Chapter 2

Basic Iterative Methods

We now consider how model problems (1.3) and (1.5) might be treated using conventional iterative or relaxation methods. We first establish the notation for this and all remaining chapters. Let

$$A\mathbf{u} = \mathbf{f}$$

denote a system of linear equations such as (1.3) or (1.6). We always use $\mathbf{u}$ to denote the exact solution of this system and $\mathbf{v}$ to denote an approximation to the exact solution, perhaps generated by some iterative method. Bold symbols, such as $\mathbf{u}$ and $\mathbf{v}$, represent vectors, while the jth components of these vectors are denoted by u_j and v_j. In later chapters, we need to associate $\mathbf{u}$ and $\mathbf{v}$ with a particular grid, say Ω^h. In this case, the notation $\mathbf{u}^h$ and $\mathbf{v}^h$ is used.

Suppose that the system $A\mathbf{u} = \mathbf{f}$ has a unique solution and that $\mathbf{v}$ is a computed approximation to $\mathbf{u}$. There are two important measures of $\mathbf{v}$ as an approximation to $\mathbf{u}$. One is the *error* (or *algebraic error*) and is given simply by

$$\mathbf{e} = \mathbf{u} - \mathbf{v}.$$

The error is also a vector and its magnitude may be measured by any of the standard vector norms. The most commonly used norms for this purpose are the maximum (or infinity) norm and the Euclidean or 2-norm, defined, respectively, by

$$\|\mathbf{e}\|_\infty = \max_{1 \le j \le n} |e_j| \quad \text{and} \quad \|\mathbf{e}\|_2 = \left\{ \sum_{j=1}^{n} e_j^2 \right\}^{1/2}.$$

Unfortunately, the error is just as inaccessible as the exact solution itself. However, a computable measure of how well $\mathbf{v}$ approximates $\mathbf{u}$ is the *residual*, given by

$$\mathbf{r} = \mathbf{f} - A\mathbf{v}.$$

The residual is simply the amount by which the approximation $\mathbf{v}$ fails to satisfy the original problem $A\mathbf{u} = \mathbf{f}$. It is also a vector and its size may be measured by the same norm used for the error. By the uniqueness of the solution, $\mathbf{r} = \mathbf{0}$ if and only if $\mathbf{e} = \mathbf{0}$. However, it may *not* be true that when $\mathbf{r}$ is small in norm, $\mathbf{e}$ is also small in norm.

Residuals and Errors. A residual may be defined for any numerical approximation and, in many cases, a small residual does *not* necessarily imply a small error. This is certainly true for systems of linear equations, as shown by the following two problems:

$$\begin{pmatrix} 1 & -1 \\ 21 & -20 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -1 \\ -19 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & -1 \\ 3 & -1 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Both systems have the exact solution $\mathbf{u} = (1,2)^T$. Suppose we have computed the approximation $\mathbf{v} = (1.95, 3)^T$. The error in this approximation is $\mathbf{e} = (-0.95, -1)^T$, for which $\|\mathbf{e}\|_2 = 1.379$. The norm of the residual in $\mathbf{v}$ for the first system is $\|\mathbf{r_1}\|_2 = 0.071$, while the residual norm for the second system is $\|\mathbf{r_2}\|_2 = 1.851$. Clearly, the relatively small residual for the first system does not reflect the rather large error. See Exercise 18 for an important relationship between error and residual norms.

Remembering that $A\mathbf{u} = \mathbf{f}$ and using the definitions of $\mathbf{r}$ and $\mathbf{e}$, we can derive an extremely important relationship between the error and the residual (Exercise 2):

$$A\mathbf{e} = \mathbf{r}.$$

We call this relationship the *residual equation*. It says that the error satisfies the same set of equations as the unknown $\mathbf{u}$ when $\mathbf{f}$ is replaced by the residual $\mathbf{r}$. The residual equation plays a vital role in multigrid methods and it is used repeatedly throughout this tutorial.

We can now anticipate, in an imprecise way, how the residual equation can be used to great advantage. Suppose that an approximation $\mathbf{v}$ has been computed by some method. It is easy to compute the residual $\mathbf{r} = \mathbf{f} - A\mathbf{v}$. To improve the approximation $\mathbf{v}$, we might solve the residual equation for $\mathbf{e}$ and then compute a new approximation using the definition of the error

$$\mathbf{u} = \mathbf{v} + \mathbf{e}.$$

In practice, this method must be applied more carefully than we have indicated. Nevertheless, this idea of residual correction is very important in all that follows.

We now turn to relaxation methods for our first model problem (1.3) with $\sigma = 0$. Multiplying that equation by h^2 for convenience, the discrete problem becomes

$$\begin{aligned} -u_{j-1} + 2u_j - u_{j+1} &= h^2 f_j, \qquad 1 \le j \le n-1, \\ u_0 = u_n &= 0. \end{aligned} \tag{2.1}$$

One of the simplest schemes is the *Jacobi* (or simultaneous displacement) method. It is produced by solving the jth equation of (2.1) for the jth unknown and using the current approximation for the $(j-1)$st and $(j+1)$st unknowns. Applied to the vector of current approximations, this produces an iteration scheme that may be written in component form as

$$v_j^{(1)} = \frac{1}{2}\left(v_{j-1}^{(0)} + v_{j+1}^{(0)} + h^2 f_j\right), \qquad 1 \le j \le n-1.$$

To keep the notation as simple as possible, the current approximation (or the initial guess on the first iteration) is denoted $\mathbf{v}^{(0)}$, while the new, updated approximation is denoted $\mathbf{v}^{(1)}$. In practice, once all of the $\mathbf{v}^{(1)}$ components have been computed, the procedure is repeated, with $\mathbf{v}^{(1)}$ playing the role of $\mathbf{v}^{(0)}$. These iteration sweeps are continued until (ideally) convergence to the solution is obtained.

It is important to express these relaxation schemes in matrix form, as well as component form. We split the matrix A in the form

$$A = D - L - U,$$

where D is the diagonal of A, and $-L$ and $-U$ are the strictly lower and upper triangular parts of A, respectively. Including the h^2 term in the vector $\mathbf{f}$, then $A\mathbf{u} = \mathbf{f}$ becomes

$$(D - L - U)\mathbf{u} = \mathbf{f}.$$

Isolating the diagonal terms of A, we have

$$D\mathbf{u} = (L + U)\mathbf{u} + \mathbf{f}$$

or

$$\mathbf{u} = D^{-1}(L + U)\mathbf{u} + D^{-1}\mathbf{f}.$$

Multiplying by D^{-1} corresponds exactly to solving the jth equation for u_j, for $1 \le j \le n-1$. If we define the Jacobi iteration matrix by

$$R_J = D^{-1}(L + U),$$

then the Jacobi method appears in matrix form as

$$\mathbf{v}^{(1)} = R_J\mathbf{v}^{(0)} + D^{-1}\mathbf{f}.$$

There is a simple but important modification that can be made to the Jacobi iteration. As before, we compute the new Jacobi iterates using

$$v_j^* = \frac{1}{2}\left(v_{j-1}^{(0)} + v_{j+1}^{(0)} + h^2 f_j\right), \qquad 1 \le j \le n-1.$$

However, v_j^* is now only an intermediate value. The new iterate is given by the weighted average

$$v_j^{(1)} = (1-\omega)v_j^{(0)} + \omega v_j^* = v_j^{(0)} + \omega(v_j^* - v_j^{(0)}), \qquad 1 \le j \le n-1,$$

where $\omega \in \mathbf{R}$ is a weighting factor that may be chosen. This generates an entire family of iterations called the *weighted* or *damped Jacobi* method. Notice that $\omega = 1$ yields the original Jacobi iteration.

In matrix form, the weighted Jacobi method is given by (Exercise 3)

$$\mathbf{v}^{(1)} = [(1-\omega)I + \omega R_J]\mathbf{v}^{(0)} + \omega D^{-1}\mathbf{f}.$$

If we define the weighted Jacobi iteration matrix by

$$R_\omega = (1-\omega)I + \omega R_J,$$

then the method may be expressed as (Exercise 3)

$$\mathbf{v}^{(1)} = R_\omega \mathbf{v}^{(0)} + \omega D^{-1}\mathbf{f}.$$

We should note in passing that the weighted Jacobi iteration can also be written in the form (Exercise 3)

$$\mathbf{v}^{(1)} = \mathbf{v}^{(0)} + \omega D^{-1}\mathbf{r}^{(0)}.$$

This says that the new approximation is obtained from the current one by adding an appropriate weighting of the residual.

This is just one example of a *stationary linear iteration.* This term refers to the fact that the update rule is linear in the unknown $\mathbf{v}$ and does not change from one iteration to the next. We can say more about such iterations in general. Recalling that $\mathbf{e} = \mathbf{u} - \mathbf{v}$ and $A\mathbf{e} = \mathbf{r}$, we have

$$\mathbf{u} - \mathbf{v} = A^{-1}\mathbf{r}.$$

Identifying $\mathbf{v}$ with the current approximation $\mathbf{v}^{(0)}$ and $\mathbf{u}$ with the new approximation $\mathbf{v}^{(1)}$, an iteration may be formed by taking

$$\mathbf{v}^{(1)} = \mathbf{v}^{(0)} + B\mathbf{r}^{(0)}, \tag{2.2}$$

where B is an approximation to A^{-1}. If B can be chosen "close" to A^{-1}, then the iteration should be effective.

