Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Chapter 13
Linear Iterative Methods

In this chapter we consider the class of iterative methods known as linear methods, concentrating primarily on the class of methods related to successive overrelaxation. These methods are relatively easy to implement and require minimal computer storage and, for these reasons, are very widely used in the numerical solution of elliptic equations.

13.1 Solving Finite Difference Schemes for Laplace's Equation in a Rectangle

We begin by considering methods for solving Laplace's equation (12.1.2) in a rectangular domain. The basic method can be extended to solve general elliptic equations such as (12.1.6) on general regions, as discussed in Section 12.7.

Consider Laplace's equation (12.1.2) on the unit square with Dirichlet boundary conditions (12.1.3). For the finite difference scheme we use the standard second-order accurate five-point Laplacian with equal grid spacing in the x and y directions. This has the finite difference formula

$$v_{\ell+1,m} + v_{\ell-1,m} + v_{\ell,m+1} + v_{\ell,m-1} - 4v_{\ell,m} = 0 \tag{13.1.1}$$

for all interior points (x_ℓ, y_m). For the Dirichlet boundary condition (12.1.3) we assume that the values of $v_{\ell,m}$ in (13.1.1) are given when (x_ℓ, y_m) is a boundary point. The Neumann boundary condition is considered in Section 13.7.

Equations (13.1.1) comprise a system of linear equations for the interior values of $v_{\ell,m}$ with the boundary $v_{\ell,m}$ values prescribed. These equations can be written in the standard matrix notation

$$Ax = b, \tag{13.1.2}$$

where the vector x consists of the interior values $v_{\ell,m}$ and b is composed from the values of $v_{\ell,m}$ on the boundary, i.e., the known values.

We could solve (13.1.2) by standard methods for systems of linear equations, such as Gaussian elimination. However, the matrix A in (13.1.2) is a very sparse matrix and is often quite large. For example, if the grid spacing in the unit square is N^{-1}, then A is an $(N-1)^2 \times (N-1)^2$ matrix, and each row contains at most five nonzero elements. If N is taken to be about 40, then only about 0.3% of the elements are nonzero. Gaussian elimination is not efficient for such sparse matrices, and so direct methods such as Gaussian elimination are not often used to solve (13.1.1). Instead, iterative methods are usually employed. Because matrix A has a well-defined structure, due to the finite difference

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

scheme, using a good iterative method is usually more efficient than the use of general sparse matrix methods for Gaussian elimination.

The Jacobi Method

The first iterative method we will consider is the Jacobi algorithm. It is given by the formula

$$v_{\ell,m}^{k+1} = \tfrac{1}{4}\left(v_{\ell+1,m}^{k} + v_{\ell-1,m}^{k} + v_{\ell,m+1}^{k} + v_{\ell,m-1}^{k}\right) \tag{13.1.3}$$

for all interior points. This formula describes how we proceed from an initial approximation $v_{\ell,m}^{0}$ to successive approximations $v_{\ell,m}^{k}$. Given the values of $v_{\ell,m}^{k}$ for all grid points, equation (13.1.3) shows how to compute $v_{\ell,m}^{k+1}$ at each interior grid point. Having computed v^{k+1} for all the grid points, the iterative process can be continued to compute v^{k+2}, and so on. Of course, throughout the computation the values of $v_{\ell,m}^{k}$ on the boundary all remain at their prescribed values.

The Jacobi algorithm (13.1.3) converges as k increases, and we stop the iterations when some criterion is satisfied. For example, one criterion would be to stop when the maximum value of $|v_{\ell,m}^{k+1} - v_{\ell,m}^{k}|$ taken over all values of (ℓ, m) is less than some prescribed tolerance.

The Gauss–Seidel Method

The Jacobi algorithm has been described as the slowest of all converging methods; it certainly is not hard to improve on it. A method that converges twice as fast as (13.1.3) is the Gauss–Seidel algorithm, given by

$$v_{\ell,m}^{k+1} = \tfrac{1}{4}\left(v_{\ell+1,m}^{k} + v_{\ell-1,m}^{k+1} + v_{\ell,m+1}^{k} + v_{\ell,m-1}^{k+1}\right). \tag{13.1.4}$$

In this formula we see that if we proceed through the grid of points in the natural order, then we do not need to keep two copies of the solution, one for the "old" values at iteration k and another for "new" values at iteration $k+1$. In (13.1.4) we can use *immediate replacement*; i.e., when $v_{\ell,m}^{k+1}$ is computed it can be stored in the location where $v_{\ell,m}^{k}$ was stored. Thus (13.1.4) uses less storage than (13.1.3) and, as shown in Section 13.3, it is twice as fast.

The natural order of progressing through the grid points is also called the *lexicographic order*. It is the order we obtain in programming using two nested loops, the inner loop being on ℓ and the outer loop being on m.

The SOR Method

A method that improves on (13.1.4) is successive overrelaxation (SOR), given by

$$v_{\ell,m}^{k+1} = v_{\ell,m}^{k} + \omega\left[\tfrac{1}{4}(v_{\ell+1,m}^{k} + v_{\ell-1,m}^{k+1} + v_{\ell,m+1}^{k} + v_{\ell,m-1}^{k+1}) - v_{\ell,m}^{k}\right]. \tag{13.1.5}$$

If the parameter ω is chosen properly, then (13.1.5) can be very much faster than (13.1.4). Notice that when ω is equal to 1, then SOR reduces to the Gauss–Seidel algorithm. SOR also uses immediate replacement.

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

In the next sections we analyze each of the preceding methods to determine their relative rates of convergence. We also present other versions of SOR.

Analysis of General Linear Iterative Methods

There is an extensive literature on iterative methods for solving linear systems of equations, and we give only an introduction to these methods. More exhaustive discussions are contained in the books by Young [73], Varga [65], Wachpress [67], and Hageman and Young [29]. The Jacobi, Gauss–Seidel, and SOR methods are particular cases of the general class of methods called *linear iterative methods.* The general linear iterative method for solving a linear system

$$Ax = b \tag{13.1.6}$$

involves decomposing the matrix A by writing it as

$$A = B - C \tag{13.1.7}$$

and then iteratively solving the system of equations

$$Bx^{k+1} = Cx^k + b. \tag{13.1.8}$$

Of course, we wish to choose B so that (13.1.8) can be easily solved. As we will show, the Jacobi, Gauss–Seidel, and SOR methods are different ways of splitting the linear system for the five-point Laplacian. Since the exact solution satisfies (13.1.6), we obtain from (13.1.8), the equation for the error,

$$Be^{k+1} = Ce^k$$

or, equivalently,

$$e^{k+1} = B^{-1}Ce^k. \tag{13.1.9}$$

The matrix $B^{-1}C$ is called the *iteration matrix* for the algorithm.

A necessary and sufficient condition for the error given by (13.1.9) to converge to zero is that all the eigenvalues of $B^{-1}C$ are less than 1 in magnitude. For a matrix M, its spectral radius $\rho(M)$ is defined by

$$\rho(M) = \max_i |\lambda_i|,$$

where the λ_i are the eigenvalues of M; see Appendix A. Thus (13.1.8) is a convergent method if and only if

$$\rho\left(B^{-1}C\right) < 1.$$

The quantity $\rho\left(B^{-1}C\right)$ is a measure of the error reduction per step of the iteration. Furthermore, the speed of convergence of the method is dependent on the size of $\rho(B^{-1}C)$.

If we have two different splittings of A,

$$\begin{aligned} A &= B_1 - C_1 \\ &= B_2 - C_2 \end{aligned}$$

and

$$\rho\left(B_2^{-1}C_2\right) < \rho\left(B_1^{-1}C_1\right),$$

then the second method, with the smaller spectral radius, converges faster than does the first.

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Exercises

13.1.1. For a linear system of the form $(A_1 + A_2)x = b$, consider the iterative method

$$\begin{aligned}(I + \mu A_1)\tilde{x} &= (I - \mu A_2)x^k + \mu b,\\ (I + \mu A_2)x^{k+1} &= (I - \mu A_1)\tilde{x} + \mu b,\end{aligned} \tag{13.1.10}$$

where μ is a parameter. Show that this iterative method can be put in the form (13.1.8) and determine the iteration matrix for the method. (This method is based on the ADI method discussed in Section 7.3.)

13.1.2. Show for the system

$$\begin{aligned}x_j - x_{j+1} &= b_j \qquad \text{for } j = 1, \ldots, K-1,\\ x_K &= b_K\end{aligned}$$

that the iterative method

$$\begin{aligned}x_j^{k+1} &= x_{j+1}^k + b_j \qquad \text{for } j = 1, \ldots, K-1,\\ x_K^{k+1} &= b_K\end{aligned}$$

converges in K steps. Show also that $\rho(B^{-1}C)$ is zero.

13.1.3. Prove that a linear iterative method converges in a finite number of steps if and only if $\rho(B^{-1}C) = 0$.

13.2 Eigenvalues of the Discrete Laplacian

In the analysis of the numerical methods introduced in the previous section we will require the eigenvalues of the discrete Laplacian operator. In this section we will derive formulas for these eigenvalues.

The equation for eigenvalues of the discrete Laplacian is

$$\nabla^2 v_{\ell,m} = -\lambda v_{\ell,m},$$

where $v_{\ell,m}$ is a grid function that is identically zero on the boundary of the region, but is not identically zero. We will determine the eigenfunctions and eigenvectors for a rectangular grid for a region with $0 < x < X$ and $0 < y < Y$. We have $\Delta x = X/L$, $\Delta y = Y/M$, and

$$\frac{v_{\ell-1,m} - 2v_{\ell,m} + v_{\ell+1,m}}{\Delta x^2} + \frac{v_{\ell,m-1} - 2v_{\ell,m} + v_{\ell,m+1}}{\Delta y^2} = -\lambda v_{\ell,m} \tag{13.2.1}$$

for all interior points and $v_{\ell,m}$ equal to zero on the boundaries.

It is important to make a distinction between the eigenvector $\bar{v}$, which has unknowns corresponding to the $(L-1)(M-1)$ interior grid points, and the grid function v, which

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

has $(L+1)(M+1)$ values corresponding to both the interior and boundary points. Because we specify that the boundary values of v are zero, we can write the simple formula (13.2.1). The equations for $\bar{v}$ are different than (13.2.1) if (ℓ, m) is next to a boundary, in which case at least one of the terms on the left-hand side of (13.2.1) would not be present.

We begin by looking for solutions of the form

$$v_{\ell,m} = A(\ell)B(m), \tag{13.2.2}$$

where $A(\cdot)$ and $B(\cdot)$ are functions of one integer variable. (Note that it is not clear a priori that we can obtain such solutions.) By substituting the relation (13.2.2) in (13.2.1) and then dividing by $A(\ell)B(m)$, we obtain the equation

$$\frac{A(\ell-1)-2A(\ell)+A(\ell+1)}{\Delta x^2 A(\ell)} + \frac{B(m-1)-2B(m)+B(m+1)}{\Delta y^2 B(m)} = -\lambda.$$

In this relation we see that we have an expression that depends on ℓ and one that depends on m and their sum is a value independent of both ℓ and m. This can only occur if both of these expressions are actually constant. That is, we have

$$\begin{aligned} A(\ell-1)-2A(\ell)+A(\ell+1) &= -2(1-\alpha)A(\ell), \\ B(m-1)-2B(m)+B(m+1) &= -2(1-\beta)B(m) \end{aligned} \tag{13.2.3}$$

for some complex values α and β related by

$$\lambda = 2\frac{1-\alpha}{\Delta x^2} + 2\frac{1-\beta}{\Delta y^2}.$$

Since the equation for $B(\cdot)$ is similar to that of $A(\cdot)$, we consider only the equation for $A(\cdot)$. To solve the equation for $A(\cdot)$, a recurrence relation, we substitute

$$A(\ell) = \zeta^\ell$$

in the first equation in (13.2.3). We obtain the quadratic equation

$$\zeta^2 - 2\alpha\zeta + 1 = 0$$

for the two values of ζ. The two roots are

$$\zeta_\pm = \alpha \pm \sqrt{\alpha^2 - 1}.$$

Note that $\zeta_- = 1/\zeta_+$. Thus the equation for $A(\cdot)$ is of the form

$$A(\ell) = A_+\zeta_+^\ell + A_-\zeta_-^\ell$$

for some constants A_+ and A_-.

