Poisson equation with zero Dirichlet boundary conditions on a hexagon. Develop a seven-point Laplacian using mesh points lying at the vertices of a grid of equilateral triangles, as shown below. Prove convergence of the method and exhibit the rate of convergence.



(5) Find a weak formulation for the one-dimensional boundary value problem

$$-(au')' + bu' + cu = f \text{ in } (0,1), \quad u(0) = \alpha, \quad u(1) + 2u'(1) = \beta,$$

where $a, b, c, f : [0,1] \to \mathbb{R}$ are given.

(6) Consider the solution to one-dimensional Poission equation

$$u'' = f \text{ in } (0,1), \quad u(0) = u(1) = 0,$$

using piecewise linear finite element on an arbitrary partition of the interval. Prove that the finite element solution u_h coincides with the interpolant $\Pi_h u$. Note: this result is special to the Poisson equation in one-dimension. The analogue in two dimensions is certainly false.

(7) a) Let $I = (a, b)$ be an interval. Prove the one-dimensional Poincaré inequality

$$\|u\|_{L^2(I)} \le c\|u'\|_{L^2(I)}$$

for $u \in H^1(I)$ such that $u(a) = 0$. Try to get the correct value for the constant c. In particular make clear how it depends on the length of the interval.

b) Prove the Poincaré inequality for $\mathring{H}^1$ functions on the unit square.

(8) a) Let $\hat{T}$ be the triangle with vertices $\hat{\boldsymbol{a}}_1 = (0,0)$, $\hat{\boldsymbol{a}}_2 = (1,0)$, $\hat{\boldsymbol{a}}_3 = (0,1)$. Find the formulas for the three linear nodal basis function $\hat{\lambda}_i$ which satisfy $\hat{\lambda}_i(\hat{\boldsymbol{a}}_i) = \delta_{ij}$.

b) Same problem but now find all six quadratic nodal basis functions on $\hat{T}$.

c) Now let T be an arbitrary triangle with vertices $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$. Find the formulas for the linear nodal basis functions in this case.

(9) Obtain stability and convergence for the centered difference/backward difference method for the heat equation using the discrete max norm in time and space.

(10) Consider the use of centered differences in space and Crank-Nicolson in time for the advection equation. Use von Neumann analysis to show unconditional stability of this method.

(11) The reverse Lax–Friedrichs scheme for the advection equation is

$$\frac{(U_{n+1}^{j+1} + U_{n-1}^{j+1})/2 - U_n^{j+1}}{k} + c\,\frac{U_{n+1}^{j+1} - U_{n-1}^{j+1}}{2h} = 0.$$

Investigate the consistency, order, and stability of this scheme.

CHAPTER 7

Some Iterative Methods of Numerical Linear Algebra

1. Introduction

In this section we return to the question of solving linear systems $Au = f$, $A \in \mathbb{R}^{n\times n}$, $f \in \mathbb{R}^n$ given, and $u \in \mathbb{R}^n$ is sought. We have in mind mostly the case where A arises from discretization of a PDE (which is why we have chosen to use u and f rather than x and b for the vectors). For example, A might be $-D_h^2$ or $-\Delta_h$, the discrete Laplacian in one or two dimensions, or it might be the stiffness matrix from a finite element method, or the linear system which arises at each time step from the discretization of an evolutionary PDE by an implicit method such as backward differences in time or Crank-Nicolson. Therefore A may be a very large matrix, but it is also very sparse. In fact the number of non-zero elements in the matrix A is generally $O(n)$ rather than $O(n^2)$. E.g., for the 5-point Laplacian it is about $5n$. We shall only consider the case where A is symmetric positive definite (SPD). This is both because that is an important case, and because the theory is far simpler and better developed in that case.

First let us recall earlier results. We may of course use Cholesky factorization. This takes roughly $n^3/6$ multiplications and additions. However this ignores the sparsity of A. If A has a banded structure with semibandwidth d, that is $a_{ij} = 0$ if $|i-j| \geq d$, then each step of the Cholesky algorithm preserves this structure, and we can suppress the steps that zero entries d or more positions below the diagonal. Careful operation counting reveals that a Cholesky factorization then costs about $nd^2/4$ operations. For example, if $A = -D_h^2$ so $d = 1$, and we have only $O(n)$ operations, and so have an optimal algorithm (no operation could require less than one operation—it must take at least one operation to compute each component of the solution). However, if we consider the 5-point Laplacian on the unit square using a mesh spacing of $h = 1/N$, then the matrix size is $n \times n$ with $n = (N-1)^2 \approx h^{-2}$ and the semibandwidth is $N - 1 \approx h^{-1}$. Thus Cholesky factorization can be implemented in about $h^{-4}/4 = O(n^2)$ operations. This is significantly better than what would occur if we were to ignore sparsity—then we would require $h^{-6}/6 = O(n^3)$ operations—but it is not optimal.

In this chapter we will consider iterative methods, which start with an initial guess u_0 and construct iterates $u_1, u_2, \ldots$ which—hopefully—converge to the exact solution.