It is useful to examine this general form of iteration a bit further. Rewriting expression (2.2), we see that

$$\begin{aligned} \mathbf{v}^{(1)} = \mathbf{v}^{(0)} + B\mathbf{r}^{(0)} &= \mathbf{v}^{(0)} + B(f - A\mathbf{v}^{(0)}) \\ &= (I - BA)\mathbf{v}^{(0)} + B\mathbf{f} \\ &\equiv R\mathbf{v}^{(0)} + B\mathbf{f}, \end{aligned}$$

where we have defined the general iteration matrix as $R = I - BA$. It can also be shown (Exercise 4) that m sweeps of this iteration result in

$$\mathbf{v}^{(m)} = R^m \mathbf{v}^{(0)} + C(\mathbf{f}),$$

where $C(\mathbf{f})$ represents a series of operations on $\mathbf{f}$. We return to this general form in Chapter 5.

Before analyzing or implementing these methods, we present a few more of the basic iterative schemes. Weighted Jacobi computes all components of the new approximation before using any of them. This requires $2n$ storage locations for the approximation vector. It also means that new information cannot be used as soon as it is available.

The *Gauss–Seidel* method incorporates a simple change: components of the new approximation are used as soon as they are computed. This means that components of the approximation vector $\mathbf{v}$ are overwritten as soon as they are updated. This small change reduces the storage requirement for the approximation vector to only n locations. The Gauss–Seidel method is also equivalent to successively setting each component of the residual vector to zero and solving for the corresponding component of the solution (Exercise 5). When applied to the model problem, this method may be expressed in component form as

$$v_j \longleftarrow \frac{1}{2}\left(v_{j-1} + v_{j+1} + h^2 f_j\right), \qquad 1 \le j \le n-1,$$

where the arrow notation stands for replacement or overwriting.

Once again it is useful to express this method in matrix form. Splitting the matrix A in the form $A = D - L - U$, we can now write the original system of equations as

$$(D - L)\mathbf{u} = U\mathbf{u} + \mathbf{f}$$

or

$$\mathbf{u} = (D - L)^{-1}U\mathbf{u} + (D - L)^{-1}\mathbf{f}.$$

This representation corresponds to solving the jth equation for u_j and using new approximations for components $1, 2, \ldots, j-1$. Defining the Gauss–Seidel iteration matrix by

$$R_G = (D - L)^{-1}U,$$

we can express the method as

$$\mathbf{v} \longleftarrow R_G\mathbf{v} + (D - L)^{-1}\mathbf{f}.$$

Finally, we look at one important variation on the Gauss–Seidel iteration. For weighted Jacobi, the order in which the components of $\mathbf{v}$ are updated is immaterial, since components are never overwritten. However, for Gauss–Seidel, the order of updating is significant. Instead of sweeping through the components (equivalently, the grid points) in ascending order, we could sweep through the components in descending order or we might alternate between ascending and descending orders. The latter procedure is called the *symmetric Gauss–Seidel* method.

Another effective alternative is to update all the even components first by the expression

$$v_{2j} \longleftarrow \frac{1}{2}\big(v_{2j-1} + v_{2j+1} + h^2 f_{2j}\big),$$

and then update all the odd components using

$$v_{2j+1} \longleftarrow \frac{1}{2}\big(v_{2j} + v_{2j+2} + h^2 f_{2j+1}\big).$$

This strategy leads to the *red-black Gauss–Seidel* method, which is illustrated in Fig. 2.1 for both one-dimensional and two-dimensional grids. Notice that the red points correspond to even-indexed points in one dimension and to points whose index sum is even in two dimensions (assuming that $i = 0$ and $j = 0$ corresponds to a boundary). The red points also correspond to what we soon call coarse-grid points.

The advantages of red-black over regular Gauss–Seidel are not immediately apparent; the issue is often problem-dependent. However, red-black Gauss–Seidel does have a clear advantage in terms of parallel computation. The red points need only the black points for their updating and may therefore be updated in any order. This work represents $\frac{n}{2}$ (or $\frac{n^2}{2}$ in two dimensions) independent tasks that can be distributed among several independent processors. In a similar way, the black sweep can also be done by several independent processors. (The Jacobi iteration is also well-suited to parallel computation.)

There are many more basic iterative methods. However, we have seen enough of the essential methods to move ahead toward multigrid. First, it is important to gain some understanding of how these basic iterations perform. We proceed both by analysis and by experimentation.

When studying stationary linear iterations, it is sufficient to work with the homogeneous linear system $A\mathbf{u} = \mathbf{0}$ and use arbitrary initial guesses to start the

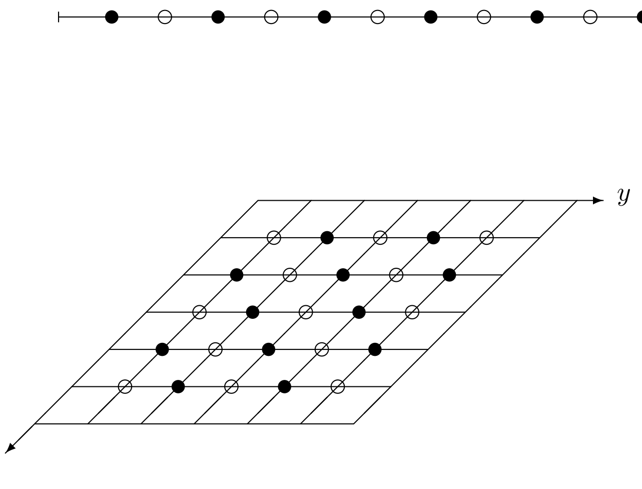

Figure 2.1: *A one-dimensional grid (top) and a two-dimensional grid (bottom), showing the red points* (○) *and the black points* (●) *for red-black relaxation.*

relaxation scheme (Exercise 6). One reason for doing this is that the exact solution is known ($\mathbf{u} = \mathbf{0}$) and the error in an approximation $\mathbf{v}$ is simply $-\mathbf{v}$. Therefore, we return to the one-dimensional model problem with $\mathbf{f} = \mathbf{0}$. It appears as

$$\begin{aligned} -u_{j-1} + 2u_j - u_{j+1} &= 0, \qquad 1 \le j \le n-1, \\ u_0 = u_n &= 0. \end{aligned} \tag{2.3}$$

We obtain some valuable insight by applying various iterations to this system of equations with an initial guess consisting of the vectors (or *Fourier modes*)

$$v_j = \sin\left(\frac{jk\pi}{n}\right), \quad 0 \le j \le n, \;\; 1 \le k \le n-1.$$

Recall that j denotes the component (or associated grid point) of the vector $\mathbf{v}$. The integer k now makes its first appearance. It is called the *wavenumber* (or *frequency*) and it indicates the number of half sine waves that constitute $\mathbf{v}$ on the domain of the problem. We use $\mathbf{v}_k$ to designate the entire vector $\mathbf{v}$ with wavenumber k. Figure 2.2 illustrates initial guesses $\mathbf{v}_1$, $\mathbf{v}_3$, and $\mathbf{v}_6$. Notice that small values of k correspond to long, smooth waves, while large values of k correspond to highly oscillatory waves. We now explore how Fourier modes behave under iteration.

We first apply the weighted Jacobi iteration with $\omega = \frac{2}{3}$ to problem (2.3) on a grid with $n = 64$ points. Beginning with initial guesses of $\mathbf{v}_1$, $\mathbf{v}_3$, and $\mathbf{v}_6$, the iteration is applied 100 times. Recall that the error is just $-\mathbf{v}$. Figure 2.3(a) shows a plot of the maximum norm of the error versus the iteration number.

For the moment, only the qualitative behavior of the iteration is important. The error clearly decreases with each relaxation sweep and the rate of decrease is larger for the higher wavenumbers. Figures 2.3(b, c) show analogous plots for the regular and red-black Gauss–Seidel iterations, where we see a similar relationship among the error, the number of iterations, and the wavenumber. (The complete situation is not quite so simple with red-black Gauss–Seidel, as illustrated in Exercise 20.)

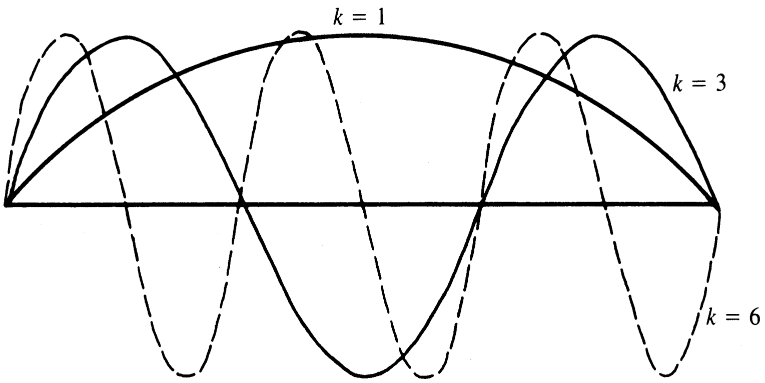

Figure 2.2: *The modes* $v_j = \sin\left(\frac{jk\pi}{n}\right)$, $0 \le j \le n$, *with wavenumbers* $k = 1, 3, 6$. *The* k*th mode consists of* $\frac{k}{2}$ *full sine waves on the interval.*

The experiment of Fig. 2.3(a) is presented in a slightly different light in Fig. 2.4. In this figure, the log of the maximum norm of the error for the weighted Jacobi method is plotted against the iteration number for various wavenumbers. This plot clearly shows a linear decrease in the log of the error norm, indicating that the error itself decreases geometrically with each iteration. If we let $\mathbf{e}^{(0)}$ be the error in the initial guess and $\mathbf{e}^{(m)}$ be the error in the mth iterate, then we might expect to describe the error by a relationship of the form

$$\|\mathbf{e}^{(m)}\|_\infty = c_k^m \|\mathbf{e}^{(0)}\|_\infty,$$

where c_k is a constant that depends on the wavenumber. We will see that the theory confirms this conjecture.

