Chapter 2

Some Iteration Methods

In this chapter the conjugate gradient (CG), minimal residual (MINRES), and generalized minimal residual (GMRES) algorithms are derived. These algorithms, designed for Hermitian positive definite, Hermitian indefinite, and general non-Hermitian matrices, respectively, each generate the "optimal" (in a sense, to be described later) approximation from the Krylov space (1.1).

The algorithms are first derived from other simpler iterative methods such as the method of steepest descent and Orthomin. This corresponds (roughly) to the historical development of the methods, with the optimal versions being developed as improvements upon nonoptimal algorithms. It shows how these algorithms are related to other iterative methods for solving linear systems.

Once it is recognized, however, that the goal in designing an iterative method is to generate the optimal approximation from the space (1.1), these methods can be derived from standard linear algebra techniques for locating the closest vector in a subspace to a given vector. This derivation is carried out in the latter part of section 2.4 for GMRES and in section 2.5 for MINRES and CG. This derivation has a number of advantages, such as demonstrating that the possible failure of CG for indefinite matrices corresponds to a singular tridiagonal matrix in the Lanczos algorithm and providing a basis for the derivation of other iterative techniques, such as those described in Chapter 5.

2.1. Simple Iteration.

Given a preconditioner M for the linear system $Ax = b$, a natural idea for generating approximate solutions is the following. Since a preconditioner is designed so that $M^{-1}A$ in some sense approximates the identity, $M^{-1}(b - Ax_k)$ can be expected to approximate the error $A^{-1}b - x_k$ in an approximate solution x_k. A better approximate solution x_{k+1} might therefore be obtained by taking

$$x_{k+1} = x_k + M^{-1}(b - Ax_k). \tag{2.1}$$

This procedure of starting with an initial guess x_0 for the solution and generating successive approximations using (2.1) for $k = 0, 1, \ldots$ is sometimes called *simple iteration*, but more often it goes by different names according to the choice of M. For M equal to the diagonal of A, it is called *Jacobi* iteration;

for M equal to the lower triangle of A, it is the *Gauss–Seidel* method; for M of the form $\omega^{-1}D - L$, where D is the diagonal of A, L is the strict lower triangle of A, and ω is a relaxation parameter, it is the successive overrelaxation or *SOR* method. Preconditioners will be discussed in Part II of this book, but our concern in this section is to describe the behavior of iteration (2.1) for a given preconditioner M in terms of properties of the preconditioned matrix $M^{-1}A$.

We will see in later sections that the simple iteration procedure (2.1) can be improved upon in a number of ways. Still, it is not to be abandoned. All of these improvements require some extra work, and if the iteration matrix $M^{-1}A$ is sufficiently close to the identity, this extra work may not be necessary. Multigrid methods, which will be discussed in Chapter 12, can be thought of as very sophisticated preconditioners used with the simple iteration (2.1).

An actual implementation of (2.1) might use the following algorithm.

Algorithm 1. Simple Iteration.

Given an initial guess x_0, compute $r_0 = b - Ax_0$, and solve $Mz_0 = r_0$.
For $k = 1, 2, \ldots,$

Set $x_k = x_{k-1} + z_{k-1}$.

Compute $r_k = b - Ax_k$.

Solve $Mz_k = r_k$.

Let $e_k \equiv A^{-1}b - x_k$ denote the error in the approximation x_k. It follows from (2.1) that

$$e_k = (I - M^{-1}A)e_{k-1} = \cdots = (I - M^{-1}A)^k e_0. \tag{2.2}$$

Taking norms on both sides in (2.2), we find that

$$|||e_k||| \leq |||(I - M^{-1}A)^k||| \cdot |||e_0|||, \tag{2.3}$$

where $|||\cdot|||$ can be any vector norm provided that the matrix norm is taken to be the one *induced* by the vector norm $|||B||| \equiv \max_{|||y|||=1} |||By|||$. In this case, the bound in (2.3) is *sharp*, since, for each k, there is an initial error e_0 for which equality holds.

LEMMA 2.1.1. *The norm of the error in iteration* (2.1) *will approach zero and* x_k *will approach* $A^{-1}b$ *for* every *initial error* e_0 *if and only if*

$$\lim_{k\to\infty} |||(I - M^{-1}A)^k||| = 0.$$

Proof. It is clear from (2.3) that if $\lim_{k\to\infty} |||(I - M^{-1}A)^k||| = 0$ then $\lim_{k\to\infty} |||e_k||| = 0$. Conversely, suppose $|||(I - M^{-1}A)^k||| > \alpha > 0$ for infinitely

many values of k. The vectors $e_{0,k}$ with norm 1 for which equality holds in (2.3) form a bounded infinite set in $\mathbf{C}^n$, so, by the Bolzano–Weierstrass theorem, they contain a convergent subsequence. Let e_0 be the limit of this subsequence. Then for k sufficiently large in this subsequence, we have $|||e_0 - e_{0,k}||| \leq \epsilon < 1$, and

$$\begin{aligned} |||(I - M^1A)^k e_0||| &\geq |||(I - M^1A)^k e_{0,k}||| - |||(I - M^1A)^k (e_{0,k} - e_0)||| \\ &\geq |||(I - M^1A)^k|||(1 - \epsilon) \geq \alpha(1 - \epsilon). \end{aligned}$$

Since this holds for infinitely many values of k, it follows that $\lim_{k\to\infty} |||(I - M^{-1}A)^k e_0|||$, if it exists, is greater than 0. □

It was shown in section 1.3.1 that, independent of the matrix norm used in (2.3), the quantity $|||(I - M^{-1}A)^k|||^{1/k}$ approaches the *spectral radius*, $\rho(I - M^{-1}A)$ as $k \to \infty$. Thus we have the following result.

THEOREM 2.1.1. *The iteration* (2.1) *converges to* $A^{-1}b$ *for every initial error* e_0 *if and only if* $\rho(I - M^{-1}A) < 1$.

Proof. If $\rho(I - M^{-1}A) < 1$, then

$$\lim_{k\to\infty} |||(I - M^{-1}A)^k||| = \lim_{k\to\infty} \rho(I - M^{-1}A)^k = 0,$$

while if $\rho(I - M^{-1}A) \geq 1$, then $\lim_{k\to\infty} |||(I - M^{-1}A)^k|||$, if it exists, must be greater than or equal to 1. In either case the result then follows from Lemma 2.1.1. □

Having established necessary and sufficient conditions for convergence, we must now consider the *rate* of convergence. How many iterations will be required to obtain an approximation that is within, say, δ of the true solution? In general, this question is not so easy to answer.

Taking norms on each side in (2.2), we can write

$$|||e_k||| \leq |||I - M^{-1}A||| \cdot |||e_{k-1}|||, \tag{2.4}$$

from which it follows that if $|||I - M^{-1}A||| < 1$, then the error is reduced by at least this factor at each iteration. The error will satisfy $|||e_k|||/|||e_0||| \leq \delta$ provided that

$$k \geq \log \delta / \log |||I - M^{-1}A|||.$$

It was shown in section 1.3.1 that for any $\epsilon > 0$, there is a matrix norm such that $|||I - M^{-1}A||| < \rho(I - M^{-1}A) + \epsilon$. Hence if $\rho(I - M^{-1}A) < 1$, then there is a norm for which the error is reduced *monotonically*, and convergence is at least *linear* with a reduction factor approximately equal to $\rho(I - M^{-1}A)$. Unfortunately, however, this norm is sometimes a very strange one (as might be deduced from the proof of Theorem 1.3.3, since the matrix D_t involved an exponential scaling), and it is unlikely that one would really want to measure convergence in terms of this norm!