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

To determine A_+ and A_- and also α, we consider the boundary conditions for $A(\cdot)$. These are

$$A(0) = 0 \quad \text{and} \quad A(L) = 0.$$

The condition $A(0) = 0$ is satisfied if $A_+ + A_- = 0$, so

$$A(\ell) = A_+(\zeta_+^\ell - \zeta_-^\ell). \tag{13.2.4}$$

Note that we cannot determine a value for A_+ since the equation for $A(\cdot)$ in (13.2.3) is a homogeneous equation.

The boundary condition $A(L) = 0$ is equivalent to

$$A(L) = A_+(\zeta_+^L - \zeta_-^L) = 0$$

or

$$\left(\frac{\zeta_+}{\zeta_-}\right)^L = 1.$$

Also, since $\zeta_- = 1/\zeta_+$, we have

$$\zeta_+^{2L} = 1.$$

Thus ζ_+ (and ζ_-) is a $2L$th root of unity, i.e.,

$$\zeta_+ = e^{i\pi a/L}, \tag{13.2.5}$$

for some integer a ranging from 0 to $2L - 1$. Since $\zeta_- = 1/\zeta_+$ we can restrict a so that $0 < a < L$ and (13.2.4) gives all of the $L - 1$ nontrivial solutions of the equation for $A(\cdot)$ in (13.2.3).

Moreover,

$$\alpha = \frac{1 + \zeta_+^2}{2\zeta_+} = \cos\frac{\pi a}{L} \quad \text{for some integer } a, \quad 0 < a < L.$$

Similarly,

$$\beta = \cos\frac{\pi b}{M} \quad \text{for some integer } b, \quad 0 < b < M.$$

From equation (13.2.3), we have that the eigenvalue corresponding to the pair of integers (a, b) is

$$\lambda^{a,b} = 2\frac{1 - \cos\frac{\pi a}{L}}{\Delta x^2} + 2\frac{1 - \cos\frac{\pi b}{M}}{\Delta y^2} = 4\frac{\sin\frac{\pi a}{2L}}{\Delta x^2} + 4\frac{\sin\frac{\pi b}{2M}}{\Delta y^2} \tag{13.2.6}$$

for integers (a, b) with $0 < a < L$ and $0 < b < M$. Moreover, also from (13.2.2) and (13.2.4), we obtain that the corresponding eigenvector is given by

$$\bar{v}_{\ell,m}^{a,b} = \sin\left(\frac{a\ell\pi}{L}\right) \sin\left(\frac{bm\pi}{M}\right). \tag{13.2.7}$$

So there are $(L - 1)(M - 1)$ eigenvalues, and each corresponds to a distinct eigenvector. This shows that the discrete Laplacian has a complete set of eigenvalues and eigenvectors given by (13.2.6) and (13.2.7), respectively.

13.3 Analysis of the Jacobi and Gauss–Seidel Methods

In this section we analyze the Jacobi and Gauss–Seidel methods for the five-point Laplacian. For simplicity of exposition, we restrict to a square with the same spacing in both directions with $\Delta x = \Delta y = h$ and N points in each direction.

To analyze the Jacobi and Gauss–Seidel methods, we rewrite (13.1.1) as

$$v_{\ell,m} - \tfrac{1}{4}v_{\ell-1,m} - \tfrac{1}{4}v_{\ell,m-1} - \tfrac{1}{4}v_{\ell+1,m} - \tfrac{1}{4}v_{\ell,m+1} = 0 \tag{13.3.1}$$

for all interior points. If this were written in the form (13.1.2), then all values of $v_{\ell\pm1,m}$ and $v_{\ell,m\pm1}$ that correspond to boundary points would have to be placed on the right-hand side of the equation. For example, if $(\ell, m-1)$ is a boundary grid point, then instead of (13.3.1) we have

$$v_{\ell,m} - \tfrac{1}{4}v_{\ell-1,m} - \tfrac{1}{4}v_{\ell+1,m} - \tfrac{1}{4}v_{\ell,m+1} = \tfrac{1}{4}v_{\ell,m-1}.$$

Using the natural ordering of the grid points, we can write (13.3.1) as

$$Ax = b$$

with

$$A = I - L - U,$$

where L is a lower triangular matrix and U is an upper triangular matrix.

It is important to realize that the vector x is indexed with pairs of indices corresponding to the grid points $v_{\ell,m}$, and the matrix A is indexed with pairs of pairs. In particular,

$$\begin{aligned} A_{(\ell,m),(\ell,m)} &= 1, & A_{(\ell,m),(\ell+1,m)} &= -\tfrac{1}{4}, & A_{(\ell,m),(\ell-1,m)} &= -\tfrac{1}{4}, \\ A_{(\ell,m),(\ell,m+1)} &= -\tfrac{1}{4}, & A_{(\ell,m),(\ell,m-1)} &= -\tfrac{1}{4} \end{aligned}$$

when these elements are defined. All other elements are 0. If the grid spacing is given by $h = 1/N$, then the matrices have order K equal to $(N-1)^2$.

We now consider the splittings corresponding to the two methods that we are studying in this section. Notice that the B matrix multiplies the unknowns of iteration $k+1$ and the C matrix multiplies those of index k.

For the Jacobi method we see that

$$B = I \quad \text{and} \quad C = L + U. \tag{13.3.2}$$

The splitting for the Gauss–Seidel method depends on the order of the unknowns. We take the order in which the unknowns are updated to be the same as that used in the vector x. With this proviso, the Gauss–Seidel method has the splitting

$$B = I - L \quad \text{and} \quad C = U. \tag{13.3.3}$$

The matrix decomposition (13.3.2) for the Jacobi method is a restatement of (13.1.3), which shows that the updated variables, those multiplied by B, are only the diagonal

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

elements. The variables evaluated at step k in formula (13.1.3) are those corresponding to the off-diagonal elements of the matrix. Similarly, the decomposition (13.3.2) for the Gauss–Seidel method is a restatement of (13.1.4) in which the variables evaluated at step $k+1$ are those corresponding to the elements of the matrix on the diagonal and below. Notice that the matrix B, being a lower triangular matrix, is easy to invert.

It is important to realize that in the actual implementation of these methods in a computer program, we do not store the matrices A, B, and C. They are all quite sparse and it is very inefficient to store them as matrices. The matrices are useful in the analysis, but the implementation can be done without explicit reference to them. That is, a computer implementation should not have an $(N-1)^2 \times (N-1)^2$ array for storage of these matrices. Instead the implementation should use a form such as (13.1.4), in which only the current values of $v^k_{\ell,m}$ are stored. There is no reason to store other arrays.

Analysis of the Jacobi Method

To determine the spectral radius of the iteration matrix for each of these methods applied to the five-point Laplacian, we first find the eigenvalues and eigenvectors of the iteration matrix for the Jacobi method (13.1.3). That is, we must find a vector $\bar{v}$ and value μ such that

$$\mu \bar{v} = (L+U)\bar{v}.$$

If we represent $\bar{v}$ as a grid function with indices from 0 to N in each direction, with the indices 0 and N corresponding to the boundaries, we have

$$\mu v_{\ell,m} = \tfrac{1}{4}\left(v_{\ell-1,m} + v_{\ell,m-1} + v_{\ell+1,m} + v_{\ell,m+1}\right) \tag{13.3.4}$$

for all interior points and $v_{\ell,m}$ equal to zero on the boundaries.

As mentioned after equation (13.2.1), it is important to make a distinction between the eigenvector $\bar{v}$, which has unknowns corresponding to the $(N-1)^2$ interior grid points, and the grid function v, which has $(N+1)^2$ values corresponding to both the interior and boundary points. Because we specify that the boundary values of v are zero, we can write the simple formula (13.3.4). The equations for $\bar{v}$ are different than (13.3.4) if (ℓ, m) is next to a boundary, in which case at least one of the terms on the right-hand side of (13.3.4) would not be present. The use of the grid function v in place of the eigenvector $\bar{v}$ allows for a simpler way to write the equations.

Since $L+U$ is an $(N-1)^2 \times (N-1)^2$ matrix, there should be $(N-1)^2$ eigenvalues and eigenvectors.

Comparing equation (13.3.4) with (13.2.1), we see that the eigenvalues of the Jacobi method are related to those of the Laplacian by

$$\mu = 1 - \tfrac{1}{4}h^2\lambda.$$

So the eigenvalues are

$$\mu^{a,b} = \frac{1}{2}\left[\cos\left(\frac{a\pi}{N}\right) + \cos\left(\frac{b\pi}{N}\right)\right] \tag{13.3.5}$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

for $1 \le a, b \le N-1$. By equation (13.2.7) the eigenvectors are given by

$$v_{\ell,m}^{a,b} = \sin\left(\frac{a\ell\pi}{N}\right) \sin\left(\frac{bm\pi}{N}\right). \tag{13.3.6}$$

This gives all $(N-1)^2$ eigenvalues and eigenvectors for the Jacobi iteration matrix. See also Exercise 13.3.1.

From the formula (13.3.5), we see that

$$\rho(B^{-1}C) = \rho(L+U) = \cos\frac{\pi}{N} = \mu^{1,1}.$$

Since $\rho(L+U)$ is less than 1, the Jacobi method will converge; however, since $\rho(L+U)$ is very close to 1, i.e.,

$$\cos\frac{\pi}{N} \approx 1 - \frac{\pi^2}{2N^2},$$

we see that the convergence will be slow.

The relationship $\mu^{N-a,N-b} = -\mu^{a,b}$ shows that the nonzero eigenvalues occur in pairs and that if μ is an eigenvalue, then $-\mu$ is also an eigenvalue. Notice also that the eigenvalues $\mu^{a,N-a}$ for a between 1 and $N-1$ are all equal to 0 and these are the only eigenvalues equal to 0. Thus there are $N-1$ eigenvalues of $L+U$ that are zero, and consequently there are $(N-1)(N-2)$ nonzero eigenvalues.