A method which we have already discussed is the conjugate gradient method, which we derived as a line-search method for minimizing $F(x) = u^T Au/2 - u^T f$ (the minimum occurs exactly at the solution of $Au = f$). Due to the sparsity of A, each iteration of the conjugate gradient method only requires $O(n)$ operations. For this method we proved that the error $e = u - u_i$ converges linearly to 0:

$$\|e_i\| \leq Cr^i,$$

where the rate

$$r = \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1},$$

and, if the norm $\|u\|_A := \sqrt{u^T A u}$ is used $C = 2\|e_0\|_A$. (Recall that the linear convergence in one norm implies linear convergence in all norms with the same rate.) Now each iteration of conjugate gradients involves a matrix multiplication and some dot products and SAXPYs, and so, for the 5-point Laplacian and similar matrices, can be completed in $O(n) = O(h^{-2})$ operations. However the condition number of the discrete Laplacian (in one or two dimensions) is $O(h^{-2})$ and so the rate of linear convergence is only $1 - O(h)$, i.e., $r \approx 1 - p \approx e^{-p}$ where $p \approx ch$ for some $c > 0$. Thus to reduce the initial error by some given factor, say by a factor of 10, we need to make $O(h^{-1})$ iterations. (To reduce the error by a factor of 10^6 then would take 6 times as many iterations, so still $O(h^{-1})$.) Thus the total work for $A = -\Delta_h$ is $O(h^{-3}) = O(n^{3/2})$ operations. This is a notable improvement on a direct solve using Cholesky decomposition, and should pretty much convince us that conjugate gradients will beat the direct solver for h sufficiently small. Computational experience suggests that there is a cross-over point: for fairly small problems direct solvers are often faster.

We also discussed previously improving the speed of conjugate gradients by preconditioning. We shall return to this presently.

2. Classical iterations

Now we consider some classical iterative methods to solve $Au = f$. One way to motivate such methods is to note that if u_0 is some approximate solution, then the exact solution u may be written $u = u_0 + e$ and the error $e = u - u_0$ is related to the residual $r = f - Au_0$ by the equation $Ae = r$. That is, we can express u as a *residual correction* to u_0: $u = u_0 + A^{-1}(f - Au_0)$. Of course, this is not a practical solution method since computing $e = A^{-1}(f - Au_0)$ by solving $Ae = r$ is as difficult as the original problem of solving for u. But suppose we have some nonsingular matrix B which approximates A^{-1} but is less costly to apply. We are then led to the iteration $u_1 = u_0 + B(f - Au_0)$, which can be repeated to give

$$u_{i+1} = u_i + B(f - Au_i), \quad i = 0, 1, 2, \ldots. \tag{7.1}$$

Of course the effectiveness of such a method will depend on the choice of B. For speed of convergence, we want B to be close to A^{-1}. For efficiency, we want B to be easy to apply. Some typical choices of B are:

- $B = \omega I$ for some $\omega > 0$. This is just the method of steepest descents with a constant step size ω. As we shall see, this method will converge for positive definite A if ω is a sufficiently small positive number. This iteration is often called the Richardson method.
- $B = D^{-1}$ where D is the diagonal matrix with the same diagonal elements as A. Then this is called the Jacobi method.
- $B = E^{-1}$ where E is the lower triangular matrix with the same diagonal and subdiagonal elements of A. This is the Gauss–Seidel method.

REMARK. Another way to derive the classical iterative methods is to give a splitting of A as $P + Q$ for two matrices P and Q where P is in some sense close to A but much easier to invert. We then write the equations as $Pu = f - Qu$, which suggests the iteration

$$u_{m+1} = P^{-1}(f - Qu_m).$$

Since $Q = A - P$, this iteration may also be written

$$u_{i+1} = u_i + P^{-1}(f - Au_i).$$

Thus this iteration coincides with (7.1) when $B = P^{-1}$.

Sometimes the iteration (7.1) is modified to

$$u_{i+1} = (1 - \alpha)u_i + \alpha[u_i + B(f - Au_i)], \quad i = 0, 1, 2, \ldots,$$

for a real parameter α. If $\alpha = 1$, this is the unmodified iteration. For $0 < \alpha < 1$ the iteration has been *damped*, while for $\alpha > 1$ the iteration is *amplified*. The damped Jacobi method will come up below when we study multigrid. The amplified Gauss–Seidel method is known as SOR (successive over-relaxation). This terminology is explained in the next two paragraphs.

Before investigating their convergence, let us particularize the classical iterations to the discrete Laplacian $-D_h^2$ in one or two dimensions. In one dimension, the equations are

$$\frac{-u^{m+1} + 2u^m - u^{m-1}}{h^2} = f^m, \quad m = 1, \ldots, N - 1,$$

where $h = 1/N$ and $u^0 = u^N = 0$. The Jacobi iteration is then simply

$$u_{i+1}^m = \frac{u_i^{m-1} + u_i^{m+1}}{2} + \frac{h^2}{2} f^m, \quad m = 1, \ldots, N - 1,$$

The error satisfies

$$e_{i+1}^m = \frac{e_i^{m-1} + e_i^{m+1}}{2},$$

so at each iteration the error at a point is set equal to the average of the errors at the neighboring points at the previous iteration. The same holds true for the 5-point Laplacian in two dimensions, except that now there are four neighboring points. In an old terminology, updating the value at a point based on the values at the neighboring points is called *relaxing* the value at the point.