It is usually the 2-norm or the ∞-norm or some closely related norm of the error that is of interest. For the class of *normal* matrices (diagonalizable

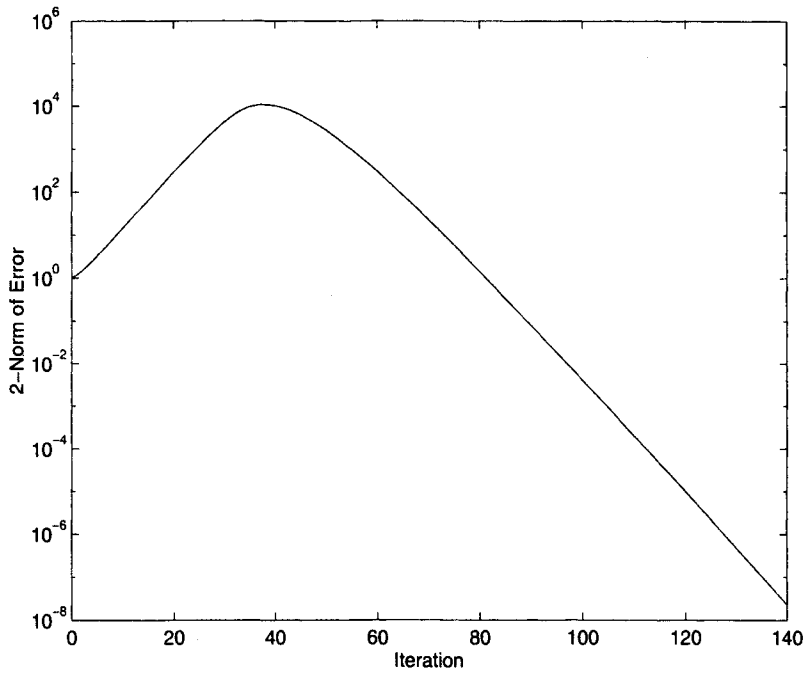


FIG. 2.1. *Simple iteration for a highly nonnormal matrix* $\rho = .74$.

matrices with a complete set of orthonormal eigenvectors), the 2-norm and the spectral radius coincide. Thus if $I - M^{-1}A$ is a normal matrix, then the 2-norm of the error is reduced by at least the factor $\rho(I - M^{-1}A)$ at each step. For nonnormal matrices, however, it is often the case that $\rho(I - M^{-1}A) < 1 < \|I - M^{-1}A\|$. In this case, the error may grow over some finite number of steps, and it is impossible to predict the number of iterations required to obtain a given level of accuracy while knowing only the spectral radius.

An example is shown in Figure 2.1. The matrix A was taken to be

$$A = \begin{pmatrix} 1 & -1.16 & & \\ .16 & \ddots & \ddots & \\ & \ddots & \ddots & -1.16 \\ & & .16 & 1 \end{pmatrix},$$

and M was taken to be the lower triangle of A. For problem size $n = 30$, the spectral radius of $I - M^{-1}A$ is about .74, while the 2-norm of $I - M^{-1}A$ is about 1.4. As Figure 2.1 shows, the 2-norm of the error increases by about four orders of magnitude over its original value before starting to decrease.

While the spectral radius generally does not determine the convergence rate of early iterations, it does describe the *asymptotic* convergence rate of (2.1). We will prove this only in the case where $M^{-1}A$ is diagonalizable and has a single eigenvalue of largest absolute value. Then, writing the eigendecomposition of $M^{-1}A$ as $V\Lambda V^{-1}$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and,

say, $\rho(I - M^{-1}A) = |1 - \lambda_1| > \max_{i>1} |1 - \lambda_i|$, we have

$$V^{-1}e_k = (I - \Lambda)^k (V^{-1}e_0).$$

Assuming that the first component of $V^{-1}e_0$ is nonzero, for k sufficiently large, the largest component of $V^{-1}e_k$ will be the first, $(V^{-1}e_k)_1$. At each subsequent iteration, this dominant component is multiplied by the factor $(1 - \lambda_1)$, and so we have

$$\begin{aligned} |||e_{k+j}||| &\approx |(V^{-1}e_{k+j})_1| = \rho(I - M^{-1}A)\ |(V^{-1}e_{k+j-1})_1| \\ &\approx \rho(I - M^{-1}A)\ |||e_{k+j-1}|||, \quad j = 1, 2, \ldots. \end{aligned}$$

Instead of considering the error ratios at *successive* steps as defining the asymptotic convergence rate, one might consider ratios of errors $(|||e_{k+j}|||/|||e_k|||)^{1/j}$ for any k and for j sufficiently large. Then, a more general proof that this quantity approaches the spectral radius $\rho(I - M^{-1}A)$ as $j \to \infty$ can be carried out using the Jordan form, as described in section 1.3.2. Note in Figure 2.1 that eventually the error decreases by about the factor $\rho(I - M^{-1}A) = .74$ at each step, since this matrix is diagonalizable with a single eigenvalue of largest absolute value.

2.2. Orthomin(1) and Steepest Descent.

In this section we discuss methods to improve on the simple iteration (2.1) by introducing dynamically computed parameters into the iteration. For ease of notation here and throughout the remainder of Part I of this book, we avoid explicit reference to the preconditioner M and consider A and b to be the coefficient matrix and right-hand side vector for the already preconditioned system. Sometimes we will assume that the preconditioned matrix is Hermitian. If the original coefficient matrix is Hermitian and the preconditioner M is Hermitian and *positive definite*, then one obtains a Hermitian preconditioned matrix by (implicitly) working with the modified linear system

$$L^{-1}AL^{-H}y = L^{-1}b, \quad x = L^{-H}y,$$

where $M = LL^H$ and the superscript ^{H} denotes the Hermitian transpose. If the original problem is Hermitian but the preconditioner is *indefinite*, then we consider this as a non-Hermitian problem.

One might hope to improve the iteration (2.1) by introducing a parameter a_k and setting

$$x_{k+1} = x_k + a_k(b - Ax_k). \tag{2.5}$$

Since the residual satisfies $r_{k+1} = r_k - a_k Ar_k$, one can minimize the 2-norm of r_{k+1} by choosing

$$a_k = \frac{\langle r_k, Ar_k \rangle}{\langle Ar_k, Ar_k \rangle}. \tag{2.6}$$

If the matrix A is Hermitian and positive definite, one might instead minimize the A-norm of the error, $\|e_{k+1}\|_A \equiv \langle e_{k+1}, Ae_{k+1}\rangle^{1/2}$. Since the error satisfies $e_{k+1} = e_k - a_k r_k$, the coefficient that minimizes this error norm is

$$a_k = \frac{\langle e_k, Ar_k\rangle}{\langle r_k, Ar_k\rangle} = \frac{\langle r_k, r_k\rangle}{\langle r_k, Ar_k\rangle}. \tag{2.7}$$

For Hermitian positive definite problems, the iteration (2.5) with coefficient formula (2.7) is called the method of *steepest descent* because if the problem of solving the linear system is identified with that of minimizing the quadratic form $x^H Ax - 2b^H x$ (which has its minimum where $Ax = b$), then the negative gradient or direction of steepest descent of this function at $x = x_k$ is $r_k = b - Ax_k$. The coefficient formula (2.6), which can be used with arbitrary nonsingular matrices A, does not have a special name like steepest descent but is a special case of a number of different methods. In particular, it can be called Orthomin(1).

By choosing a_k as in (2.6), the Orthomin(1) method produces a residual r_{k+1} that is equal to r_k minus its *projection* onto Ar_k. It follows that $\|r_{k+1}\| \leq \|r_k\|$ with equality if and only if r_k is already orthogonal to Ar_k. Recall the definition of the *field of values* $\mathcal{F}(B)$ of a matrix B as the set of all complex numbers of the form $y^H By/y^H y$, where y is any complex vector other than the zero vector.

THEOREM 2.2.1. *The 2-norm of the residual in iteration* (2.5) *with coefficient formula* (2.6) *decreases strictly monotonically for every initial vector* r_0 *if and only if* $0 \notin \mathcal{F}(A^H)$.

Proof. If $0 \in \mathcal{F}(A^H)$ and r_0 is a nonzero vector satisfying $\langle r_0, Ar_0\rangle \equiv r_0^H A^H r_0 = 0$, then $\|r_1\| = \|r_0\|$. On the other hand, if $0 \notin \mathcal{F}(A^H)$, then $\langle r_k, Ar_k\rangle$ cannot be 0 for any k and $\|r_{k+1}\| < \|r_k\|$. □

Since the field of values of A^H is just the complex conjugate of the field of values of A, the condition in the theorem can be replaced by $0 \notin \mathcal{F}(A)$.

Suppose $0 \notin \mathcal{F}(A^H)$. To show that the method (2.5–2.6) converges to the solution $A^{-1}b$, we will show not only that the 2-norm of the residual is reduced at each step but that it is reduced by at least some fixed factor, independent of k. Since the field of values is a closed set, if $0 \notin \mathcal{F}(A^H)$ then there is a positive number d—the distance of $\mathcal{F}(A^H)$ from the origin—such that $|\frac{y^H A^H y}{y^H y}| \geq d$ for all complex vectors $y \neq 0$. From (2.5) it follows that

$$r_{k+1} = r_k - a_k Ar_k,$$

and, taking the inner product of r_{k+1} with itself, we have

$$\langle r_{k+1}, r_{k+1}\rangle = \langle r_k, r_k\rangle - \frac{|\langle r_k, Ar_k\rangle|^2}{\langle Ar_k, Ar_k\rangle},$$

which can be written in the form

$$\|r_{k+1}\|^2 = \|r_k\|^2 \left(1 - \left|\frac{r_k^H A^H r_k}{r_k^H r_k}\right|^2 \cdot \left(\frac{\|r_k\|}{\|Ar_k\|}\right)^2\right). \tag{2.8}$$

Bounding the last two factors in (2.8) independently, in terms of d and $\|A\|$, we have

$$\|r_{k+1}\|^2 \leq \|r_k\|^2 \, (1 - d^2/\|A\|^2).$$

We have proved the following theorem.