Analysis of the Gauss–Seidel Method

We now consider the Gauss-Seidel method. An eigenvector $\bar{v}$ of the iteration matrix $(I-L)^{-1}U$ with eigenvalue λ satisfies

$$\lambda(I-L)\bar{v} = U\bar{v},$$

or, for the grid function $v_{\ell,m}$, we have

$$\lambda v_{\ell,m} = \tfrac{1}{4}\left(\lambda v_{\ell-1,m} + \lambda v_{\ell,m-1} + v_{\ell+1,m} + v_{\ell,m+1}\right) \tag{13.3.7}$$

for all interior points and $v_{\ell,m}$ equal to zero on the boundaries. Notice that the coefficient λ in (13.3.7) multiplies only the variables with superscript of $k+1$ in the formula (13.1.4). In the form (13.3.7) the formula is rather intractable; however, there is a substitution that reduces the analysis of this case to that of the Jacobi method. If we set

$$v_{\ell,m} = \lambda^{(\ell+m)/2} u_{\ell,m} \tag{13.3.8}$$

for each nonzero eigenvalue λ, we obtain, after dividing by $\lambda^{(\ell+m+1)/2}$,

$$\lambda^{1/2} u_{\ell,m} = \tfrac{1}{4}\left(u_{\ell-1,m} + u_{\ell,m-1} + u_{\ell+1,m} + u_{\ell,m+1}\right). \tag{13.3.9}$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

By comparing (13.3.9) with (13.3.4), we see that the nonzero eigenvalues λ of the Gauss–Seidel method are related to the eigenvalues μ of the Jacobi method by

$$\lambda^{a,b} = (\mu^{a,b})^2 = \frac{1}{4}\left(\cos\frac{a\pi}{N} + \cos\frac{b\pi}{N}\right)^2. \tag{13.3.10}$$

In particular,

$$\rho[(I-L)^{-1}U] = \rho(L+U)^2,$$

which shows that the Gauss–Seidel method converges twice as fast as the Jacobi method for the five-point Laplacian.

The eigenvalues of the Gauss–Seidel iteration matrix from equation (13.3.10) give only $(N-1)(N-2)/2$ eigenvalues corresponding to the $(N-1)(N-2)$ nonzero eigenvalues of the Jacobi iteration matrix. An examination of the corresponding eigenvectors for the Gauss–Seidel method shows that they are given by

$$v_{\ell,m}^{a,b} = (\mu^{a,b})^{\ell+m} \sin\left(\frac{a\ell\pi}{N}\right) \sin\left(\frac{bm\pi}{N}\right)$$

with $v_{\ell,m}^{N-a,N-b} = v_{\ell,m}^{a,b}$. All other eigenvalues are zero, and they are not semisimple. (See Appendix A for the definition of a semisimple eigenvalue.)

An alternative way to describe the preceding analysis is to consider the equation

$$\det[\lambda I - (I-L)^{-1}U] = 0$$

for the eigenvalues of the Gauss–Seidel iteration matrix. We have the relationship

$$0 = \det[\lambda I - (I-L)^{-1}U] = \det(\lambda I - \lambda L - U)\det(I-L)^{-1}.$$

The value of $\det(I-L)^{-1}$ is 1, since L is strictly lower triangular. We next transform the matrix $\lambda I - \lambda L - U$ by a similarity transformation using the diagonal matrix S, where the (ℓ, m)th entry on the diagonal is $\lambda^{(\ell+m)/2}$. (Recall that the rows and columns of L, U, and S, are indexed by the ordered pairs of integers corresponding to the grid indices.) We then have

$$S^{-1}(\lambda I - \lambda L - U)S = \lambda I - \lambda^{1/2}(L+U)$$

corresponding to (13.3.9). Thus

$$\begin{aligned}
\det(\lambda I - \lambda L - U) &= \det[\lambda I - \lambda^{1/2}(L+U)] \\
&= \lambda^{(N-1)^2/2}\det[\lambda^{1/2} I - (L+U)] \\
&= \lambda^{(N-1)^2/2} \prod_{1\le a,b\le N-1} (\lambda^{1/2} - \mu^{a,b}) \\
&= \lambda^{N(N-1)/2} \prod_{2\le a+b\le N-1} [\lambda - (\mu^{a,b})^2],
\end{aligned}$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

where in the last product we used the facts that $\mu^{a,N-a}$ is zero for each a and $\mu^{N-a,N-b} = -\mu^{a,b}$. This last formula confirms our previous conclusion that there are $N(N-1)/2$ zero eigenvalues and shows that (13.3.10) gives the $(N-1)(N-2)/2$ nonzero eigenvalues.

An examination of why the substitution (13.3.8) works shows that the updating of values in the Gauss–Seidel method can be organized either in the standard lexicographic order or in the order of increasing values of $\ell + m$. When one updates a value at a grid point with indices (ℓ, m), the computation involves only points of lower value for the sum of the indices, the points with "new" values, and points of larger value for the sum of the indices, the points with "old" values.

The Jacobi method can also be regarded as solving the heat equation

$$u_t = u_{xx} + u_{yy}$$

until a steady-state solution is reached using forward-time central-space differencing and $\Delta t = \frac{1}{4}h^2$. In general it seems that finding steady-state solutions by solving the corresponding time-dependent equations is less efficient than using special methods for the steady-state equations. The Gauss–Seidel method can be regarded as a finite difference approximation for the time-dependent evolution for the equation

$$u_t = u_{xx} + u_{yy} - \varepsilon(u_{xt} + u_{yt}),$$

where $\Delta t = \frac{1}{2}h^2$ and $\varepsilon = \frac{1}{4}h$. The equation should be discretized about (t, x, y) equal to $((n + \frac{1}{2})\Delta t,\ \ell h,\ mh)$ to obtain (13.1.4).

Methods for Diagonally Dominant Matrices

We now state and prove a theorem about the Gauss–Seidel and Jacobi methods for the class of diagonally dominant matrices. Many schemes for second-order elliptic equations, including the five-point Laplacian, give rise to diagonally dominant matrices.

Definition 13.3.1. *A matrix is diagonally dominant if*

$$\sum_{j \neq i} |a_{ij}| \leq |a_{ii}| \tag{13.3.11}$$

for each value of i. A row is strictly diagonally dominant if the inequality in (13.3.11) is a strict inequality and a matrix is strictly diagonally dominant if each row is strictly diagonally dominant.

By a permutation of a matrix A, we mean a simultaneous permutation of the rows and columns of the matrix; i.e., a_{ij} is replaced by $a_{\sigma(i),\sigma(j)}$ for some permutation σ.

Definition 13.3.2. *A matrix is reducible if there is a permutation σ under which A has the structure*

$$\begin{pmatrix} A_1 & O \\ A_{12} & A_2 \end{pmatrix}, \tag{13.3.12}$$

where A_1 and A_2 are square matrices. A matrix is irreducible if it is not reducible.

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

For an arbitrary matrix A the Jacobi iterative method for equation (13.1.1) is

$$\begin{aligned} x^{k+1} &= D^{-1}((D-A)x^k + b) \\ &= (I - D^{-1}A)x^k + D^{-1}b, \end{aligned} \tag{13.3.13}$$

where D is the diagonal matrix with the same diagonal elements as A. If A is written as

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

where L and U are strictly lower and upper triangular matrices, respectively, then the Gauss–Seidel method for (13.1.2) is

$$(D-L)x^{k+1} = Ux^k + b. \tag{13.3.14}$$

Notice that the diagonal dominance of a matrix is preserved if the rows and columns of the matrix are permuted simultaneously. The Gauss–Seidel method is dependent on the permutations of the matrix, whereas the Jacobi method is not, and a matrix is reducible if in using the Jacobi method it is possible to have certain components of x^k be zero for all values of k while x^0 is not identically zero (see Exercises 13.3.4 and 13.3.5).

Theorem 13.3.1. *If A is an irreducible matrix that is diagonally dominant, with at least one row being strictly diagonally dominant, then the Jacobi and Gauss–Seidel methods are convergent.*

Proof. We prove the theorem only for the Gauss–Seidel method; the proof for the Jacobi method is easier. Our proof is based on that of James [31]. We begin by assuming that there is an eigenvalue of the iteration matrix, λ, that satisfies $|\lambda| \geq 1$. Let x be an eigenvector of the iteration matrix with eigenvalue λ, and we normalize x so that $\|x\|_\infty$ is 1.

Let x_i be a component of x with $|x_i|$ equal to 1; then we have the series of inequalities

$$\begin{aligned} |\lambda||a_{ii}||x_i| &= \left| \lambda \sum_{j=1}^{i-1} a_{ij}x_j + \sum_{j=i+1}^{n} a_{ij}x_j \right| \\ &\leq |\lambda| \sum_{j=1}^{i-1} |a_{ij}||x_j| + \sum_{j=i+1}^{n} |a_{ij}||x_j| \\ &\leq |\lambda| \sum_{j \neq i} |a_{ij}||x_j| \\ &\leq |\lambda| \sum_{j \neq i} |a_{ij}||x_i| \leq |\lambda||a_{ii}||x_i|. \end{aligned} \tag{13.3.15}$$

Since the first and last expressions are the same, each inequality in the preceding sequence must be an equality. This implies that for each j, either $|x_j|$ is 1 or a_{ij} is zero. This conclusion follows for each i with $|x_i|$ equal to 1.

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

If we permute the indices of A so that the components with $|x_j|$ equal to 1 are placed first and the others, for which $|a_{ij}|$ is zero, are last, then the structure of A is of form (13.3.12). Since A is irreducible, we conclude that $|x_j|$ is 1 for each value of j.

By choosing a row that is strictly diagonally dominant, the last inequality of (13.3.15) is then a strict inequality, which leads to a contradiction. This implies that the assumption that λ satisfies $|\lambda| \geq 1$ is false. Therefore, $|\lambda|$ is less than 1 for the iteration matrix, and the Gauss–Seidel method is convergent. □

Exercises

13.3.1. Verify by direct substitution that the eigenvalues and eigenvectors for the Jacobi iteration matrix are given by (13.3.5) and (13.3.6), respectively.

13.3.2. Determine the eigenvalues of the Jacobi iteration matrix when applied to the "diagonal" five-point Laplacian scheme given by

$$\frac{1}{2h^2}\left(v_{\ell-1,m-1} + v_{\ell+1,m-1} + v_{\ell-1,m+1} + v_{\ell+1,m+1} - 4v_{\ell,m}\right) = f_{\ell,m} \quad (13.3.16)$$

on a uniform grid with $\Delta x = \Delta y = h$. *Hint:* The eigenvectors for this Jacobi method are the same as for the Jacobi method for the usual five-point Laplacian. The eigenvalues, however, are different.

13.3.3. Verify that zero is not a semisimple eigenvalue of the iteration matrix for the Gauss–Seidel method for the five-point Laplacian on the unit square.

13.3.4. Show that the Jacobi method (13.3.13) is not affected by a simultaneous reordering of the rows and columns of a matrix, whereas the Gauss–Seidel method (13.3.14) is affected. Note that such a permutation is equivalent to applying a similarity transformation using a permutation matrix P to the matrix A resulting in the matrix PAP^{-1}.

13.3.5. Show that a matrix is reducible if in using the Jacobi method, it is possible to have certain components of x^k be zero for all values of k while x^0 is not identically zero (see Exercise 13.3.4).

