Chapter 3
Model Problems and Preconditioning

Abstract Model problem ADI iteration is discussed for three distinct classes of problems. The first is discretized elliptic systems with separable coefficients so that difference equations may be split into two commuting matrices. The second is where the model ADI problem approximates the actual nonseparable problem and serves as a preconditioner. The third is an entirely different class of problems than initially considered. These are Lyapunov and Sylvester matrix equations in which commuting operations are inherent.

3.1 Five-Point Laplacians

The heat-diffusion problem in two space dimensions was treated by Peaceman and Rachford (1955) in their seminal work on ADI iteration. They considered both time-dependent parabolic problems and steady-state elliptic problems. The Laplacian operator may be discretized over a rectangular region by standard differencing over a grid with spacing h in both the x and y directions. If one multiplies the equations by h^2, one obtains five-point interior equations with diagonal coefficient of 4 and off-diagonal coefficients of -1 connecting each interior node to its four nearest neighbors. Boundary conditions are incorporated in the difference equations. This is a model ADI problem when the boundary condition on each side is uniform. Given values need not be constant on a side, but one cannot have given value on part of the side and another condition like zero normal derivative on the remainder of the side. It was shown by Birkhoff, Varga and Young (1962) that there must be a full rectangular grid in order that model conditions prevail. For the Dirichlet problem (with values given on all boundaries), the horizontal coupling for a grid with m rows and n columns of unknowns when the equations are row-ordered is

$$H = \mathrm{diag}_m[L_n], \tag{1.1}$$

$$L_n = \mathrm{tridiag}_n[-1, 2, -1]. \tag{1.2}$$

E. Wachspress, *The ADI Model Problem*, DOI 10.1007/978-1-4614-5122-8_3,

The subscripts designate the orders of the matrices. The vertical coupling is similar with m and n interchanged when the equations are column-ordered. When row-ordered this coupling is

$$V = \mathrm{tridiag}_m[-I_n, 2I_n, -I_n], \tag{2}$$

where I_n is the identity matrix of order n. Matrices H and V commute and the simultaneous eigenvectors for $r = 1, 2, \ldots, m$ and $s = 1, 2, \ldots, n$ have components at the node in column i and row j of

$$v(r, s; i, j) = \sin\frac{ir\pi}{m+1}\,\sin\frac{js\pi}{n+1}. \tag{3}$$

The corresponding eigenvalues are

$$\lambda(H) = 2\left(1 - \cos\frac{s\pi}{n+1}\right), \tag{4.1}$$

$$\gamma(V) = 2\left(1 - \cos\frac{r\pi}{m+1}\right). \tag{4.2}$$

When the spacing is h along the x-axis and k along the y-axis, one may multiply the difference equations by the mesh-box area hk to yield matrices $H' = \frac{k}{h}H$ and $V' = \frac{h}{k}V$. The eigenvectors remain the same but the eigenvalues are now multiplied by these mesh ratios. It is seen that when the ratio of these increments (the "aspect ratio") differs greatly from unity, the spectra for the two directions differ significantly even when $m = n$. For optimal use of ADI iteration, one must consider the two-variable problem and apply Jordan's transformation to obtain parameters for use in the generalized equations, Eqs. 3 of Chap. 1.

Now consider variable increments, h_i between columns i and $i+1$ and k_j between rows j and $j+1$. The equation at node i, j may be normalized by the mesh-box area: $\frac{1}{4}(h_{i-1} + h_i)(k_{j-1} + k_j)$. Then

$$L_n = \mathrm{tridiag}_n\left[-\frac{1}{h_{i-1}(h_{i-1}+h_i)}, \frac{1}{2h_{i-1}h_i}, -\frac{1}{h_i(h_{i-1}+h_i)}\right]. \tag{5}$$

Note that the elements of L_n do not depend on the row index j. The eigenvalues of matrix H are now the eigenvalues of tridiagonal matrix L_n, each of multiplicity m. The Jordan normal form of this matrix is diagonal since it is the product of a positive diagonal matrix and a symmetric matrix. Bounds on these eigenvalues must be computed in order to determine optimum iteration parameters. If the V matrix is ordered by columns, then the corresponding diagonal blocks of order m are tridiagonal matrices with k_j replacing h_i in Eq. 5. Thus, column-ordered $V = \mathrm{tridiag}_n[S_m]$, with

$$S_m = \mathrm{tridiag}_m\left[-\frac{1}{k_{j-1}(k_{j-1}+k_j)}, \frac{1}{2k_{j-1}k_j}, -\frac{1}{k_j(k_{j-1}+k_j)}\right]. \tag{6}$$

Eigenvalue bounds for S_m must also be estimated for determining iteration parameters. Instead of dividing the equations by the mesh-box areas, we may retain the H and V matrices so that $H + V$ is the difference approximation to the differential operator integrated over the mesh box. We now multiply the iteration parameters by the normalizing (diagonal) matrix F whose entries are the mesh-box areas. This approach has ramifications which are beneficial in a more general context. Iteration Eq. 4 of Chap. 1 yield a matrix whose eigenvectors are independent of the iteration parameters when $HF^{-1}V - VF^{-1}H = 0$. This is evidently true for this case where $F^{-1}H$ and $F^{-1}V$ commute. Commutation is revealed by the fact that the elements in $F^{-1}H$ (which are displayed in Eq. 5) depend only on the index i while the elements in $F^{-1}V$ (which are displayed in Eq. 6) depend only on the index j. The spectra for which parameters are computed remain those of $F^{-1}H$ and $F^{-1}V$.

The ADI model-problem conditions are attainable in any orthogonal coordinate system for a full rectangular grid. When the Laplacian operator is discretized by integrating over the mesh box around node ij, the diagonal matrix of mesh-box areas is the appropriate matrix F. In fact, the first application of ADI iteration with Eq. 3 of Chap. 1 included cylindrical and polar coordinates [Wachspress, 1957].

A comparison with Fast Fourier Transform solution of such problems is revealing [Concus and Golub, 1973]. When the spacing is uniform in each direction, the eigensolutions are known. When high accuracy is desired the FFT outperforms ADI in this case. However, when only modest error reduction is demanded ADI is quite competitive. The FFT suffers somewhat when the number of rows or columns is not a power of two, but that is more a programming complication than a deficiency of the approach. Now consider variable increments. For ADI iteration we need only eigenvalue bounds. For the FFT we need the complete eigensolutions for both the H and the V matrices. This is time-consuming, and ADI in general outperforms FFT in such cases. Only when the same grid is used with many forcing vectors can FFT become competitive in this more general case. There are other "Fast Poisson Solvers" which may outperform ADI when very high accuracy is demanded [Buzbee, Golub and Nielson, 1970].

Eigenvalue bounds for the tridiagonal matrices, L_n and S_m, are relatively easy to compute. The maximum absolute row sum provides an adequate upper bound. The iteration is insensitive to loose (but conservative) upper bounds. Lower bounds can be computed with shifted inverse iteration, starting with a guess of zero. There is only one tridiagonal matrix for each direction and the time for the eigenvalue bound computation is negligible compared to the iteration time.

3.2 The Neutron Group-Diffusion Equation

The neutron group-diffusion equation is

$$-\nabla \cdot D(x,y)\nabla u(x,y) + \sigma(x,y)u(x,y) = s(x,y), \tag{7}$$

where $D(x, y) > 0$ and $\sigma(x, y) \geq 0$. This is an ADI model problem when the region is rectangular with uniform boundary condition on each side and the coefficients are separable in that

$$D(x, y) = D(x)D'(y) \text{ and } \sigma(x, y) = D(x)D'(y)[\sigma(x) + \sigma'(y)], \tag{8}$$

for we may then divide the equation by $D(x)D'(y)$ and express the operator as the sum of two commuting operators, $\mathcal{H}$ and $\mathcal{V}$, where

$$\mathcal{H} = \frac{1}{D(x)} \frac{\partial}{\partial x} D(x) \frac{\partial}{\partial x} + \sigma(x) \tag{9.1}$$

and

$$\mathcal{V} = \frac{1}{D'(y)} \frac{\partial}{\partial y} D'(y) \frac{\partial}{\partial y} + \sigma'(y). \tag{9.2}$$

This is a slight generalization of the model problem displayed by Young and Wheeler (1964) in which σ was restricted to $KD(x)D'(y)$ with K constant.