For the Gauss–Seidel method, the corresponding equations are

$$u_{i+1}^m = \frac{u_{i+1}^{m-1} + u_i^{m+1}}{2} + \frac{h^2}{2} f^m, \quad m = 1, \ldots, N - 1,$$

and

$$e_{i+1}^m = \frac{e_{i+1}^{m-1} + e_i^{m+1}}{2}, \quad m = 1, \ldots, N - 1.$$

We can think of the Jacobi method as updating the value of u at all the mesh points simultaneously based on the old values, while the Gauss–Seidel method updates the values of one point after another always using the previously updated values. For this reason the Jacobi method is sometimes referred to as *simultaneous relaxation* and the Gauss–Seidel method as *successive relaxation* (and amplified Gauss–Seidel as successive overrelaxation). Note that the Gauss–Seidel iteration gives different results if the unknowns are reordered. (In fact, from the point of view of convergence of Gauss–Seidel, there are better orderings

than just the naive orderings we have taken so far.) By contrast, the Jacobi iteration is unaffected by reordering of the unknowns.

To investigate the convergence of (7.1) we write it as

$$u_{m+1} = (I - BA)u_m + Bf = Gu_m + Bf,$$

where the *iteration matrix* $G = I - BA$. The equation $u = Gu + Bf$ is certainly satisfied (where u is the exact solution), and so the classical iterations may be viewed as one-point iterations for this fixed point equation, as studied in Chapter 4.2. The error then satisfies $e_{m+1} = Ge_m$, and the method converges for all starting values $e_0 = u - u_0$ if and only if $\lim_{m\to\infty} G^m = 0$. Recall that this holds if and only if the spectral radius $\rho(G) < 1$ (Corollary 4.4), and then the rate of convergence is $\rho(G)$. Since $G = I - BA$, the convergence condition is that all the eigenvalues of BA must lie strictly inside the unit circle in the complex plane centered at 1. If BA has real eigenvalues (which will always be the case if B is symmetric, since then BA is symmetric with respect to the inner product $\langle u, v\rangle_A = u^T Av$), then the condition becomes that all the eigenvalues of BA belong to the interval $(0, 2)$.

As a first example, we consider the convergence of the Richardson method. The matrix A, being SPD, has a full set of eigenvalues

$$0 < \lambda_{\min}(A) = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n = \lambda_{\max}(A) = \rho(A).$$

The eigenvalues of $BA = \omega A$ are then $\omega\lambda_i$, $i = 1, \dots, n$, and the iteration converges if and only if $0 < \omega < 2/\lambda_{\max}$.

THEOREM 7.1. *Let A be an SPD matrix. Then the Richardson iteration $u_{m+1} = u_m + \omega(f - Au_m)$ is convergent for all choices of u_0 if and only if $0 < \omega < 2/\lambda_{max}(A)$. In this case the rate of convergence is*

$$\max(|1 - \omega\lambda_{max}(A)|, |1 - \omega\lambda_{min}(A)|).$$

For example, if we consider $A = -D_h^2$, the discrete Laplacian in one dimension, and choose $\omega = h^2/4$ (so $\omega \approx 1/\lambda_{\max}(A)$), the Richardson iteration converges with rate $1 - h^2\lambda_{\min}(A)/4 = 1 - O(h^2)$. Thus the converge is very slow (we will need $O(h^{-2})$ iterations).

Note that for $A = -D_h^2$ the Jacobi method coincides with the Richardson method with $\omega = h^2/2$. Since $\lambda_{\max}(A) < 4/h^2$ the Jacobi method is convergent. But again convergence is very slow, with a rate of $1 - O(h^2)$. In fact for any $0 < \alpha \le 1$, the damped Jacobi method is convergent, since it coincides with the Richardson method with $\omega = \alpha h^2/2$.

For the Richardson, Jacobi, and damped Jacobi iterations, the approximate inverse B is symmetric, but this is not the case for Gauss–Seidel, in which B is the inverse of the lower triangle of A. Of course we get a similar method if we use B^T, the upper triangle of A. If we take two steps of Gauss–Seidel, one with the lower triangle and one with the upper triangle, the iteration matrix is

$$(I - B^T A)(I - BA) = I - (B^T + B - B^T AB)A,$$

so this double iteration is itself a classical iteration with the approximate inverse

$$\bar{B} := B^T + B - B^T AB. \tag{7.2}$$

This iteration is called *symmetric Gauss–Seidel*. Now, from the definition of $\bar{B}$, we get the identity

$$\|v\|_A^2 - \|(I - BA)v\|_A^2 = \langle \bar{B}Av, v\rangle_A. \tag{7.3}$$

It follows that $\langle \bar{B}Av, v\rangle_A \leq \|v\|_A^2$, and hence that $\lambda_{\max}(\bar{B}A) \leq 1$. Thus the symmetrized Gauss–Seidel iteration is convergent if and only if $\lambda_{\min}(\bar{B}A) > 0$, i.e., if and only if $\bar{B}A$ is SPD with respect to the A inner product. This is easily checked to be equivalent to $\bar{B}$ being SPD with respect to the usual inner product. When this is the case (7.3) implies that $\|(I - BA)v\|_A < \|v\|_A$ for all nonzero v, and hence the original iteration is convergent as well.