THEOREM 2.2.2. *The iteration* (2.5) *with coefficient formula* (2.6) *converges to the solution* $A^{-1}b$ *for all initial vectors* r_0 *if and only if* $0 \notin \mathcal{F}(A^H)$. *In this case, the* 2*-norm of the residual satisfies*

$$\|r_{k+1}\| \leq \sqrt{1 - d^2/\|A\|^2} \, \|r_k\| \tag{2.9}$$

for all k, *where* d *is the distance from the origin to the field of values of* A^H.

In the special case where A is real and the Hermitian part of A, $H(A) \equiv (A + A^H)/2$, is positive definite, the distance d in (2.9) is just the smallest eigenvalue of $H(A)$. This is because the field of values of $H(A)$ is the real part of the field of values of A^H, which is convex and symmetric about the real axis and hence has its closest point to the origin on the real axis.

The bound (2.9) on the rate at which the 2-norm of the residual is reduced is not necessarily sharp, since the vectors r_k for which the first factor in (2.8) is equal to d^2 are not necessarily the ones for which the second factor is $1/\|A\|^2$. Sometimes a stronger bound can be obtained by noting that, because of the choice of a_k,

$$\|r_{k+1}\| \leq \|I - \alpha A\| \cdot \|r_k\| \tag{2.10}$$

for *any* coefficient α. In the special case where A is Hermitian and positive definite, consider $\alpha = 2/(\lambda_n + \lambda_1)$, where λ_n is the largest and λ_1 the smallest eigenvalue of A. Inequality (2.10) then implies that

$$\|r_{k+1}\| \leq \max_{i=1,\ldots,n} \left|1 - \frac{2\lambda_i}{\lambda_n + \lambda_1}\right| \cdot \|r_k\| \leq \left(\frac{\kappa - 1}{\kappa + 1}\right) \|r_k\|, \tag{2.11}$$

where $\kappa = \lambda_n/\lambda_1$ is the condition number of A. (For the Hermitian positive definite case, expression (2.9) gives the significantly weaker bound $\|r_{k+1}\| \leq \sqrt{1 - \kappa^{-2}} \, \|r_k\| \approx (1 - \frac{1}{2}\kappa^{-2}) \, \|r_k\|$.)

The same argument applied to the *steepest descent* method for Hermitian positive definite problems shows that for that algorithm,

$$\|e_{k+1}\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right) \|e_k\|_A. \tag{2.12}$$

In the more general non-Hermitian case, suppose the field of values of A^H is contained in a disk $\mathbf{D} = \{z \in \mathbf{C} : \ |z - \bar{c}| \leq s\}$ which does not contain the origin. Consider the choice $\alpha = 1/c$ in (2.10). It follows from (1.16-1.17) that

$$\mathcal{F}(I - (1/\bar{c})A^H) = 1 - (1/\bar{c})\mathcal{F}(A^H) \subseteq \{z \in \mathbf{C} : \ |z| \leq s/|c|\}.$$

Using relation (1.21) between the numerical radius and the norm of a matrix, we conclude that for this choice of α

$$\|I - \alpha A\| = \|I - (1/\bar{c})A^H\| \leq 2\frac{s}{|c|},$$

and hence

$$\|r_{k+1}\| \leq 2\,\frac{s}{|c|}\,\|r_k\|. \tag{2.13}$$

This estimate may be stronger or weaker than that in (2.9), depending on the exact size and shape of the field of values. For example, if $\mathcal{F}(A)$ is contained in a disk of radius $s = (\|A\| - d)/2$ centered at $c = (\|A\| + d)/2$, then (2.13) implies

$$\|r_{k+1}\| \leq 2\frac{1 - d/\|A\|}{1 + d/\|A\|}\,\|r_k\|.$$

This is smaller than the bound in (2.9) if $d/\|A\|$ is greater than about .37; otherwise, (2.9) is smaller.

Stronger results have recently been proved by Eiermann [38, 39]. Suppose $\mathcal{F}(A) \subseteq \Omega$, where Ω is a compact convex set with $0 \notin \Omega$. Eiermann has shown that if $\varphi_m(z) = F_m(z)/F_m(0)$, where $F_m(z)$ is the mth-degree Faber polynomial for the set Ω (the analytic part of $(\Phi(z))^m$, where $\Phi(z)$ maps the exterior of Ω to the exterior of the unit disk), then

$$\nu(\varphi_m(A)) \leq \max_{z\in\Omega} |\varphi_m(z)| \leq c_m \min_{p_m} \max_{z\in\Omega} |p_m(z)|, \tag{2.14}$$

where the minimum is over all mth-degree polynomials with value one at the origin and the constant c_m depends on Ω but is independent of A. For $m = 1$, as in (2.10), inequality (2.14) is of limited use because the constant c_m may be larger than the inverse of the norm of the first degree minimax polynomial on Ω. We will see later, however, that for methods such as (restarted) GMRES involving higher degree polynomials, estimate (2.14) can sometimes lead to very good bounds on the norm of the residual.

Orthomin(1) can be generalized by using different inner products in (2.6). That is, if B is a Hermitian positive definite matrix, then, instead of minimizing the 2-norm of the residual, one could minimize the B-norm of the residual by taking

$$a_k = \frac{\langle r_k, BAr_k\rangle}{\langle Ar_k, BAr_k\rangle}.$$

Writing a_k in the equivalent form

$$a_k = \frac{\langle B^{1/2}r_k, (B^{1/2}AB^{-1/2})B^{1/2}r_k\rangle}{\langle B^{1/2}Ar_k, B^{1/2}Ar_k\rangle},$$

it is clear that for this variant the B-norm of the residual decreases strictly monotonically for all r_0 if and only if $0 \notin \mathcal{F}((B^{1/2}AB^{-1/2})^H)$. Using arguments similar to those used to establish Theorem 2.2.2, it can be seen that

$$\|r_{k+1}\|_B \leq \sqrt{1 - d_B^2/\|B^{1/2}AB^{-1/2}\|^2}\ \|r_k\|_B, \tag{2.15}$$

where d_B is the distance from the origin to the field of values of $(B^{1/2}AB^{-1/2})^H$. If $0 \in \mathcal{F}(A^H)$, it may still be possible to find a Hermitian positive definite matrix B such that $0 \notin \mathcal{F}((B^{1/2}AB^{-1/2})^H)$.

2.3. Orthomin(2) and CG.

The Orthomin(1) and steepest descent iterations of the previous section can be written in the form

$$x_{k+1} = x_k + a_k p_k,$$

where the *direction vector* p_k is equal to the residual r_k. The new residual and error vectors then satisfy

$$r_{k+1} = r_k - a_k A p_k, \quad e_{k+1} = e_k - a_k p_k,$$

where the coefficient a_k is chosen so that either r_{k+1} is orthogonal to Ap_k (Orthomin(1)) or, in the case of Hermitian positive definite A, so that e_{k+1} is A-orthogonal to p_k (steepest descent). Note, however, that r_{k+1} is not orthogonal to the previous vector Ap_{k-1} in the Orthomin(1) method and e_{k+1} is not A-orthogonal to the previous vector p_{k-1} in the steepest descent method.

If, in the Orthomin iteration, instead of subtracting off the projection of r_k in the direction Ar_k we subtracted off its projection in a direction like Ar_k but orthogonal to Ap_{k-1}, i.e., in the direction $A\tilde{p}_k$, where

$$\tilde{p}_k = r_k - \frac{\langle Ar_k, Ap_{k-1}\rangle}{\langle Ap_{k-1}, Ap_{k-1}\rangle} p_{k-1},$$

then we would have

$$\langle r_{k+1}, A\tilde{p}_k\rangle = 0 \text{ and } \langle r_{k+1}, Ap_{k-1}\rangle = \langle r_k, Ap_{k-1}\rangle - a_k\langle A\tilde{p}_k, Ap_{k-1}\rangle = 0.$$

Now the residual norm is minimized in the *plane* spanned by Ar_k and Ap_{k-1}, since we can write

$$r_{k+1} = r_k - a_k A r_k + a_k b_{k-1} A p_{k-1},$$

and the coefficients force orthogonality between r_{k+1} and $\text{span}\{Ar_k, Ap_{k-1}\}$.