13.3.6. Show that the matrix for the five-point Laplacian on the unit square is irreducible.

13.4 Convergence Analysis of Point SOR

We now analyze the convergence of SOR for the five-point Laplacian as given by (13.1.5). We have to determine the splitting matrices B and C. As before, B multiplies the unknowns at iteration $k+1$ and C multiplies those at iteration k. We also have the condition that $A = B - C = I - L - U$. After rearranging the formula (13.1.5) and dividing by ω, we obtain that the splitting is given by

$$B = \frac{1}{\omega} I - L, \quad C = \frac{1-\omega}{\omega} I + U.$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

By the same reasoning used with the other methods, from (13.1.5) we obtain that the eigenvalues λ are given as the solutions to

$$\omega^{-1}(\lambda+\omega-1)\,v_{\ell,m} = \tfrac{1}{4}\left(\lambda v_{\ell-1,m} + \lambda v_{\ell,m-1} + v_{\ell+1,m} + v_{\ell,m+1}\right) \tag{13.4.1}$$

for interior grid points, with $v_{\ell,m} = 0$ on the boundary. We use the substitution (13.3.8) for the nonzero eigenvalues, which we used on (13.3.7), obtaining

$$\frac{\lambda+\omega-1}{\omega\lambda^{1/2}}u_{\ell,m} = \frac{1}{4}\left(u_{\ell-1,m} + u_{\ell,m-1} + u_{\ell+1,m} + u_{\ell,m+1}\right).$$

From this relation we see that the nonzero eigenvalues for SOR are related to those of the Jacobi method by

$$\frac{\lambda+\omega-1}{\omega\lambda^{1/2}} = \mu$$

for each eigenvalue μ of the Jacobi iteration matrix. We rewrite this relationship as

$$\lambda - \lambda^{1/2}\omega\mu + \omega - 1 = 0, \tag{13.4.2}$$

which is a quadratic equation in $\lambda^{1/2}$.

First note that the iteration matrix for SOR is nonsingular for ω not equal to 1. We have

$$\begin{aligned}\det B^{-1}C &= \det(\omega^{-1}I - L)^{-1}\det[\omega^{-1}(1-\omega)I + U]\\ &= \omega^K[\omega^{-1}(1-\omega)]^K = (1-\omega)^K,\end{aligned}$$

where K is $(N-1)^2$, the order of the matrix. In particular, zero is not an eigenvalue of the iteration matrix for SOR when ω is not equal to 1.

Equation (13.4.2) relates each eigenvalue of the Jacobi iteration matrix to two eigenvalues of the SOR iteration matrix. Since $\mu^{a,b} = -\mu^{N-a,N-b}$ and there is an ambiguity in the sign of $\lambda^{1/2}$, there is actually a one-to-one correspondence between the pair of nonzero eigenvalues $\{\mu^{a,b}, \mu^{N-a,N-b}\}$ of the Jacobi iteration matrix and the pair of solutions of equation (13.4.2) with μ equal to $\mu^{a,b}$. For $\mu^{a,b}$ equal to zero, there corresponds the one eigenvalue λ equal to $1-\omega$. Thus equation (13.4.2) determines the $(N-1)^2$ eigenvalues of the SOR iteration matrix from the $(N-1)^2$ eigenvalues of the Jacobi iteration matrix.

Since we wish to have both roots of (13.4.2) less than 1 in magnitude and the product of the roots is $\omega - 1$, we see that a necessary condition for the convergence of SOR is

$$|\omega - 1| < 1,$$

or, equivalently,

$$0 < \omega < 2. \tag{13.4.3}$$

This same conclusion is reached for the $N-1$ eigenvalues corresponding to $\mu^{a,b} = 0$. Solving (13.4.2), we obtain

$$\lambda^{1/2} = \tfrac{1}{2}\left[\omega\mu + \sqrt{\omega^2\mu^2 - 4(\omega-1)}\,\right]. \tag{13.4.4}$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

We choose the nonnegative square root when the square root is real in this formula so that when ω equals 1, then $\lambda = \mu^2$ for positive μ and λ is zero for negative μ. This correspondence is somewhat arbitrary, but since SOR reduces to the Gauss–Seidel method for ω equal to 1, it is useful to relate the eigenvalues in this way.

We now assume, without loss of generality, that μ is positive, and we wish to find the value of ω that minimizes the magnitude of $\lambda^{1/2}$ when $\lambda^{1/2}$ is real, i.e., when the quantity inside the square root in (13.4.4) satisfies

$$\omega^2\mu^2 - 4\omega + 4 = \left(\omega\mu - \frac{2}{\mu}\right)^2 - 4\left(\frac{1}{\mu^2} - 1\right) \geq 0.$$

To determine how $\lambda^{1/2}$ varies as a function of ω, we take the derivative of (13.4.4):

$$\begin{aligned}\frac{\partial}{\partial\omega}\lambda^{1/2} &= \frac{1}{2}\mu + \frac{1}{2}(\omega\mu^2 - 2)(\omega^2\mu^2 - 4\omega + 4)^{-1/2} \\ &= \frac{\mu}{2}\left[1 - \frac{2/\mu - \omega\mu}{\sqrt{(2/\mu - \omega\mu)^2 - 4(\mu^{-2} - 1)}}\right] < 0.\end{aligned}$$

Since this derivative is negative, we see that to decrease the size of $\lambda^{1/2}$ we must increase ω. The maximum value of ω for which $\lambda^{1/2}$ is real is the root of

$$\omega^2\mu^2 - 4\omega + 4 = 0 \tag{13.4.5}$$

that satisfies (13.4.3).

When μ is negative and $\lambda^{1/2}$ is real, then $\lambda^{1/2}$ is less than the value of $\lambda^{1/2}$ corresponding to $|\mu|$ and thus does not affect the spectral radius of the iteration matrix. Since we are ultimately concerned with determining the spectral radius of the iteration matrix, we need not consider this case in detail.

Now consider the case when $\lambda^{1/2}$ is complex. Notice that since the polynomial in (13.4.2) has real coefficients, the two values of λ corresponding to μ and $-\mu$ are complex conjugates of each other. The magnitude of λ can be computed from (13.4.4) as follows:

$$|\lambda| = |\lambda^{1/2}|^2 = \tfrac{1}{4}\left[\omega^2\mu^2 + 4(\omega - 1) - \omega^2\mu^2\right] = \omega - 1.$$

From this relationship we see that to decrease $|\lambda|$ we must decrease ω. The minimum value of ω for which $\lambda^{1/2}$ is complex is again the root of (13.4.5) satisfying (13.4.3).

We now consider the eigenvalues $\lambda(\mu^{a,b})$ for the SOR iteration matrix for all eigenvalues $\mu^{a,b}$ of $L + U$. The spectral radius for the SOR iteration matrix is the maximum magnitude of all the $\lambda(\mu^{a,b})$. We wish to choose ω in order to minimize the spectral radius.

First, consider ω very close to 2—so close that

$$\omega^2(\mu^{a,b})^2 - 4\omega + 4$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

is negative for all eigenvalues $\mu^{a,b}$. By our previous discussion, all the λ corresponding to nonzero values of $\mu^{a,b}$ are complex with magnitude equal to $\omega - 1$. Those λ corresponding to $\mu^{a,b}$ that are equal to zero have the value $-(\omega - 1)$, which means all eigenvalues have the same magnitude. The spectral radius is therefore $\omega - 1$. As we decrease ω, we will reach some value ω^* at which some $\lambda(\mu^{a,b})$ that is complex will become real. It is easy to see that this must happen for $\mu^{a,b}$ equal to $\bar{\mu}$, the largest eigenvalue of $L + U$ in magnitude. For ω less than ω^*, the spectral radius will now increase because $\partial\lambda^{1/2}/\partial\omega$ is negative for λ corresponding to $\bar{\mu}$. Thus the optimal choice for ω is ω^*, where ω^* satisfies

$$\omega^{*2}\bar{\mu}^2 - 4\omega^* + 4 = 0$$

and (13.4.3), which gives the optimal value as

$$\omega^* = \frac{2}{1 + \sqrt{1 - \bar{\mu}^2}}. \tag{13.4.6}$$

Since for Laplace's equation $\bar{\mu} = \cos \pi/N$, the value of ω^* for Laplace's equation is

$$\omega^* = \frac{2}{1 + \sin \pi/N}$$

and the spectral radius is

$$\rho^* = \omega^* - 1 = \frac{1 - \sin \pi/N}{1 + \sin \pi/N} \approx 1 - \frac{2\pi}{N}.$$

The behavior of the spectral radius as a function of ω is displayed in Figure 13.1 for $N = 10$. The optimal value of ω is the lowest point on the graph. For ω larger than ω^*, the spectral radius is seen to be the linear relation $\rho = \omega - 1$. Otherwise, the spectral radius is obtained from (13.4.4) for $\mu = \bar{\mu}$.

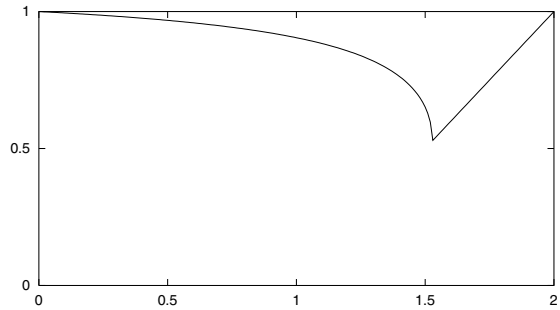

Figure 13.1. *The spectral radius ρ as a function of ω for $N = 10$.*

It is also useful to consider the behavior of all of the eigenvalues of the iteration matrix for SOR as functions of ω as ω increases from 1. For ω equal to 1, there are $N(N-1)/2$ eigenvalues that are 0, and the rest are real and located between 0 and 1, given by (13.3.10). As ω is taken to be larger than 1, these eigenvalues between 0 and

1 all decrease in magnitude. Of the eigenvalues that are 0 for ω equal to 1, $N-1$ of them become negative for ω larger than 1 and have the value $1-\omega$, and the rest become positive and increase as ω increases. When an eigenvalue from the group that is decreasing with ω coalesces with an eigenvalue from the group that is increasing with ω, they become a pair of complex conjugates of magnitude $\omega - 1$. The optimal value of ω is that value where only two eigenvalues in the interval $(0, 1)$ are real and are equal to each other. This value is given by (13.4.6).

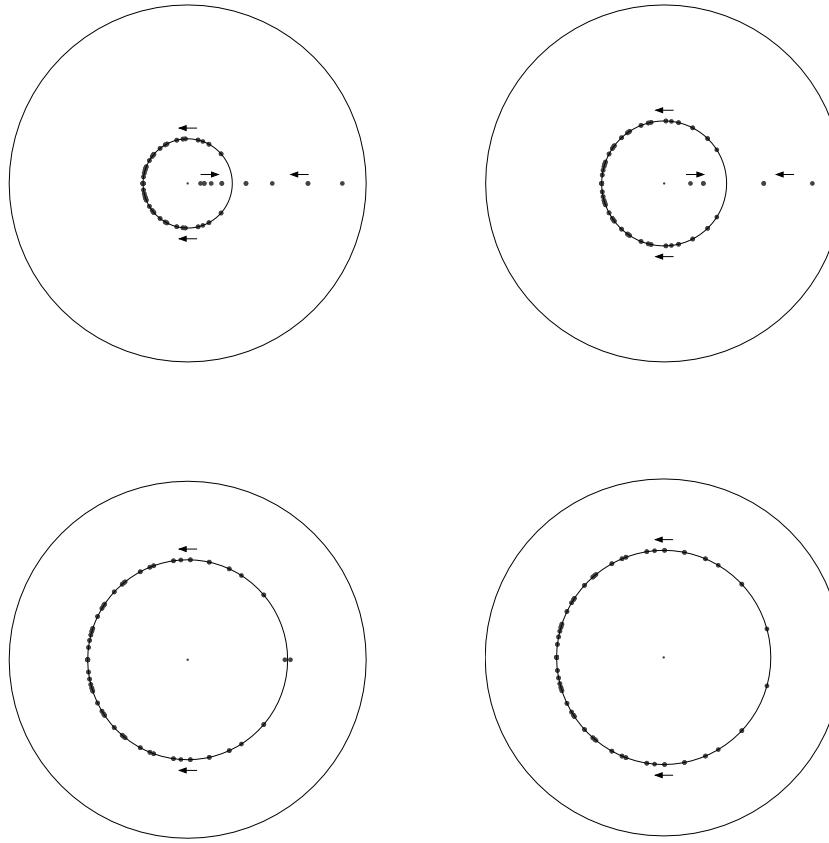

Figure 13.2. *Eigenvalues for* $\omega = 1.25, 1.35, 1.56, 1.60$ *with* $N = 11$.