In general, most initial guesses (or, equivalently, most right-side vectors $\mathbf{f}$) would not consist of a single mode. Consider a slightly more realistic situation in which the initial guess (hence, the error) consists of three modes: a low-frequency wave ($k = 1$), a medium-frequency wave ($k = 6$), and a high-frequency wave ($k = 32$) on a grid with $n = 64$ points; it is given by

$$v_j = \frac{1}{3}\left[\sin\left(\frac{j\pi}{n}\right) + \sin\left(\frac{6j\pi}{n}\right) + \sin\left(\frac{32j\pi}{n}\right)\right].$$

Figure 2.5 shows the maximum norm of the error plotted against the number of iterations. The error decreases rapidly within the first five iterations, after which it decreases much more slowly. The initial decrease corresponds to the quick elimination of the high-frequency modes of the error. The slow decrease is due to the presence of persistent low-frequency modes. The important observation is that the standard iterations converge very quickly as long as the error has high-frequency components. However, the slower elimination of the low-frequency components degrades the performance of these methods.

With some experimental evidence in hand, we now turn to a more analytical approach. Each of the methods discussed so far may be represented in the form

$$\mathbf{v}^{(1)} = R\mathbf{v}^{(0)} + \mathbf{g},$$

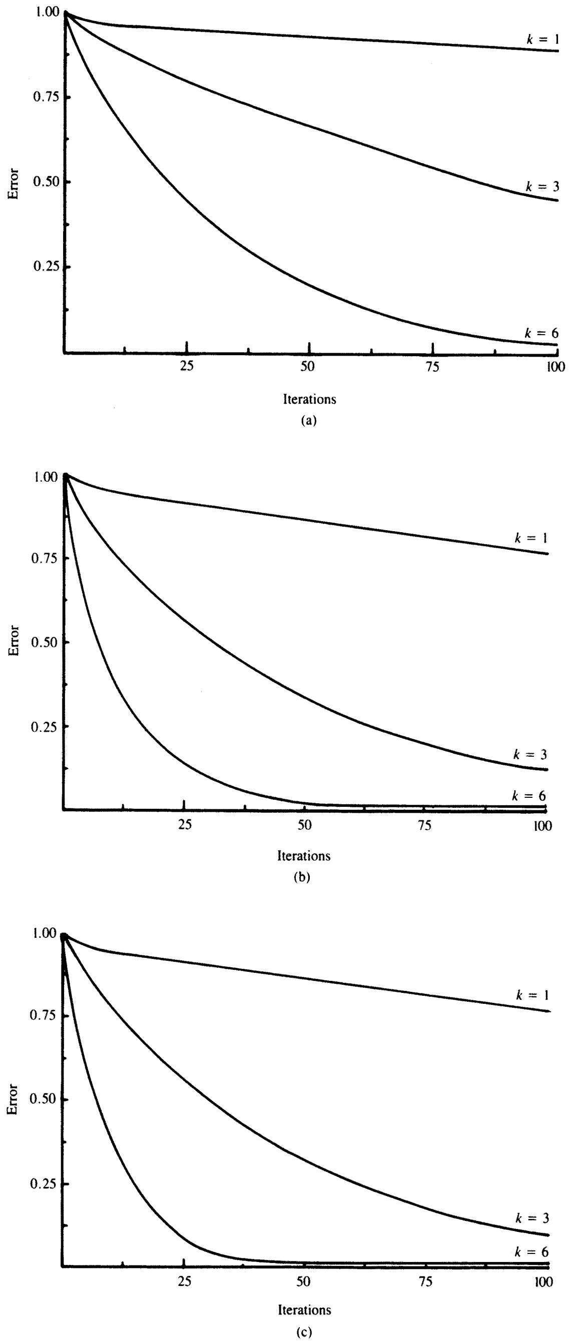

Figure 2.3: (a) *Weighted Jacobi iteration with* $\omega = \frac{2}{3}$, (b) *regular Gauss–Seidel iteration, and* (c) *red-black Gauss–Seidel iteration applied to the one-dimensional model problem with* $n = 64$ *points and with initial guesses* $\mathbf{v}_1$, $\mathbf{v}_3$, *and* $\mathbf{v}_6$. *The maximum norm of the error,* $\|\mathbf{e}\|_\infty$, *is plotted against the iteration number for* 100 *iterations.*

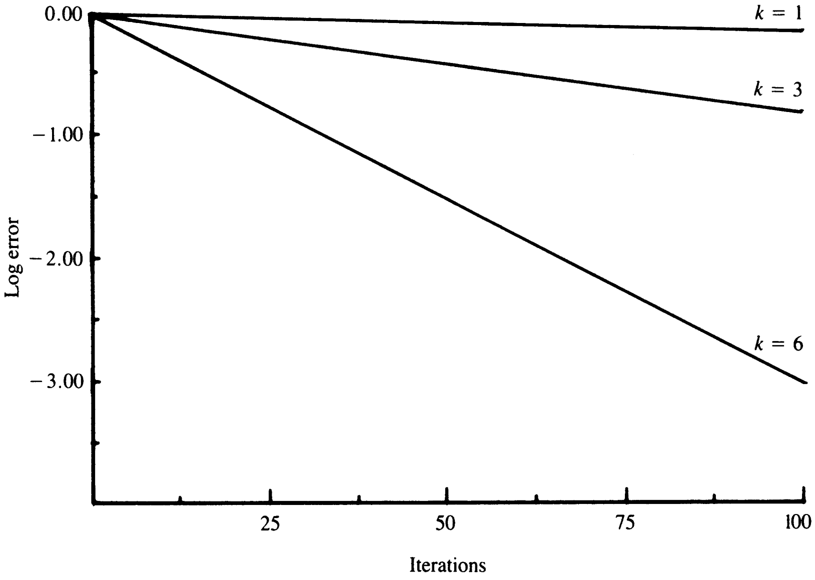

Figure 2.4: *Weighted Jacobi iteration with $\omega = \frac{2}{3}$ applied to the one-dimensional model problem with $n = 64$ points and with initial guesses $\mathbf{v}_1$, $\mathbf{v}_3$, and $\mathbf{v}_6$. The log of $\|\mathbf{e}\|_\infty$ is plotted against the iteration number for 100 iterations.*

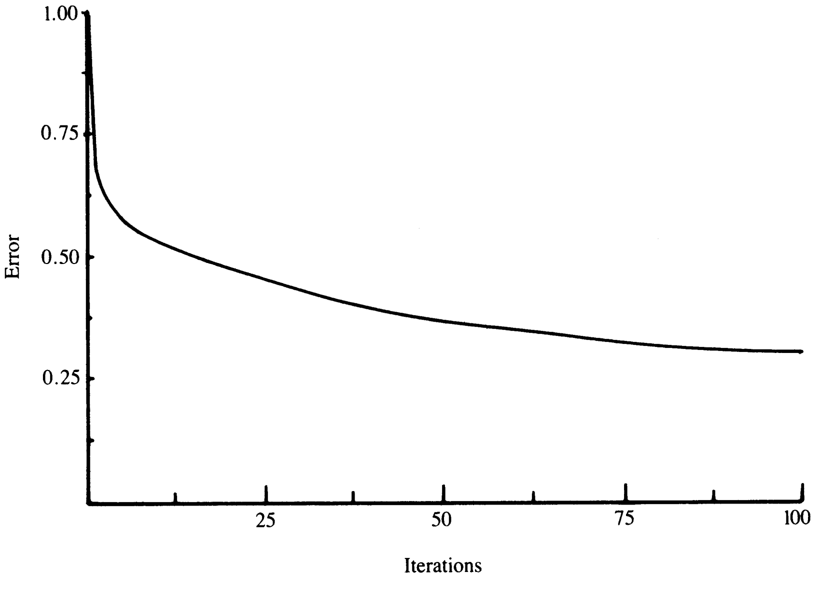

Figure 2.5: *Weighted Jacobi method with $\omega = \frac{2}{3}$ applied to the one-dimensional model problem with $n = 64$ points and an initial guess $(\mathbf{v}_1 + \mathbf{v}_6 + \mathbf{v}_{32})/3$. The maximum norm of the error, $\|\mathbf{e}\|_\infty$, is plotted against the iteration number for 100 iterations.*

where R is one of the iteration matrices derived earlier. Furthermore, all of these methods are designed such that the exact solution, $\mathbf{u}$, is a fixed point of the iteration (Exercise 4). This means that iteration does not change the exact solution:

$$\mathbf{u} = R\mathbf{u} + \mathbf{g}.$$

Subtracting these last two expressions, we find that

$$\mathbf{e}^{(1)} = R\mathbf{e}^{(0)}.$$

Repeating this argument, it follows that after m relaxation sweeps, the error in the mth approximation is given by

$$\mathbf{e}^{(m)} = R^m\mathbf{e}^{(0)}.$$

Matrix Norms. Matrix norms can be defined in terms of the commonly used vector norms. Let A be an $n \times n$ matrix with elements a_{ij}. Consider the vector norm $\|\mathbf{x}\|_p$ defined by

$$\begin{aligned} \|\mathbf{x}\|_p &= \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}, \quad 1 \le p < \infty, \\ \|\mathbf{x}\|_\infty &= \sup_{1\le i\le n} |x_i|. \end{aligned}$$

The matrix norm *induced* by the vector norm $\|\cdot\|_p$ is defined by

$$\|A\|_p = \sup_{\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p}.$$

While not obvious without some computation, this definition leads to the following matrix norms induced by the vector norms $\|\cdot\|_1$, $\|\cdot\|_\infty$, and $\|\cdot\|_2$:

$\|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|$ (maximum column sum),

$\|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$ (maximum row sum),

$\|A\|_2 = \sqrt{\text{spectral radius of } A^TA}$.