The new algorithm, known as Orthomin(2), is the following.

Given an initial guess x_0, compute $r_0 = b - Ax_0$ and set $p_0 = r_0$.
For $k = 1, 2, \ldots,$
Compute Ap_{k-1}.
Set $x_k = x_{k-1} + a_{k-1}p_{k-1}$, where $a_{k-1} = \frac{\langle r_{k-1}, Ap_{k-1}\rangle}{\langle Ap_{k-1}, Ap_{k-1}\rangle}$.
Compute $r_k = r_{k-1} - a_{k-1}Ap_{k-1}$.
Set $p_k = r_k - b_{k-1}p_{k-1}$, where $b_{k-1} = \frac{\langle Ar_k, Ap_{k-1}\rangle}{\langle Ap_{k-1}, Ap_{k-1}\rangle}$.

The new algorithm can also fail. If $\langle r_0, Ar_0\rangle = 0$, then r_1 will be equal to r_0, and p_1 will be 0. An attempt to compute the coefficient a_1 will result in dividing 0 by 0. As for Orthomin(1), however, it can be shown that Orthomin(2) cannot fail if $0 \notin \mathcal{F}(A^H)$. If an Orthomin(2) step does succeed, then the 2-norm of the residual is reduced by at least as much as it would be reduced by an Orthomin(1) step from the same point. This is because the residual norm is minimized over a larger space—$r_k + \text{span}\{Ar_k, Ap_{k-1}\}$ instead of $r_k + \text{span}\{Ar_k\}$. It follows that the bound (2.9) holds also for Orthomin(2) when $0 \notin \mathcal{F}(A^H)$. Unfortunately, no stronger a priori bounds on the residual norm are known for Orthomin(2) applied to a general matrix whose field of values does not contain the origin although, in practice, it may perform significantly better than Orthomin(1).

In the special case when A is *Hermitian*, if the vectors at steps 1 through $k+1$ of the Orthomin(2) algorithm are defined, then r_{k+1} is minimized not just over the two-dimensional space $r_k + \text{span}\{Ar_k, Ap_{k-1}\}$ but over the $(k+1)$-dimensional space $r_0 + \text{span}\{Ap_0, \ldots, Ap_k\}$.

THEOREM 2.3.1. *Suppose that A is Hermitian, the coefficients $a_0, \ldots, a_{k-1}$ are nonzero, and the vectors $r_1, \ldots, r_{k+1}$ and $p_1, \ldots, p_{k+1}$ in the Orthomin(2) algorithm are defined. Then*

$$\langle r_{k+1}, Ap_j\rangle = \langle Ap_{k+1}, Ap_j\rangle = 0 \quad \forall j \le k.$$

It follows that of all vectors in the affine space

$$r_0 + \text{span}\{Ar_0, A^2r_0, \ldots, A^{k+1}r_0\}, \tag{2.16}$$

r_{k+1} has the smallest Euclidean norm. It also follows that if $a_0, \ldots, a_{n-2}$ are nonzero and $r_1, \ldots, r_n$ and $p_1, \ldots, p_n$ are defined, then $r_n = 0$.

Proof. By construction, we have $\langle r_1, Ap_0\rangle = \langle Ap_1, Ap_0\rangle = 0$. Assume that $\langle r_k, Ap_j\rangle = \langle Ap_k, Ap_j\rangle = 0 \; \forall j \le k-1$. The coefficients at step $k+1$ are chosen to force $\langle r_{k+1}, Ap_k\rangle = \langle Ap_{k+1}, Ap_k\rangle = 0$. For $j \le k-1$, we have

$$\langle r_{k+1}, Ap_j\rangle = \langle r_k - a_k Ap_k, Ap_j\rangle = 0$$

by the induction hypothesis. Also,

$$\begin{aligned}
\langle Ap_{k+1}, Ap_j\rangle &= \langle A(r_{k+1} - b_k p_k), Ap_j\rangle \\
&= \langle Ar_{k+1}, a_j^{-1}(r_j - r_{j+1})\rangle \\
&= \bar{a}_j^{-1}\langle Ar_{k+1}, p_j + b_{j-1}p_{j-1} - p_{j+1} - b_j p_j\rangle \\
&= \bar{a}_j^{-1}\langle r_{k+1}, A(p_j + b_{j-1}p_{j-1} - p_{j+1} - b_j p_j)\rangle \\
&= 0,
\end{aligned}$$

with the next-to-last equality holding because $A = A^H$. It is justified to write a_j^{-1} since, by assumption, $a_j \neq 0$.

It is easily checked by induction that r_{k+1} lies in the space (2.16) and that $\text{span}\{Ap_0, \ldots, Ap_k\} = \text{span}\{Ar_0, \ldots, A^{k+1}r_0\}$. Since r_{k+1} is orthogonal

to span$\{Ap_0, \ldots, Ap_k\}$, it follows that r_{k+1} is the vector in the space (2.16) with minimal Euclidean norm. For $k = n - 1$, this implies that $r_n = 0$. □

The assumption in Theorem 2.3.1 that $r_1, \ldots, r_{k+1}$ and $p_1, \ldots, p_{k+1}$ are defined is actually implied by the other hypothesis. It can be shown that these vectors are defined provided that $a_0, \ldots, a_{k-1}$ are defined and nonzero and r_k is nonzero.

An algorithm that approximates the solution of a Hermitian linear system $Ax = b$ by minimizing the residual over the affine space (2.16) is known as the *MINRES* algorithm. It should not be implemented in the form given here, however, unless the matrix is positive (or negative) definite because, as noted, this iteration can fail if $0 \in \mathcal{F}(A)$. An appropriate implementation of the MINRES algorithm for Hermitian indefinite linear systems is derived in section 2.5.

In a similar way, the steepest descent method for Hermitian positive definite matrices can be modified so that it eliminates the A-projection of the error in a direction that is already A-orthogonal to the previous direction vector, i.e., in the direction

$$\tilde{p}_k = r_k - \frac{\langle r_k, Ap_{k-1}\rangle}{\langle p_{k-1}, Ap_{k-1}\rangle} p_{k-1}.$$

Then we have

$$\langle e_{k+1}, A\tilde{p}_k\rangle = \langle e_{k+1}, Ap_{k-1}\rangle = 0,$$

and the A-norm of the error is minimized over the two-dimensional affine space $e_k + \text{span}\{r_k, p_{k-1}\}$. The algorithm that does this is called the *CG* method. It is usually implemented with slightly different (but equivalent) coefficient formulas, as shown in Algorithm 2.

Algorithm 2. Conjugate Gradient Method (CG)
(for Hermitian positive definite problems)

Given an initial guess x_0, compute $r_0 = b - Ax_0$ and set $p_0 = r_0$.
For $k = 1, 2, \ldots,$

Compute Ap_{k-1}.

Set $x_k = x_{k-1} + a_{k-1}p_{k-1}$, where $a_{k-1} = \frac{\langle r_{k-1}, r_{k-1}\rangle}{\langle p_{k-1}, Ap_{k-1}\rangle}$.

Compute $r_k = r_{k-1} - a_{k-1}Ap_{k-1}$.

Set $p_k = r_k + b_{k-1}p_{k-1}$, where $b_{k-1} = \frac{\langle r_k, r_k\rangle}{\langle r_{k-1}, r_{k-1}\rangle}$.

It is left as an exercise for the reader to prove that these coefficient formulas are equivalent to the more obvious expressions

$$a_{k-1} = \frac{\langle r_{k-1}, p_{k-1}\rangle}{\langle p_{k-1}, Ap_{k-1}\rangle}, \quad b_{k-1} = -\frac{\langle r_k, Ap_{k-1}\rangle}{\langle p_{k-1}, Ap_{k-1}\rangle}. \tag{2.17}$$

Since the CG algorithm is used only with positive definite matrices, the coefficients are always defined, and it can be shown, analogous to the MINRES method, that the A-norm of the error is actually minimized over the much larger affine space $e_0 + \text{span}\{p_0, p_1, \ldots, p_k\}$.

THEOREM 2.3.2. *Assume that A is Hermitian and positive definite. The CG algorithm generates the exact solution to the linear system $Ax = b$ in at most n steps. The error, residual, and direction vectors generated before the exact solution is obtained are well defined and satisfy*

$$\langle e_{k+1}, Ap_j \rangle = \langle p_{k+1}, Ap_j \rangle = \langle r_{k+1}, r_j \rangle = 0 \quad \forall j \leq k.$$

It follows that of all vectors in the affine space

$$e_0 + \text{span}\{Ae_0, A^2e_0, \ldots, A^{k+1}e_0\}, \tag{2.18}$$

e_{k+1} has the smallest A-norm.