Remark. In fact the above argument didn't use any properties of the original approximate inverse B. So what we have really proved is that given *any* approximate inverse matrix B, if we symmetrize by the formula (7.2) then the symmetrized iteration $u_{i+1} = u_i + \bar{B}(f - Au_i)$ is convergent if and only if $\bar{B}$ is SPD, and, in that case, the original iteration $u_{i+1} = u_i + B(f - Au_i)$ is convergent as well.

For Gauss–Seidel, let us write $A = D - L - L^T$ where D is diagonal and L strictly lower diagonal. Then the approximate inverse is $B = (D - L)^{-1}$ and

$$\begin{aligned}\bar{B} = B^T + B - B^T AB = B^T(B^{-1} + B^{-T} - A)B \\ = B^T[(D - L) + (D - L^T) - (D - L - L^T)]B = B^T DB,\end{aligned}$$

which is clearly SPD whenever A is. Thus we have proven:

Theorem 7.2. *The Gauss–Seidel and symmetric Gauss–Seidel iterations are convergent for any symmetric positive definite linear system.*

It is worth remarking that the same result is *not* true for the Jacobi iteration: although convergence can be proven for many of the SPD matrices that arise from discretizations of PDE, there exists SPD matrices for which Jacobi iteration does not converge. As to the speed of convergence for Gauss–Seidel when applied to the discrete Laplacian, the analysis is much trickier than for Jacobi, but it can again be proven (or convincingly demonstrated via simple numerical experiments) that for $A = -D_h^2$ the rate of convergence is again just $1 - O(h^2)$, as for Jacobi, although the constant in the $O(h^2)$ term is about twice as big for Gauss–Seidel as for Jacobi.

Thus while the classical iterations have a certain elegance, and do converge for typical SPD problems arising from elliptic PDE, they converge very slowly. In fact, they are not competitive with the conjugate gradient method. One good use of the classical iterations, however, is to precondition conjugate gradients. As long as the approximate inverse B is SPD, we may use it as a preconditioner. The Jacobi preconditioner, also known as diagonal preconditioning often has minimal effect. Indeed for the discrete Laplacian it has no effect at all, since the diagonal is constant. The symmetric Gauss–Seidel preconditioner is a bit more helpful.

In fact, we can show that conjugate gradients preconditioned by some SPD approximate inverse always converges faster than the corresponding classical iterative method. For if λ is an eigenvalue of BA, then $-\rho \leq 1 - \lambda \leq \rho$ where ρ is the spectral radius of $I - BA$, and so

$$\lambda_{\min}(BA) \geq 1 - \rho, \quad \lambda_{\max}(BA) \leq 1 + \rho, \quad \kappa(BA) \leq \frac{1 + \rho}{1 - \rho}.$$

TABLE 7.1. Operation counts for solving linear systems for the discrete Laplacian in one and two dimensions. For the iterative methods, the total number of operations to reduce the error by a given factor is the product of the number of operations per iteration times the number of iterations required.

Method	1D: $A = -D_h^2$, $n = O(h^{-1})$			2D: $A = -\Delta_h$, $n = O(h^{-2})$		
	ops./iter.	no. iters.	total ops.	ops./iter.	no. iters.	total ops.
Cholesky factorization	–	–	$O(h^{-1})$	–	–	$O(h^{-4})$
Conjugate gradients	$O(h^{-1})$	$O(h^{-1})$	$O(h^{-2})$	$O(h^{-2})$	$O(h^{-1})$	$O(h^{-3})$
Richardson, Jacobi	$O(h^{-1})$	$O(h^{-2})$	$O(h^{-3})$	$O(h^{-2})$	$O(h^{-2})$	$O(h^{-4})$
Gauss–Seidel	$O(h^{-1})$	$O(h^{-2})$	$O(h^{-3})$	$O(h^{-2})$	$O(h^{-2})$	$O(h^{-4})$

Thus the rate of convergence for the PCG method is at most

$$\frac{\sqrt{\kappa(BA)}-1}{\sqrt{\kappa(BA)}+1} \le \frac{\sqrt{\frac{1+\rho}{1-\rho}}-1}{\sqrt{\frac{1+\rho}{1-\rho}}+1} = \frac{1-\sqrt{1-\rho^2}}{\rho}.$$

The last quantity is strictly less than ρ for all $\rho \in (0,1)$. (For ρ small it is about $\rho/2$, while for $\rho \approx 1-\epsilon$ with ϵ small, it is approximately $1-\sqrt{2\epsilon}$.) Thus the rate of convergence of PCG with B as a preconditioner is better than that of the classical iteration with B as approximate inverse.

3. Multigrid methods

Figure 7.1 shows the result of solving a discrete system of the form $-D_h^2 u_h = f$ using the Gauss–Seidel iteration. We have take $h = 1/128$, and chosen a smooth right-hand side vector f which results in the vector u_h which is plotted as a dashed line in each of the plots. The initial iterate u_0, which is shown in the first plot, was chosen at random, and then the iterates u_1, u_2, u_5, u_{50}, and u_{500} are shown in the subsequent plots. In Figure 7.2, the relative error $\|u_h - u_i\|/\|u_h\|$ is plotted for $i = 0, 1, \ldots, 50$, in both the l^∞ and the l^2 norms.