Recall that the *spectral radius* of a matrix is given by

$$\rho(A) = \max|\lambda(A)|,$$

where $\lambda(A)$ denotes the eigenvalues of A. For symmetric matrices, the matrix 2-norm is just the spectral radius of A:

$$\|A\|_2 = \sqrt{\rho(A^TA)} = \sqrt{\rho(A^2)} = \rho(A).$$

If we now choose a particular vector norm and its associated matrix norm, it is possible to bound the error after m iterations by

$$\|\mathbf{e}^{(m)}\| \le \|R\|^m \|\mathbf{e}^{(0)}\|.$$

This leads us to conclude that if $\|R\| < 1$, then the error is forced to zero as the iteration proceeds.

It is shown in many standard texts [9, 20, 24, 26] that

$$\lim_{m\to\infty} R^m = 0 \quad \text{if and only if} \quad \rho(R) < 1.$$

Therefore, it follows that the iteration associated with the matrix R converges for all initial guesses if and only if $\rho(R) < 1$.

The spectral radius $\rho(R)$ is also called the *asymptotic convergence factor* when it appears in the context of iterative methods. It has some useful interpretations. First, it is roughly the worst factor by which the error is reduced with each relaxation sweep. By the following argument, it also tells us approximately how many iterations are required to reduce the error by a factor of 10^{-d}. Let m be the smallest integer that satisfies

$$\frac{\|\mathbf{e}^{(m)}\|}{\|\mathbf{e}^{(0)}\|} \le 10^{-d}.$$

This condition will be approximately satisfied if

$$[\rho(R)]^m \le 10^{-d}.$$

Solving for m, we have

$$m \ge -\frac{d}{\log_{10}[\rho(R)]}\,.$$

The quantity $-\log_{10}(\rho(R))$ is called the *asymptotic convergence rate.* Its reciprocal gives the approximate number of iterations required to reduce the error by one decimal digit. We see that as $\rho(R)$ approaches 1, the convergence rate decreases. Small values of $\rho(R)$ (that is, $\rho(R)$ positive and near zero) give a high convergence rate.

We have established the importance of the spectral radius of the iteration matrix in analyzing the convergence properties of relaxation methods. Now it is time to compute some spectral radii. Consider the weighted Jacobi iteration applied to the one-dimensional model problem. Recalling that $R_\omega = (1-\omega)I + \omega R_J$, we have (Exercise 3)

$$R_\omega = I - \frac{\omega}{2}\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & -1 \\ & & & & -1 & 2 \end{bmatrix}.$$

Written in this form, it follows that the eigenvalues of R_ω and A are related by

$$\lambda(R_\omega) = 1 - \frac{\omega}{2}\lambda(A).$$

The problem becomes one of finding the eigenvalues of the original matrix A. This useful exercise (Exercise 8) may be done in several different ways. The result is

Interpreting the Spectral Radius. The spectral radius is considered to be an *asymptotic* measure of convergence because it predicts the worst-case error reduction over many iterations. It can be shown [9, 20] that, in any vector norm,

$$\rho(R) = \lim_{m\to\infty} \|R^m\|^{1/m}.$$

Therefore, in terms of error reduction, we have

$$\rho(R) = \lim_{m\to\infty} \sup_{\mathbf{e}^{(0)}} \left(\frac{\|\mathbf{e}^{(m)}\|}{\|\mathbf{e}^{(0)}\|} \right)^{1/m}.$$

However, the spectral radius does not, in general, predict the behavior of the error norm for a single iteration. For example, consider the matrix

$$R = \begin{pmatrix} 0 & 100 \\ 0 & 0 \end{pmatrix}.$$

Clearly, $\rho(R) = 0$. But if we start with $\mathbf{e}^{(0)} = (0,1)^T$ and compute $\mathbf{e}^{(1)} = R\mathbf{e}^{(0)}$, then the convergence factor is

$$\frac{\|\mathbf{e}^{(1)}\|_2}{\|\mathbf{e}^{(0)}\|_2} = 100.$$

The next iterate achieves the asymptotic estimate, $\rho(R) = 0$, because $\mathbf{e}^{(2)} = 0$. A better worst-case estimate of error reduction for one or a few iterations is given by the matrix norm $\|R\|_2$. For the above example, we have $\|R\|_2 = 100$. The discrepancy between the asymptotic convergence factor, $\rho(R)$, and the worst-case estimate, $\|R\|_2$, disappears when R is symmetric because then $\rho(R) = \|R\|_2$.

that the eigenvalues of A are

$$\lambda_k(A) = 4\sin^2\left(\frac{k\pi}{2n}\right), \qquad 1 \le k \le n-1.$$

Also of interest are the corresponding eigenvectors of A. In all that follows, we let $w_{k,j}$ be the jth component of the kth eigenvector, $\mathbf{w}_k$. The eigenvectors of A are then given by (Exercise 9)

$$w_{k,j} = \sin\left(\frac{jk\pi}{n}\right), \qquad 1 \le k \le n-1, \quad 0 \le j \le n.$$

We see that the eigenvectors of A are simply the Fourier modes discussed earlier.

With these results, we find that the eigenvalues of R_ω are

$$\lambda_k(R_\omega) = 1 - 2\omega\sin^2\left(\frac{k\pi}{2n}\right), \qquad 1 \le k \le n-1,$$

while the eigenvectors of R_ω are the same as the eigenvectors of A (Exercise 10). It is important to note that if $0 < \omega \le 1$, then $|\lambda_k(R_\omega)| < 1$ and the weighted Jacobi iteration converges. We return to these convergence properties in more detail after a small detour.

The eigenvectors of the matrix A are important in much of the following discussion. They correspond very closely to the eigenfunctions of the continuous model problem. Just as we can expand fairly arbitrary functions using this set of eigenfunctions, it is also possible to expand arbitrary vectors in terms of a set of eigenvectors. Let $\mathbf{e}^{(0)}$ be the error in an initial guess used in the weighted Jacobi method. Then it is possible to represent $\mathbf{e}^{(0)}$ using the eigenvectors of A in the form

$$\mathbf{e}^{(0)} = \sum_{k=1}^{n-1} c_k \mathbf{w}_k,$$

where the coefficients $c_k \in \mathbf{R}$ give the “amount” of each mode in the error. We have seen that after m sweeps of the iteration, the error is given by

$$\mathbf{e}^{(m)} = R_\omega^m \mathbf{e}^{(0)}.$$

Using the eigenvector expansion for $\mathbf{e}^{(0)}$, we have

$$\mathbf{e}^{(m)} = R_\omega^m \mathbf{e}^{(0)} = \sum_{k=1}^{n-1} c_k R_\omega^m \mathbf{w}_k = \sum_{k=1}^{n-1} c_k \lambda_k^m(R_\omega)\mathbf{w}_k.$$

The last equality follows because the eigenvectors of A and R_ω are the same; therefore, $R_\omega \mathbf{w}_k = \lambda_k(R_\omega)\mathbf{w}_k$.

This expansion for $\mathbf{e}^{(m)}$ shows that after m iterations, the kth mode of the initial error has been reduced by a factor of $\lambda_k^m(R_\omega)$. It should also be noted that the weighted Jacobi method does not mix modes: when applied to a single mode, the iteration can change the amplitude of that mode, but it cannot convert that mode into different modes. In other words, the Fourier modes are also eigenvectors of the iteration matrix. As we will see, this property is not shared by all stationary iterations.

To develop some familiarity with these Fourier modes, Fig. 2.6 shows them on a grid with $n = 12$ points. Notice that the kth mode consists of $\frac{k}{2}$ full sine waves and has a wavelength of $\ell = \frac{24h}{k} = \frac{2}{k}$ (the entire interval has length 1). The $k = \frac{n}{2}$ mode has a wavelength of $\ell = 4h$ and the $k = n-1$ mode has a wavelength of almost $\ell = 2h$. Waves with wavenumbers greater than n (wavelengths less than $2h$) cannot be represented on the grid. In fact (Exercise 12), through the phenomenon of *aliasing*, a wave with a wavelength less than $2h$ actually appears on the grid with a wavelength greater than $2h$.

At this point, it is important to establish some terminology that is used throughout the remainder of the tutorial. We need some qualitative terms for the various Fourier modes that have been discussed. The modes in the lower half of the spectrum, with wavenumbers in the range $1 \le k < \frac{n}{2}$, are called *low-frequency* or *smooth* modes. The modes in the upper half of the spectrum, with $\frac{n}{2} \le k \le n-1$, are called *high-frequency* or *oscillatory* modes.

Having taken this excursion through Fourier modes, we now return to the analysis of the weighted Jacobi method. We established that the eigenvalues of the iteration matrix are given by

$$\lambda_k(R_\omega) = 1 - 2\omega \sin^2\left(\frac{k\pi}{2n}\right), \qquad 1 \le k \le n-1.$$

What choice of ω gives the best iterative scheme?