This is illustrated in the plots in Figure 13.2 that show the eigenvalues $\lambda^{a,b}$ as functions of ω for N equal to 11. There are 100 eigenvalues in all. For ω equal to 1.25, the figure in the upper left shows four real eigenvalues larger than $\omega - 1$ and four positive real eigenvalues less than this value. In addition, there is the real eigenvalue $-\omega + 1$ of multiplicity 10. The other 82 eigenvalues are complex. The arrows at the top and bottom of the circles show the direction that the eigenvalues move as ω increases. The eigenvalues

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

on the positive real axis move toward the circle of radius $\omega - 1$ as ω increases. The plot at the upper right shows the positions of the eigenvalues for ω equal to 1.35. For this case, there are only two pairs of positive real eigenvalues. As ω increases through the values 1.25 to 1.35 the magnitude of the largest positive eigenvalue decreases and one pair of real eigenvalues becomes complex. At 1.56, which is just slightly less than ω^*, there remain only two positive eigenvalues, and they are very close to $\omega - 1$. For ω equal to 1.60, shown in the lower right plot, all eigenvalues are of magnitude $\omega - 1$.

Because of the relationship $\mu^{a,b} = \mu^{b,a}$, which holds if $\Delta x = \Delta y$, the *set* of eigenvalues has fewer than $(N-1)(N-2)+1$ elements. If N is even, the set of eigenvalues has $N(N-2)/4+1$ elements, and if N is odd, the set of eigenvalues has $(N-1)^2/4+1$ elements.

We now examine how the number of iterations for an iterative method to achieve a certain error tolerance is related to the spectral radius. Suppose an iterative method has spectral radius ρ and we wish to know how many iterations, I, it will take to reduce the norm of the error to a certain multiple, ε, of the initial error. From (13.1.9) we see that we must have

$$\rho^I \approx \varepsilon$$

or

$$I \approx \frac{-\log \varepsilon}{-\log \rho}.$$

If ρ is close to 1, then

$$I \approx \frac{\log \varepsilon^{-1}}{1-\rho}.$$

So, for the Gauss–Seidel method, from $\rho = (\cos(\pi/N))^2 \approx 1 - \pi^2/N^2$,

$$I \approx \frac{N^2}{\pi^2} \log \varepsilon^{-1},$$

and for SOR with $\omega = \omega^*$,

$$I \approx \frac{N}{2\pi} \log \varepsilon^{-1}.$$

These formulas show that for the Gauss–Seidel and Jacobi methods, the number of iterations is proportional to N^2, whereas for SOR it is proportional to N. This is why SOR is a dramatic improvement in efficiency over the Gauss–Seidel method for even small values of N.

Exercise

13.4.1. Determine the optimal value of ω for SOR applied to the "diagonal" five-point Laplacian (13.3.16).

13.5 Consistently Ordered Matrices

In relating the eigenvalues of the Gauss–Seidel and SOR methods to the eigenvalues of the Jacobi method, we made use of the fact that if α is an eigenvalue of $\lambda L + U$, then $\alpha\lambda^{-1/2}$ is an eigenvalue of $L + U$. (See the discussion relating to (13.3.7) and (13.4.1).) If $L + U$ has this special property, it is said to be *consistently ordered.*

Definition 13.5.1. *A matrix of the form $I - L - U$ is consistently ordered if whenever α is an eigenvalue of $\lambda L + U$, then $\alpha\lambda^{-1/2}$ is an eigenvalue of $L + U$.*

An examination of our analysis shows that we have proved that if $I - L - U$ is consistently ordered, then the Gauss–Seidel method will converge if and only if the Jacobi method converges, and the Gauss–Seidel method will converge twice as fast. We have also shown that SOR will converge under these conditions, and the optimal value of ω is given by (13.4.6). The reader should check that in deriving (13.4.6) we used nothing special about the matrix $I - L - U$ other than that it was consistently ordered and that its eigenvalues are real (see Exercise 13.5.6). Thus we have proved the following theorem.

Theorem 13.5.1. *If the matrix A, which is equal to $I - L - U$, is consistently ordered and has real eigenvalues, then the SOR method, given by*

$$(\omega^{-1}I - L)x^{k+1} = (\omega^{-1}(1 - \omega)I + U)x^k + b,$$

converges to the solution of $Ax = b$ for all ω in the interval $(0, 2)$ if and only if the Jacobi method converges. Moreover, the optimal value of ω is given by formula (13.4.6), where $\bar{\mu}$ is the eigenvalue of $L + U$ with largest magnitude.

In case matrix $I - L - U$ is consistently ordered but with complex eigenvalues, we can determine those values of ω for which the SOR method converges, but it is more difficult to determine the optimal value of ω.

Theorem 13.5.2. *If the matrix A, given by $I - L - U$, is consistently ordered, then the SOR method converges for those values of ω in the interval $(0, 2)$ that satisfy*

$$(\text{Re } \mu_i)^2 + \left(\frac{\omega}{2 - \omega}\right)^2 (\text{Im } \mu_i)^2 < 1 \tag{13.5.1}$$

for each eigenvalue μ_i of $L + U$. In particular, if $|\text{Re } \mu_i| < 1$ for each μ_i, then there is an interval $(0, \bar{\omega})$ of values of ω for which SOR converges.

Proof. Let $\tau_i = \lambda_i^{1/2}$. Then equation (13.4.2) can be written

$$\zeta_i = \frac{1}{2}\left(\tau_i - \frac{1 - \omega}{\tau_i}\right) = \frac{\omega\mu_i}{2}.$$

We consider the mapping of the complex plane that takes the complex variable τ to $\zeta = (\tau - (1 - \omega)/\tau)/2$. This mapping takes circles in the complex τ plane to ellipses in the

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

complex ζ plane. The circle $|\tau| = |1-\omega|^{1/2}$ is mapped to the degenerate ellipse given by

$$\text{Re } \zeta = 0, \qquad |\text{Im } \zeta| \le \sqrt{1-\omega}$$

when $0 < \omega < 1$ and

$$\text{Im } \zeta = 0, \qquad |\text{Re } \zeta| < \sqrt{\omega - 1}$$

when $1 < \omega < 2$. In either case the annulus $|\omega - 1|^{1/2} \le |\tau| < 1$ is mapped onto the ellipse

$$\left(\frac{\text{Re } \zeta}{\omega/2}\right)^2 + \left(\frac{\text{Im } \zeta}{1-\omega/2}\right)^2 < 1. \tag{13.5.2}$$

For each value of ζ we obtain two roots; if τ_1 is one root, then $\tau_2 = (\omega - 1)/\tau_1$ is the other root. It is therefore necessary that $|\omega - 1| = |\tau_1 \tau_2|$ must be less than 1. We also see that one root, say τ_1, must satisfy $|\omega - 1|^{1/2} \le |\tau_1| < 1$. If we set $\zeta = \omega\mu_i/2$ in (13.5.2) we obtain (13.5.1), which proves the first assertion of the theorem. We also see that if $|\text{Re } \mu_i|$ is less than 1 for all μ_i, then there are values of ω near 0 that satisfy (13.5.1). This proves the theorem. □

Estimating the Optimal Value of ω

SOR often converges when $I - L - U$ is not consistently ordered, for example, when used on more general elliptic equations with variable coefficients. Even though formula (13.4.6) is not valid, we often find that the optimal ω is close to 2. In fact, the relation

$$\omega^* = \frac{2}{1 + Ch} \tag{13.5.3}$$

is often nearly true, where h is some measure of the grid spacing and C is some constant. This formula is computationally very useful and can be employed as follows. First, for a coarse grid we find a good estimate for ω^*, the optimal ω, by experimentation, i.e., by making several calculations with different values of ω. Given this ω^* and h, we can determine C and then use (13.5.3) to estimate ω^* for smaller values of h. This formula can considerably reduce computational effort.

Garabedian [22] showed that the optimal value of ω for Poisson's equation on a domain other than the square can be approximated by

$$\omega^* \approx \frac{2}{1 + k_1 h/\sqrt{2}},$$

where h is the mesh width and k_1 is the first eigenvalue of the Laplacian, i.e., the least positive value k_1 such that

$$\nabla^2 u + k_1^2 u = 0$$

has a nontrivial solution with u equal to zero on the boundary. He also pointed out that the value of k_1 can be estimated from below by the Faber–Krahn inequality

$$k_1 \ge k_1^* \left(\frac{\pi}{A}\right)^{1/2},$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

where A is the area of the domain and k_1^* is the first eigenvalue for a circle of radius 1. The constant k_1^* is the first zero of the Bessel function J_0 and is approximately 2.4. Because the Faber–Krahn inequality is sharp for circular domains and less sharp for elongated and nonconvex regions, we can estimate k_1 as a multiple of $k_1^*(\pi/A)^{1/2}$, the multiplying factor being determined by experiment. In ways similar to this, we can usually estimate the optimal value of ω quite well in situations for which it cannot be explicitly determined.

In estimating the optimal value of ω, it is important to realize that it is better to overestimate ω^* than it is to underestimate. This is because, as shown in Figure 13.1 for ω larger than ω^*, the spectral radius varies linearly with ω, but the derivative with respect to ω of $\lambda(\bar{\mu})$ for ω less than ω^*, as given in (13.4.4), is infinite for the optimal value of ω.

Variations of SOR

There are several variations of SOR. The one we have considered is often called point SOR with *natural ordering*. One variation is to use a different ordering of the points. If we update all the points with $\ell + m$ equal to an even number, followed by an update of all those with $\ell + m$ equal to an odd number, we have point SOR with *checkerboard ordering*.

We can also do one iteration of point SOR with natural ordering followed by one iteration of point SOR with reverse natural ordering. This is called *symmetric* SOR, or SSOR.

Line SOR, or LSOR, updates one line of grid points at a time. The formula is

$$\tilde{v}_{\ell-1,m} - 4\tilde{v}_{\ell,m} + \tilde{v}_{\ell+1,m} = -\omega \left(v_{\ell-1,m}^k + v_{\ell+1,m}^k + v_{\ell,m-1}^{k+1} + v_{\ell,m+1}^k - 4v_{\ell,m}^k \right),$$

$$v_{\ell,m}^{k+1} = v_{\ell,m}^k + \tilde{v}_{\ell,m} \tag{13.5.4}$$

when taking the lines in the usual order. LSOR requires that a tridiagonal system be solved for each grid line. This extra work is offset by a smaller spectral radius of the iterative method. Generally it is considered to be faster than point SOR by a factor of $\sqrt{2}$; see Exercise 13.5.7.

