4

SOLUTION OF DISCRETE CONVECTION–DIFFUSION PROBLEMS

As shown in Chapter 3, the coefficient matrix arising from discretization of the convection–diffusion equation is nonsymmetric. To develop iterative solution algorithms for these problems, as well as those arising in other settings such as the Navier–Stokes equations, the algorithms discussed in Chapter 2 must be adapted to handle nonsymmetric systems of linear equations. In this chapter, we outline the strategies and issues associated with Krylov subspace iteration for general nonsymmetric systems, together with specific details for convection–diffusion systems associated with preconditioning and multigrid methods.

4.1 Krylov subspace methods

We are considering iterative methods for solving a system $F\mathbf{u} = \mathbf{f}$, where, for the moment (i.e. in this section), F represents an arbitrary nonsymmetric matrix of order n. Recall that for symmetric positive-definite systems, the conjugate gradient method has two properties that make it an effective iterative solution algorithm. It is *optimal*, in the sense that at the kth step, the energy norm of the error is minimized with respect to the k-dimensional Krylov space $\mathcal{K}_k(F, \mathbf{r}^{(0)})$. (Equivalently, the error is orthogonal to $\mathcal{K}_k(F, \mathbf{r}^{(0)})$ with respect to the energy inner product.) In addition, it is *inexpensive*: the number of arithmetic operations required at each step of the iteration is independent of the iteration count k. This also means that the storage requirements are fixed. Unfortunately, there are no generalizations of CG directly applicable to arbitrary nonsymmetric systems that have both of these properties. A Krylov subspace method for nonsymmetric systems of equations can display at most one of them: it can retain optimality but allow the cost per iteration to increase as the number of iterations grows, or it can require a fixed amount of computational work at each step but sacrifice optimality.

Before discussing what can be done, we note that one way to apply Krylov subspace methods to a nonsymmetric system is to simply create a symmetric positive definite one such as that defined by the normal equations $F^T F\mathbf{u} = F^T\mathbf{f}$. The problem could then be solved by applying the conjugate gradient method to the new system. This approach clearly inherits some of the favorable features of CG. However, the Krylov subspace generated is $\mathcal{K}_k(F^T F, F^T\mathbf{r}^{(0)})$ and therefore convergence will depend on properties of $F^T F$. For example, recall Theorem 2.4, which specifies a bound that depends on the condition number of the coefficient matrix. Since the condition number of F^TF is the square of that of F, this suggests that using CG in this way may be less effective than when it is applied directly to symmetric positive-definite systems. In our experience with problems arising in fluid mechanics such as the convection–diffusion equation, this is indeed the case; convergence of CG applied to the normal equations is slower than alternative approaches designed to be applied directly to nonsymmetric problems.

Let us consider instead iterative methods for systems with nonsymmetric coefficient matrices that generate a basis for $\mathcal{K}_k(F, \mathbf{r}^{(0)})$. Effective strategies are derived by exploiting the connection between algorithms for estimating eigenvalues of matrices (more precisely, for constructing nearly invariant subspaces of matrices) and those for solving systems. This connection was introduced in Section 2.4, where we established a relation between the conjugate gradient method and the Lanczos method for eigenvalues: the CG iterate is a linear combination of vectors generated by the Lanczos algorithm that constitute a basis for $\mathcal{K}_k(F, \mathbf{r}^{(0)})$. Here we will show how generalizations and variants of the Lanczos method for nonsymmetric matrices can be exploited in an analogous way.

4.1.1 *GMRES*

Our starting point is the *generalized minimum residual method* (GMRES), defined below. This algorithm, developed by Saad & Schultz [165], represents the standard approach for constructing iterates satisfying an optimality condition. It is derived by replacing the symmetric Lanczos recurrence (2.29) with the variant for nonsymmetric matrices known as the Arnoldi algorithm.