When the neutron group-diffusion equation is discretized by the box-integration method, the difference forms of Eqs. 9 are each three-point equations. We need not divide the equations by $D(x, y)$ if we define the F matrix by

$$F = \text{diag}[(i, j)] = GG' = \text{diag}[g(i)]\,\text{diag}[g'(j)], \tag{10}$$

where

$$g(i) = \frac{1}{2}[D_i h_i + D_{i-1} h_{i-1}], \tag{11.1}$$

and

$$g'(j) = \frac{1}{2}[D'_j k_j + D'_{j-1} k_{j-1}]. \tag{11.2}$$

In these equations, $D_i = D(x)$ between columns i and $i + 1$ while $D'_j = D'(y)$ between rows j and $j + 1$. The coefficient matrix obtained by box-integration can now be expressed as

$$A = LG' + L'G, \tag{12}$$

where for row-ordered equations

$$L \equiv \text{diagonal}_m[L_n], \tag{13}$$

with the matrix L_n repeated as the m diagonal blocks in L given by

$$L_n = \text{tridiagonal}\left\{-\frac{D_{i-1}}{h_{i-1}}, \left[D_{i-1}\left(\frac{1}{h_{i-1}} + \frac{h_{i-1}\sigma_{i-1}}{2}\right) + D_i\left(\frac{1}{h_i} + \frac{h_i\sigma_i}{2}\right)\right], -\frac{D_i}{h_i}\right\}, \tag{14}$$

and for column-ordered equations

$$L' \equiv \text{diagonal}_n[L'_m], \tag{15}$$

with the matrix L'_m repeated as the n diagonal blocks in L' given by

$$L'_m = \text{tridiagonal}\left\{-\frac{D'_{j-1}}{k_{j-1}}, \left[D'_{j-1}\left(\frac{1}{k_{j-1}}+\frac{k_{j-1}\sigma'_{j-1}}{2}\right)+D'_j\left(\frac{1}{k_j}+\frac{k_j\sigma'_j}{2}\right)\right], -\frac{D'_j}{k_j}\right\}. \tag{16}$$

Here, σ_i is the value between columns i and $i+1$ while σ'_j is the value between rows j and $j+1$.

The primed and unprimed matrices of order mn commute. The ADI equations can be expressed in the form

$$(LG' + w_s GG')\mathbf{u}_{s-\frac{1}{2}} = -(L'G - w_s GG')\mathbf{u}_{s-1} + \mathbf{s}, \tag{17.1}$$

$$(L'G + w'_s GG')\mathbf{u}_s = -(LG' - w'_s GG')\mathbf{u}_{s-\frac{1}{2}} + \mathbf{s}, \tag{17.2}$$

$$s = 1, 2, \ldots, J.$$

The right-hand side of Eq. 17.1 may be computed with the column-ordered block diagonal matrix L' and column-ordered $\mathbf{u}$ and $\mathbf{s}$. The resulting vector may then be reordered by rows as the forcing term for Eq. 17.1 with row ordering. Similarly, the right-hand side of Eq. 17.2 may be computed in row order and transposed to column order.

Eigenvalue bounds must be computed for the commuting tridiagonal matrices $G_n^{-1}L_n$ and $G_m'^{-1}L'_m$ for determining optimum parameters and associated convergence. These matrices are similar to SPD matrices and methods described for the model Laplace equation suffice for computing these eigenvalue bounds.

3.3 Nine-Point (FEM) Equations

When the Laplace or neutron group-diffusion operator is discretized by the finite element method over a rectangular mesh with bilinear basis functions, the equations are nine-point rather than five-point. It is by no means obvious that these are model ADI problems. Although Peaceman and Rachford introduced ADI iteration in the 1950s and the theory relating to convergence and choice of optimum parameters was in place by 1963, it was not until 1983 that I discovered how to express the nine-point equations as a model ADI problem [Wachspress, 1984]. The catalyst for this generalization was the analysis of the generalized five-point model problem discussed in Sect. 3.2 and in particular the form of the ADI iteration in Eqs. 17. This method was first implemented in 1990 [Dodds, Sofu and Wachspress], roughly 45 years after the seminal work by Peaceman and Rachford. One might question the practical worth of such effort in view of the restrictions imposed by the model conditions. However, application of model-problem analysis to more general problems will be exposed in Sect. 3.4.

Finite element discretization is based on a variational principle applied with a set of basis functions over each element. The basis functions from which the nine-point equations over a rectangular grid are obtained are bilinear. These nine-point finite element equations are related to the five-point box-integration equations.

A detailed analysis reveals that when the model conditions of Eq. 8 are satisfied, the finite element equations can be expressed as in Eq. 12:

$$A\mathbf{u} \equiv (LG' + L'G)\mathbf{u} = \mathbf{s}, \tag{18}$$

where we define the unprimed matrices when the equations are ordered by rows as

$$L \equiv \text{diagonal}_m[L_n], \tag{19.1}$$

$$G \equiv \text{diagonal}_m[G_n], \tag{19.2}$$

with tridiagonal matrices repeated as diagonal blocks:

$$L_n = \text{tridiagonal}\left\{ D_{i-1}\left(\frac{h_{i-1}\sigma_{i-1}}{6} - \frac{1}{h_{i-1}}\right), \right.$$
$$\left. \left[D_{i-1}\left(\frac{h_{i-1}\sigma_{i-1}}{3} + \frac{1}{h_{i-1}}\right) + D_i\left(\frac{h_i\sigma_i}{3} + \frac{1}{h_i}\right)\right],\ D_i\left(\frac{h_i\sigma_i}{6} - \frac{1}{h_i}\right)\right\} \tag{20}$$

and

$$G_n = \text{tridiagonal}[D_{i-1}h_{i-1},\ 2(D_{i-1}h_{i-1} + D_ih_i),\ D_ih_i]/6. \tag{21}$$

The primed matrices are of the same form when the equations are ordered by columns:

$$L' \equiv \text{diagonal}_n[L'_m], \tag{22.1}$$

$$G' \equiv \text{diagonal}_n[G'_m], \tag{22.2}$$

with tridiagonal matrices:

$$L'_m = \text{tridiagonal}\left\{ D'_{j-1}(\frac{k_{j-1}\sigma'_{j-1}}{6} - \frac{1}{k_{j-1}}), \right.$$
$$\left. \left[D'_{j-1}\left(\frac{k_{j-1}\sigma'_{j-1}}{3} + \frac{1}{k_{j-1}}\right) + D'_j\left(\frac{k_j\sigma'_j}{3} + \frac{1}{k_j}\right)\right],\ D'_j\left(\frac{k_j\sigma'_j}{6} - \frac{1}{k_j}\right)\right\} \tag{23}$$

and

$$G'_m = \text{tridiagonal}[D'_{j-1}k_{j-1},\ 2(D'_{j-1}k_{j-1} + D'_jk_j),\ D'_jk_j]/6. \tag{24}$$

The σ terms in the L and L' matrices are characteristic of finite element rather than box-integration equations, but this difference is sometimes eliminated by the "lumped mass" finite element approach which reduces the σ contribution to the box-integration diagonal contribution. Matrices L_n and L'_m in Eqs. 20 and 23 are then identical to matrices L_n and L'_m in Eqs. 14 and 16. This has no effect on the ADI analysis. The G and G' matrices are now tridiagonal diffusion-coefficient-weighted Simpson rule quadrature matrices. The fact that these matrices are tridiagonal rather than diagonal seems to preclude efficient ADI iteration, but we shall soon show how this is remedied.

We consider the ADI-type iteration defined in Eq. 17:

$$(LG' + w_s GG')\mathbf{u}_{s-\frac{1}{2}} = -(L'G - w_s GG')\mathbf{u}_{s-1} + \mathbf{s}, \tag{25.1}$$

$$(L'G + w'_s GG')\mathbf{u}_s = -(LF' - w'_s GG')\mathbf{u}_{s-\frac{1}{2}} + \mathbf{s}, \tag{25.2}$$

$$s = 1, 2, \ldots, J.$$

Since G and G' are tridiagonal rather than diagonal, the systems to be solved in each step are not block tridiagonal but have the same structure as the coefficient matrix A. They are systems of nine-point equations. We must somehow reduce these iteration equations to the form of Eqs. 1–3 with tridiagonal systems on the left-hand sides. For this purpose we define the vectors

$$\mathbf{v}_{s-\frac{1}{2}} = G'\mathbf{u}_{s-\frac{1}{2}} \tag{26.1}$$

and

$$\mathbf{v}_s = G\mathbf{u}_s. \tag{26.2}$$

One starts the iteration by computing $\mathbf{v}_0 = G\mathbf{u}_0$ and by virtue of commutativity of primed and unprimed matrices rewrites Eqs. 25 as

$$(L + w_s G)\mathbf{v}_{s-\frac{1}{2}} = -(L' - w_s G')\mathbf{v}_{s-1} + \mathbf{s}, \tag{27.1}$$

$$(L' + w'_s G')\mathbf{v}_s = -(L - w'_s G)\mathbf{v}_{s-\frac{1}{2}} + \mathbf{s}, \tag{27.2}$$

$$s = 1, 2, \ldots, J.$$

These equations are almost the same as the five-point iteration equations. They differ only in that the iteration parameters are multiplied by tridiagonal rather than diagonal matrices. However, the matrices on each side of these equations have the same structure as the corresponding five-point matrices. The coefficient matrix on the left side of Eq. 27.1 for update of all rows is the tridiagonal matrix $(L_n + w_s G_n)$, and the coefficient matrix on the left side of Eq. 27.2 for update of all columns is the tridiagonal matrix $(L'_m + w'_s G'_m)$. The iteration is terminated with recovery of $\mathbf{u}_J$ after J iterations by solving the tridiagonal systems $G\mathbf{u}_J = \mathbf{v}_J$.