Proof. Since A is positive definite, it is clear that the coefficients in the CG algorithm are well defined unless a residual vector is zero, in which case the exact solution has been found. Assume that $r_0, \ldots, r_k$ are nonzero. By the choice of a_0, it is clear that $\langle r_1, r_0 \rangle = \langle e_1, Ap_0 \rangle = 0$, and from the choice of b_0 it follows that

$$\begin{aligned} \langle p_1, Ap_0 \rangle &= \langle r_1, Ap_0 \rangle + \frac{\langle r_1, r_1 \rangle}{\langle r_0, r_0 \rangle} \langle p_0, Ap_0 \rangle \\ &= \langle r_1, a_0^{-1}(r_0 - r_1) \rangle + a_0^{-1} \langle r_1, r_1 \rangle = 0, \end{aligned}$$

where the last equality holds because $\langle r_1, r_0 \rangle = 0$ and a_0^{-1} is real. Assume that

$$\langle e_k, Ap_j \rangle = \langle p_k, Ap_j \rangle = \langle r_k, r_j \rangle = 0 \quad \forall j \leq k-1.$$

Then we also have

$$\langle p_k, Ap_k \rangle = \langle r_k + b_{k-1}p_{k-1}, Ap_k \rangle = \langle r_k, Ap_k \rangle,$$

$$\langle r_k, p_k \rangle = \langle r_k, r_k + b_{k-1}p_{k-1} \rangle = \langle r_k, r_k \rangle,$$

so, by the choice of a_k, it follows that

$$\langle r_{k+1}, r_k \rangle = \langle r_k, r_k \rangle - a_k \langle r_k, Ap_k \rangle = \langle r_k, r_k \rangle - \langle r_k, r_k \rangle = 0,$$

$$\langle e_{k+1}, Ap_k \rangle = \langle r_{k+1}, p_k \rangle = \langle r_k, p_k \rangle - a_k \langle Ap_k, p_k \rangle = \langle r_k, r_k \rangle - \langle r_k, r_k \rangle = 0.$$

From the choice of b_k, we have

$$\langle p_{k+1}, Ap_k \rangle = \langle r_{k+1}, Ap_k \rangle + \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle} \langle p_k, Ap_k \rangle$$

$$= \langle r_{k+1}, a_k^{-1}(r_k - r_{k+1}) \rangle + a_k^{-1} \langle r_{k+1}, r_{k+1} \rangle = 0.$$

For $j \leq k-1$, we have

$$\langle e_{k+1}, Ap_j \rangle = \langle e_k - a_k p_k, Ap_j \rangle = 0,$$

$$\langle r_{k+1}, r_j \rangle = \langle r_k - a_k Ap_k, r_j \rangle = -a_k \langle p_k, A(p_j - b_{j-1}p_{j-1}) \rangle = 0,$$

$$\langle p_{k+1}, Ap_j \rangle = \langle r_{k+1} + b_k p_k, Ap_j \rangle = \langle r_{k+1}, a_j^{-1}(r_j - r_{j+1}) \rangle = 0,$$

so, by induction, the desired equalities are established.

It is easily checked by induction that e_{k+1} lies in the space (2.18) and that $\text{span}\{p_0, \ldots, p_k\} = \text{span}\{Ae_0, \ldots, A^{k+1}e_0\}$. Since e_{k+1} is A-orthogonal to $\text{span}\{p_0, \ldots, p_k\}$, it follows that e_{k+1} is the vector in the space (2.18) with minimal A-norm, and it also follows that if the exact solution is not obtained before step n, then $e_n = 0$. □

2.4. Orthodir, MINRES, and GMRES.

Returning to the case of general matrices A, the idea of minimizing over a larger subspace can be extended, at the price of having to save and orthogonalize against additional vectors at each step. To minimize the 2-norm of the residual r_{k+1} over the j-dimensional affine space

$$r_k + \text{span}\{Ap_k, Ap_{k-1}, \ldots, Ap_{k-j+1}\} = r_k + \text{span}\{Ar_k, Ap_{k-1}, \ldots, Ap_{k-j+1}\},$$

set

$$p_k = r_k - \sum_{\ell=1}^{j-1} b_{k-\ell}^{(k)} p_{k-\ell}, \quad b_{k-\ell}^{(k)} = \frac{\langle Ar_k, Ap_{k-\ell} \rangle}{\langle Ap_{k-\ell}, Ap_{k-\ell} \rangle}. \tag{2.19}$$

This defines the Orthomin(j) procedure. Unfortunately, the algorithm can still fail if $0 \in \mathcal{F}(A^H)$, and again the only proven a priori bound on the residual norm is estimate (2.9), although this bound is often pessimistic.

It turns out that the possibility of failure can be eliminated by replacing r_k in formula (2.19) with Ap_{k-1}. This algorithm, known as *Orthodir*, generally has worse convergence behavior than Orthomin for $j < n$, however. The bound (2.9) can no longer be established because the space over which the norm of r_{k+1} is minimized may not contain the vector Ar_k.

An exception is the case of *Hermitian* matrices, where it can be shown that for $j = 3$, the Orthodir(j) algorithm minimizes the 2-norm of the residual over the entire affine space

$$r_0 + \text{span}\{Ap_0, Ap_1, \ldots, Ap_k\} = r_0 + \text{span}\{Ar_0, A^2r_0, \ldots, A^k r_0\}. \tag{2.20}$$

This provides a reasonable implementation of the MINRES algorithm described in section 2.3.

Given an initial guess x_0, compute $r_0 = b - Ax_0$ and set $p_0 = r_0$. Compute $s_0 = Ap_0$. For $k = 1, 2, \ldots,$

Set $x_k = x_{k-1} + a_{k-1}p_{k-1}$, where $a_{k-1} = \frac{\langle r_{k-1}, s_{k-1}\rangle}{\langle s_{k-1}, s_{k-1}\rangle}$.

Compute $r_k = r_{k-1} - a_{k-1}s_{k-1}$.

Set $p_k = s_{k-1}$, $s_k = As_{k-1}$. For $\ell = 1, 2$,

$$b_{k-\ell}^{(k)} = \frac{\langle s_k, s_{k-\ell}\rangle}{\langle s_{k-\ell}, s_{k-\ell}\rangle},$$
$$p_k \longleftarrow p_k - b_{k-\ell}^{(k)} p_{k-\ell},$$
$$s_k \longleftarrow s_k - b_{k-\ell}^{(k)} s_{k-\ell}.$$

A difficulty with this algorithm is that in finite precision arithmetic, the vectors s_k, which are supposed to be equal to Ap_k, may differ from this if there is much cancellation in the computation of s_k. This could be corrected with an occasional extra matrix–vector multiplication to explicitly set $s_k = Ap_k$ at the end of an iteration. Another possible implementation is given in section 2.5.

For general non-Hermitian matrices, if $j = n$, then the Orthodir(n) algorithm minimizes the 2-norm of the residual at each step k over the affine space in (2.20). It follows that the exact solution is obtained in n or fewer steps (assuming exact arithmetic) but at the cost of storing up to n search directions p_k (as well as auxiliary vectors $s_k \equiv Ap_k$) and orthogonalizing against k direction vectors at each step $k = 1, \ldots, n$. If the full n steps are required, then Orthodir(n) requires $O(n^2)$ storage and $O(n^3)$ work, just as would be required by a standard dense Gaussian elimination routine. The power of the method lies in the fact that at each step the residual norm is minimized over the space (2.20) so that, hopefully, an acceptably good approximate solution can be obtained in far fewer than n steps.

There is another way to compute the approximation x_k for which the norm of r_k is minimized over the space (2.20). This method requires about half the storage of Orthodir(n) (no auxiliary vectors) and has better numerical properties. It is the *GMRES method.*

The GMRES method uses the modified Gram–Schmidt process to construct an orthonormal basis for the Krylov space span$\{r_0, Ar_0, \ldots, A^k r_0\}$. When the modified Gram–Schmidt process is applied to this space in the form given below it is called *Arnoldi's method.*

Arnoldi Algorithm.

Given q_1 with $\|q_1\| = 1$. For $j = 1, 2, \ldots,$

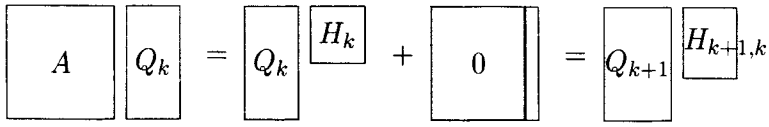
$\tilde{q}_{j+1} = Aq_j$. For $i = 1, \ldots, j$, $h_{ij} = \langle \tilde{q}_{j+1}, q_i\rangle$, $\tilde{q}_{j+1} \longleftarrow \tilde{q}_{j+1} - h_{ij}q_i$.