These numerical experiments illustrate the following qualitative properties, which are typical of the Gauss–Seidel iteration applied to matrices arising from the discretization of elliptic PDEs.

- If we start with a random error, the norm of the error will be reduced fairly quickly for the first few iterations, but the error reduction occurs much more slowly after that.
- After several iterations the error is much smoother, but not much smaller, than initially. Otherwise put, the highly oscillatory modes of the error are suppressed much more quickly by the iteration than the low frequency modes.

FIGURE 7.1. Iterative solution to $-D_h^2 u_h = f$, $h = 1/128$, using Gauss–Seidel. The random initial iterate is rapidly smoothed, but approaches the solution u_h only very slowly.

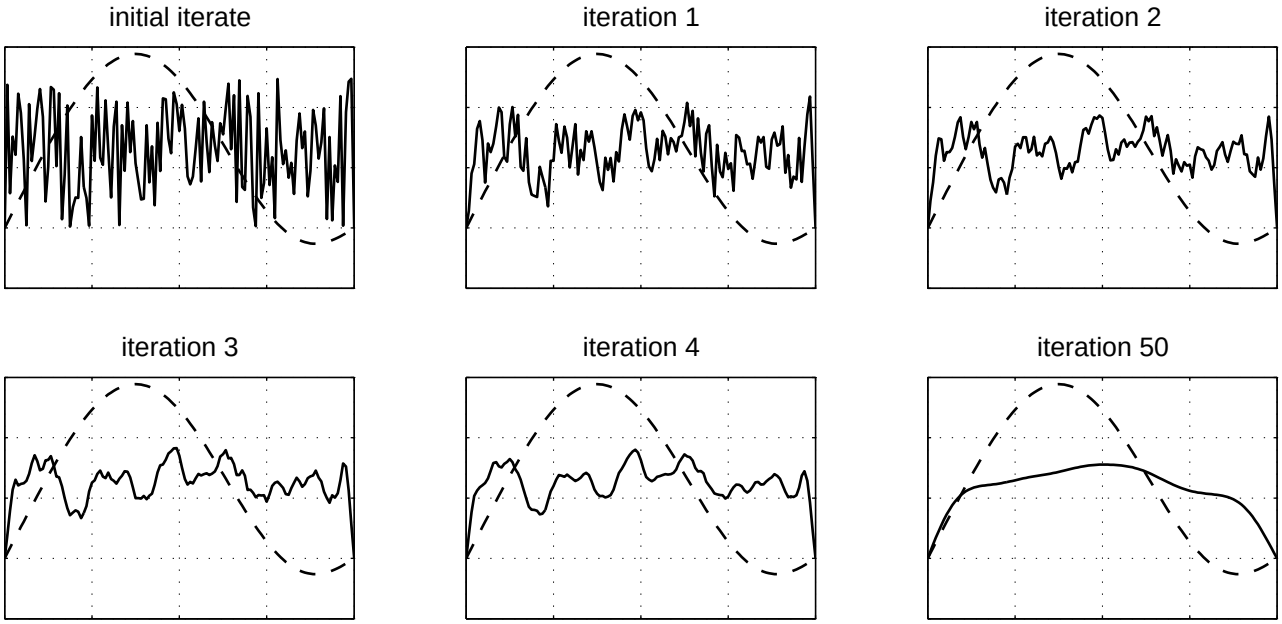
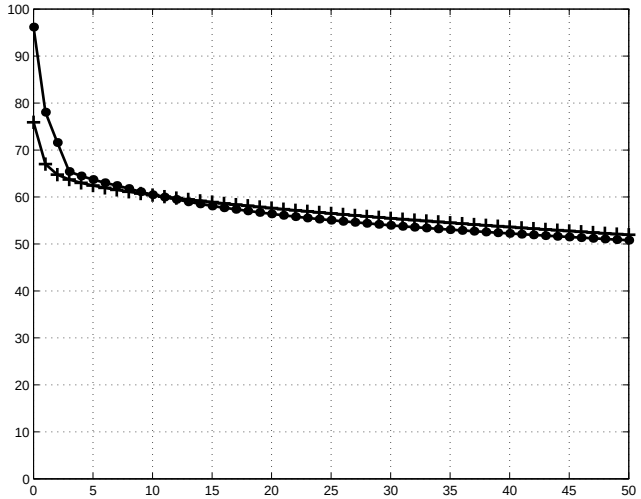


FIGURE 7.2. Relative error in percent in the Gauss–Seidel iterates 0 through 50 in the l^∞ (•) and l^2 (+) norms.



The first observation is valid for all the methods we have studied: Richardson, Jacobi, damped Jacobi, and Gauss–Seidel. The second obervation—that Gauss–Seidel iteration *smooths* the error—is shared damped Jacobi with $\alpha < 1$, but not by Jacobi itself.