To show how this method works, we identify its relation to the Arnoldi method for eigenvalue computation. Starting with the initial vector $\mathbf{v}^{(1)}$, the main loop (on k) of Algorithm 4.1 constructs an orthonormal basis

$$\left\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(k)}\right\}$$

for the Krylov space $\mathcal{K}_k(F, \mathbf{v}^{(1)})$. To make $\mathbf{v}^{(k+1)}$ orthogonal to $\mathcal{K}_k(F, \mathbf{v}^{(1)})$, it is necessary to use all previously constructed vectors $\{\mathbf{v}^{(j)}\}_{j=1}^{k}$ in the computation. The construction in Algorithm 4.1 is analogous to the modified Gram–Schmidt process for generating an orthogonal basis. Let $V_k = [\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(k)}]$ denote the matrix containing $\mathbf{v}^{(j)}$ in its jth column, for $j = 1, \ldots, k$, and let $H_k = [h_{ij}]$, $1 \le i, j \le k$, where entries of H_k not specified in the Algorithm are zero. Thus, H_k is an upper-Hessenberg matrix (i.e. $h_{ij} = 0$ for $j < i - 1$), and

$$\begin{aligned} FV_k &= V_k H_k + h_{k+1,k}\,[0, \ldots, 0, \mathbf{v}^{(k+1)}] \\ H_k &= V_k^T F V_k \,. \end{aligned} \tag{4.1}$$

The Arnoldi method for eigenvalues is to use the eigenvalues of H_k as estimates for those of F. This technique is a generalization of the Lanczos method that is applicable to nonsymmetric matrices. When F is symmetric, H_k reduces to the tridiagonal matrix produced by the Lanczos algorithm, and (4.1) is identical to (2.30).

```
Algorithm 4.1: The Gmres Method
Choose $\mathbf{u}^{(0)}$, compute $\mathbf{r}^{(0)} = \mathbf{f} - F\mathbf{u}^{(0)}$, $\beta_0 = \|\mathbf{r}^{(0)}\|$, $\mathbf{v}^{(1)} = \mathbf{r}^{(0)}/\beta_0$
for $k = 1, 2, \ldots$ until $\beta_k < \tau\beta_0$ do
    $\mathbf{w}_0^{(k+1)} = F\mathbf{v}^{(k)}$
    for $l = 1$ to $k$ do
        $h_{lk} = \langle \mathbf{w}_l^{(k+1)}, \mathbf{v}^{(l)} \rangle$
        $\mathbf{w}_{l+1}^{(k+1)} = \mathbf{w}_l^{(k+1)} - h_{lk}\mathbf{v}^{(l)}$
    enddo
    $h_{k+1,k} = \|\mathbf{w}_{k+1}^{(k+1)}\|$
    $\mathbf{v}^{(k+1)} = \mathbf{w}_{k+1}^{(k+1)}/h_{k+1,k}$
    Compute $\mathbf{y}^{(k)}$ such that $\beta_k = \left\|\beta_0\mathbf{e}_1 - \widehat{H}_k\mathbf{y}^{(k)}\right\|$ is minimized,
        where $\widehat{H}_k = [h_{ij}]_{1\le i\le k+1, 1\le j\le k}$
enddo
$\mathbf{u}^{(k)} = \mathbf{u}^{(0)} + V_k\mathbf{y}^{(k)}$
```

To derive a Krylov subspace iteration for solving systems of equations, we let $\mathbf{u}^{(k)} \in \mathbf{u}^{(0)} + \mathcal{K}_k(F, \mathbf{r}^{(0)})$. For the choice $\mathbf{v}^{(1)} = \mathbf{r}^{(0)}/\beta_0$, with $\beta_0 = \|\mathbf{r}^{(0)}\|$ as in Algorithm 4.1, this is equivalent to

$$\mathbf{u}^{(k)} = \mathbf{u}^{(0)} + V_k\mathbf{y}^{(k)} \tag{4.2}$$

for some k-dimensional vector $\mathbf{y}^{(k)}$. But the first line of (4.1) can be rewritten as $FV_k = V_{k+1}\widehat{H}_k$, and this implies that the residual satisfies