The eigenvalue bounds for $G_n^{-1} L_n$ and $G'^{-1}_m L'_m$ must be computed. These may be treated as generalized eigenvalue problems: $L_n\mathbf{e} = \lambda G_n\mathbf{e}$ and $L'_m\mathbf{e}' = \gamma G'_m\mathbf{e}'$. Shifted inverse iteration has been used to compute upper and lower bounds for these eigenvalues. Some simple observations facilitate the computation. Matrices L_n and L'_m have positive inverses [Varga, 1962] and matrices G_n and G'_m are irreducible and nonnegative. Therefore, matrices $L_n^{-1} G_n$ and $L'^{-1}_m G'_m$ are positive. The Perron theorem asserts that the largest eigenvalues of these matrices have positive eigenvectors. If we choose $\mathbf{e}_0$ as a vector with all components equal to unity and solve the tridiagonal systems $L_n \mathbf{e}_1 = G_n \mathbf{e}_0$ and $L'_m \mathbf{e}'_1 = G'_m \mathbf{e}_0$, then the largest components of $\mathbf{e}_1$ and $\mathbf{e}'_1$ are upper bounds on the largest eigenvalues of these positive matrices. Their reciprocals are therefore lower bounds for the smallest eigenvalues of $G_n^{-1} L_n$ and $G'^{-1}_m L'_m$, respectively. These bounds may be used as a first shift in the computation of the lower eigenvalue bounds. First estimates for upper bounds may be computed with Rayleigh quotients $\frac{\mathbf{f}_0^T, L_n \mathbf{f}_0}{\mathbf{f}_0^T, G_n \mathbf{f}_0}$ and $\frac{\mathbf{f}_0^T, L'_m \mathbf{f}_0}{\mathbf{f}_0^T, G'_m \mathbf{f}_0}$, where the components of $\mathbf{f}_0$ alternate between plus one and minus one.

3.4 ADI Model-Problem Limitations

We have described a class of boundary value problems to which ADI model-problem theory applies. There is no other iterative method for which precise convergence prediction is possible that has the logarithmic dependence on problem condition. (We measure problem condition of an SPD system by the ratio of maximum to minimum eigenvalue of the coefficient matrix. This condition often varies as the number of nodes in the grid when spacing retains the same uniformity as the grid is refined.) Preconditioned conjugate gradient and multigrid computation may be competitive and even superior for some of these problems, but convergence theory is less definitive. Successive overrelaxation and Chebyshev extrapolation converge as the square root of the condition of the problem. For moderately difficult ADI model problems, the ADI iteration is more efficient. For example, the five-point Laplace problem with equal spacing and a 100×100 grid requires about 150 SOR iterations and only 10 ADI iterations for an error reduction by a factor of 10^{-4}. One model-problem ADI iteration, including both sweeps, requires about twice the work of one SOR iteration, but ADI has a clear advantage here. This advantage tends to manifest itself with smaller grids when mesh spacing is not uniform.

The greatest failing of ADI iteration is not in solution of model problems, but rather in restrictions imposed by the model conditions. Practitioners often demand methods which are applicable to a greater variety of problems. ADI iteration is often applied to problems for which model conditions are not met. Although considerable success has been realized for a variety of problems, departure from model conditions can lead to significant deterioration of the rapid convergence characteristic of ADI applied to model problems. Varga (1962) illustrated this with a simple problem contrived so that ADI iteration diverges with parameters chosen as though model conditions are satisfied when in reality they are not. Theory relating to parameter selection for general problems is sketchy. Although convergence can be guaranteed with some choices, the rate of convergence can rarely be predicted with variable parameters when the model conditions are not satisfied. It is this lack of sound theoretical foundations that motivated restriction of this monograph to application of ADI iteration only to model problems. In the next section we describe how model-problem ADI iteration may be applied to solve problems for which model conditions are not satisfied.

3.5 Model-Problem Preconditioners

3.5.1 Preconditioned Iteration

Several significant concepts were Introduced in Wachspress (1963). The Peaceman–Rachford ADI equations (Eq. 1 of Chap. 1) were generalized with different parameters for the two sweeps each iteration (Eqs. 1–3) to improve efficiency in solution of problems with different spectral intervals for the two directions. The earlier AGM algorithm for computing parameters when $J = 2^n$ (Sect. 1.4) was extended to this generalized iteration. This algorithm motivated Jordan's transformation of variables (Sect. 2–1.3). Both the variable transformation and Jordan's elliptic-function solution to the minimax problem were published for the first time as an appendix in Wachspress (1963).

The method now known as "preconditioned conjugate gradients" was also introduced in this paper as "compound iteration." Studies performed in 1962 established the potency of this new procedure, but the sparse numerical studies reported in this paper stimulated little interest and the method lay dormant for several years. It was rediscovered, was enhanced with a variety of preconditioners, and is now one of the more universally used methods for solving large elliptic type systems.

Compound iteration with ADI inner iteration was introduced by D'Yakonov (1961) to extend application of model problem ADI iteration to problems for which the model conditions were violated. The model problem was thus used as a "preconditioner" for the true problem. The term preconditioner was not introduced until several years after D'Yakonov's paper appeared. D'Yakonov used a two-term "outer" iteration with a constant extrapolation that converged about the same as Gauss–Seidel applied to the preconditioned system. The combination of ADI preconditioning and Lanczos-type[1] outer iteration was the new aspect of the analysis in my 1963 paper. This is in general much more efficient than Gauss–Seidel iteration.

[1] Nowadays, a variety of names are attached to variants of the Lanczos recursion formulas derived by minimizing different functionals. Forty years ago Gabe Horvay (a GE mechanics expert and one of my associates at KAPL) introduced me to this new approach developed by his friend Lanczos and, influenced strongly by Gabe, I became accustomed to referring to all these schemes as "Lanczos algorithms." Hence, the method of "conjugate gradients" is often referred to as "Lanczos' method" in my early works.

The following description of compound iteration is taken directly from the 1963 paper. The wording parallels that of modern texts on this method. I have been unable to find an earlier published account of preconditioned conjugate gradients, and refer the reader to the comprehensive historical review by Golub and O'Leary (1987).

3.5.2 Compound Iteration (Quotations from Wachspress, 1963)

"Application of compound iteration with inner iteration other than ADI was described by Cesari (1937) and by Engeli et al. (1959). Use of ADI iteration in this manner was discussed first by D'Yakonov (1961). We wish to solve the matrix equation $A\mathbf{z} = \mathbf{s}$ for $\mathbf{z}$ when given the vector $\mathbf{s}$ and the real, positive definite matrix A.[2] It is not often possible to express A as the sum of two symmetric commuting matrices, H and V, such that the matrix inversions in [Eq. 3 of Chap. 1] are readily performed. There may, however, be a model problem matrix M which approximates A in the sense that $p(M^{-1}A) << p(A)$, where p is the p-condition number, equal in this case to the ratio of the maximum to minimum eigenvalues. The closer $p(M^{-1}A)$ is to unity, the more efficient compound iteration becomes."

The paper continued with proof of a theorem on the effect of termination of the ADI model problem iteration with error reduction ε on the condition of this compound iteration. The ADI iteration actually replaces M by an SPD matrix B_ε. A more detailed proof with useful innovations will be given in Sect. 3.6. The theorem asserts that the effective condition is

$$p(B_\varepsilon^{-1}A) \leq \frac{1+\varepsilon}{1-\varepsilon} p(M^{-1}A). \tag{28}$$

Next, details were given for a symmetric conjugate gradient algorithm applied directly to the system $B^{-1}A$. This was the first published account of applicability of this algorithm to a product of SPD matrices. Hestenes and Stieffel (1952) discussed preconditioning of nonsymmetric systems with their transposes to yield symmetric systems. The observation that these algorithms could be applied to a product of SPD matrices is trivial and can be cast as application to A with inner products defined as $(\mathbf{w}\,,\,\mathbf{z}) = \mathbf{w}^T B^{-1}\mathbf{z}$. When B is SPD one can define a norm consistent with this inner product as $\|\mathbf{u}\| = (\mathbf{u}\,,\,\mathbf{u})^{\frac{1}{2}}$. The conjugate gradient algorithm then minimizes the norm of the residual vector.