If we take the Richardson method with $\omega = 1/\lambda_{\max}(A)$ for the operator $A = -D_h^2$, it is very easy to see how the smoothing property comes about. The initial error can be

expanded in terms of the eigenfunctions of A: $e_0 = \sum_{m=1}^{n} c_i \sin m\pi x$. The mth component in this expansion is multiplied by $1 - \lambda_m/\lambda_{\max} = 1 - \lambda_m/\lambda_n$ at each iteration. Thus the high frequency components, $m \approx n$, are multiplied by something near to 0 at each iteration, and so are damped very quickly. Even the intermediate eigenvalues, $\lambda_m \approx \lambda_n/2$ are damped reasonably quickly (by a factor of about $1/2$ at each iteration). But the low frequency modes, for which $\lambda_m \ll \lambda_n$, decrease very slowly.

This also explains the first observation, that the norm of the error decreases quickly at first, and then more slowly. The norm of the error has contributions from all modes present in the initial error. Those associated to the higher frequency modes disappear in a few iterations, bringing the error down by a significant fraction. But after that the error is dominated by the low frequency modes, and so decays very slowly.

The same analysis applies to damped Jacobi with positive damping, and shows that undamped Jacobi doesn't have the smoothing property: the mth mode is multiplied by about $1 - 2\lambda_m/\lambda_n$, and so convergence is very slow for low frequency modes and also the highest frequency modes $\lambda_m \approx \lambda_n$. For the intermediate modes, $\lambda_m \approx \lambda_n/2$, convergence is very fast.

Establishing the smoothing property for Gauss–Seidel is more complicated, since the eigenfunctions of the Gauss–Seidel iteration don't coincide with those of A even for $A = -D_h^2$. However both numerical study and careful analysis show that Gauss–Seidel does indeed have the smoothing property for discretized elliptic operators.

The idea behind the multigrid method is to create an iterative method which reduces all components of the residual quickly by putting together two steps. First it applies the approximate inverse from Gauss–Seidel or another classical iterative method with the smoothing property to the residual. This greatly reduces the high frequency components of the residual, but barely reduces the low frequency components. The new residual, being relatively smooth, can then be accurately approximated on a coarser mesh. So, for the second step, the residual is (somehow) transferred to a coarser mesh, and the equation solved there, thus reducing the low frequency components. On the coarser mesh, it is of course less expensive to solve. For simplicity, we assume for now that an exact solver is used on the coarse mesh. Finally this coarse mesh solution to the residual problem is somehow transferred back to the fine mesh where it can be added back to our smoothed approximation.

Thus we have motivated the following rough outline of an algorithm:

(1) Starting from an initial guess u_0 apply a fine mesh smoothing iteration to get an improved approximation $\bar{u}$.
(2) Transfer the residual in $\bar{u}$ to a coarser mesh, solve a coarse mesh version of the problem there, transfer the solution back to the fine mesh, and add it back to $\bar{u}$ to get $\bar{\bar{u}}$.

Taking $\bar{\bar{u}}$ for u_1 and thus have described an iteration to get from u_0 to u_1 (which we can then apply again to get from u_1 to u_2, and so on). In fact it is much more common to also apply a fine mesh smoothing at the end of the iteration, i.e., to add a third step:

3. Starting from $\bar{\bar{u}}$ apply the smoothing iteration to get an improved approximation $\bar{\bar{\bar{u}}}$.

The point of including the third step is that it leads to a multigrid iteration which is symmetric, which is often advantageous (e.g., the iteration can be used as a preconditioner for conjugate gradients). If the approximation inverse B used for the first smoothing step is not

symmetric, we need to apply B^T (which is also an approximate inverse, since A is symmetric) to obtain a symmetric iteration.

We have just described a *two-grid* iteration. The true multigrid method will involve not just the original mesh and one coarser mesh, but a whole sequence of meshes. However, once we understand the two-grid iteration, the multigrid iteration will follow easily.

To make the two-grid method more precise we need to explain step 2 more fully, namely (a) how do we transfer the residual from the fine mesh to the coarse mesh?; (b) what problem do we solve on the coarse mesh?; and (c) how do we transfer the solution of that problem from the coarse mesh to the fine mesh? For simplicity, we suppose that $N = 1/h$ is even and that we are interested in solving $A_h u = f$ where $A = -D_h^2$. Let $H = 2h = 1/(2N)$. We will use the mesh of size H as our coarse mesh. The first step of our multigrid iteration is then just

$$\bar{u} = u_0 + B_h(f - A_h u_0),$$

where B_h is just the approximate inverse of A_h from Gauss–Seidel or some other smoothing iteration. The resulting residual is $f - A_h\bar{u}$. This is a function on the fine mesh points $h, 2h, \ldots, (N-1)h$, and a natural way to transfer it to the coarse mesh is restrict it to the even grid points $2h, 4h, \ldots, (N-2)h = H, 2H, \ldots, (N/2-1)H$, which are exactly the coarse mesh grid points. Denoting this *restriction operator* from fine grid to coarse grid functions (i.e., from $\mathbb{R}^{N-1} \to \mathbb{R}^{N/2-1}$) by P_H, we then solve $A_H e_H = P_H(f - A_h\bar{u}_h)$ where, of course, $A_H = -D_H^2$ is the 3-point difference operator on the coarse mesh. To transfer the solution e_H, a coarse grid function, to the fine grid, we need a *prolongation operator* $Q_H : \mathbb{R}^{N/2-1} \to \mathbb{R}^{N-1}$. It is natural to set $Q_H e_H(jh) = e_H(jh)$ if j is even. But what about when j is odd: how should we define $Q_H e_H$ at the midpoint of two adjacent coarse mesh points? A natural choice, which is simple to implement, is $Q_H e_H(jh) = [e_H((j-1)h) + e((j+1)h)]/2$. With these two operators second step is

