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 [166], 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

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

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