$h_{j+1,j} = \|\tilde{q}_{j+1}\|$, $q_{j+1} = \tilde{q}_{j+1}/h_{j+1,j}$.

If Q_k is the n-by-k matrix with the orthonormal basis vectors $q_1, \ldots, q_k$ as columns, then the Arnoldi iteration can be written in matrix form as

$$AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} \xi_k^T = Q_{k+1} H_{k+1,k}. \tag{2.21}$$

Here H_k is the k-by-k upper Hessenberg matrix with (i,j)-element equal to $h_{i,j}$ for $j = 1,\ldots,k$, $i = 1,\ldots,\min\{j+1,k\}$, and all other elements zero. The vector ξ_k is the kth unit k-vector, $(0,\ldots,0,1)^T$. The $k+1$-by-k matrix $H_{k+1,k}$ is the matrix whose top k-by-k block is H_k and whose last row is zero except for the $(k+1,k)$-element, which is $h_{k+1,k}$. Pictorially, the matrix equation (2.21) looks like



In the GMRES method, the approximate solution x_k is taken to be of the form $x_k = x_0 + Q_k y_k$ for some vector y_k; that is, x_k is x_0 plus some linear combination of the orthonormal basis vectors for the Krylov space. To obtain the approximation for which $r_k = r_0 - AQ_k y_k$ has a minimal 2-norm, the vector y_k must solve the least squares problem

$$\min_y \|r_0 - AQ_k y\| = \min_y \|r_0 - Q_{k+1} H_{k+1,k} y\|$$

$$= \min_y \|Q_{k+1}(\beta\xi_1 - H_{k+1,k} y)\| = \min_y \|\beta\xi_1 - H_{k+1,k} y\|,$$

where $\beta = \|r_0\|$, ξ_1 is the first unit $(k+1)$-vector $(1,0\ldots,0)^T$, and the second equality is obtained by using the fact that $Q_{k+1}\xi_1$, the first orthonormal basis vector, is just r_0/β.

The basic steps of the GMRES algorithm are as follows.

> Given x_0, compute $r_0 = b - Ax_0$ and set $q_1 = r_0/\|r_0\|$. For $k = 1, 2, \ldots,$
>
> Compute q_{k+1} and $h_{i,k}$, $i = 1,\ldots,k+1$ using the Arnoldi algorithm.
>
> Form $x_k = x_0 + Q_k y_k$, where y_k is the solution to the least squares problem $\min_y \|\beta\xi_1 - H_{k+1,k} y\|$.

A standard method for solving the least squares problem $\min_y \|\beta\xi_1 - H_{k+1,k} y\|$ is to factor the $k+1$-by-k matrix $H_{k+1,k}$ into the product of a $k+1$-by-$k+1$ unitary matrix F^H and a $k+1$-by-k upper triangular matrix R (that is, the top k-by-k block is upper triangular and the last row is 0). This factorization, known as the QR factorization, can be accomplished using plane rotations. The solution y_k is then obtained by solving the upper triangular system

$$R_{k\times k}\, y = \beta\, (F\xi_1)_{k\times 1}, \tag{2.22}$$

where $R_{k\times k}$ is the top k-by-k block of R and $(F\xi_1)_{k\times 1}$ is the top k entries of the first column of F.

Given the QR factorization of $H_{k+1,k}$, we would like to be able to compute the QR factorization of the next matrix $H_{k+2,k+1}$ with as little work as possible. To see how this can be done, let F_i denote the rotation matrix that rotates the unit vectors ξ_i and ξ_{i+1} through the angle θ_i:

$$F_i = \begin{pmatrix} I & & & \\ & c_i & s_i & \\ & -\bar{s}_i & c_i & \\ & & & I \end{pmatrix},$$

where $c_i \equiv \cos(\theta_i)$ and $s_i \equiv \sin(\theta_i)$. The dimension of the matrix F_i, that is, the size of the second identity block, will depend on the context in which it is used. Assume that the rotations F_i, $i = 1, \ldots, k$ have previously been applied to $H_{k+1,k}$ so that

$$(F_k F_{k-1} \cdots F_1)\, H_{k+1,k} = R^{(k)} = \begin{pmatrix} x & x & \ldots & x \\ & x & \ldots & x \\ & & \ddots & \vdots \\ & & & x \\ 0 & 0 & \ldots & 0 \end{pmatrix},$$

where the x's denote nonzeros. In order to obtain $R^{(k+1)}$, the upper triangular factor for $H_{k+2,k+1}$, first premultiply the last column of $H_{k+2,k+1}$ by the previous rotations to obtain

$$(F_k F_{k-1} \cdots F_1)\, H_{k+2,k+1} = \begin{pmatrix} x & x & \ldots & x & x \\ & x & \ldots & x & x \\ & & \ddots & \vdots & \vdots \\ & & & x & x \\ 0 & 0 & \ldots & 0 & d \\ 0 & 0 & \ldots & 0 & h \end{pmatrix},$$

where the $(k+2, k+1)$-entry, h, is just $h_{k+2,k+1}$, since this entry is unaffected by the previous rotations. The next rotation, F_{k+1}, is chosen to eliminate this entry by setting $c_{k+1} = |d|/\sqrt{|d|^2 + |h|^2}$, $\bar{s}_{k+1} = c_{k+1}h/d$ if $d \neq 0$, and $c_{k+1} = 0$, $s_{k+1} = 1$ if $d = 0$. Note that the $(k+1)$st diagonal entry of $R^{(k+1)}$ is nonzero since h is nonzero (assuming the exact solution to the linear system has not already been computed), and, if $d \neq 0$, then this diagonal element is $(d/|d|)\sqrt{|d|^2 + |h|^2}$ while if $d = 0$, the diagonal element is h.

The right-hand side in (2.22) is computed by applying each of the rotations $F_1, \ldots, F_k$ to the unit vector ξ_1. The absolute value of the last entry of this $(k+1)$-vector, multiplied by β, is the 2-norm of the residual at step k since

$$\|b - Ax_k\| = \|\beta\xi_1 - F^H R y_k\| = \|\beta F\xi_1 - R y_k\|,$$

and $\beta F\xi_1 - R y_k$ is zero except for its bottom entry, which is just the bottom entry of $\beta F\xi_1$.

The GMRES algorithm can be written in the following form.

Algorithm 3. Generalized Minimal Residual Algorithm (GMRES).

Given x_0, compute $r_0 = b - Ax_0$ and set $q_1 = r_0/\|r_0\|$.
Initialize $\xi = (1, 0, \ldots, 0)^T$, $\beta = \|r_0\|$. For $k = 1, 2, \ldots$,

Compute q_{k+1} and $h_{i,k} \equiv H(i,k)$, $i = 1, \ldots, k+1$, using the Arnoldi algorithm.

Apply $F_1, \ldots, F_{k-1}$ to the last column of H; that is,
For $i = 1, \ldots, k-1$,

$$\begin{pmatrix} H(i,k) \\ H(i+1,k) \end{pmatrix} \leftarrow \begin{pmatrix} c_i & s_i \\ -\bar{s}_i & c_i \end{pmatrix} \begin{pmatrix} H(i,k) \\ H(i+1,k) \end{pmatrix}$$

Compute the kth rotation, c_k and s_k, to annihilate the $(k+1,k)$ entry of H.[1]

Apply kth rotation to ξ and to last column of H:

$$\begin{pmatrix} \xi(k) \\ \xi(k+1) \end{pmatrix} \leftarrow \begin{pmatrix} c_k & s_k \\ -\bar{s}_k & c_k \end{pmatrix} \begin{pmatrix} \xi(k) \\ 0 \end{pmatrix}$$

$$H(k,k) \leftarrow c_k H(k,k) + s_k H(k+1,k), \quad H(k+1,k) \leftarrow 0.$$

If residual norm estimate $\beta|\xi(k+1)|$ is sufficiently small, then
Solve upper triangular system $H_{k\times k}\, y_k = \beta\, \xi_{k\times 1}$.
Compute $x_k = x_0 + Q_k y_k$.

The (full) GMRES algorithm described above may be impractical because of increasing storage and work requirements, if the number of iterations needed to solve the linear system is large. The GMRES(j) algorithm is defined by simply restarting GMRES every j steps, using the latest iterate as the initial guess for the next GMRES cycle. In Chapter 3, we discuss the convergence rate of full and restarted GMRES.