After giving the conjugate gradient algorithm inner products and recursion formulas, my 1963 paper continued with: "The number of Lanczos iterations for a prescribed error reduction varies as $\sqrt{p(B^{-1}A)}$. [A footnote attributed this result to Lanczos being at least as efficient as Chebyshev extrapolation.] To gain some

[2] My definition of positive definite in those days implied symmetry. More recently, the term has been used by some with a different definition so that it is now customary to impose symmetry and denote A as "SPD" for "symmetric and positive definite." I still prefer the old definition in Wachspress, 1966, but approve wholeheartedly of the use of SPD to resolve any doubt.

insight regarding best strategy for compound iteration, we ... observe that the total number of ADI iterations ... varies as

$$J\sqrt{\frac{1+\varepsilon_J}{1-\varepsilon_J}p(M^{-1}A)}. \tag{29}$$

...When Jordan's [parameter selection] is used, J is optimum when ε_J is approximately equal to 0.36. In numerical application, however one must consider relative time requirements of inner and outer iterations...It may then be best to choose J so that ε_J is an order of magnitude smaller. This may increase the total number of inner (ADI) iterations, but the overall time may be reduced significantly...A desirable feature of compound iteration is that, having decided upon strategy according to machine limitations, one may find efficient iteration parameters with negligible computation time."

The paper continued with analysis of dependence on mesh spacing as a function of normalization of A in an attempt to approach model conditions and with numerical studies comparing different normalizations. The paper concluded with the statement that "Numerical results support prediction based on theory of rapid convergence rates in the numerical solution of the diffusion equation over a rectangular domain. Further studies are contemplated, including extension to nonrectangular domains." This latter study was pursued with a few examples in my 1966 book.

3.5.3 Updated Analysis of Compound Iteration

Although much of the early analysis is still valid, developments during the past 25 years have shed new light on this approach and have led to improvements. We first consider generation of a model problem. The early studies were done with the Laplace operator as a model for the diffusion operator with diffusion coefficient $D(x,y)$. D'Yakonov proved that $p(M^{-1}A)$ is equal to the ratio of the maximum to minimum values of $D(x,y)$. This is independent of grid geometry. Thus, the number of outer iterations is independent of spacing h as $h \to 0$. Computation time per iteration increases as h^{-2} and the number of inner ADI iterations per outer iteration to achieve a fixed error reduction increases as $\log\frac{1}{h}$.

In my 1984 paper an algorithm was presented for choosing a separable model problem to solve the diffusion equation in the absence of the σ term. This requires a "best" approximation to $D(x,y)$ by the separable coefficient $D(x)D'(y)$. If one considers the approximation of $\ln D(x,y)$ by $\ln D(x) + \ln D'(y)$, one has the problem treated by Diliberto and Strauss (1951): "On the approximation of a function of several variables by a sum of functions of fewer variables." In our application we have a precise measure of merit in that now

$$p(M^{-1}A) \le \frac{\max \frac{D(x,y)}{D(x)D'(y)}}{\min \frac{D(x,y)}{D(x)D'(y)}}. \tag{30}$$

[3]The algorithm for determining separable diffusion coefficients entails alternating improvement of $D(x)$ and $D'(y)$ until further improvement yields negligible reduction in p. The algorithm is:

1. For $i = 1, 2, \ldots, m$, set $D_i = 1.0$.
2. For $j = 1, 2, \ldots, n$, set
$$D'_j = \left(\max_i D_{ij} \cdot \min_i D_{ij} \right)^{\frac{1}{2}}.$$
3. For $i = 1, 2, \ldots, m$, set
$$D_i = \left(\max_j \frac{D_{ij}}{D'_j} \cdot \min_j \frac{D_{ij}}{D'_j} \right)^{\frac{1}{2}}.$$
4. For $j = 1, 2, \ldots, n$, set
$$D'_j = \left(\max_i \frac{D_{ij}}{D_i} \cdot \min_i \frac{D_{ij}}{D_i} \right)^{\frac{1}{2}}.$$
5. Cycle through steps 3 and 4 until values do not change appreciably. Convergence is quite rapid and high accuracy is not required. Two or three iterations often suffice.

The example given in [Wachspress, 1984] was for the pattern of diffusion coefficients in the matrix

$$D_{ij} = \begin{array}{ccc} 9 & 25 & 1 \\ 16 & 100 & 1600. \\ 1 & 4 & 36 \end{array}$$

The values for D_i and D'_j obtained by two cycles of the algorithm were

$D_1 = 6.931 \quad D_2 = 28.88 \quad D_3 = 23.10$
$D'_1 = 0.465 \quad D'_2 = 12.65 \quad D'_3 = 0.237$

This resulted in

$$D_i D'_j = \begin{array}{ccc} 1.643 & 6.845 & 5.475 \\ 87.677 & 365.332 & 292.215. \\ 3.223 & 13.429 & 10.742 \end{array}$$

The ratios of diffusion coefficients were then

$$\frac{D_{ij}}{D_i D'_j} = \begin{array}{ccc} 5.478 & 3.652 & 0.183 \\ 0.183 & 0.274 & 5.475. \\ 0.310 & 0.298 & 3.351 \end{array}$$

[3] Al Schatz (Cornell) advised me when I was preparing work on this preconditioner for publication that he had considered a related approximation for solving finite element problems but I have not yet seen a published reference to this work. His effort was devoted more to approximating equations over nonrectangular grids by preconditioning equations over rectangular grids.

Thus, $p(M^{-1}A) = \frac{5.478}{0.183} = 29.93$ in contrast with the Laplacian model-problem value of 1600. Since the solution effort varies as the square root of p, there is a gain by a factor greater than seven through use of the best separable problem. Note that the "best" D_i and $D'{}_j$ are not necessarily unique. In this example, $D'{}_1$ may vary within the interval $[0.286, 0.760]$ without increasing p.

For the more general diffusion equation with removal σ, we first compute the separable diffusion coefficient as above and then approximate $\tau_{ij} \equiv \frac{\sigma_{ij}}{D_i D'_j}$ by $\tau_i + \tau'_j$. One scheme which has been used successfully is to approximate $\exp(\tau_{ij})$ by the product $\exp(\tau_i)\exp(\tau'_j)$, using the same algorithm as for approximating the nonseparable diffusion coefficient. Care must be taken to disallow negative removal. This can be accomplished by replacing an exponential value less than unity by unity in the algorithm.

If $\alpha > \frac{D_{ij}}{D_i D'_j} > \frac{1}{\alpha}$ and $\beta > \frac{\tau_{ij}}{\tau_i + \tau'_j} > \frac{1}{\beta}$, then $p(M^{-1}A)$ is bounded by $(\alpha + \beta)^2$. Competition between diffusion and removal is a function of the geometry and changes with mesh spacing. The removal term will have its maximum effect on eigenvectors associated with smaller eigenvalues of the matrix A. The geometric buckling of a rectangle of length X and height Y is defined as $B^2 = (\frac{\pi^2}{X^2} + \frac{\pi^2}{Y^2})$. A reasonable estimate for p is

$$p(M^{-1}A) \doteq \frac{\max\limits_{ij} \frac{B^2 D_{ij} + \sigma_{ij}}{(B^2 + \tau_i + \tau'_j) D_i D'_j}}{\min\limits_{ij} \frac{B^2 D_{ij} + \sigma_{ij}}{(B^2 + \tau_i + \tau'_j) D_i D'_j}}. \tag{31}$$

The value computed in the absence of removal is precise when there is an interior node in each region of constant $D_i D'_j$. There is an eigenvector of $M^{-1}A$ with a component of unity at each such node and zero elsewhere belonging to the eigenvalue $\frac{D_{ij}}{D_i D'_j}$. In the absence of such interior nodes, the value computed is a close upper bound on p. The value in Eq. 31 is only an estimate that can be used to assess the model problem prior to the actual iteration. In the absence of removal, precise bounds are computable for the eigenvalues of $M^{-1}A$. This facilitates use of Chebyshev extrapolation as the outer iteration. In the absence of such bounds, conjugate gradient iteration seems preferable. The cost of the additional inner products is not significant.

3.6 Interaction of Inner and Outer Iteration

Let A be the coefficient matrix of the discretized diffusion operator $-\nabla \cdot D(x, y)\nabla$ over a rectangular partitioning of a rectangle, resulting from either five-point differencing or nine-point bilinear finite elements. The vector $\mathbf{u}$ whose components are the approximations to the desired field vector at the grid nodes is obtained as the solution to the linear system

$$A\mathbf{u} = \mathbf{b}, \tag{32}$$

where $\mathbf{b}$ is a given vector. Let B be the corresponding matrix with the separable diffusion coefficient $D(x)D'(y)$, and let the model-problem matrix equation be

$$B\mathbf{v} = \mathbf{r}\,. \tag{33}$$

Let F be the SPD normalizing matrix defined in Sect. 3.1 for which the matrix splitting $B = H + V$ satisfies $HF^{-1}V - VF^{-1}H = 0$. It follows that $F^{-1}B$ commutes with $F^{-1}H$ and $F^{-1}V$. For any matrix X, define $\tilde{X} = F^{-\frac{1}{2}}\, X\, F^{-\frac{1}{2}}$. Then if we define

$$\tilde{\mathbf{v}} \equiv F^{\frac{1}{2}}\mathbf{v},\;\; \tilde{\mathbf{r}} \equiv F^{-\frac{1}{2}}\mathbf{r}\,,\; \tilde{\mathbf{b}} \equiv F^{-\frac{1}{2}}\mathbf{b}\,,\;\; \text{and}\; \tilde{\mathbf{u}} \equiv F^{\frac{1}{2}}\mathbf{u}, \tag{34}$$

we have the transformed problem to be solved:

$$\tilde{A}\tilde{\mathbf{u}} = \tilde{\mathbf{b}}\,, \tag{35}$$

and the corresponding model problem:

$$\tilde{B}\tilde{\mathbf{v}} = \tilde{\mathbf{r}} \tag{36}$$

with $\tilde{B} = \tilde{H} + \tilde{V}$, where

$$\tilde{H}\tilde{V} - \tilde{V}\tilde{H} = F^{-\frac{1}{2}}[HF^{-1}V - VF^{-1}H]F^{-\frac{1}{2}} = 0. \tag{37}$$