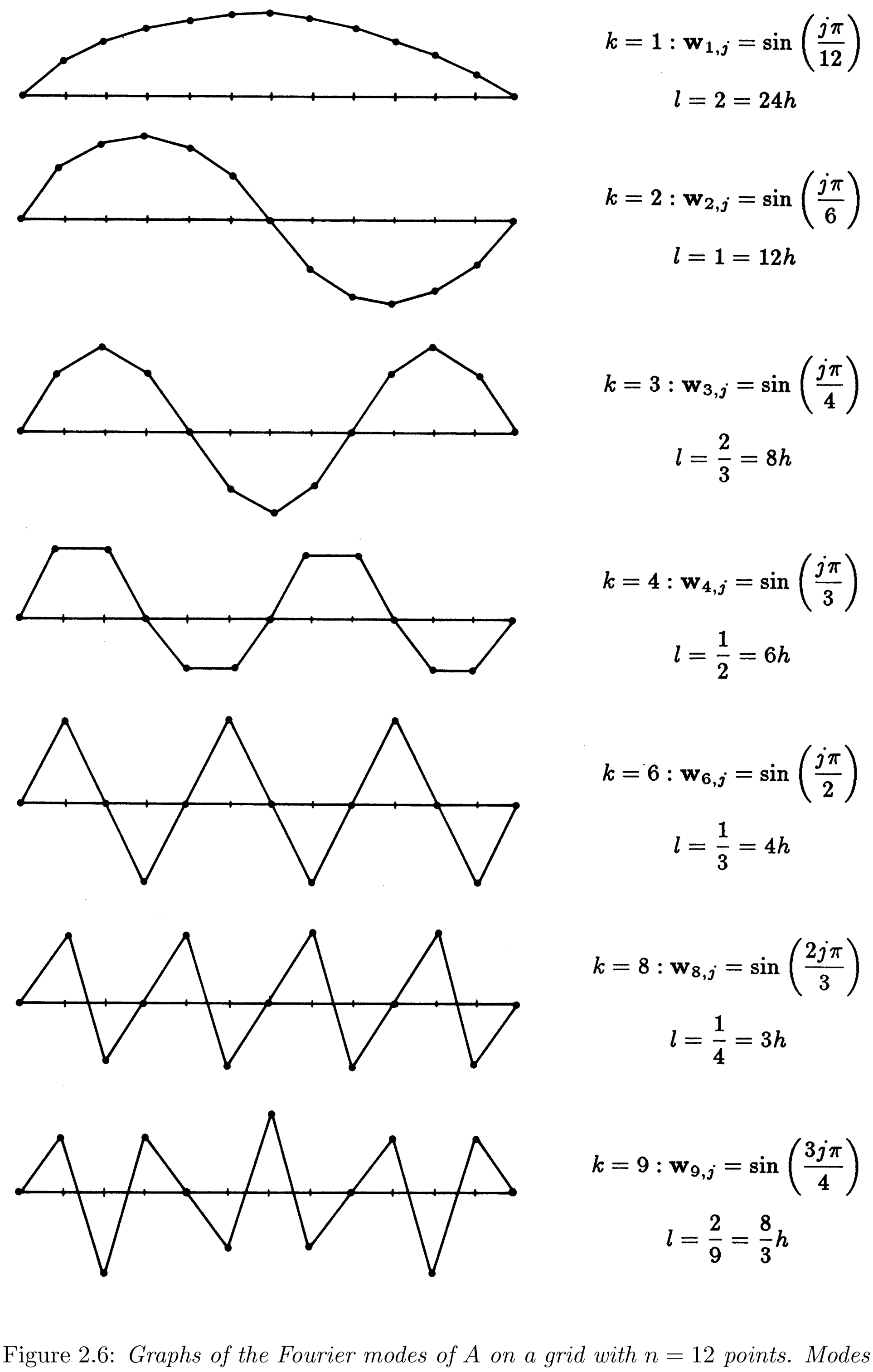

Figure 2.6: *Graphs of the Fourier modes of A on a grid with $n = 12$ points. Modes with wavenumbers $k = 1, 2, 3, 4, 6, 8, 9$ are shown. The wavelength of the kth mode is $\ell = \frac{24h}{k}$.*

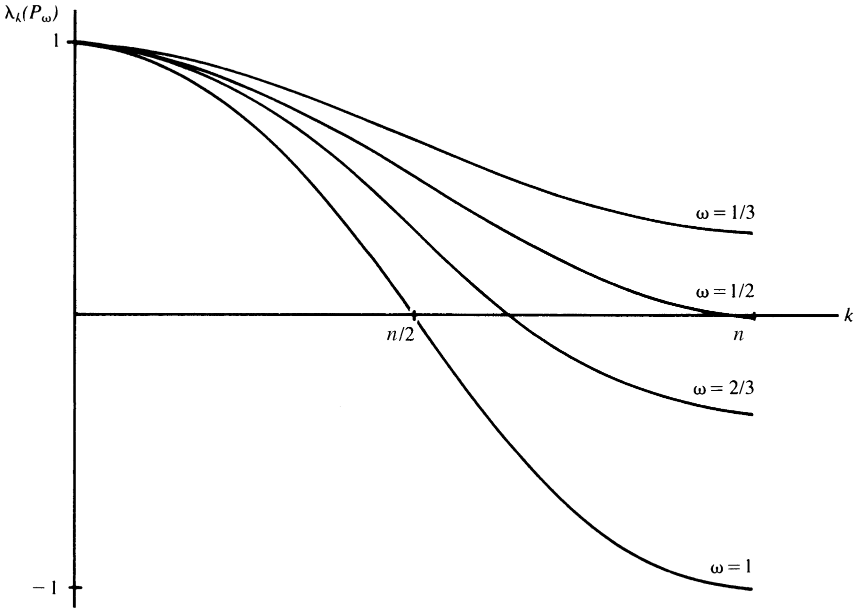

Figure 2.7: *Eigenvalues of the iteration matrix* R_ω *for* $\omega = \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, 1$*. The eigenvalues* $\lambda_k = 1 - 2\omega \sin^2\left(\frac{k\pi}{2n}\right)$ *are plotted as if* k *were a continuous variable on the interval* $0 \le k \le n$*. In fact,* $1 \le k \le n-1$ *takes only integer values.*

Recall that for $0 < \omega \le 1$, we have $|\lambda_k(R_\omega)| < 1$. We would like to find the value of ω that makes $|\lambda_k(R_\omega)|$ as small as possible for all $1 \le k \le n-1$. Figure 2.7 is a plot of the eigenvalues λ_k for four different values of ω. Notice that for all values of ω satisfying $0 < \omega \le 1$,

$$\lambda_1 = 1 - 2\omega \sin^2\left(\frac{\pi}{2n}\right) = 1 - 2\omega \sin^2\left(\frac{\pi h}{2}\right) \approx 1 - \frac{\omega \pi^2 h^2}{2}.$$

This fact implies that λ_1, the eigenvalue associated with the smoothest mode, will always be close to 1. Therefore, no value of ω will reduce the smooth components of the error effectively. Furthermore, the smaller the grid spacing h, the closer λ_1 is to 1. Any attempt to improve the accuracy of the solution (by decreasing the grid spacing) will only worsen the convergence of the smooth components of the error. Most basic relaxation schemes share this ironic limitation.

Having accepted the fact that no value of ω damps the smooth components satisfactorily, we ask what value of ω provides the best damping of the oscillatory components (those with $\frac{n}{2} \le k \le n-1$). We could impose this condition by requiring that

$$\lambda_{n/2}(R_\omega) = -\lambda_n(R_\omega).$$

Solving this equation for ω leads to the optimal value $\omega = \frac{2}{3}$.

We also find (Exercise 13) that with $\omega = \frac{2}{3}$, $|\lambda_k| < \frac{1}{3}$ for all $\frac{n}{2} \le k \le n-1$. This says that the oscillatory components are reduced at least by a factor of three with each relaxation. This damping factor for the oscillatory modes is an important property of any relaxation scheme and is called the *smoothing factor* of the scheme. An important property of the basic relaxation scheme that underlies much of the power of multigrid methods is that the smoothing factor is not only small, but also independent of the grid spacing h.

We now turn to some numerical experiments to illustrate the analytical results that have just been obtained. Once again, the weighted Jacobi method is applied to the one-dimensional model problem $A\mathbf{u} = \mathbf{0}$ on a grid with $n = 64$ points. We use initial guesses (which are also initial errors) consisting of single modes with wavenumbers $1 \leq k \leq n - 1$. Figure 2.8 shows how the method performs in terms of different wavenumbers. Specifically, the wavenumber of the initial error is plotted against the number of iterations required to reduce the norm of the initial error by a factor of 100. This experiment is done for weighting factors of $\omega = 1$ and $\omega = \frac{2}{3}$.

With $\omega = 1$, both the high- and low-frequency components of the error are damped very slowly. Components with wavenumbers near $\frac{n}{2}$ are damped rapidly. This behavior is consistent with the eigenvalue curves of Fig. 2.7. We see a quite different behavior in Fig. 2.8(b) with $\omega = \frac{2}{3}$. Recall that $\omega = \frac{2}{3}$ was chosen to give preferential damping to the oscillatory components. Indeed, the smooth waves are damped very slowly, while the upper half of the spectrum ($k \geq \frac{n}{2}$) shows rapid convergence. Again, this is consistent with Fig. 2.7.