2.5. Derivation of MINRES and CG from the Lanczos Algorithm.

When the matrix A is Hermitian, the Arnoldi algorithm of the previous section can be simplified to a 3-term recurrence known as the Lanczos algorithm. Slightly different (but mathematically equivalent) coefficient formulas are normally used in the Hermitian case.

[1]The formula is $c_k = |H(k,k)|/\sqrt{|H(k,k)|^2 + |H(k+1,k)|^2}$, $\bar{s}_k = c_k H(k+1,k)/H(k,k)$, but a more robust implementation should be used. See, for example, BLAS routine DROTG [32].

Lanczos Algorithm (for Hermitian matrices A).

Given q_1 with $\|q_1\| = 1$, set $\beta_0 = 0$. For $j = 1, 2, \ldots,$

$\tilde{q}_{j+1} = Aq_j - \beta_{j-1}q_{j-1}$. Set $\alpha_j = \langle \tilde{q}_{j+1}, q_j \rangle$, $\tilde{q}_{j+1} \longleftarrow \tilde{q}_{j+1} - \alpha_j q_j$.

$\beta_j = \|\tilde{q}_{j+1}\|$, $q_{j+1} = \tilde{q}_{j+1}/\beta_j$.

To see that the vectors constructed by this algorithm are the same as those constructed by the Arnoldi algorithm when the matrix A is Hermitian, we must show that they form an orthonormal basis for the Krylov space formed from A and q_1. It is clear that the vectors lie in this Krylov space and each vector has norm one because of the choice of the β_j's. From the formula for α_j, it follows that $\langle q_{j+1}, q_j \rangle = 0$. Suppose $\langle q_k, q_i \rangle = 0$ for $i \neq k$ whenever $k, i \leq j$. Then

$$\begin{aligned} \langle \tilde{q}_{j+1}, q_{j-1} \rangle &= \langle Aq_j - \alpha_j q_j - \beta_{j-1}q_{j-1}, q_{j-1} \rangle = \langle Aq_j, q_{j-1} \rangle - \beta_{j-1} \\ &= \langle q_j, Aq_{j-1} \rangle - \beta_{j-1} \\ &= \langle q_j, \tilde{q}_j + \alpha_{j-1}q_{j-1} + \beta_{j-2}q_{j-2} \rangle - \beta_{j-1} = \langle q_j, \tilde{q}_j \rangle - \beta_{j-1} = 0. \end{aligned}$$

For $i < j - 1$, we have

$$\begin{aligned} \langle \tilde{q}_{j+1}, q_i \rangle &= \langle Aq_j - \alpha_j q_j - \beta_{j-1}q_{j-1}, q_i \rangle = \langle Aq_j, q_i \rangle \\ &= \langle q_j, Aq_i \rangle \\ &= \langle q_j, \tilde{q}_{i+1} + \alpha_i q_i + \beta_{i-1}q_{i-1} \rangle = 0. \end{aligned}$$

Thus the vectors $q_1, \ldots, q_{j+1}$ form an orthonormal basis for the Krylov space $\text{span}\{q_1, Aq_1, \ldots, A^j q_1\}$.

The Lanczos algorithm can be written in matrix form as

$$AQ_k = Q_k T_k + \beta_k q_{k+1} \xi_k^T = Q_{k+1} T_{k+1,k}, \tag{2.23}$$

where Q_k is the n-by-k matrix whose columns are the orthonormal basis vectors $q_1, \ldots, q_k$, ξ_k is the kth unit k-vector, and T_k is the k-by-k Hermitian tridiagonal matrix of recurrence coefficients:

$$T_k = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{pmatrix}. \tag{2.24}$$

The $k+1$-by-k matrix $T_{k+1,k}$ has T_k as its upper k-by-k block and $\beta_k \xi_k^T$ as its last row.

It was shown in section 2.3 that the MINRES and CG algorithms generate the Krylov space approximations x_k for which the 2-norm of the residual and the A-norm of the error, respectively, are minimal. That is, if q_1 in the

Lanczos algorithm is taken to be r_0/β, $\beta = \|r_0\|$, then the algorithms generate approximate solutions of the form $x_k = x_0 + Q_k y_k$, where y_k is chosen to minimize the appropriate error norm. For the MINRES algorithm, y_k solves the least squares problem

$$\begin{aligned} \min_y \|r_0 - AQ_k y\| &= \min_y \|r_0 - Q_{k+1} T_{k+1,k} y\| \\ &= \min_y \|Q_{k+1}(\beta\xi_1 - T_{k+1,k} y)\| \\ &= \min_y \|\beta\xi_1 - T_{k+1,k} y\|, \end{aligned} \tag{2.25}$$

similar to the GMRES algorithm of the previous section.

For the MINRES algorithm, however, there is no need to save the orthonormal basis vectors generated by the Lanczos algorithm. Hence a different formula is needed to compute the approximate solution x_k. Let $R_{k\times k}$ be the upper k-by-k block of the triangular factor R in the QR decomposition of $T_{k+1,k} = F^H R$, as described in the previous section. Since $T_{k+1,k}$ is tridiagonal, $R_{k\times k}$ has only three nonzero diagonals. Define $P_k \equiv (p_0, \ldots, p_{k-1}) \equiv Q_k R_{k\times k}^{-1}$. Then p_0 is a multiple of q_1 and successive columns of P_k can be computed using the fact that $P_k R_{k\times k} = Q_k$:

$$p_{k-1} = \left(q_k - b_{k-2}^{(k-1)} p_{k-2} - b_{k-3}^{(k-1)} p_{k-3}\right) / b_{k-1}^{(k-1)},$$

where $b_{k-\ell}^{(k-1)}$ is the $(k-\ell+1, k)$-entry of $R_{k\times k}$. Recall from the arguments of the previous section that $b_{k-1}^{(k-1)}$ is nonzero, provided that the exact solution to the linear system has not been obtained already. The approximate solution x_k can be updated from x_{k-1} since

$$x_k = x_0 + P_k \beta\, (F\xi_1)_{k\times 1} = x_{k-1} + a_{k-1} p_{k-1},$$

where a_{k-1} is the kth entry of $\beta(F\xi_1)$. This leads to the following implementation of the MINRES algorithm.

Algorithm 4. Minimal Residual Algorithm (MINRES)
(for Hermitian Problems).

Given x_0, compute $r_0 = b - Ax_0$ and set $q_1 = r_0/\|r_0\|$.
Initialize $\xi = (1, 0, \ldots, 0)^T$, $\beta = \|r_0\|$. For $k = 1, 2, \ldots$,

Compute q_{k+1}, $\alpha_k \equiv T(k,k)$, and $\beta_k \equiv T(k+1,k) \equiv T(k,k+1)$
using the Lanczos algorithm.

Apply F_{k-2} and F_{k-1} to the last column of T; that is,

$$\begin{pmatrix} T(k-2,k) \\ T(k-1,k) \end{pmatrix} \leftarrow \begin{pmatrix} c_{k-2} & s_{k-2} \\ -\bar{s}_{k-2} & c_{k-2} \end{pmatrix} \begin{pmatrix} 0 \\ T(k-1,k) \end{pmatrix}, \quad \text{if } k > 2,$$

$$\begin{pmatrix} T(k-1,k) \\ T(k,k) \end{pmatrix} \leftarrow \begin{pmatrix} c_{k-1} & s_{k-1} \\ -\bar{s}_{k-1} & c_{k-1} \end{pmatrix} \begin{pmatrix} T(k-1,k) \\ T(k,k) \end{pmatrix}, \quad \text{if } k > 1.$$

Compute the kth rotation, c_k and s_k, to annihilate the $(k+1,k)$-entry of T.[2]

Apply kth rotation to ξ and to last column of T:

$$\begin{pmatrix} \xi(k) \\ \xi(k+1) \end{pmatrix} \leftarrow \begin{pmatrix} c_k & s_k \\ -\bar{s}_k & c_k \end{pmatrix} \begin{pmatrix} \xi(k) \\ 0 \end{pmatrix}.$$

$$T(k,k) \leftarrow c_k T(k,k) + s_k T(k+1,k), \quad T(k+1,k) \leftarrow 0.$$

Compute $p_{k-1} = [q_k - T(k-1,k)p_{k-2} - T(k-2,k)p_{k-3}]/T(k,k)$,
where undefined terms are zero for $k \leq 2$.

Set $x_k = x_{k-1} + a_{k-1}p_{k-1}$, where $a_{k-1} = \beta\xi(k)$.