Matrices $\tilde{A}$, $\tilde{B}$, $\tilde{H}$, and $\tilde{V}$ are all SPD. Let $\tilde{T}$ be the ADI iteration matrix for the symmetric normalized equations. This iteration matrix is symmetric with eigenvalues in the interval $[-\varepsilon, \varepsilon]$. The base matrix on which the outer iteration acts is

$$\tilde{W} = (I - \tilde{T})\tilde{B}^{-1}\tilde{A}, \tag{38}$$

where

$$\tilde{T}\tilde{B} - \tilde{B}\tilde{T} = 0. \tag{39}$$

A similarity transformation with $\tilde{B}^{\frac{1}{2}}$ yields

$$\tilde{W} \sim G \equiv (I - \tilde{T})\tilde{B}^{-\frac{1}{2}}\tilde{A}\tilde{B}^{-\frac{1}{2}}\,. \tag{40}$$

G is the product of two SPD matrices, $(I - \tilde{T})$ and $\tilde{B}^{-\frac{1}{2}}\tilde{A}\tilde{B}^{-\frac{1}{2}}$. Therefore, the eigenvalues of G are all real and positive and its Jordan normal form is diagonal. Let

$$b' \equiv \lambda_{\max}(\tilde{B}^{-1}\tilde{A}) = \lambda_{\max}(\tilde{B}^{-\frac{1}{2}}\tilde{A}\tilde{B}^{-\frac{1}{2}}), \tag{41}$$

and let

$$a' \equiv \lambda_{\min}(\tilde{B}^{-1}\tilde{A}) = \lambda_{\min}(\tilde{B}^{-\frac{1}{2}}\tilde{A}\tilde{B}^{-\frac{1}{2}}). \tag{42}$$

Let $b \equiv \lambda_{\max}(\tilde{W})$ and $a \equiv \lambda_{\min}(\tilde{W})$. Then

$$b \;\le\; \|G\| \;\le\; \|I - \tilde{T}\|\; \|\tilde{B}^{-\frac{1}{2}}\tilde{A}\tilde{B}^{-\frac{1}{2}}\| = (1+\varepsilon)b' \tag{43}$$

and

$$a \geq \|G^{-1}\|^{-1} \geq [\|(I-\tilde{T})^{-1}\| \, \|\tilde{B}^{\frac{1}{2}}\tilde{A}^{-1}\tilde{B}^{\frac{1}{2}}\|]^{-1} = (1-\varepsilon)a'. \tag{44}$$

Thus, we have as rigorous bounds on the eigenvalues of $\tilde{W}$:

$$a = (1-\varepsilon)a' \text{ and } b = (1+\varepsilon)b' \,. \tag{45}$$

The ADI equations are not normalized with the square-root matrix. The matrix on which the outer iteration acts is now $W = (I - T)B^{-1}A$. However, a similarity transformation with $F^{\frac{1}{2}}$ reveals that $W \sim \tilde{W}$. Hence, the eigenvalues of W are all real and positive with the same bounds, and the Jordan form of W is also diagonal. Let K be the matrix of eigenvectors of W. Then $W = K\mathbf{\Lambda}K^{-1}$ where $\mathbf{\Lambda}$ is the positive diagonal matrix of eigenvalues of W. Any polynomial $P_n(W)$ can be expressed as $P_n(W) = KP_n(\mathbf{\Lambda})K^{-1}$. Therefore,

$$\|P_n(W)\| \leq \|K\|\|K^{-1}\| \max_{\lambda}|P_n(\lambda)| = \kappa(K) \max_{a\leq\lambda\leq b}|P_n(\lambda)| \,, \tag{46}$$

where κ is the condition number of matrix K. When Chebyshev extrapolation is used for the outer iteration with the eigenvalue bounds a and b,

$$\max_{\lambda}|P_n(\lambda)| = \left(\cosh\left[n\cosh^{-1}\left(\frac{b+a}{b-a}\right)\right]\right)^{-1} . \tag{47}$$

Thus, the norm of the error reduction after n outer iterations, with inner ADI error reduction ε each outer iteration, is bounded by

$$\sigma = \kappa\left(\cosh\left[n\cosh^{-1}\left(\frac{b+a}{b-a}\right)\right]\right)^{-1} , \tag{48}$$

where the dependence on ε occurs through $a = (1-\varepsilon)a'$ and $b = (1+\varepsilon)b'$. Rigorous bounds on b' and a' are found readily. In finite element discretization, the contribution from rectangle q to $\mathbf{x}^\top A\ \mathbf{x}$ divided by the contribution to $\mathbf{x}^\top B\ \mathbf{x}$ is $\frac{D(x,y)}{D(x)D'(y)}|_q$. Therefore, the maximum eigenvalue of $B^{-1}A$ is equal to

$$b' = \max_{x,y}\frac{D(x,y)}{D(x)D'(y)}. \tag{49}$$

Similarly,

$$a' = \min_{x,y}\frac{D(x,y)}{D(x)D'(y)}. \tag{50}$$

Let point i, j be interior to a region of constant $D(x,y)$ and $D(x)D'(y)$. Then the vector with nonzero value only at i, j is an eigenvector of $B^{-1}A$ with eigenvalue equal to $\frac{D(x,y)}{D(x)D'(y)}$. Thus, the computed bounds are actually achieved in the presence of interior nodes. The other eigenvectors are in general not easily found and have components which are mostly nonzero. The separable model problem is generated to minimize the ratio b/a. Although the ADI inner iterations required to attain a prescribed error reduction increases logarithmically with grid refinement, the number of outer iterations remains fixed. Conjugate gradient outer iteration seems appropriate in the presence of space-dependent removal terms (as in neutron diffusion problems), but when accurate eigenvalue bounds are easily found Chebyshev extrapolation may be slightly more efficient since one then avoids the need for computing two inner products per iteration.

Optimum choice of the number of inner iterations per outer may be determined in advance by minimizing the work required for a prescribed accuracy. Each inner iteration requires about the same work as the residual evaluation for the next outer iteration. Let t be the number of inners per outer and s the number of outers. Then the total work varies as $f(t) = s(1+t)$. For significant error reduction, s varies as

$$s = C\sqrt{\frac{1+\varepsilon_t}{1-\varepsilon_t}}. \tag{51}$$

Optimum strategy often requires few inners per outer so that asymptotic inner iteration convergence estimates are not valid. The AGM algorithm for $t = 2^n$ is useful in this analysis. We define

$$\theta_1 \equiv \sqrt{k'}, \tag{52.1}$$

$$\theta_m \equiv \left[\frac{2\theta_{m-1}}{1+\theta_{m-1}^2}\right]^{\frac{1}{2}}. \tag{52.2}$$

The inner iteration error reduction for t iterations is

$$\varepsilon(t) = \left(\frac{1-\theta_t}{1+\theta_t}\right)^2. \tag{53}$$

The number of outer iterations s varies as $(\theta + \frac{1}{\theta})^{1/2}$, and

$$f(t=2^n) = C'(1+t)\left(\theta_t + \frac{1}{\theta_t}\right)^{\frac{1}{2}}. \tag{54}$$

The most efficient strategy depends on the value of k', and we examine a range of values (Table 3.1):

For most problems of interest, $k' << 0.01$ and a value close to $t = 4$ is optimum. One may compute $\varepsilon(t)$ by one of the methods described in Sect. 1.6 to optimize. For example, when $k' = 10^{-6}$, Eq. 1–54 gives

$$\begin{aligned}\varepsilon(t) &= 4\exp\left[-\frac{\pi^2 t}{\ln(4/10^{-6})}\right]\\ &= 4(0.5224)^t.\end{aligned} \tag{55}$$

For comparison with the values in the above table,

Table 3.1 Inner–outer iteration

k'	t	θ_t	$f(t)/C'$	t(opt)	ε
0.1	1	0.3162	3.73	1	0.2699
0.1	2	0.7582	6.23		
0.01	1	0.1000	6.36		
0.01	2	0.4450	4.92	2	0.1475
0.01	4	0.8619	7.11		
10^{-4}	1	0.01	20.0		
10^{-4}	2	0.1414	8.06		
10^{-4}	4	0.5317	7.76	4	0.093
10^{-4}	8	0.9105	12.8		
10^{-6}	1	0.001	63.3		
10^{-6}	2	0.0447	14.2		
10^{-6}	4	0.2991	9.54	4	0.291
10^{-6}	8	0.7410	13.0		
10^{-8}	4	0.1682	12.4	4	0.507

$$f(t) = C'\sqrt{2}(1+t)\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\frac{1}{2}}, \tag{56}$$

and we compute $f(3)/C' \doteq 10.82$ and $f(5)/C' \doteq 9.93$. For a fair comparison we reevaluate $f(4)/C'$ with this approximation as $f(4)/C' \doteq 9.61$. This does not differ appreciably from the value of 9.54 in the table. In this case, $t = 4$ is indeed optimal.