$$\mathbf{r}^{(k)} = \mathbf{r}^{(0)} - AV_k\mathbf{y}^{(k)} = V_{k+1}\left(\beta_0\mathbf{e}_1 - \widehat{H}_k\mathbf{y}^{(k)}\right), \tag{4.3}$$

where $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$ is the unit vector of size k. The vectors $\{\mathbf{v}^{(j)}\}$ are pairwise mutually orthogonal, so that

$$\|\mathbf{r}^{(k)}\| = \beta_k = \left\|\beta_0\mathbf{e}_1 - \widehat{H}_k\mathbf{y}^{(k)}\right\|. \tag{4.4}$$

In particular, the residual of the iterate (4.2) with smallest Euclidean norm is determined by the choice of $\mathbf{y}^{(k)}$ that minimizes the expression on the right side of (4.4).

This upper-Hessenberg least squares problem can be solved by transforming $\widehat{H}_k$ into upper triangular form $\binom{R_k}{0}$, where R_k is upper triangular, using $k+1$ plane rotations (which are also applied to $\beta_0\mathbf{e}_1$). Here, $\widehat{H}_k$ contains $\widehat{H}_{k-1}$ as a submatrix, so that in a practical implementation, R_k can be updated from R_{k-1}. Moreover, by an analysis similar to that leading to (2.37), it can be shown that $\|\mathbf{r}^{(k)}\|$ is available at essentially no cost. Hence, a step of the GMRES algorithm consists of constructing a new Arnoldi vector $\mathbf{v}^{(k+1)}$, determining the residual norm of the iterate $\mathbf{r}^{(k)}$ that would be obtained from $\mathcal{K}_k(F, \mathbf{r}^{(0)})$, and then either constructing $\mathbf{u}^{(k)}$ if the stopping criterion is satisfied, or proceeding to the next step otherwise.

By construction, the iterate $\mathbf{u}^{(k)}$ generated by the GMRES method is the member of the translated Krylov space

$$\mathbf{u}^{(0)} + \mathcal{K}_k(F, \mathbf{r}^{(0)})$$

for which the Euclidean norm of the residual vector is minimal. That is,

$$\|\mathbf{r}^{(k)}\| = \min_{p_k \in \Pi_k,\, p_k(0)=1} \|p_k(F)\mathbf{r}^{(0)}\|\,. \tag{4.5}$$

As in the analysis of the CG method, the Cayley–Hamilton theorem implies that the exact solution is obtained in at most n steps. Bounds on the norm of the residuals associated with the GMRES iterates are derived from the optimality condition.

Theorem 4.1. *Let $\mathbf{u}^{(k)}$ denote the iterate generated after k steps of* GMRES *iteration, with residual $\mathbf{r}^{(k)}$. If F is diagonalizable, that is, $F = V\Lambda V^{-1}$ where Λ is the diagonal matrix of eigenvalues of F, and V is the matrix whose columns are the eigenvectors, then*

$$\frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{r}^{(0)}\|} \le \kappa(V) \min_{p_k \in \Pi_k,\, p_k(0)=1} \max_{\lambda_j} |p_k(\lambda_j)|, \tag{4.6}$$

where $\kappa(V) = \|V\|\,\|V^{-1}\|$ is the condition number of V. If, in addition, $\mathcal{E}$ is any set that contains the eigenvalues of F, then

$$\frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{r}^{(0)}\|} \le \kappa(V) \min_{p_k \in \Pi_k,\, p_k(0)=1} \max_{\lambda \in \mathcal{E}} |p_k(\lambda)|. \tag{4.7}$$