Another perspective on these convergence properties is provided in Figure 2.9. This time the actual approximations are plotted. The weighted Jacobi method with $\omega = \frac{2}{3}$ is applied to the same model problem on a grid with $n = 64$ points. Figure 2.9(a) shows the error with wavenumber $k = 3$ after one relaxation sweep (left plot) and after 10 relaxation sweeps (right plot). This smooth component is damped very slowly. Figure 2.9(b) shows a more oscillatory error ($k = 16$) after one and after 10 iterations. The damping is now much more dramatic. Notice also, as mentioned before, that the weighted Jacobi method preserves modes: once a $k = 3$ mode, always a $k = 3$ mode.

Figure 2.9(c) illustrates the selectivity of the damping property. This experiment uses an initial guess consisting of two modes with $k = 2$ and $k = 16$. After 10 relaxation sweeps, the high-frequency modulation on the long wave has been nearly eliminated. However, the original smooth component persists.

We have belabored the discussion of the weighted Jacobi method because it is easy to analyze and because it shares many properties with other basic relaxation schemes. In much less detail, let us look at the Gauss–Seidel iteration. We can show (Exercise 14) that the Gauss–Seidel iteration matrix for the model problem (matrix A) has eigenvalues

$$\lambda_k(R_G) = \cos^2\left(\frac{k\pi}{n}\right), \qquad 1 \leq k \leq n - 1.$$

These eigenvalues, which are plotted in Fig. 2.10, must be interpreted carefully. We see that when k is close to 1 or n, the corresponding eigenvalues are close to 1 and convergence is slow. However, the eigenvectors of R_G are given by (Exercise 14)

$$w_{k,j} = \left[\cos\left(\frac{k\pi}{n}\right)\right]^j \sin\left(\frac{jk\pi}{n}\right),$$

where $0 \leq j \leq n$ and $1 \leq k \leq n - 1$. These eigenvectors do not coincide with the eigenvectors of A. Therefore, $\lambda_k(R_G)$ gives the convergence rate, not for the kth mode of A, but for the kth eigenvector of R_G.

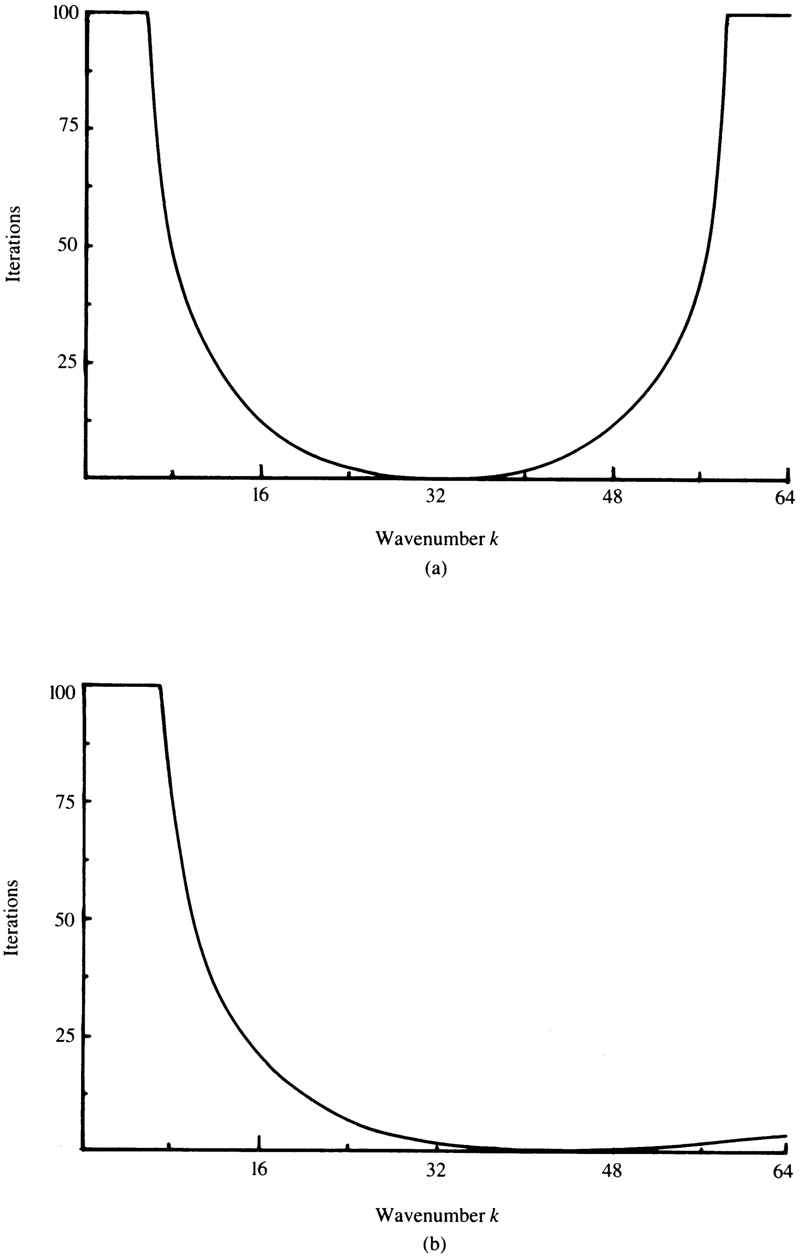

Figure 2.8: *Weighted Jacobi method with* (a) $\omega = 1$ *and* (b) $\omega = \frac{2}{3}$ *applied to the one-dimensional model problem with* $n = 64$ *points. The initial guesses consist of the modes* $\mathbf{w}_k$ *for* $1 \leq k \leq 63$. *The graphs show the number of iterations required to reduce the norm of the initial error by a factor of* 100 *for each* $\mathbf{w}_k$. *Note that for* $\omega = \frac{2}{3}$, *the damping is strongest for the oscillatory modes (*$32 \leq k \leq 63$*).*

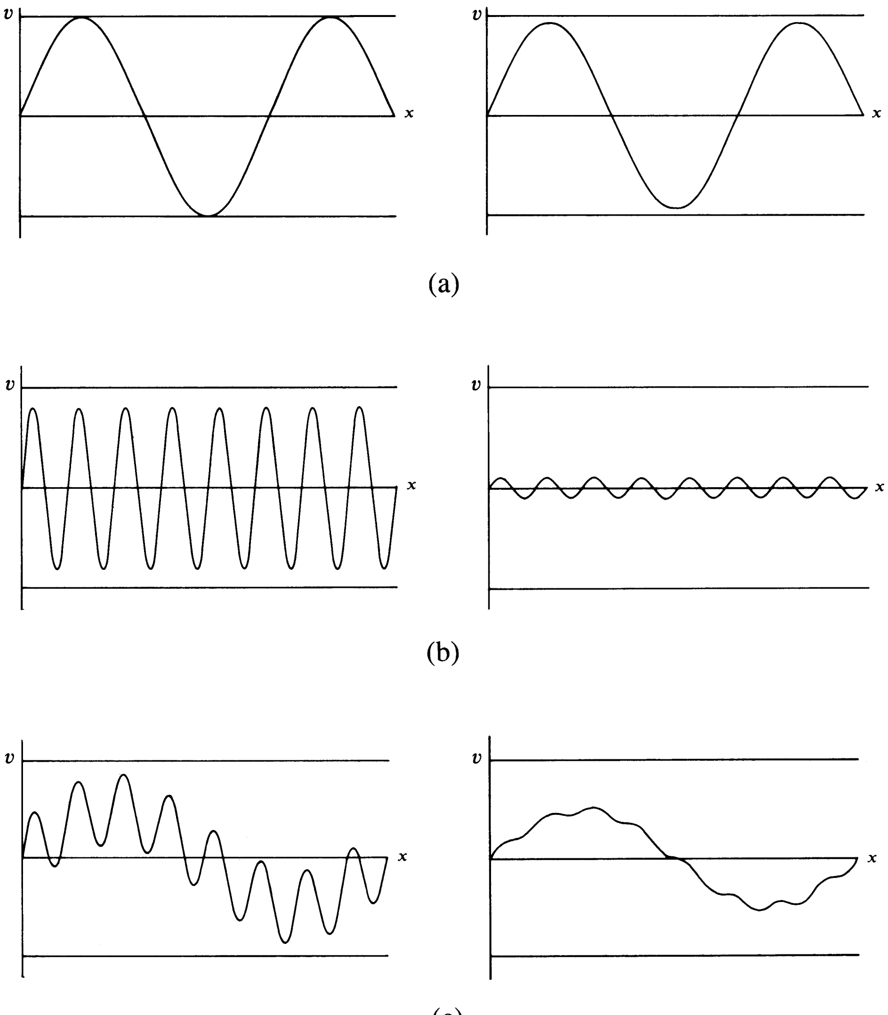

Figure 2.9: *Weighted Jacobi method with* $\omega = \frac{2}{3}$ *applied to the one-dimensional model problem with* $n = 64$ *points and with an initial guess consisting of* (a) $\mathbf{w}_3$, (b) $\mathbf{w}_{16}$, *and* (c) $(\mathbf{w}_2 + \mathbf{w}_{16})/2$. *The figures show the approximation after one iteration (left side) and after* 10 *iterations (right side).*

This distinction is illustrated in Fig. 2.11. As before, the wavenumber k is plotted against the number of iterations required to reduce the norm of the initial error by a factor of 100. In Fig. 2.11(a), the initial guess (and error) consists of the eigenvectors of R_G with wavenumbers $1 \leq k \leq 63$. The graph looks similar to the eigenvalue graph of Fig. 2.10. In Fig. 2.11(b), the initial guess consists of the eigenvectors of the original matrix A. The structure of this graph would be much more difficult to anticipate analytically. We see that when convergence of the Gauss–Seidel method is described in terms of the modes of A, then once again the smooth modes are damped slowly, while the oscillatory modes show rapid decay.