Having established that the number of outer iterations varies as $\sqrt{p(B^{-1}A)}$ and a means for relating the number of inner iterations per outer to k', we return to the question of whether or not a nine-point model preconditioner is more efficient than a five-point model preconditioner when A is a nine-point finite element matrix. The smallest eigenvalues of B_5 and B_9 do not differ significantly. However, the largest eigenvalues differ significantly in general. One can compute these values before actually deciding on the preconditioner for a particular problem. Some insight is gained by considering the discrete Laplacian with equal mesh spacing. The B_5 and B_9 matrices have common eigenvectors in this case. However, their eigenvalues differ. The maximum absolute row sum in B_5 is 8 and the corresponding value in B_9 is $16/3$. It follows that $p({B_5}^{-1}A) \doteq 1.5p({B_9}^{-1}A)$. The additional work of significance when the nine-point preconditioner is used is the recovery of the solution vector from the last iteration each cycle. This requires three flops per node. Each ADI inner iteration requires ten flops per node. Thus, if t inners are performed per outer, the additional work per outer with B_9 is by a factor of $(10t+3)/10t$. The work ratio of nine-point to five-point iteration is then approximately equal to $(10t+3)/(10t\sqrt{1.5}) = (10t+3)/12.247t$. This is greater than one only when $t = 1$. When $t = 4$, which is often close to optimal, the work saving through use of the nine-point preconditioner is by a factor of approximately 1.14. One must weigh the complexity of programming a nine-point preconditioner against the gain of approximately 14 % in computation efficiency. The effect of unequal spacing should be investigated.

3.7 Cell-Centered Nodes

The five-point Laplacian discussed in Sect. 3–3.1 and the nine-point FEM discretization described in Sect. 3–3.3 are both associated with vector components computed at intersections of grid lines. An alternative cell-centered formulation also enjoys widespread application. The discretization technique is exposed by considering the operator $-\frac{\mathrm{d}}{\mathrm{d}x}D(x)\frac{\mathrm{d}}{\mathrm{d}x}$ at segment i of width h_i and diffusion coefficient D_i. The right neighboring segment is of width h_{i+1} and has diffusion coefficient D_{i+1}. The equation is integrated over segment i. The coupling between i and $i+1$ is the two-point approximation to $[-D\frac{\mathrm{d}}{\mathrm{d}x}]$ at the right end of segment i. We assume a continuous piecewise linear solution between the cell centers with joint at the segment junction. Continuity of value and current $[-D\frac{\mathrm{d}}{\mathrm{d}x}]$ at this junction yields a value there in terms of the cell-centered values of

$$u_o = \frac{D_i h_{i+1} u_i + D_{i+1} h_i u_{i+1}}{D_i h_{i+1} + D_{i+1} h_i}. \tag{57}$$

The current $[-D\frac{\mathrm{d}}{\mathrm{d}x}]$ at the junction is then approximated by

$$D_i \frac{2(u_i - u_o)}{h_i} = \frac{2 D_i D_{i+1}}{D_i h_{i+1} + D_{i+1} h_i}(u_i - u_{i+1}). \tag{58}$$

We now consider solution of Poisson's equation with a separable approximation as a preconditioner: $-\bigtriangledown \cdot D(x,y) \bigtriangledown \mathbf{u}$ approximated by $-\bigtriangledown \cdot D(x)D(y) \bigtriangledown \mathbf{u}$. We prove that when the nonseparable cell diffusion coefficient $D_{i,j}$ is approximated by the separable $D_i D_j$, the eigenvalue bounds in Eqs. 49–50 are valid. Let $\alpha_{i,j} \equiv \frac{D_{i,j}}{D_i D_j}$ and let $\alpha \le \alpha_{i,j} \le 1/\alpha$. The ratio of the true coupling between nodes i,j and $i+1,j$ and the separable approximation is

$$\begin{aligned} R(i,j) &= \alpha_{i,j}\alpha_{i+1,j}\frac{D_j(h_{i+1}D_i + h_i D_{i+1})}{h_{i+1}D_{i,j} + h_i D_{i+1,j}} \\ &= \alpha_{i,j}\alpha_{i+1,j}\frac{h_{i+1}D_i + h_i D_{i+1}}{\alpha_{i,j}h_{i+1}D_i + \alpha_{i+1,j}h_i D_{i+1}}. \end{aligned} \tag{59}$$

It follows that $R(i,j)$ is in the interval $[\alpha_{i,j}, \alpha_{i+1,j}]$. All coefficient ratios satisfy similar relationships. Hence, the eigenvalues of $B^{-1}A$ are in the interval $\alpha, 1/\alpha$ as asserted when cell-centered equations are used for the true and the model problem.

3.8 The Lyapunov Matrix Equation

Let the $n \times n$ matrix A and the SPD $n \times n$ matrix C be given. Then the Lyapunov matrix problem is to find the symmetric matrix X such that

$$AX + XA^{\top} = C. \tag{60}$$

That this Lyapunov matrix equation (and more generally the Sylvester matrix equation $AX + XB = C$, where A is of order n and B of order m) is a model ADI problem was discovered in 1982 in connection with determination of "infinitesimal scaling" impedance matrices [Hurwitz, 1984] and [Wachspress, 1988a]. Although ADI was developed for application to SPD systems with real spectra, the iteration equations do not rely on symmetry. The model condition that the component matrices commute is retained. However, the SPD condition may be relaxed to require only that the eigenvalues of the coefficient matrix lie in the positive-real half plane. Such matrices are said to be "N-stable." (The eigenvalues of a "stable" matrix are in the negative-real half plane. The "N" in N-stable is for negative and this notation implies the double negative which flips the eigenvalues into the positive-real half plane.) When A is N-stable, it is known that Eq. 51 has a unique SPD solution matrix, X. A major deterrent to use of ADI iteration for solving elliptic partial differential equations is possible loss in convergence in the absence of a convenient commuting splitting. The N-stable Lyapunov matrix problem is seen to be a model ADI problem when one recognizes that this is equivalent to a linear operator $\mathcal{A}$ mapping X into C where $\mathcal{A}$ is the sum of the commuting operators: premultiplication of X by A and postmultiplication by $A^\top$. Thus, commutation is inherent in the Lyapunov application.

The ADI equations applied directly to Eq. 60 are

$$X_0 = \mathbf{0}, \tag{61.1}$$

$$(A + p_j I)X_{j-\frac{1}{2}} = C - X_{j-1}(A^\top - p_j I), \tag{61.2}$$

$$(A + p_j I)X_j = C - X^\top_{j-\frac{1}{2}}(A^\top - p_j I), \tag{61.3}$$

$$\text{with } j = 1, 2, \ldots, J.$$

Matrix X is not in general symmetric after the first sweep of each iteration, but the result of the double sweep is symmetric. Each row of grid points in ADI solution of a Laplacian-type system corresponds to a column of the matrix X and each column of the Laplace grid corresponds to a row of matrix X. Equation 61.3 is actually the transpose of the conventional ADI second step. An iterative method introduced by Smith (1968) is closely related to ADI with all the p_j the same. Each of Smith's iterations effectively doubles J at the expense of three matrix multiplications.

Application of ADI iteration to N-stable Lyapunov matrix equations requires generalization of the ADI theory into the complex plane. This is described in depth in Chap. 4. The initial work concerned generalization of the elliptic-function theory and was reported in a series of papers by Ellner (nee Saltzman), Lu, and Wachspress (1986–1991). This analysis centered around embedding a given spectrum in a region bounded by a curve of the form

$$\Gamma = \{z = b\, dn[u \pm ir, k] | 0 \le u \le 1\}. \tag{62}$$

Such regions were denoted as "elliptic-function" regions. Additional theory relating to ADI iteration with complex spectra and methods for determining optimal ADI parameters for spectra not well represented by the elliptic-function regions used in the earlier work were reported by Starke (1989). Alternative effective parameters for rectangular spectra were developed in [Wachspress, 1991]. Subsequent analysis by Istace and Thiran (1993) applied nonlinear optimization techniques to this problem.