$$\bar{\bar{u}} = \bar{u} + Q_H A_H^{-1} P_H(f - A_h\bar{u}).$$

And then final post-smoothing step is

$$\bar{\bar{\bar{u}}} = \bar{\bar{u}} + B_h^T(f - A_h\bar{\bar{u}}).$$

Actually this does not give a symmetric iteration. To obtain symmetry we need $Q_h = cP_H^T$ and that is not the case for the grid transfer operators we defined. We have

$$Q_H = \begin{pmatrix} 1/2 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 1/2 & 1/2 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1/2 & 1/2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1/2 \end{pmatrix}, \tag{7.4}$$

but P_H as we described it, consists only of 0's and 1's. Therefore one commonly takes a different choice for P_H, namely $P_H = (1/2)Q_H^T$. This means that the transferred coarse grid function doesn't just take the value of the corresponding fine grid function at the coarse grid point, but rather uses a weighted average of the fine grid function's values at the point in question and the fine grid points to the left and right (with weights 1/4, 1/2, 1/4). With

this choice, $Q_H A_h P_H$ is symmetric; in fact, $Q_H A_h P_H = A_H$. This is a useful formula. For operators other than the $A_h = -D_h^2$, we can use the same intergrid transfer operators, namely Q_H given by (7.4) and $P_H = (1/2)Q_H^T$, and then define the coarse grid operator by $A_H = Q_H A_h P_H$.

REMARK. In a finite element context, the situation is simpler. If the fine mesh is a refinement of the coarse mesh, then a coarse mesh function is already a fine mesh function. Therefore, the operator Q_H can be taken simply to be the inclusion operator of the coarse mesh space into the fine mesh space. The residual in $u_0 \in S_h$ is most naturally viewed as a functional on S_h: $v \mapsto (f, v) - B(u_0, v)$. It is then natural to transfer the residual to the coarse mesh simply by restricting the test function v to S_H. This operation $S_h^T \to S_H^T$ is exactly the adjoint of the inclusion operator $S_H \to S_h$. Thus the second step, solving the coarse mesh problem for the restricted residual is obvious in the finite element case: we find $e_H \in S_H$ such that

$$B(e_H, v) = (f, v) - B(\bar{u}, v), \quad v \in S_H,$$

and then we set $\bar{\bar{u}} = \bar{u} + e_H \in S_h$.

Returning to the case of finite differences we have arrived at the following two-grid iterative method to solve $A_h u_h = f_h$.

$u_h = \text{twogrid}(h, A_h, f_h, u_0)$
input: h, mesh size ($h = 1/n$ with n even)
A_h, operator on mesh functions
f_h, mesh function (right-hand side)
u_0, mesh function (initial iterate)
output: u_h, mesh function (approximate solution)

---

for $i = 0, 1, \ldots$ until satisfied
1. presmoothing: $\bar{u} = u_i + B_h(f_h - A_h u_i)$
2. coarse grid correction:
   2.1. residual computation: $r_h = f_h - A_h \bar{u}$
   2.2. restriction: $H = 2h$, $r_H = P_H r_h$, $A_H = P_H A_h Q_H$
   2.3. coarse mesh solve: solve $A_H e_H = r_H$
   2.4. prolongation: $e_h = Q_H e_H$
   2.5. correction: $\bar{\bar{u}} = \bar{u} + e_h$
3. postsmoothing: $u_h \leftarrow u_{i+1} = \bar{\bar{u}} + B_h^T(f_h - A_h \bar{\bar{u}})$

end

Algorithm 7.1: Two-grid iteration for approximately solving $A_h u_h = f_h$.

In the smoothing steps, the matrix B_h could be, for example, $(D - L)^{-1}$ where D is diagonal, L strictly lower triangular, and $A_h = D - L - L^T$. This would be a Gauss–Seidel smoother, but there are other possibilities as well. Besides these steps, the major work is in the coarse mesh solve. To obtain a more efficient algorithm, we may also solve on the coarse mesh using a two-grid iteration, and so involving an even coarser grid. In the following

multigrid algorithm, we apply this idea recursively, using multigrid to solve at each mesh level, until we get to a sufficiently coarse mesh, $h = 1/2$, at which point we do an exact solve (with a 1×1 matrix!).