We have looked in detail at the convergence properties of some basic relaxation schemes. The experiments we presented reflect the experience of many practi-

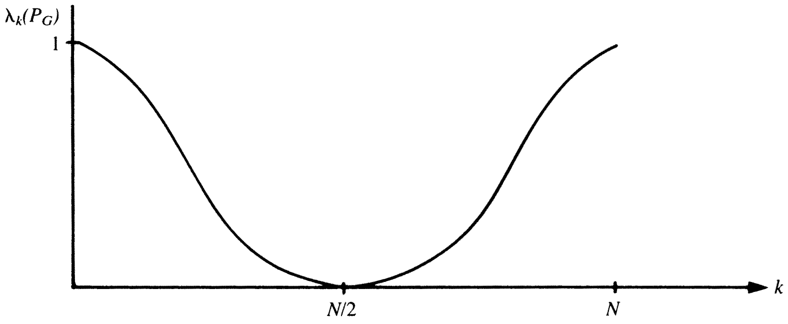

Figure 2.10: *Eigenvalues of the Gauss–Seidel iteration matrix. The eigenvalues* $\lambda_k = \cos^2\left(\frac{k\pi}{n}\right)$ *are plotted as if* k *were a continuous variable on the interval* $0 \leq k \leq n$.

tioners. These schemes work very well for the first several iterations. Inevitably, however, convergence slows and the entire scheme appears to stall. We have found a simple explanation for this phenomenon: the rapid decrease in error during the early iterations is due to the efficient elimination of the oscillatory modes of that error; but once the oscillatory modes have been removed, the iteration is much less effective in reducing the remaining smooth components.

There is also a good physical explanation for why smooth error modes are so resistant to relaxation. Recall from (2.2) that stationary linear iterations can be written in the form

$$\mathbf{v}^{(1)} = \mathbf{v}^{(0)} + B\mathbf{r}^{(0)}.$$

Subtracting this equation from the exact solution $\mathbf{u}$, the error at the next step is

$$\mathbf{e}^{(1)} = \mathbf{e}^{(0)} - B\mathbf{r}^{(0)}.$$

We see that changes in the error are made with *spatially local* corrections expressed through the residual. If the residual is small relative to the error itself, then changes in the error will be correspondingly small. At least for the model problems we have posed, smooth error modes have relatively small residuals (Exercise 19), so the error decreases slowly. Conversely, oscillatory errors tend to have relatively large residuals and the corrections to the error with a single relaxation sweep can be significant.

Many relaxation schemes possess this property of eliminating the oscillatory modes and leaving the smooth modes. This so-called *smoothing property* is a serious limitation of conventional relaxation methods. However, this limitation can be overcome and the remedy is one of the pathways to multigrid.

In one very brief chapter, we have barely touched upon the wealth of lore and theory surrounding iterative methods. The subject constitutes a large and important domain of classical numerical analysis. It is also filled with very elegant mathematics from both linear algebra and analysis. However, esoteric iterative methods are not required for the development of multigrid. The most effective multigrid techniques are usually built upon the simple relaxation schemes presented in this chapter. We now use these few basic schemes and develop them into far more powerful methods.

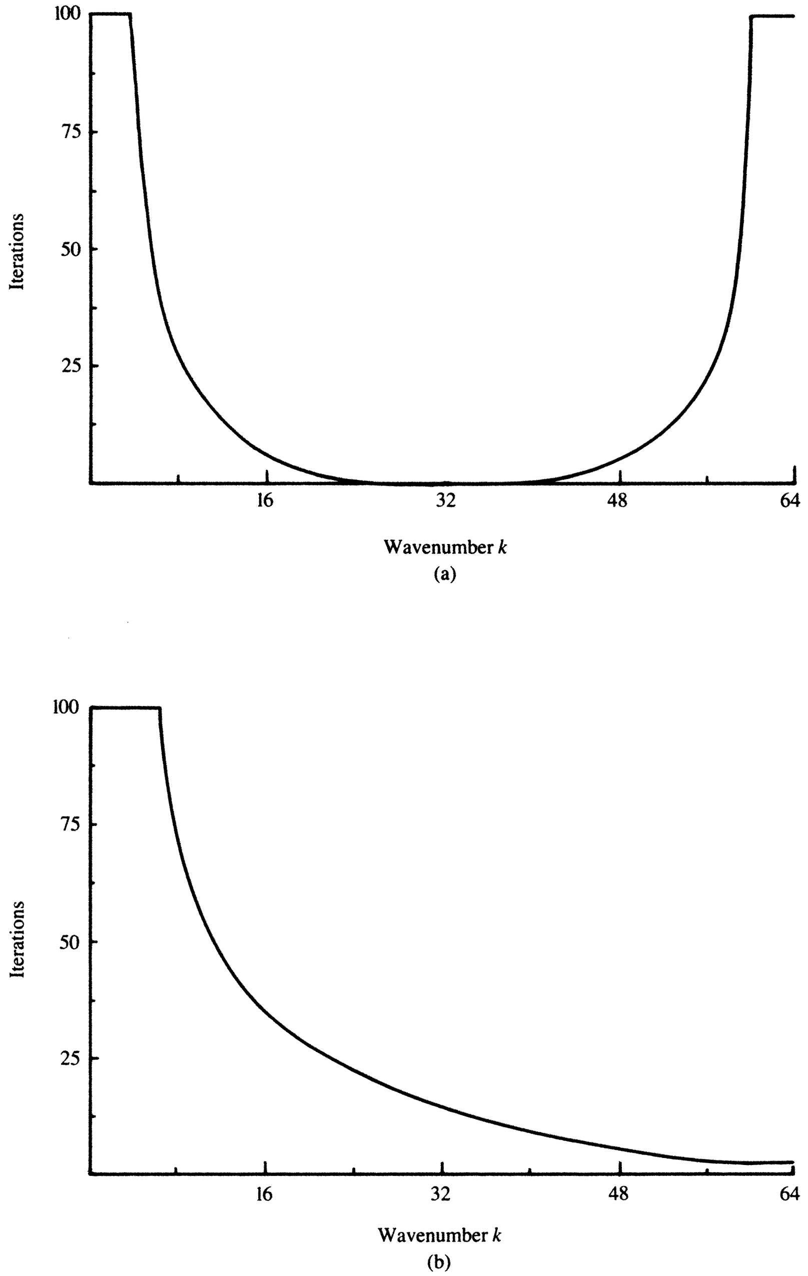

Figure 2.11: *Gauss–Seidel iteration matrix applied to the model problem with* $n = 64$ *points. The initial guesses consist of* (a) *the eigenvectors of the iteration matrix* R_G *with wavenumbers* $1 \leq k \leq 63$ *and* (b) *the eigenvectors of* A *with wavenumbers* $1 \leq k \leq 63$. *The figure shows the number of iterations required to reduce the norm of the initial error by a factor of* 100 *for each initial guess.*

Exercises

1. **Residual vs. error.** Consider the two systems of linear equations given in the box on residuals and errors in this chapter. Make a sketch showing the pair of lines represented by each system. Mark the exact solution $\mathbf{u}$ and the approximation $\mathbf{v}$. Explain why, even though the error is the same in both cases, the residual is small in one case and large in the other.

2. **Residual equation.** Use the definition of the algebraic error and the residual to derive the residual equation $A\mathbf{e} = \mathbf{r}$.

3. **Weighted Jacobi iteration.**

   (a) Starting with the component form of the weighted Jacobi method, show that it can be written in matrix form as $\mathbf{v}^{(1)} = [(1-\omega)I + \omega R_J]\mathbf{v}^{(0)} + \omega D^{-1}\mathbf{f}$.

   (b) Show that the weighted Jacobi method may also be written in the form
   $$\mathbf{v}^{(1)} = R_\omega \mathbf{v}^{(0)} + \omega D^{-1}\mathbf{f}.$$

   (c) Show that the weighted Jacobi iteration may also be expressed in the form
   $$\mathbf{v}^{(1)} = \mathbf{v}^{(0)} + \omega D^{-1}\mathbf{r}^{(0)},$$
   where $\mathbf{r}^{(0)}$ is the residual associated with the approximation $\mathbf{v}^{(0)}$.

   (d) Assume that A is the matrix associated with the model problem. Show that the weighted Jacobi iteration matrix can be expressed as
   $$R_\omega = I - \frac{\omega}{2}A.$$

4. **General stationary linear iteration.** It was shown that a general stationary linear iteration can be expressed in the form
   $$\mathbf{v}^{(1)} = (I - BA)\mathbf{v}^{(0)} + B\mathbf{f} \equiv R\mathbf{v}^{(0)} + B\mathbf{f}.$$

   (a) Show that m sweeps of the iteration has the form
   $$\mathbf{v}^{(1)} = R^m \mathbf{v}^{(0)} + C(\mathbf{f}).$$
   Find an expression for $C(\mathbf{f})$.

   (b) Show that the form of the iteration given above is equivalent to
   $$\mathbf{v}^{(1)} = \mathbf{v}^{(0)} + B\mathbf{r}^{(0)},$$
   where $\mathbf{r}^{(0)}$ is the initial residual. Use this form to argue that the exact solution to the linear system, $\mathbf{u}$, is unchanged by (and is therefore a fixed point of) the iteration.