Popular techniques for solving Eq. 60 include the method proposed by Smith (1968) and the B–S scheme developed by Bartels & Stewart (1972). The B–S algorithm requires about $15N^3$ flops to solve for X. In many applications, neither Smith's method nor ADI iteration is competitive with B–S when applied directly to a full matrix. Even if A has a known real spectrum so that the ADI theory is precise and convergence is rapid, each iteration requires several n^3 flops. It was found that one feasible technique which makes ADI iteration competitive is to first reduce the system to banded form. ADI iterative solution when A has bandwidth $b << n$ requires only $O(bn^2)$ flops. An additional advantage of this method is that the spectrum can be determined with little increase in computation time. This facilitates choice of iteration parameters for specific spectra.

Any similarity transformation with a matrix G reduces the Lyapunov equation to

$$SZ + ZS^\top = D, \tag{63}$$

where

$$S = GAG^{-1} \quad Z = GXG^\top \quad \text{and } D = GCG^\top. \tag{64}$$

Once Z is found, X may be recovered from

$$X = G^{-1}ZG^{-\top}. \tag{65}$$

Reduction to diagonal form yields the solution $z_{ij} = d_{ij}/(g_{ii} + g_{jj})$, but this reduction is too costly. It is equivalent to finding all the eigenvalues and eigenvectors of matrix A. When A is symmetric, Householder reduction to tridiagonal form is efficient and robust. The spectrum is real and ADI iteration rests on theory already described. When A is not symmetric, Householder reduction may be used to transform A into upper Hessenberg form, H. ADI iteration with H is often not competitive with B–S. One may attempt to reduce H to tridiagonal form with gaussian transformations. This is a classical problem in linear algebra, known to have many pitfalls [Wilkinson, 1965]. Large multipliers often arise and these lead to rapid loss in accuracy. Several researchers addressed this problem in seeking efficient means for finding the eigenvalues of A. [Dax and Kaniel, 1981; Hare and Tang, 1989; Tang, 1988; Watkins, 1988]. Once A or H is reduced to tridiagonal form, shifted LR transformations which preserve the band structure yield the eigenvalues more efficiently than the shifted QR transformations conventionally applied to H for this purpose. Wilkinson and later researchers showed that multipliers as large as $2^{\frac{t}{3}}$ would not detract from eigenvalue accuracy for calculations performed with roundoff error of order 2^t. However, for solution of the Lyapunov equation, more stringent bounds are needed.

In the first numerical studies of ADI applied to the Lyapunov equation, three features were introduced. First, the gaussian reduction was applied to the Hessenberg matrix by columns, starting at the last column. Second, a recovery algorithm was applied when a large multiplier was encountered. This consisted in creating a bulge at the $(n-2,n)$ element and chasing the bulge up to the "breakdown" column [Wachspress, 1988b]. Although this often succeeded, there were situations where this did not remedy the problem. To ensure robustness, on failure of the recovery algorithm, the offending column was left intact and the algorithm was continued. This resulted in a tridiagonal system (from bounded gaussian transformations) with a few added vertical "spikes" above the diagonal. Although this was reasonably successful for the ADI iteration, it was not suitable for the eigenvalue computation since the LR iterations fill in to a full Hessenberg matrix when there are spikes. The ADI iteration lost efficiency due to insufficient spectral knowledge.

[4] A significant variant introduced in [Geist, 1989] reduced rows and columns of A sequentially from row/col 1 to row/col n. Before each row/col reduction he permuted rows and columns in an attempt to reduce the magnitude of the gaussian multipliers. Such permutations were not possible when reducing from Hessenberg form. When the row and column to be reduced are close to orthogonal large multipliers cannot be avoided. Al's program was made robust by abandoning the reduction at the point of breakdown, applying a random Householder transformation to matrix A, and restarting the reduction. With a grant from ORNL, my graduate student at the University of Tennessee (An Lu) incorporated Geist's program ATOTRI into our ADI Lyapunov solver [Geist, Lu and Wachspress, 1989]. Geist's shifted LR eigenvalue solver was then used to determine the matrix spectrum for the ADI parameter optimization.

Although most problems are solved efficiently with this procedure, the lack of robustness and the computation time expended in recovery from breakdown detract from the method. Subsequently, motivated by discussions with Al and me at ORNL, Howell (1994) handled breakdown by allowing the bandwidth to expand above the diagonal. The row reduction lagged behind the column reduction with an increase in upper-half bandwidth each time another large multiplier was encountered. In the worst case, matrix A was reduced to upper Hessenberg form by stable gaussian transformations. Howell's program BHESS [Howell and Diaa, 2005] is well suited for ADI solution of the Lyapunov equation.

The success of Geist's permutation to reduce large multipliers was puzzling since after reducing the column (which was arbitrarily reduced first) the pivot for the row is small when the row and column to be reduced are nearly orthogonal. No initial permutation can change the product of the two pivots. Large multipliers from different row/col reductions can interact to yield large norms for the composite transformation matrix and its inverse. For the Lyapunov application one should monitor the accumulated condition number of the transformation matrix.

[4] While at the University of Tennessee in Knoxville I interacted with Al Geist at Oak Ridge and awakened his interest in gaussian reduction to tridiagonal form. Our work stimulated renewed interest by several mathematicians with whom we communicated.

In 1994 I suggested a BHESS modification (described in the Howell et al. paper) which could possibly reduce interaction of large multipliers and thereby improve stability. This has been realized and is developed in Chap. 5 with application to Lyapunov and Sylvester equations.

When excessive multiplication factors do not occur, theoretical improvement over the B–S method by a factor of around two is possible with combination of reduction to banded form followed by ADI iteration. The iterative method facilitates approximate solution when solving nonlinear (Riccati) equations with Newton iteration. Each Newton iteration requires solution of a Lyapunov equation. Another beneficial property of the iterative method is that it appears to be more readily parallelizable than the B–S method, in which QR transformations consume significant computer time. The ADI iteration itself on the banded equation requires $O(bn^2)$ flops. The arithmetic associated with the similarity transformation adds up to about $7n^3$ flops.

The B–S algorithm applies to all nonsingular matrices A. The ADI iteration applies directly only when A is N-stable. When A is nonsingular but not N-stable, it is possible to transform the problem into an equivalent N-stable system [Watkins, 1988]. However, this transformation may be too expensive to justify the entire procedure. The B–S scheme seems preferable in such cases. Fortunately for the ADI alternative, many of the problems encountered are N-stable. This is evidenced by widespread use over the years of the Smith algorithm which also requires N-stability.

The minimax theory was extended for the ADI problem in analogous fashion to the polynomial approximation problem [Opfer and Schober, 1984] with Rouché's theorem replacing the Chebyshev alternating extremes property. Elliptic-function regions play the role in ADI iteration of the ellipses in polynomial approximation. The logarithms of the elliptic-function regions are close to elliptical in shape. The theory is quite definitive and yields close to optimal parameters when the spectrum can be embedded in an elliptic-function region without excessive expansion. These regions have logarithmic symmetry with respect to the real and translated imaginary axes. When such regions are not appropriate one must seek alternative parameters [Starke, 1989; Wachspress, 1991; Istace and Thiran, 1993]. Fortunately, elliptic-function regions and unions of such regions apply to many problems of concern. The ADI minimax problem is more tractable than the corresponding polynomial problem in that when the parameters are positive or appear as conjugate pairs with positive real part the spectral radius of the iteration matrix is bounded by unity.

3.9 The Sylvester Matrix Equation

The Sylvester matrix equation

$$AX + XB = C \tag{66}$$

has a unique solution X for any C when there is no combination of eigenvalues $\lambda(A)$ and $\gamma(B)$ which sum to zero. The system is then said to be nonsingular. The ADI iteration is applicable only when the sum of the real parts is positive for all combinations. Although it is possible to construct from any nonsingular system another system with the same solution for which all real part combinations are positive, this construction often involves prohibitive computation. ADI iteration does not seem to be viable for such problems.

If A and B are symmetric, solution by the method of Golub, Nash and VanLoan (1979) is quite efficient, and ADI iteration is not competitive in this case. Reduction of one of A and B to tridiagonal form and the other to diagonal form with the symmetric QR algorithm provides a robust and elegant basis for solution of the Sylvester equation. On the other hand, when A and B are not symmetric, the Householder reduction to Hessenberg form does not yield a tridiagonal matrix. The method of Golub et al. requires further reduction of only one of these Hessenberg matrices to Schur form. Nevertheless, the additional work associated with reduction to Schur form of a matrix of order n takes about $13n^3$ flops. Thus, considerable time savings may be realized through use of gaussian reduction to banded form and ADI iterative solution of the reduced equations.

Let the similarity transformations that reduce A and B to the banded matrices S and T be G and H, respectively. Then the Sylvester equation reduces to

$$SZ + ZT = F, \tag{67.1}$$

$$\text{where } S = GAG^{-1}, \tag{67.2}$$

$$T = HBH^{-1}, \tag{67.3}$$

$$F = GCH^{-1}, \tag{67.4}$$

$$\text{and } Z = GXH^{-1}. \tag{67.5}$$

The spectra for A and $A^\top$ in the Lyapunov equation were the same. Hence, parameters p_j and q_j in Eq. 62 were the same for the two steps of each iteration applied to the Lyapunov equation. Here, the spectra of A and B differ in most cases and the more general two-variable ADI theory is applicable. A generalization to complex spectra of the transformation of W.B. Jordan described in Chap. 2 will be exposed in Chap. 4. This transformation provides a basis for choice of parameters p_j and q_j.