$u_h = \text{multigrid}(h, A_h, f_h, u_0)$
input: h, mesh size ($h = 1/n$ with n a power of 2)
A_h, operator on mesh functions
f_h, mesh function (right-hand side)
u_0, mesh function (initial iterate)
output: u_h, mesh function (approximate solution)

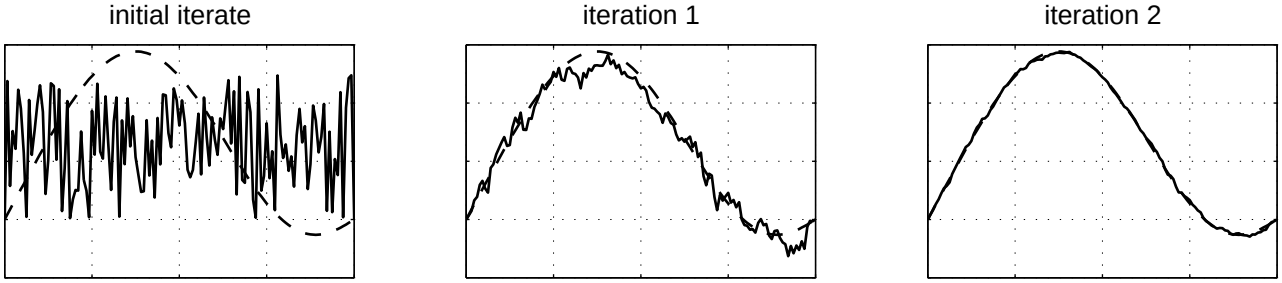
---

if $h = 1/2$ **then**
  $u_h = A_h^{-1} f_h$
else
  for $i = 0, 1, \ldots$ until satisfied
    1. presmoothing: $\bar{u} = u_i + B_h(f - A_h u_i)$
    2. coarse grid correction:
      2.1. residual computation: $r_h = f_h - A_h \bar{u}$
      2.2. restriction: $H = 2h$, $r_H = P_H r_h$, $A_H = P_H A_h Q_H$
      2.3. coarse mesh solve: $e_H = \text{multigrid}(H, A_H, r_H, 0)$
      2.4. prolongation: $e_h = Q_H e_H$
      2.5. correction: $\bar{\bar{u}} = \bar{u} + e_h$
    3. postsmoothing: $u_h \leftarrow u_{i+1} = \bar{\bar{u}} + B_h^T(f - A_h \bar{\bar{u}})$
  end
end if

Algorithm 7.2: Multigrid iteration for approximately solving $A_h u_h = f$.

Figure 7.3 shows two iterations of this multigrid algorithm for solving the system $-D_h^2 u_h = f$, $h = 1/128$, considered at the beginning of this section. Compare with Figure 7.1. The fast convergence of the multigrid algorithm is remarkable. Indeed, for the multigrid method discussed here it is possible to show that the iteration is linearly convergent with a rate independent of the mesh size. This means that the number of iterations needed to obtain a desired accuracy remains bounded independent of h. It is also easy to count the number of operations per iteration. Each iteration involves two applications of the smoothing iteration, plus computation of the residual, restriction, prolongation, and correction on the finest mesh level. All those procedures cost $O(n)$ operations. But then, during the coarse grid solve, the same procedures are applied on the grid of size $2h$, incurring an additional cost of $O(n/2)$. Via the recursion the work will be incurred for each mesh size $h, 2h, 4h, \ldots$. Thus the total work per iteration will be $O(n + n/2 + n/4 + \ldots + 1) = O(n)$ (since the geometric series sums to $2n$). Thus the total work to obtain the solution of the discrete system to any desired accuracy is itself $O(n)$, i.e., optimal. For a fuller introduction to multigrid, including the theoretical analysis, see [**1**].

FIGURE 7.3. Iterative solution to $-D_h^2 u_h = f$, $h = 1/128$, using multigrid.



Other things that could go here. Some more numerical results showing independence of h. V-cycle diagram, discussion of multiple smoothings, W-cycle; more precise convergence statement. Full multigrid. Multigrid for finite elements; connection with adaptivity. Numerical results, e.g., from black hole research.

SOR theory might make some good exercises.

EXERCISES

(1) Let A be a tridiagonal matrix with all the diagonal entries equal to 3 and all the subdiagonal and superdiagonal entries equal to -1. Determine for which values of the real parameter ω the iteration $x_{i+1} = x_i + \omega(b - Ax_i)$ converges to the solution of $Ax = b$ for any choice of initial iterate x_0.

(2) Consider the linear system: find $u \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$ such that

$$Au + Bp = f, \qquad B^T u = g.$$

Here $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $B \in \mathbb{R}^{n\times m}$ has rank m, $f \in \mathbb{R}^n$, and $g \in \mathbb{R}^m$.

a) Prove that this system is nonsingular.

b) The *Uzawa iteration* for this system proceeds as follows.

1. pick an initial iterate $p_0 \in \mathbb{R}^m$
2. for $i = 0, 1, \ldots$
   solve $Au_i + Bp_i = f$ to determine $u_i \in \mathbb{R}^n$
   set $p_{i+1} = p_i + \alpha(B^T u_i - g)$

Determine for what values of the real parameter α, the Uzawa iteration converges.

Bibliography

1. Jinchao Xu, *An introduction to multilevel methods*, Wavelets, multilevel methods and elliptic PDEs (Leicester, 1996) (M. Ainsworth, J. Levesley, M. Marletta, and W. A. Light, eds.), Oxford Univ. Press, New York, 1997, pp. 213–302.