Proof Assertion (4.6) is derived from the observations that, for any polynomial p_k,

$$\begin{aligned} \|p_k(F)\mathbf{r}^{(0)}\| &= \|V p_k(\Lambda) V^{-1}\mathbf{r}^{(0)}\| \\ &\le \|V\|\,\|V^{-1}\|\,\|p_k(\Lambda)\|\,\|\mathbf{r}^{(0)}\| \\ &\le \|V\|\,\|V^{-1}\| \max_{\lambda_j} |p_k(\lambda_j)|\,\|\mathbf{r}^{(0)}\|. \end{aligned}$$

The bound (4.7) is an immediate consequence of (4.6). □

These "minimax" bounds generalize the analogous results (2.11) and (2.12) for the conjugate gradient method. There are, however, two significant differences. First, there is the presence of the condition number $\kappa(V)$ of the matrix of eigenvectors. It is difficult to bound this quantity, but its presence is unavoidable for polynomial bounds entailing the eigenvalues of F. Second, it is more difficult to derive an error bound for the GMRES iterates in a form that is as clean as

Theorem 2.4 for CG. This is partly due to the factor $\kappa(V)$, but it also depends on the need for bounds from approximation theory for $\max_\lambda |p_k(\lambda)|$.

The results of Theorem 4.1 can be used to gain insight into the convergence behavior of GMRES by taking the kth root of (either of) the bounds. In particular,

$$\left(\frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{r}^{(0)}\|}\right)^{1/k} \leq \kappa(V)^{1/k}\left(\min_{p_k\in\Pi_k,\, p_k(0)=1}\ \max_{\lambda_j}|p_k(\lambda_j)|\right)^{1/k}.$$

$\kappa(V)$ does not depend on k, and it therefore follows that $\kappa(V)^{1/k}\rightarrow 1$ as k increases. This suggests consideration of the limit

$$\rho := \lim_{k\rightarrow\infty}\left(\min_{p_k\in\Pi_k,\, p_k(0)=1}\ \max_{\lambda_j}|p_k(\lambda_j)|\right)^{1/k}. \tag{4.8}$$

Since GMRES constructs the exact solution in a finite number of steps, this does not lead to a simple statement about the error at any given step of the computation. However, it does give insight into the asymptotic behavior for large enough k: as the iteration proceeds, it can be expected that the norm of the residual will be reduced by a factor roughly equal to ρ at each step. We refer to ρ of (4.8) as the *asymptotic convergence factor* of the GMRES iteration. It is an interesting fact that asymptotic estimates for large k are often descriptive of observed convergence behavior for $k \ll n$. It is rarely the case that n iterations are necessary for an accurate solution to be obtained.

For later analysis, we mention that for a simple (stationary) iteration (as in (2.26)) with iteration matrix $T = M^{-1}R$, the norms of successive error vectors will asymptotically (for large numbers of iterations) reduce by a factor which is simply the eigenvalue of T of maximum modulus. We will therefore denote by $\rho(T)$ the eigenvalue of T of maximum modulus since this reflects the ultimate rate of convergence for a simple iteration as does ρ defined above for GMRES iteration.

Returning to GMRES, a bound on ρ can be obtained using the fact that any polynomial $\chi_k \in \Pi_k$ with $\chi_k(0) = 1$ satisfies, for any set $\mathcal{E}$ that contains the eigenvalues of F,

$$\min_{p_k\in\Pi_k,\, p_k(0)=1}\ \max_{\lambda\in\mathcal{E}}|p_k(\lambda)| \leq \max_{\lambda\in\mathcal{E}}|\chi_k(\lambda)|.$$

This was used in Theorem 2.4 to construct a bound on the error for the CG method, where χ_k was taken to be a scaled and translated Chebyshev polynomial. The same approach can be used to derive a bound on the asymptotic convergence factor for the GMRES method when the enclosing set $\mathcal{E}$ is an ellipse in the complex plane.

Theorem 4.2. *Suppose F is diagonalizable and its eigenvalues all lie in an ellipse $\mathcal{E}$ with center c, foci $c \pm d$ and semi-major axis a, and $\mathcal{E}$ does not contain the origin. Then the asymptotic convergence factor for* GMRES *iteration is*