Once A and B have been reduced to S and T of bandwidth b, one can solve the Sylvester equation by ADI iteration with $O(bnm)$ flops per iteration, where A is of order n and B is of order m. The iteration equations for the reduced system are

$$Z_0 = \mathbf{0}, \tag{68.1}$$

$$(S + p_j I_n) Z_{j-\frac{1}{2}} = F - Z_{j-1}(T - p_j I_m), \tag{68.2}$$

$$(T^\top + q_j I_m) Z_j^\top = [F - (S - q_j I_n) Z_{j-\frac{1}{2}}]^\top, \tag{68.3}$$

$$\text{for } j = 1, 2, \ldots, , J$$

Let the right-hand sides in Eqs. 68.2 and 68.3 be denoted by $G_{j-\frac{1}{2}}$ and G_j. The ADI iteration arithmetic is reduced if one computes these terms recursively:

$$\text{For the first half step, } G_{\frac{1}{2}} = F \tag{69.1}$$

$$\text{and thereafter on the half steps } G_{j-\frac{1}{2}} = F + [(p_j + q_{j-1})Z_{j-1} - G_{j-1}]^\top. \tag{69.2}$$

$$\text{For the whole steps: } G_j = [F + (p_j + q_j)Z_{j-\frac{1}{2}} - G_{j-\frac{1}{2}}]^\top. \tag{69.3}$$

A rough estimate of the number of flops required to solve the Sylvester equation when $m = n$ is $21n^3$ for the Golub et al. method and $10n^3$ for the ADI method. The savings with iteration is essentially the flops associated with reduction of A or B from Hessenberg to Schur form. The iterative method uses $\frac{5}{3}(n^3 + m^3)$ flops to reduce A and B to banded form while accumulating the gaussian transformations, $nm(n + m)$ flops to transform the right-hand side, and another $nm(n + m)$ flops to recover X from Z. The estimate of $10n^3$ flops includes an allowance for the ADI iterations and verification of the approximate solution.

3.10 The Generalized Sylvester Equations

The generalized Sylvester equations may be expressed in the form

$$AX + YB = C, \tag{70}$$

$$EX - YF = G. \tag{71}$$

Matrices A and E are $n \times n$, B and F are $m \times m$, X, Y, C, and G are $n \times m$. These equations arise in solution of eigenvalue problems [Golub, Nash and VanLoan, 1979] and in control theory [Byers, 1983]. In these applications it is often true that

$$Re\lambda(E^{-1}A) + Re\lambda(BF^{-1}) > 0. \tag{72}$$

This is a stability condition which ensures existence of a unique solution to the generalized Sylvester equations. The ADI iteration equations for numerical solution of Eqs. 70–71 are

$$Y_0 = \mathbf{0}, \tag{73.1}$$

$$(A + p_j E)]X_j = C + p_j G - Y_{j-1}(B - p_j F), \tag{73.2}$$

$$(B^\top + q_j F^\top)Y_j^\top = [(C - q_j G) + (q_j E - A)X_j]^\top, \tag{73.3}$$

$$\text{for } j = 1, 2, \ldots, J.$$

These equations may be reduced to banded form. Let $S = HAE^{-1}H^{-1}$ and $T = KF^{-1}BK^{-1}$ be of bandwidth b. These matrices are computed in approximately $\frac{7}{2}(n^3 + m^3)$ flops. One also must compute $C' = HCK^{-1}$ and $G' = HGK^{-1}$. This takes $2nm(n + m)$ flops. The reduced equations are

$$Z_0 = \mathbf{0}, \tag{74.1}$$

$$(S + p_j I_n)V_j = C' + p_j G' - Z_{j-1}^\top (T - p_j I_m), \tag{74.2}$$

$$(T^\top + q_j I_m]Z_j = [C' - q_j G' - (S - q_j I_n)V_j]^\top, \tag{74.3}$$

$$\text{for } j = 1, 2, \ldots, J.$$

Note that each iteration updates both V and Z. A simple recursive relationship may be used to reduce the arithmetic in computing successive right-hand sides, denoted by L:

$$L_{\frac{1}{2}} = C' + p_1 G', \tag{75.1}$$

$$L_{j-\frac{1}{2}} = (C' + p_j G') + [(p_j + q_{j-1})Z_{j-1} - L_{j-1}]^\top, \tag{75.2}$$

$$L_j = [C' - q_j G' - L_{j-\frac{1}{2}} + (p_j + q_j)V_j]^\top. \tag{75.3}$$

This is an $O(bnm)$ algorithm with time small compared to the $O(n^3 + m^3)$ operations performed before and after solution of the equations. Matrices X and Y are recovered from V and Z with

$$X = E^{-1}H^{-1}V_J K \tag{76.1}$$

$$\text{and } Y = H^{-1}Z^\top KF^{-1}. \tag{76.2}$$

This requires another $3nm(n + m)$ flops. When $n = m$, the total arithmetic is thus around $17n^3$ flops.

3.11 The Three-Variable Laplacian-Type Problem

In Sect. 2.3 of Chap. 2 we discussed the three-variable ADI model problem and described an iteration designed primarily as a preconditioner. We will now examine this preconditioner in more detail. If we were able to use Eqs. 29.1 and 29.2 of Chap. 2 we would obtain the usual ADI preconditioning matrix, say

$$[B(t)]^{-1} = [I - M(t)]B^{-1}, \tag{77}$$

where $B = H + V + P$ is the model-problem matrix and $M(t)$ is the standard ADI iteration matrix for t double sweeps. The analysis already presented in this chapter would then apply. However, when Eqs. 2–32 are used, the preconditioner becomes

$$[B(t)]^{-1} = \left[I - \prod_{j=1}^{t} L_j \right] B^{-1}, \tag{78}$$

where L_j includes the inner ADI iteration matrix for double-sweep j of the ADI iteration. This preconditioner must be SPD for the conjugate gradient procedure to succeed. Since the L_j and B commute with one another, the preconditioner is symmetric. The norm of the ADI iteration matrix is now the spectral radius of

$$L(t) \equiv \prod_{j=1}^{t} L_j. \tag{79}$$

The spectral radius, say ε, of $L(t)$ must be less than unity for the preconditioner to be positive definite. Sufficient inner iterations must be performed to guarantee an SPD preconditioner. In Sect. 3.6 we discussed the interaction of inner ADI and outer CG iteration. We found that a value of ε of order magnitude 0.1 was reasonable and that $t = 4$ was often near optimal.

For three-variable iteration, the optimum value for t tends to be smaller than for corresponding two-variable problems. The smaller value for k'_j for the inner ADI iterations when w_j is small tends to reduce the relative efficiency as t is increased. Precise optimization of the inner ADI, the outer ADI, and the CG iteration is possible but requires evaluation of various options. This may be clarified by example.

The Dirichlet problem with Laplace's equation over a uniform grid with 100 nodes on each side yields $k' = 0.000281$ for the outer ADI iteration. The optimal parameters for $t = 4$ are [0.000507, 0.00508, 0.05536, 0.55439] in the transformed space. The corresponding error reduction with Eqs. 29.1 and 29.2 of Chap. 2 is 0.0644. The inner ADI iterations for an error reduction of $\varepsilon_j < 2k' = 0.000562$ satisfy (Table 3.2).

Table 3.2 Inner iterations for a 3D problem

j	k'_j	Inner iterations
1	0.00039	8
2	0.00268	7
3	0.027	5
4	0.217	3

The spectral radius of the ADI iteration is bounded by $\varepsilon_4 = \sqrt{0.0644} = 0.2537$. The number of CG iterations is increased by iterative approximation of the model-problem inverse by a factor of 1.3. The number of mesh sweeps per CG iteration is 50. Each CG acceleration requires about the work of around three mesh sweeps. We therefore estimate the work factor as $(1.3)(53) = 68.9$.

A similar computation for $t = 2$ yields $\varepsilon_2 = 0.69$. This is achieved with 11 inner ADI iterations for a total of 24 mesh sweeps per CG iteration. The CG loss factor is now 2.33 and the estimated work factor is $(2.33)(27) = 62.9$. This is slightly better than $t = 4$.

For a two-variable computation with the same value for k', four ADI iterations per CG step would yield a work factor of 11 and two ADI iterations per CG step a factor of $(1.678)(7) = 11.75$. Although the optimum number of ADI iterations per step is now greater, these computations display an insensitivity of efficiency to the number of ADI iterations per CG step with relatively few ADI iterations being optimal.

In three-variable iteration, insufficient inner ADI iterations lead to growth in high mode $H + V$ error components. These are the oscillatory modes and their growth is similar to that associated with roundoff instability. One must not confuse this behavior with roundoff error.