In general, line, or block, SOR is derived by writing the system (13.1.6) as

$$-\sum_{m<j} L_{jm} x_m + D_j x_j - \sum_{m>j} U_{jm} x_m = b_j,$$

where each x_j is a vector consisting of a subset of all the components of x. The coefficients L_{jm}, U_{jm}, and D_j are matrices of the appropriate sizes. In the usual case x_j is the set of unknowns associated with the jth grid line. The line Jacobi method is given by

$$x_j^{k+1} = D_j^{-1} \left(b_j + \sum_{m<j} L_{jm} x_m^k + \sum_{m>j} U_{jm} x_m^k \right)$$

and the LSOR is given by

$$x_j^{k+1} = x_j^k + \omega D_j^{-1} \left(b_j + \sum_{m<j} L_{jm} x_m^{k+1} + \sum_{m>j} U_{jm} x_m^k - D_j x_j^k \right),$$

from which we obtain (13.5.4) for the special case of the five-point Laplacian.

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

It is easy to implement a symmetric LSOR method, in which the lines are swept in the opposite order during each successive iteration. As with point SOR, symmetric LSOR has a better convergence rate with almost no extra work.

One case where LSOR is useful is in the solution of elliptic equations on domains with polar coordinate systems (r, θ); see Section 12.6 and Exercise 12.7.2. Each "line" consists of the grid points with fixed value of r. At the center we use formula (12.6.3). The periodic tridiagonal system for each line can be solved by the methods of Section 3.5 (see also Exercise 3.5.8). We first update all the points other than the origin; then (12.6.3) can be used to compute the new value at the origin. In the SOR iterations it appears to be best to proceed from the boundary of the disk in toward the center. The equation to update the center value using (12.6.3) is

$$u_0^{k+1} = u_0^k + \omega \left[\frac{1}{J} \sum_{j=1}^{J} u_{1j}^{k+1} - u_0^k - f(0) \left(\frac{\Delta r}{2} \right)^2 \right].$$

Implementing SOR Methods

The implementation of SOR methods is quite straightforward, but there are some small details that should be mentioned. The SOR methods are usually terminated when the change in the solution is sufficiently small. One usually sets a tolerance and proceeds until the changes are smaller than that tolerance. Rather than using formula (13.1.5) it is better to use the two-step procedure

$$\begin{aligned} \tilde{v}_{\ell,m}^k &= \tfrac{1}{4}(v_{\ell+1,m}^k + v_{\ell-1,m}^{k+1} + v_{\ell,m+1}^k + v_{\ell,m-1}^{k+1}) - v_{\ell,m}^k, \\ v_{\ell,m}^{k+1} &= v_{\ell,m}^k + \omega\, \tilde{v}_{\ell,m}^k, \end{aligned} \tag{13.5.5}$$

where $\tilde{v}_{\ell,m}^k$ is used to measure the change in the solution per iteration.

Of course, since SOR uses immediate replacement, in the computer implementation there is no need to index the solution by the index k. Also, the temporary variable $\tilde{v}_{\ell,m}$ is not stored as an array; it need only be a scalar. Both steps of (13.5.5) are computed at each grid point before proceeding to the next point. The two-step procedure (13.5.5) is less sensitive to loss of significance than is the procedure of first using (13.1.5) and then determining the change by computing the difference between the successive values of $v_{\ell,m}$. The line SOR (13.5.4) is given as a two-step procedure for the same reason. For more details on the implementation, the reader is referred to Hageman and Young [29].

Here is a section of pseudocode illustrating how to implement the SOR method. Notice that it requires only the one two-dimensional array `v`.

```
Initialize solution
while change > tolerance
    change = 0
    loop on ℓ
       loop on m
```

```
          change_pt = [v(ℓ-1,m)+v(ℓ+1,m)+v(ℓ,m-1)+v(ℓ,m+1)
                              -4v(ℓ,m)]/ 4.
          v(ℓ,m) = v(ℓ,m) + omega*change_pt
          change = change + change_pt ^ 2
       end of loop on m
    end of loop on ℓ
    change = sqrt( change*h ^ 2
end of while loop
```

The changes in the solution can be measured by the L^2 norm of $\tilde{v}^k$, either with or without the factor of ω. The L^2 norm preferred by the author is

$$\|\tilde{v}^k\| = \left(\sum_{\ell,m} |\tilde{v}^k_{\ell,m}|^2 \, h^2 \right)^{1/2}. \tag{13.5.6}$$

The factor of h in the measurement of the norm causes the stopping tolerance to be relatively independent of the grid size. The results given in the examples in this book use the norm (13.5.6).

In checking for the optimal value of ω for a SOR method, it is often found that the optimal value of ω to achieve convergence for a given tolerance in the norm (13.5.6) is close to, but not the same as, that given by formula (13.4.6). One reason for this discrepancy is that the convergence criteria are different; i.e., the use of (13.5.6) is not a measurement of the spectral radius that was used in deriving (13.4.6). This discrepancy is of little concern, since formulas such as (13.4.6) and (13.5.3) can be used to give nearly optimal values for ω.

For Poisson problems, the values of $f(x, y)$ at grid points should be computed once and stored in an array, rather than be computed as needed. For standard computers, accessing an array element is much faster than a function call and its related computation.

For other information on these and other iterative methods, see the compendium of numerical methods by Barrett et al. [4].

Exercises

13.5.1. Using the point SOR method, solve Poisson's equation

$$u_{xx} + u_{yy} = -2 \cos x \sin y$$

on the unit square. The boundary conditions and exact solution are given by the formula $u = \cos x \sin y$. Use the standard five-point difference scheme with $h = \Delta x = \Delta y = 0.1,\ 0.05,$ and 0.025. The initial iterate should be zero in the interior of the square. Comment on the accuracy of the scheme and the efficiency of the method. Use $\omega = 2/(1 + \pi h)$. Stop the iterations when the changes in the solution as measured in the L^2 norm (13.5.6) are less than 10^{-7}. *Note:* For some computers the value of 10^{-7} will be too small unless double-precision variables are used.

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

13.5.2. Solve the same problem as in Exercise 13.5.1 but use the fourth-order accurate finite difference scheme (12.5.4). Comment on the efficiency and accuracy of the two methods. Even though the matrix for this scheme is not consistently ordered, the SOR method will converge, as is shown in the next section. A good estimate for the optimal value of ω is $2/(1+\pi h)$.

13.5.3. Use the results of Exercise 13.5.1 to show that the values of $\delta_{0x}v$ and $\delta_x^2 v$, where v is the computed solution, are second-order approximations to the corresponding derivatives.

13.5.4. Use the results of Exercise 13.5.2 to show that the values of the approximations to the first and second derivatives given by (3.3.3) and (3.3.7) give fourth-order approximations to the corresponding solutions.

13.5.5. Solve the same equation as in Exercise 13.5.1 but on the trapezoidal domain discussed in Section 12.7.

13.5.6. Prove Theorem 13.5.1.

13.5.7. Determine the formula for the optimal value of ω as a function of the grid spacing for LSOR on the unit square in the case of equal spacing in both directions. *Hint:* You will have to use the fact that the natural ordering of the lines is a consistent ordering and also that the eigenvectors for the line Jacobi method are the same as for the point Jacobi method. The eigenvalues, however, are different.

13.5.8. Suppose matrix A, given by $I-L-U$, is consistently ordered and $L+U$ is skew with eigenvalues μ_j. (A skew matrix is one for which $S^T=-S$.) Show that SOR is convergent if and only if ω is in the interval $\left(0,\ 2(1+\bar{\beta})^{-1}\right)$, where $\bar{\beta}=\max|\mu_j|$ and the optimal value of ω is given by

$$\omega^*=\frac{2}{1+(1+\bar{\beta}^2)^{1/2}}.$$

Notice that ω^* is less than 1.

13.5.9. Show that the fourth-order accurate finite difference scheme (12.5.4) is not consistently ordered with the natural ordering of points. Also show that it is consistently ordered for LSOR.

13.5.10. Show that the optimal value of ω for point SOR with the checkerboard ordering applied to the five-point Laplacian on the unit square is given by formula (13.4.6). *Hint:* Show that the checkerboard ordering is a consistent ordering.

13.6 Linear Iterative Methods for Symmetric, Positive Definite Matrices

We can also analyze linear iterative methods when the matrix A is symmetric and positive definite. The methods of this section can be applied to many schemes that are not consistently ordered and thus cannot be analyzed by the methods of the previous section. For

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

example, the fourth-order accurate nine-point scheme (12.5.4) is not consistently ordered for point SOR, but the matrix is symmetric and positive definite (see Exercise 13.6.3). On the one hand, the method of analysis of this section requires less detailed understanding of the matrix than is required to establish the consistent ordering of A; on the other hand, it is not apparent how to determine the optimal value of ω.

It should be pointed out that one need not write out the scheme in matrix form to determine if the matrix is symmetric. The matrix A representing the scheme is symmetric when the coefficient multiplying $v_{\ell',m'}$ in the scheme applied at grid point (ℓ, m) is the same as the coefficient multiplying $v_{\ell,m}$ in the scheme applied at grid point (ℓ', m') for each of the unknown grid function values.

The main result for symmetric, positive definite matrices is the following theorem.

Theorem 13.6.1. *If A is symmetric and positive definite, then the iterative method (13.1.8) based on the splitting (13.1.7) is convergent if*

$$\text{Re } B > \tfrac{1}{2}A \tag{13.6.1}$$

or, equivalently, that $B^T + C$ is symmetric and positive definite, i.e.,

$$B^T + C > 0. \tag{13.6.2}$$

Proof. We first establish that the two conditions in the conclusion are equivalent. The matrix Re B is $(B + B^T)/2$, and thus (13.6.1) is equivalent to

$$B^T + B - A > 0. \tag{13.6.3}$$

The defining relation of the splitting (13.1.7) shows that this is equivalent to (13.6.2) and that $B^T + C$ is symmetric.

We now begin the proof. We measure the error in the norm induced by A, i.e., $\|x\|_A = (x, Ax)^{1/2}$. In this norm we have the relation

$$\|e^{k+1}\|_A = \|B^{-1}Ce^k\|_A \le \|B^{-1}C\|_A \|e^k\|_A$$

(see Appendix A). If the norm of $B^{-1}C$ is less than 1, then the error will decrease at each iteration and the method will converge. We have that the norm of $B^{-1}C$ is given by

$$\|B^{-1}C\|_A^2 = \sup_{x \neq 0} \frac{(B^{-1}Cx, AB^{-1}Cx)}{(x, Ax)} = \sup_{x \neq 0} \frac{(x, C^T B^{-T} AB^{-1}Cx)}{(x, Ax)}.$$

Thus the condition $\|B^{-1}C\|_A < 1$ is equivalent to $C^T B^{-T} AB^{-1}C < A$, and we consider now the matrix $C^T B^{-T} AB^{-1}C$. We have, using relation (13.1.7) to eliminate C,

$$\begin{aligned} C^T B^{-T} AB^{-1}C &= (I - AB^{-T})A(I - B^{-1}A) \\ &= A - (AB^{-T}A + AB^{-1}A - AB^{-T}AB^{-1}A). \end{aligned}$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Thus we see that $C^T B^{-T} A B^{-1} C < A$ if and only if

$$AB^{-T}A + AB^{-1}A - AB^{-T}AB^{-1}A > 0. \tag{13.6.4}$$

But this last expression can be factored as

$$AB^{-T}(B + B^T - A)B^{-1}A$$

or

$$(B^{-1}A)^T(B + B^T - A)B^{-1}A.$$

Thus (13.6.4) is true if and only if (13.6.3) is true, and this implies that $\|B^{-1}C\|_A$ is less than 1 and so the method is convergent. This proves the theorem. □

Example 13.6.1. As our first application of Theorem 13.6.1 we consider SOR for a symmetric matrix A of the form

$$A = I - L - L^T. \tag{13.6.5}$$

Note that L need not be the lower triangular part of A, although in most applications it is. We have the splitting

$$B = \frac{1}{\omega}I - L, \qquad C = \frac{1-\omega}{\omega}I + L^T,$$

and the condition (13.6.2) is

$$B^T + C = \frac{2-\omega}{\omega}I > 0.$$

We conclude that SOR for the matrix (13.6.5) will converge for ω in the interval $(0, 2)$ if the matrix A is positive definite.