5. **Interpreting Gauss–Seidel.** Show that the Gauss–Seidel iteration is equivalent to successively setting each component of the residual to zero.

6. **Zero right side.** Argue that in analyzing the error in a stationary linear relaxation scheme applied to $A\mathbf{u} = \mathbf{f}$, it is sufficient to consider $A\mathbf{u} = \mathbf{0}$ with arbitrary initial guesses.

7. **Asymptotic convergence rate.** Explain why the asymptotic convergence rate,
$$-\log_{10}\rho(R),$$
is positive. Which iteration matrix gives a higher asymptotic convergence rate: one with $\rho(R) = 0.1$ or one with $\rho(R) = 0.9$? Explain.

8. **Eigenvalues of the model problem.** Compute the eigenvalues of the matrix A of the one-dimensional model problem. (Hint: Write out a typical equation of the system $A\mathbf{w} = \lambda\mathbf{w}$ with $w_0 = w_n = 0$. Notice that vectors of the form $w_j = \sin\left(\frac{jk\pi}{n}\right)$, $1 \le k \le n-1$, $0 \le j \le n$, satisfy the boundary conditions.) How many distinct eigenvalues are there? Compute λ_1, λ_2, λ_{n-2}, λ_{n-1} when $n = 32$.

9. **Eigenvectors of the model problem.** Using the results of the previous problem, find the eigenvectors of the one-dimensional model problem matrix A.

10. **Jacobi eigenvalues and eigenvectors.** Find the eigenvalues of the weighted Jacobi iteration matrix when it is applied to the one-dimensional model problem matrix A. Verify that the eigenvectors of R_ω are the same as the eigenvectors of A.

11. **Fourier modes.** Consider the interval $0 \le x \le 1$ with grid points $x_j = \frac{j}{n}$, $0 \le j \le n$. Show that the kth Fourier mode $w_{k,j} = \sin\left(\frac{jk\pi}{n}\right)$ has wavelength $\ell = \frac{2}{k}$. Which mode has wavelength $\ell = 8h$? Which mode has wavelength $\ell = \frac{1}{4}$?

12. **Aliasing.** On a grid with $n-1$ interior points, show that the mode $w_{k,j} = \sin\left(\frac{jk\pi}{n}\right)$ with $n < k < 2n$ is actually represented as the mode $\mathbf{w}_{k'}$ where $k' = 2n - k$. How is the mode with wavenumber $k = \frac{3n}{2}$ represented on the grid? How is the mode with wavelength $l = \frac{4h}{3}$ represented on the grid? Make sketches for these two examples.

13. **Optimal Jacobi.** Show that when the weighted Jacobi method is used with $\omega = \frac{2}{3}$, the smoothing factor is $\frac{1}{3}$. Show that if ω is chosen to damp the smooth modes effectively, then the oscillatory modes are actually amplified.

14. **Gauss–Seidel eigenvalues and eigenvectors.**

    (a) Show that the eigenvalue problem for the Gauss–Seidel iteration matrix, $R_G\mathbf{w} = \lambda\mathbf{w}$, may be expressed in the form $U\mathbf{w} = (D-L)\lambda I\mathbf{w}$, where U, L, D are defined in the text.

    (b) Write out the equations of this system and note the boundary condition $w_0 = w_n = 0$. Look for solutions of this system of equations of the form $w_j = \mu^j$, where $\mu \in \mathbf{C}$ must be determined. Show that the boundary conditions can be satisfied only if $\lambda = \lambda_k = \cos^2\left(\frac{k\pi}{n}\right)$, $1 \le k \le n-1$.

    (c) Show that the eigenvector associated with λ_k is $w_{k,j} = \cos(\frac{k\pi}{n})\sin\left(\frac{jk\pi}{n}\right)$.

15. **Richardson iteration.**

    (a) Recall that for real vectors $\mathbf{u}$, $\mathbf{v}$, the inner product is given by $(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T\mathbf{v}$ and $\|\mathbf{u}\|_2^2 = (\mathbf{u}, \mathbf{u})$. Furthermore, if A is symmetric positive definite, then $\|A\|_2 = \rho(A)$, the spectral radius of A. Richardson's iteration is given by

$$\mathbf{v}^{(1)} = \mathbf{v}^{(0)} + \frac{s}{\|A\|_2}\mathbf{r}^{(0)} \quad \text{for } 0 < s < 2,$$

where $\mathbf{r}^{(0)} = \mathbf{f} - A\mathbf{v}^{(0)}$ is the residual. Show that when A has a constant diagonal, this method reduces to the weighted Jacobi method.

(b) Show that the error after one sweep of Richardson's method is governed by

$$\|\mathbf{e}^{(1)}\|_2^2 \leq \left[1 - \frac{s(2-s)(A\mathbf{e}^{(0)}, \mathbf{e}^{(0)})}{\|A\|_2 \ (\mathbf{e}^{(0)}, \mathbf{e}^{(0)})}\right] \|\mathbf{e}^{(0)}\|_2^2.$$

(c) If the eigenvalues of A are ordered $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_n$ and the smallest eigenvalues correspond to the smooth modes, show that Richardson's method has the smoothing property. (Use the fact that the eigenvalues are given by the Rayleigh quotients of the eigenvectors, $\lambda_k = (A\mathbf{w}_k, \mathbf{w}_k)/(\mathbf{w}_k, \mathbf{w}_k)$, where $\mathbf{w}_k$ is the eigenvector associated with λ_k.)

16. **Properties of Gauss–Seidel.** Assume A is symmetric, positive definite.

(a) Show that the jth step of a *single* sweep of the Gauss–Seidel method applied to $A\mathbf{u} = \mathbf{f}$ may be expressed as

$$v_j \leftarrow v_j + \frac{r_j}{a_{jj}}.$$

(b) Show that the jth step of a *single* sweep of the Gauss–Seidel method can be expressed in vector form as

$$\mathbf{v} \leftarrow \mathbf{v} + \frac{(\mathbf{r}, \hat{\mathbf{e}}_j)}{(A\hat{\mathbf{e}}_j, \hat{\mathbf{e}}_j)}\,\hat{\mathbf{e}}_j, \qquad 1 \leq j \leq n,$$

where $\hat{\mathbf{e}}_j$ is the jth unit vector.

(c) Show that each sweep of Gauss–Seidel decreases the quantity $(A\mathbf{e}, \mathbf{e})$, where $\mathbf{e} = \mathbf{u} - \mathbf{v}$.

(d) Show that Gauss–Seidel is optimal in the sense that the quantity $\|\mathbf{e} - s\hat{\mathbf{e}}_j\|_A$ is minimized for each $1 \leq j \leq n$ when $s = (\mathbf{r}, \hat{\mathbf{e}}_j)/(A\hat{\mathbf{e}}_j, \hat{\mathbf{e}}_j)$, which is precisely a Gauss–Seidel step.

17. **Matrix 2-norm**. Show that the matrix 2-norm is given by

$$\|A\|_2 = \sqrt{\rho(A^T A)}.$$

Use the definition of matrix norm and the relations $\|\mathbf{x}\|_2^2 = (\mathbf{x}, \mathbf{x})$ and $\|A\mathbf{x}\|_2^2 = (A\mathbf{x}, A\mathbf{x})$.

18. **Error and residual norms.** The *condition number* of a matrix, $\text{cond}(A) = \|A\|_2\|A^{-1}\|_2$, gives an idea of how well the residual measures the error. In the following exercise, use the property of matrix and vector norms that $\|Ax\| \leq \|A\|\|x\|$.

(a) Begin with the relations $Ae = r$ and $A^{-1}f = u$. Taking norms and combining terms, show that

$$\frac{\|r\|_2}{\|f\|_2} \leq \mathrm{cond}(A)\frac{\|e\|_2}{\|u\|_2}.$$

Knowing that this bound is sharp (that is, equality can be achieved), interpret this inequality in the case that the condition number is large.

(b) Now begin with the relations $Au = f$ and $A^{-1}r = e$. Taking norms and combining terms, show that

$$\frac{\|e\|_2}{\|u\|_2} \leq \mathrm{cond}(A)\frac{\|r\|_2}{\|f\|_2}.$$

Knowing that this bound is sharp (that is, equality can be achieved), interpret this inequality in the case that the condition number is large.

(c) Combine the above bounds to form the following relations:

$$\frac{1}{\mathrm{cond}(A)}\frac{\|r\|_2}{\|f\|_2} \leq \frac{\|e\|_2}{\|u\|_2} \leq \mathrm{cond}(A)\frac{\|r\|_2}{\|f\|_2}.$$

19. Residuals of smooth errors. Consider the residual equation, $A\mathbf{e} = \mathbf{r}$, at a single point, where A is the matrix for the model problem in either one or two dimensions. Show that if $\mathbf{e}$ is smooth (for example, nearly constant), then $\mathbf{r}$ is small relative to $\|A\|\|\mathbf{e}\|$. Conversely, show that if $\mathbf{e}$ is oscillatory, then $\mathbf{r}$ is relatively large.

20. Numerical experiments. Write a short program that performs weighted Jacobi (with variable ω), Gauss–Seidel, and red-black Gauss–Seidel for the one-dimensional model problem. First reproduce the experiments shown in Fig. 2.3. Then experiment with initial guesses with different wavenumbers. Describe how each method performs as the wavenumbers increase and approach n.