For the CG method, y_k is chosen to make the residual r_k orthogonal to the columns of Q_k. For positive definite matrices A, this minimizes the A-norm of the error since $e_k = e_0 - Q_k y_k$ has minimal A-norm when it is A-orthogonal to the columns of Q_k, i.e., when $Q_k^H A e_k = Q_k^H r_k = 0$. Note that the criterion that r_k be orthogonal to the columns of Q_k can be enforced for Hermitian *indefinite* problems as well, although it does not correspond to minimizing any obvious error norm. The vector y_k for the CG algorithm satisfies

$$Q_k^H(r_0 - AQ_k y_k) = \beta\xi_1 - T_k y_k = 0; \tag{2.26}$$

that is, y_k is the solution to the k-by-k linear system $T_k y = \beta\xi_1$. While the least squares problem (2.25) always has a solution, the linear system (2.26) has a unique solution if and only if T_k is *nonsingular*. When A is positive definite, it follows from the minimax characterization of eigenvalues of a Hermitian

[2]The formula is $c_k = |T(k,k)|/\sqrt{|T(k,k)|^2 + |T(k+1,k)|^2}$, $\bar{s}_k = c_k T(k+1,k)/T(k,k)$, but a more robust implementation should be used. See, for example, BLAS routine DROTG [32].

matrix that the tridiagonal matrix $T_k = Q_k^H A Q_k$ is also positive definite, since its eigenvalues lie between the smallest and largest eigenvalues of A.

If T_k is positive definite, and sometimes even if it is not, then one way to solve (2.26) is to factor T_k in the form

$$T_k = L_k D_k L_k^H, \tag{2.27}$$

where L_k is a unit lower bidiagonal matrix and D_k is a diagonal matrix. One would like to be able to compute not only y_k but the approximate solution $x_k = x_0 + Q_k y_k$ without saving all of the basis vectors $q_1, \ldots, q_k$. The factorization (2.27) can be updated easily from one step to the next, since L_k and D_k are the k-by-k principal submatrices of L_{k+1} and D_{k+1}. If we define $P_k \equiv (p_0, \ldots p_{k-1}) \equiv Q_k L_k^{-H}$, then the columns of P_k are A-orthogonal since

$$P_k^H A P_k = L_k^{-1} Q_k^H A Q_k L_k^{-H} = L_k^{-1} T_k L_k^{-H} = D_k,$$

and since P_k satisfies $P_k L_k^H = Q_k$, the columns of P_k can be computed in succession via the recurrence

$$p_k = q_k - \bar{b}_{k-1} p_{k-1},$$

where b_{k-1} is the $(k, k-1)$-entry of L_k. It is not difficult to see that the columns of P_k are, to within constant factors, the direction vectors from the CG algorithm of section 2.3. The Lanczos vectors are normalized versions of the CG residuals, with opposite signs at every other step. With this factorization, then, x_k is given by

$$x_k = x_0 + Q_k T_k^{-1} \beta e_1 = x_0 + P_k D_k^{-1} L_k^{-1} \beta e_1,$$

and since x_{k-1} satisfies

$$x_{k-1} = x_0 + P_{k-1} D_{k-1}^{-1} L_{k-1}^{-1} \beta e_1,$$

it can be seen that x_k satisfies

$$x_k = x_{k-1} + a_{k-1} p_{k-1},$$

where $a_{k-1} = d_k^{-1} \beta (L_k^{-1})_{k,1}$ and d_k is the (k,k)-entry of D_k. The coefficient a_{k-1} is defined, provided that L_k is invertible and $d_k \neq 0$.

With this interpretation it can be seen that if the CG algorithm of section 2.3 were applied to a Hermitian *indefinite* matrix, then it would fail at step k if and only if the LDL^H factorization of T_k does not exist. If this factorization exists for $T_1, \ldots, T_{k-1}$, then it can fail to exist at step k only if T_k is singular. For indefinite problems, it is possible that T_k will be singular, but subsequent tridiagonal matrices, e.g., T_{k+1}, will be nonsingular. The CG algorithm of section 2.3 cannot recover from a singular intermediate matrix T_k. To overcome this difficulty, Paige and Saunders [111] proposed a CG-like algorithm based

on the LQ-factorization of the tridiagonal matrices. This algorithm is known as SYMMLQ. In the SYMMLQ algorithm, the 2-norm of the error is minimized, but over a different Krylov subspace. The CG iterates can be derived stably from the SYMMLQ algorithm, but for Hermitian indefinite problems they do not minimize any standard error or residual norm. We will show in Chapter 5, however, that the residuals in the CG and MINRES methods are closely related.

Comments and Additional References.

For a discussion of simple iteration methods (e.g., Jacobi, Gauss–Seidel, SOR), the classic books of Varga [135] and Young [144] are still highly recommended.

The CG algorithm was originally proposed by Hestenes and Stiefel [79] and appeared in a related work at the same time by Lanczos [90]. The Orthomin method described in this chapter was first introduced by Vinsome [137], and the Orthodir algorithm was developed by Young and Jea [145]. Subsequently, Saad and Schultz invented the GMRES algorithm [119]. Theorem 2.2.2 was proved in the special case when the Hermitian part of the matrix is positive definite by Eisenstat, Elman, and Schultz [40].

The MINRES implementation described in section 2.5 was given by Paige and Saunders [111], as was the first identification of the CG algorithm with the Lanczos process followed by LDL^H-factorization of the tridiagonal matrix. The idea of minimizing the 2-norm of the residual for Hermitian indefinite problems was contained in the original Hestenes and Stiefel paper, implemented in the form we have referred to here as Orthomin(2). This is sometimes called the *conjugate residual* method. For an implementation that uses the Orthomin(2) algorithm when it is safe to do so and switches to Orthodir(3) in other circumstances, see Chandra [25].

A useful bibliography of work on the CG and Lanczos algorithms between 1948 and 1976 is contained in [58].

Exercises.

2.1. Use the Jordan form discussed in section 1.3.2 to describe the asymptotic convergence of simple iteration for a nondiagonalizable matrix.

2.2. Show that if A and b are *real* and $0 \in \mathcal{F}(A^H)$, then there is a *real* initial vector for which the Orthomin iteration fails.

2.3. Give an example of a problem for which Orthomin(1) converges but simple iteration (with no preconditioner) does not. Give an example of a problem for which simple iteration converges but Orthomin(j) does not for any $j < n$.

2.4. Verify that the coefficient formulas in (2.17) are equivalent to those in Algorithm 2.

2.5. Show that for Hermitian matrices, the MINRES algorithm given in section 2.4 minimizes $\|r_{k+1}\|$ over the space in (2.20).

2.6. Express the entries of the tridiagonal matrix generated by the Lanczos algorithm in terms of the coefficients in the CG algorithm (Algorithm 2). (Hint: Write down the 3-term recurrence for $q_{k+1} \equiv (-1)^k r_k/\|r_k\|$ in terms of q_k and q_{k-1}.)

2.7. Derive a necessary and sufficient condition for the convergence of the restarted GMRES algorithm, GMRES(j), in terms of the *generalized* field of values defined in section 1.3.3; that is, show that GMRES(j) converges to the solution of the linear system for all initial vectors if and only if the zero vector is not contained in the set

$$\mathcal{F}_j(A, A^2, \ldots, A^j) = \left\{ \begin{pmatrix} y^H A y \\ y^H A^2 y \\ \vdots \\ y^H A^j y \end{pmatrix} \; : \; y \in C^n, \; \|y\| = 1 \right\}.$$

2.8. Show that for a *normal* matrix whose eigenvalues have real parts greater than or equal to the imaginary parts in absolute value, the GMRES(2) iteration converges for all initial vectors. (Hint: Since $r_2 = P_2(A)r_0$ for a certain second-degree polynomial P_2 with $P_2(0) = 1$ and since P_2 minimizes the 2-norm of the residual over all such polynomials, we have $\|r_2\| \leq \min_{p_2} \|p_2(A)\| \cdot \|r_0\|$. If a second-degree polynomial p_2 with value 1 at the origin can be found which satisfies $\|p_2(A)\| < 1$, then this will show that each GMRES(2) cycle reduces the residual norm by at least a constant factor. Since A is normal, its eigendecomposition can be written in the form $A = U\Lambda U^T$, where U is unitary and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$; it follows that $\|p_2(A)\| = \max_{i=1,\ldots,n} |p_2(\lambda_i)|$. Hence, find a polynomial $p_2(z)$ that is strictly less than 1 in absolute value throughout a region

$$\{z : |\mathrm{Re}(z)| \geq |\mathrm{Im}(z)|, \quad |z| \leq \max_{i=1,\ldots,n} |\lambda_i|\},$$

containing the spectrum of A.)