This result applies to the fourth-order accurate nine-point scheme (12.5.4), which is not consistently ordered; see Exercise 13.6.3. □

Example 13.6.2. For our second application we consider SSOR for a matrix in the form (13.6.5). For SSOR the splitting is

$$\begin{aligned} B &= \frac{\omega}{2-\omega}\left(\frac{1}{\omega}I - L\right)\left(\frac{1}{\omega}I - L^T\right), \\ C &= \frac{\omega}{2-\omega}\left(\frac{1-\omega}{\omega}I + L\right)\left(\frac{1-\omega}{\omega}I + L^T\right) \end{aligned} \tag{13.6.6}$$

(see Exercise 13.6.1). In this case both B and C are symmetric and

$$\begin{aligned} B + C &= \omega(2-\omega)^{-1}\left[\omega^{-2}(2 - 2\omega + \omega^2)I - L - L^T + 2LL^T\right] \\ &= \omega(2-\omega)^{-1}\left[\frac{(2-\omega)^2}{2\omega^2}I + \left(\frac{1}{\sqrt{2}}I - \sqrt{2}L\right)\left(\frac{1}{\sqrt{2}}I - \sqrt{2}L^T\right)\right] \\ &= \frac{2-\omega}{2\omega} + \frac{\omega}{2-\omega}\left(\frac{1}{\sqrt{2}} - \sqrt{2}L\right)\left(\frac{1}{\sqrt{2}} - \sqrt{2}L\right)^T, \end{aligned}$$

which is positive definite if and only if $0 < \omega < 2$. □

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

As we see from these two examples, this analysis shows rather easily that the iterative methods will converge for ω between 0 and 2, but it does not give an indication of the optimal value of ω. The method used to prove Theorem 13.6.1 can be refined to give estimates of the optimal ω, but we will not pursue this topic. Formula (13.5.3) and the discussion of that formula should suffice for most applications.

Exercises

13.6.1. Verify that the matrices in (13.6.6) define the splitting for SSOR.

13.6.2. Consider the iterative method (13.1.10) based on the ADI method and assume that the matrices A_1 and A_2 are symmetric. Use Theorem 13.6.1 to determine the values of μ for which the iterative method will converge.

13.6.3. Show that the matrix arising from the fourth-order accurate scheme (12.5.4) is positive definite when written in the form

$$\begin{aligned}\frac{10}{3}v_{\ell,m} &- \frac{2}{3}\left(v_{\ell+1,m} + v_{\ell-1,m} + v_{\ell,m+1} + v_{\ell,m-1}\right)\\ &- \frac{1}{6}\left(v_{\ell+1,m+1} + v_{\ell+1,m-1} + v_{\ell-1,m+1} + v_{\ell-1,m-1}\right)\\ &= -\frac{h^2}{12}\left(f_{\ell+1,m} + f_{\ell-1,m} + f_{\ell,m+1} + f_{\ell,m-1} + 8f_{\ell,m}\right).\end{aligned}$$

13.7 The Neumann Boundary Value Problem

In this section we examine second-order elliptic equations with the Neumann boundary condition (12.1.4). More specifically, we confine ourselves to equations of the form

$$a(x,y)u_{xx} + 2b(x,y)u_{xy} + c(x,y)u_{yy} + d_1(x,y)u_x + d_2(x,y)u_y = f(x,y) \quad (13.7.1)$$

on a domain Ω with the boundary condition

$$\frac{\partial u}{\partial n} = b(x,y) \quad \text{on } \partial\Omega, \quad (13.7.2)$$

which is the same as (12.1.4). Notice that equation (13.7.1) depends on u only through its derivatives. As opposed to the Dirichlet boundary value problem for equation (13.7.1), the solution to (13.7.1) and (13.7.2) is not unique. Indeed, if u is any solution to (13.7.1) and (13.7.2), then for any constant c the function u_c given by $u_c(x,y) = u(x,y) + c$ is also a solution. The solution of this boundary value problem is unique to within the additive constant; that is, any two solutions differ by a constant (see Exercise 13.7.2). (The nonuniqueness of the solution of elliptic equations can occur for any type of boundary condition; see Example 12.3.2.)

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

In addition to the solution not being unique, a solution may not exist unless the data, f and b in (13.7.1), satisfy a linear constraint. For many applications, especially symmetric problems, we can easily determine the constraint to be satisfied, but for some problems it may be quite difficult to determine this constraint. For Poisson's equation (12.1.1) with the Neumann boundary condition (13.7.2), the constraint on the data is equation (12.1.5).

As an example of an equation for which it is difficult to determine the constraint, we have

$$u_{xx} + e^{xy}u_{yy} = f$$

with the Neumann boundary condition (see Exercise 12.7.3). The solutions of this boundary value problem are unique to within an additive constant, and numerical evidence confirms that there is a constraint on the data.

The nonuniqueness of the solution of the differential equation boundary value problem and possible nonexistence of a solution causes some difficulties in obtaining the numerical solution. A careful examination of the difficulties leads to effective strategies to surmount them.

We now consider using a finite difference scheme to obtain an approximate solution of the Neumann problem. As an example, we consider solving the Neumann problem for the Laplacian on the unit square. Either the five-point Laplacian (12.5.1) or the nine-point Laplacian (12.5.4) might be used to approximate the differential equation. For the boundary condition, suitable approximations are

$$\frac{\partial u}{\partial x}(0, y_m) \approx \frac{-3v_{0m} + 4v_{1m} - v_{2m}}{2\Delta x} = b(0, y_m) \tag{13.7.3}$$

or

$$\frac{\partial u}{\partial x}(0, y_m) \approx \frac{v_{1m} - v_{0m}}{\Delta x} = b(0, y_m). \tag{13.7.4}$$

The approximation (13.7.3) is second-order accurate, whereas (13.7.4) is first-order accurate. For each of these methods we obtain one equation for each unknown $v_{\ell,m}$, $0 \le \ell, m \le N$. The linear system can be written as

$$Ax = b \tag{13.7.5}$$

as for the Dirichlet boundary conditions, except in this case the vector of unknowns, x, also contains the components of $v_{\ell,m}$ on the boundary. Thus, K, the order of the system (13.7.5), is $(N+1)^2$.

The nonuniqueness of the solution of the Neumann problem for (13.7.1) implies that the matrix A in (13.7.5) is singular or nearly singular. Because the solution of (13.7.1) with the Neumann boundary conditions is unique only up to a constant, most difference schemes for (13.7.1) and the boundary conditions will also be unique only to within an additive constant. That is, if x is a solution to (13.7.5), then

$$A(x + \alpha x_0) = b$$

is also true, where x_0 is the vector all of whose components are 1 and α is any real number. Comparing this equation with (13.7.5), we see that x_0 is a null vector of A, i.e.,

$$Ax_0 = 0.$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

We will assume that the null space of the matrix A is one-dimensional. (The null space of a matrix is the linear subspace of vectors z such that Az is the zero vector.) The matrix A is said to have a (column) rank deficiency of 1. This is a reasonable assumption, since the null space of the differential operator is also one-dimensional.

A fundamental result of linear algebra is that the row rank of a matrix is equal to its column rank. Thus there is a nonzero vector y_0 such that $y_0^T A$ is the zero vector. The vector y_0 represents the constraint that the data in (13.7.5) must satisfy in order for a solution to exist. We have

$$0 = (y_0^T A)x = y_0^T (Ax) = y_0^T b \tag{13.7.6}$$

if a solution x exists for (13.7.5). If A is symmetric, then y_0 may be taken to be x_0.

There are two problems concerning constraint (13.7.6). The first is that we may not know the constraint vector y_0, and the second is that the constraint (13.7.6) may not be satisfied exactly for the known or given data, either because of errors in the physical data or through truncation errors. One solution to these difficulties is to use only simple boundary condition discretizations that maintain the symmetry of A, when that is possible. Unfortunately, this usually results in only first-order accurate boundary conditions (see Exercise 13.7.1).

If we delete one equation from the linear system (13.7.5) and arbitrarily fix one component of x, then the resulting system will usually be nonsingular. However, the accuracy of the solution will depend on which equation is deleted.

An approach that does not single out any particular equation or variable is to use the concept of a factor space. We consider two vectors v_1 and v_2 to be equivalent if their difference, $v_1 - v_2$, is a multiple of the null vector x_0. We consider equation (13.7.5) for solutions in the resulting factor space, which we denote by $R^K/\langle x_0 \rangle$. If we consider the data in the factor space $R^K/\langle y_0 \rangle$, then the system is nonsingular. If we do not know y_0, we can consider the data in $R^K/\langle x_0 \rangle$, and the system will be nonsingular as long as $y_0^T x_0$ is nonzero (see Exercise 13.7.3). We will assume that $y_0^T x_0$ is nonzero for each system we discuss.

This abstract reasoning is useful only if it leads to a useful and convenient algorithm. In this case it does, as we now illustrate. The norm of a vector x in $R^K/\langle x_0 \rangle$, where x_0 is the vector with all components equal to 1, is given by

$$\|x\| = \left(\sum_{\nu=1}^{K} (x_\nu - \overline{x})^2 \right)^{1/2},$$

where $\overline{x}$ is the average of the components x_ν. The equation being solved is no longer (13.7.5), but rather

$$Ax = b - \gamma x_0, \tag{13.7.7}$$

where γ is the average of $b - Ax$, i.e., the average residual. When a solution to (13.7.7) is obtained, the value of γ is an indication of how closely the data vector b satisfies the constraint. A nonzero value of γ can be due either to errors in the data or to the truncation errors implicit in the use of finite difference schemes.

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

We now give formulas for using this method on an elliptic equation. First, we write the finite difference equation at each grid point (ℓ, m) in the form

$$v_{\ell,m} - \sum L_{(\ell,m)(\ell',m')} v_{\ell',m'} - \sum U_{(\ell,m)(\ell',m')} v_{\ell',m'} = b_{\ell,m},$$

where L and U refer to the lower and upper triangular parts of the matrix. One sweep of SOR applied to this system may be described as follows. At each grid point (ℓ, m), the value of $r^k_{\ell,m}$, the update is computed:

$$r^k_{\ell,m} = \sum L_{(\ell,m)(\ell',m')} v^{k+1}_{\ell',m'} + \sum U_{(\ell,m)(\ell',m')} v^k_{\ell',m'} - v^k_{\ell,m} + b_{\ell,m}.$$

The value of $v^{k+1}_{\ell,m}$ is obtained as

$$v^{k+1}_{\ell,m} = v^k_{\ell,m} + \omega r^k_{\ell,m}.$$

The iteration continues until the updates are essentially constant, independent of (ℓ, m), i.e., until $\|r - \bar{r}\|$, the norm of the update in the factor space, is sufficiently small. To make the method efficient requires a convenient means of computing the average of the update and computing $\|r - \bar{r}\|$.

We now show how to compute both the average of the update and the norm of the update in the factor space. The algorithm for computing the averages and norms is due to West [70], who introduced it as an efficient means of computing averages and variances of statistical quantities. First, the variables $\bar{r}^{k+1}_0$ and $\bar{v}^{k+1}_0$, which will accumulate the average values of the update and v, respectively, are set to zero along with the variables R^k_0 and V^k_0, which will accumulate the norms of these quantities. It is also convenient to use the variable J to count the total number of points that have been updated.

At each grid point the accumulators of the norms are computed as

$$R^{k+1}_{J+1} = R^{k+1}_J + (r^{k+1}_{\ell,m} - \bar{r}^{k+1}_J)^2 \frac{J}{J+1},$$
$$V^{k+1}_{J+1} = V^{k+1}_J + (v^{k+1}_{\ell,m} - \bar{v}^{k+1}_J)^2 \frac{J}{J+1},$$

and then the averages are computed:

$$\bar{r}^{k+1}_{J+1} = \bar{r}^{k+1}_J + \frac{r^{k+1}_{\ell,m} - \bar{r}^{k+1}_J}{J+1},$$
$$\bar{v}^{k+1}_{J+1} = \bar{v}^{k+1}_J + \frac{v^{k+1}_{\ell,m} - \bar{v}^{k+1}_J}{J+1}.$$

The value of J is then incremented by 1, and the computation proceeds to the next grid point.

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

At the completion of one SOR sweep, the value of J will be equal to the total number of grid points at which values have been updated, which is K. The value of $\bar{r}_K^{k+1}$ will be equal to the average update and $\bar{v}_K^{k+1}$ will be the average value of v^{k+1}. The norms

$$\|v^{k+1} - \bar{v}^{k+1}\| = \left(\sum_{\ell,m} (v_{\ell,m}^{k+1} - \bar{v}^{k+1})^2 \Delta x \Delta y \right)^{1/2}$$

and

$$\|r^{k+1} - \bar{r}^{k+1}\| = \left(\sum_{\ell,m} (r_{\ell,m}^{k+1} - \bar{r}^{k+1})^2 \Delta x \Delta y \right)^{1/2}$$

will be equal to $(V_J^{k+1} \Delta x \Delta y)^{1/2}$ and $(R_J^{k+1} \Delta x \Delta y)^{1/2}$, respectively. The SOR iterations can be stopped when $\|r^{k+1} - \bar{r}^{k+1}\|$ is sufficiently small.

Example 13.7.1. We show results of using the factor space method and the method in which a specified variable is fixed in Table 13.7.1. The equation being solved is Poisson's equation

$$u_{xx} + u_{yy} = -5 \sin (x + 2y)$$

on the unit square with the normal derivative data being consistent with the solution

$$u(x, y) = \sin (x + 2y) + C. \tag{13.7.8}$$

The five-point Laplacian was used, and the boundary conditions were approximated by the second-order approximation (13.7.3).

The finite difference grid used equal grid spacing in each direction. The three different grid spacings are displayed in the first column of the table. The next columns show the number of iterations required to obtain a converged solution and the error in the solutions.

Table 13.7.1
Comparison of using factor space or fixing the center value.

	Factor method		Fixed center value		
h	Iterations	Error*	Iterations	Error*	Error**
0.100	55	3.40–3	95	4.47–3	2.38–2
0.050	93	9.38–4	241	1.13–3	7.70–3
0.025	200	2.47–5	958	2.87–4	2.37–3

*In the factor space L^2 norm.
**In the usual L^2 norm.

For each method the initial iterate was the grid function, which was identically zero. Each method was terminated when the appropriate norm of the change in the solution was less than 10^{-7}. This convergence criterion was sufficient to produce results for which the

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

error was primarily due to the truncation error. For the factor space method, the iteration parameter ω was chosen as $2/(1+\pi h/\sqrt{2})$, since π is the smallest eigenvalue for the Laplacian on the square with Neumann boundary conditions.

For the fixed-value method, the value of ω was $2/(1+h)$ for h equal to $1/10$, it was $2/(1+1.1h)$ for h equal to $1/20$, and it was $2/(1+2h)$ for h equal to $1/40$. These values give convergence but are not optimal. For this method, the exact value of the solution was fixed at the center point of the square; the constant in (13.7.8) was chosen so that $u(1/2, 1/2)$ was zero.

The solutions show the second-order accuracy of the finite difference methods when measured in the factor space norm. Notice that the error in the factor space norm is significantly smaller than in the usual L^2 norm. □

Example 13.7.2. Table 13.7.2 shows the results of using the factor space method on equation

$$e^{xy}u_{xx} + u_{yy} = f \tag{13.7.9}$$

on the unit square with Neumann boundary data. The values of f and the boundary data are determined by the exact solution

$$u(x, y) = e^{-xy}.$$

The last column gives the average update for the last iteration. It can be seen that the average update is quite small compared with the error. The results clearly show that the solution is second-order accurate.

This example is interesting because the integrability constraint is unknown. The integrability condition is discussed in Section 12.1 and is a linear relationship involving the boundary data and the data f in equation (13.7.9). The integrability condition must be satisfied for a solution to exist.

In spite of not knowing the integrability condition, the solution can be computed. The integrability constraint for this equation is difficult to obtain because this equation cannot be put into divergence form; see Exercise 12.7.3.

Table 13.7.2
The factor space method for a nonsymmetric equation.

h	Iteration	Error	$\bar{r}$
0.100	60	2.05–4	1.13–5
0.050	103	5.07–5	1.56–6
0.025	233	1.26–5	1.98–7

If a nonzero constant, say 1, is added to the value of f in (13.7.9), then the integrability condition is not satisfied. This method will compute a solution in the factor space, but the value of the average update, corresponding to γ in (13.7.7), will not be small, since the constraint is not close to being satisfied. □

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Exercises

13.7.1. Show that the five-point Laplacian and first-order accurate boundary condition (13.7.4) on the unit square give a symmetric matrix if the equations are scaled properly.

13.7.2. Using the maximum principle, show that equation (13.7.1) with boundary condition (13.7.2) has a unique solution to within an additive constant.

13.7.3. Consider a $K \times K$ matrix A that is singular with rank deficiency 1 and with a left null vector y_0 and right null vector x_0. Show that when considered as a linear mapping from the factor space $R^K/\langle x_0 \rangle$ to the factor space $R^K/\langle x_0 \rangle$, A is nonsingular if and only if the inner product of x_0 and y_0 is nonzero.

13.7.4. Solve Poisson's equation

$$u_{xx} + u_{yy} = -2\pi^2 \cos \pi x \cos \pi y$$

on the unit square with the Neumann boundary condition

$$\frac{\partial u}{\partial n} = 0.$$

The exact solution is $u(x, y) = \cos \pi x \cos \pi y$. Use both the first-order accurate approximation (13.7.4) and the second-order accurate approximation (13.7.3) to approximate the boundary conditions. Use equal grid spacing for both directions, and use grid spacings of $1/10$, $1/20$, and $1/40$. Use $\omega = 2/(1 + \pi h/\sqrt{2})$.

13.7.5. Consider the Jacobi iteration given by the five-point Laplacian on the unit square given by

$$v_{\ell,m}^{k+1} = \tfrac{1}{4}\,(v_{\ell+1,m}^{k} + v_{\ell-1,m}^{k} + v_{\ell,m+1}^{k} + v_{\ell,m-1}^{k})$$

for $\ell = 0, \ldots, N$ and $m = 0, \ldots, N$ with grid spacing h equal to N^{-1}. The boundary conditions are used to eliminate the variables $v_{\ell,m}$ with ℓ or m less than 0 or greater than N, with the relations

$$\begin{aligned} v_{-1,m} &= v_{1,m} && \text{for } m = 0, \ldots, N, \\ v_{\ell,-1} &= v_{\ell,1} && \text{for } \ell = 0, \ldots, N, \\ v_{N+1,m} &= v_{N-1,m} && \text{for } m = 0, \ldots, N, \\ v_{\ell,N+1} &= v_{\ell,N-1} && \text{for } \ell = 0, \ldots, N. \end{aligned}$$

Show that the eigenvalues are given by

$$\mu^{a,b} = \frac{1}{2}\left[\cos\left(\frac{a\pi}{N}\right) + \cos\left(\frac{b\pi}{N}\right)\right]$$

for $0 \le a, b \le N$ and the corresponding eigenvectors are

$$v_{\ell,m}^{a,b} = \cos\left(\frac{a\ell\pi}{N}\right) \cos\left(\frac{bm\pi}{N}\right).$$

Show that the Jacobi method will not converge in the factor space $R^K/\langle x_0\rangle$ in which x_0 is the vector with all components equal to 1. Show also that the Gauss–Seidel method will converge. This result does not contradict Theorem 13.5.1, since the Jacobi method in the factor space is not the true Jacobi method.

13.7.6. Show that the optimal value of ω for point SOR applied to the equations in Exercise 13.7.5 in the factor space is

$$\begin{aligned}\omega^* &= \frac{2}{1+\sin(\pi/2N)\sqrt{1+\cos^2(\pi/2N)}} \\ &\approx \frac{2}{1+\pi h/\sqrt{2}}.\end{aligned}$$

13.7.7. Verify that the following algorithm can be used to compute norms and vector products in the factor space $R^K/\langle x_0\rangle$, where x_0 is the vector with all components equal to 1:

Given vectors x and y in R^K, let σ_K and τ_K denote the factor space norms of x and y, respectively, and their inner product will be denoted by π_K. The quantities $\bar{x}_K$ and $\bar{y}_K$ are the averages of x and y, respectively.

The algorithm is: Set $\sigma_0 = 0$, $\tau_0 = 0$, $\pi_0 = 0$, $\bar{x}_0 = 0$, $\bar{y}_0 = 0$. Then for k from 0 to $K-1$, compute the quantities

$$\begin{aligned}\bar{x}_{k+1} &= \bar{x}_k + (x_{k+1} - \bar{x}_k)/(k+1), & \bar{y}_{k+1} &= \bar{y}_k + (y_{k+1} - \bar{y}_k)/(k+1), \\ \sigma_{k+1} &= \sigma_k + (x_{k+1} - \bar{x}_k)^2 k/(k+1), & \tau_{k+1} &= \tau_k + (y_{k+1} - \bar{y}_k)^2 k/(k+1), \\ \pi_{k+1} &= \pi_k + (x_{k+1} - \bar{x}_k)(y_{k+1} - \bar{y}_k)k/(k+1).\end{aligned}$$

Then, at the conclusion of the algorithm,

$$\begin{aligned}\bar{x}_K &= \textstyle\sum_{j=1}^K x_j/K = \bar{x}, & \bar{y}_K &= \textstyle\sum_{j=1}^K y_j/K = \bar{y}, \\ \sigma_K &= \textstyle\sum_{j=1}^K (x_j - \bar{x})^2, & \tau_K &= \textstyle\sum_{j=1}^K (y_j - \bar{y})^2, \\ \pi_K &= \textstyle\sum_{j=1}^K (x_j - \bar{x})(y_j - \bar{y}).\end{aligned}$$

Downloaded 01/06/21 to 129.241.228.92. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms