CHAPTER 2

Iterative Krylov-Subspace Solvers

Preconditioned Krylov-subspace iterations are a key ingredient in many modern linear solvers, including in solvers that employ support preconditioners. This chapter presents Krylov-subspace iterations for symmetric semidefinite matrices. In particular, we analyze the convergence behavior of these solvers. Understanding what determines the convergence rate is key to designing effective preconditioners.

1. The Minimal Residual Method

The minimal-residual method (minres) is an iterative algorithm that finds in each iteration the vector \( x^{(t)} \) that minimizes the residual \( \|Ax^{(t)} - b\|_2 \) in a subspace \( K_t \) of \( \mathbb{R}^n \). These subspaces, called the Krylov subspaces, are nested, \( K_t \subseteq K_{t+1} \), and the dimension of the subspace usually grows by one in every iteration, so the accuracy of the approximate solution \( x^{(t)} \) tends to improve from one iteration to the next. The construction of the spaces \( K_t \) is designed to allow an efficient computation of the approximate solution in every iteration.

**Definition 1.1.** The Krylov subspace \( K_t \) is the subspace that is spanned by the columns of the matrix \( K_t = [b \ A b^2 \ A^2 b \ \cdots \ A^{t-1}] \). That is,

\[
K_t = \{ K_t y : y \in \mathbb{R}^t \}.
\]

Once we have a basis for \( K_t \), we can express the minimization of \( \|Ax - b\|_2 \) in \( K_t \) as an unconstrained linear least-squares problem, since

\[
\min_{x \in K_t} \|Ax - b\|_2 = \min_{y \in \mathbb{R}^t} \|AK_t y - b\|_2.
\]

There are three fundamental tools in the solution of least squares problems with full-rank \( m \times n \) coefficient matrices with \( m \geq n \). The first tool is unitary transformations. A unitary matrix \( Q \) preserves the 2-norm of vectors, \( \|Qx\|_2 = \|x\|_2 \) for any \( x \) (a square matrix with orthonormal columns, or equivalently, a matrix such that \( QQ^* = I \)). This allows us to transform least squares problem into equivalent forms

\[
(1) \quad \min_x \|Ax - b\|_2 = \min_y \|QAx - Qb\|_2.
\]
An insight about matrices that contain only zeros in rows $n+1$ through $m$ is the second tool. Consider the least squares problem

$$\min_x \left\| \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right\|_2.$$ 

If the coefficient matrix has full rank, then the square block $R_1$ must be invertible. The solution to the problem is the vector $x$ that minimizes the Euclidean distance between $\begin{bmatrix} R_1 \\ 0 \end{bmatrix} x$ and $\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ is minimal. But it is clear what is this vector: $x = R_1^{-1} b_1$. This vector yields $\begin{bmatrix} R_1 \\ 0 \end{bmatrix} x = \begin{bmatrix} b_1 \\ 0 \end{bmatrix}$, and we clearly cannot any closer. The third tool combines the first two into an algorithm. If we factor $A$ into a product $QR$ of a unitary matrix $Q$ and an upper trapezoidal matrix $R = [R_{1:0}]^*$, which is always possible, then we can multiply the system by $Q^*$ to obtain

$$\min_x \| Ax - b \|_2 = \min_x \| QR x - b \|_2 = \min_x \| Rx - Q^* b \|_2 = \min_x \left\| \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - Q^* b \right\|_2.$$ 

We can now compute the minimizer $x$ by substitution. Furthermore, the norm of the residual is given by $\| (Q^* b)_{n+1:m} \|_2$.

In principle, we could solve (1) for $y$ using a $QR$ factorization of $AK_t$. Once we find $y$, we can compute $x = K_t y$. There are, however, two serious defects in this approach. First, it is inefficient, because $AK_t$ is dense. Second, as $t$ grows, $A^t b$ tends to the subspace spanned by the dominant eigenvectors of $A$ (the eigenvectors associated with the eigenvalues with maximal absolute value). This causes $K_t$ to become ill conditioned; for some vectors $x \in K_t$ with $\|x\|_2 = 1$, the vector $y$ such that $x = K_t y$ has huge elements. This phenomenon will happen even if we normalize the columns of $K_t$ to have unit norm, and it causes instabilities when the computation is carried out using floating-point arithmetic.

We need a better basis for $K_t$ for stability, and we need to exploit the special properties of $AK_t$ for efficiency. MINRES does both. It uses a stable basis that can be computed efficiently, and it it combines the three basic tools in a clever way to achieve efficiency.

An orthonormal basis $Q_t$ for $K_t$ would work better, because for $x = Q_t z$ we would have $\|z\|_2 = \|x\|_2$. There are many orthonormal bases for $K_t$. We choose a particular basis that we can compute incrementally, one basis column in each iteration. The basis that we use consists of the columns of the $Q$ factor from the $QR$ factorization of $K_t = Q_t R_t$, where
$Q_t$ is orthonormal and $R_t$ is upper triangular. We compute $Q_t$ using Gram-Schmidt orthogonalization. The Gram-Schmidt process is not always numerically stable when carried out in floating-point arithmetic, but it is the only efficient way to compute $Q_t$ one column at a time.

We now show how to compute, given $Q_t = [q_1 \cdots q_t]$, the next basis column $q_{t+1}$. The key to the efficient computation of $q_{t+1}$ is the special relationship of the matrices $Q_t$ with $A$. We assume by induction that $K_t = Q_t R_t$ and that $r_{t,t} \neq 0$. If $r_{t,t} = 0$, then it is not hard to show that the exact solution $x$ is in $K_t$, so we would have found it in one of the previous iterations. If we reached iteration $t$, then $x \notin K_t$, so $r_{t,t} \neq 0$. Expanding the last column of $K_t = Q_t R_t$, we get

$$A^{t-1}b = r_{1,t}q_1 + r_{2,t}q_2 + \cdots + r_{t-1,t}q_{t-1} + r_{t,t}q_t.$$ 

We now isolate $q_t$ to obtain

$$q_t = r_{t,t}^{-1} \left( A^{t-1}b - r_{1,t}q_1 - r_{2,t}q_2 - \cdots - r_{t-1,t}q_{t-1} \right),$$

so

$$Aq_t = r_{t,t}^{-1} \left( A^t b - r_{1,t} Aq_1 - r_{2,t} Aq_2 - \cdots - r_{t-1,t} Aq_{t-1} \right).$$

Because $q_t \in K_t$, clearly $Aq_t \in K_{t+1}$. Furthermore, if we reached iteration $t+1$, then $Aq_t \notin K_t$, because if $Aq_t \in K_t$ then $x \in K_t$. Therefore, we can orthogonalize $Aq_t$ with respect to $q_1, \ldots, q_t$ and normalize to obtain $q_{t+1}$

$$\tilde{q}_{t+1} = Aq_t - (q_t^* Aq_t) q_t - \cdots - (q_1^* Aq_t) q_1,$$

$$q_{t+1} = \tilde{q}_{t+1} / \| \tilde{q}_{t+1} \|_2.$$

These expressions allow us not only to compute $q_t$, but also to express $Aq_t$ as a linear combination of $q_1, \ldots, q_t, q_{t+1}$,

$$Aq_t = \| \tilde{q}_{t+1} \|_2 q_{t+1} + (q_t^* Aq_t) q_t + \cdots + (q_1^* Aq_t) q_1 = h_{t+1,t} q_{t+1} + h_{t,t} q_t + \cdots + h_{1,t} q_1,$$

where $h_{t+1,t} = \| \tilde{q}_{t+1} \|_2$, and where $h_{j,t} = q_j^* A q_t$ for $j \leq t$. The same argument also holds for $Aq_1, \ldots, Aq_{t-1}$, so in matrix terms,

$$AQ_t = Q_{t+1} \tilde{H}_t.$$

The matrix $\tilde{H}_t \in \mathbb{R}^{(t+1) \times t}$ is upper Hessenberg: in column $j$, rows $j+2$ to $t+1$ are zero. If we multiply both sides of this equation from the
left by $Q^*_t$, we obtain

$$Q^*_t A Q_t = Q^*_t Q_{t+1} \tilde{H}_t$$

$$= \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \tilde{H}_t$$

$$= (\tilde{H}_t)_{1:t,1:t}.$$  

We denote the first $t$ rows of $\tilde{H}_t$ by $H_t$ to obtain

$$Q^*_t A Q_t = H_t.$$  

Because $A$ is symmetric, $H_t$ must be symmetric, and hence tridiagonal. The symmetry of $H_t$ implies that in column $j$ of $H_t$ and $\tilde{H}_t$, rows 1 through $j - 2$ are also zero. This is the key to the efficiency of minres and similar algorithms, because it implies that for $j \leq t - 2$, $h_{j,t} = q^*_j A q_t = 0$; we do not need to compute these inner products and we do not need to subtract $(q^*_j A q_t) q_j$ from $A q_t$ in the orthogonalization process. As we shall shortly see, we do not even need to store $q_j$ for $j \leq t - 2$.

Now that we have an easy-to-compute orthonormal basis for $K_t$, we return to our least-squares problem

$$\min_{x \in K_t} \|Ax - b\|_2 = \min_{y \in \mathbb{R}^t} \|AK_t y - b\|_2$$

$$= \min_{z \in \mathbb{R}^t} \|AQ_t z - b\|_2$$

$$= \min_{z \in \mathbb{R}^t} \|Q_{t+1} \tilde{H}_t z - b\|_2.$$  

Our strategy now is to compute the minimizer $z$ from the expression in the last line. Once we compute $z$, the minimizer $x$ in $K_t$ is $x = Q_t z$. To compute the minimizer, we use the equality

$$\arg \min_{z \in \mathbb{R}^t} \|Q_{t+1} \tilde{H}_t z - b\|_2 = \arg \min_{z \in \mathbb{R}^t} \|\tilde{H}_t z - Q^*_t b\|_2$$

$$= \arg \min_{z \in \mathbb{R}^t} \|\tilde{H}_t z - \|b\|_2 e_1\|_2,$$

where $e_1$ is the first unit vector. The equality $Q^*_{t+1} b = \|b\|_2 e_1$ holds because $Q^*_{t+1}$ is the orthogonal factor in the $QR$ factorization of $K_{t+1}$, whose first column is $b$. To solve this least-squares problem, we will use a $QR$ factorization $\tilde{H}_t = V_t U_t$ of $\tilde{H}_t$: the minimizer $z$ is then defined by $U_t z = \|b\|_2 V^*_t e_1$. Because $\tilde{H}_t \in \mathbb{R}^{t \times t}$ is tridiagonal we can compute its $QR$ factorization with a sequence of of $t - 1$ Givens rotations, where
2. The Conjugate Gradients Algorithm

2. The Conjugate Gradients Algorithm

Minres is a variant of an older and more well-known algorithm called the Conjugate Gradients method (CG). The Conjugate Gradients method is only guaranteed to work when $A$ is symmetric positive definite, whereas minres only requires $A$ to be symmetric. Conjugate Gradients also minimizes the norm of the residuals over the same Krylov subspaces, but not the 2-norm but the $A^{-1}$-norm,

$$\|Ax^{(t)} - b\|_{A^{-1}} = \sqrt{(Ax^{(t)} - b)^* A^{-1} (Ax^{(t)} - b)} .$$

This is equivalent to minimizing the $A$-norm of the error,

$$\|x - x^{(t)}\|_A = \sqrt{(x - x^{(t)})^* A (x - x^{(t)})} .$$

Minimizing the $A$-norm of the residual may seem like an odd idea, because of the dependence on $A$ in the measurement of the residual.
\[ q_1 = b / \| b \|_2 \]  
\[ w_1 = \| b \|_2 \]  
\[ H_{t+1,t} = \| \tilde{q}_{t+1} \|_2 \]  
\[ H_{t,t} = q_t^* A q_t \]  
\[ H_{t-1,t} = q_{t-1}^* A q_t \]  
\[ q_{t+1} = \tilde{q}_{t+1} / \| \tilde{q}_{t+1} \|_2 \]  
\[ \text{Apply rotation } t - 2 \text{ to } H_{t,t} \]  
\[ R_{t-2,t} = s_{t-2} H_{t-1,t} \]  
\[ \text{if } t > 2 \text{ then } U_{t-1,t} = c_{t-2} H_{t-1,t} \text{ else } U_{t-1,t} = H_{t-1,t} \]  
\[ \text{Apply rotation } t - 1 \text{ to } H_{t,t} \]  
\[ U_{t-1,t} = c_{t-1} U_{t-1,t} + s_{t-1} H_{t,t} \]  
\[ \text{if } t > 1 \text{ then } U_{t,t} = -s_{t-1} U_{t-1,t} + c_{t-1} H_{t,t} \text{ else } U_{t,t} = H_{t,t} \]  
\[ \text{compute } [c_t \; -s_t] \text{, the Givens rotation such that } [c_t \; -s_t] \begin{bmatrix} U_{t,t} \\ H_{t+1,t} \end{bmatrix} = \begin{bmatrix} \text{anything} \\ 0 \end{bmatrix} \]  
\[ \text{Apply rotation } t \text{ to } H_{t,t} \]  
\[ U_{t,t} = c_t U_{t,t} + s_t H_{t+1,t} \]  
\[ \text{Apply rotation } t \text{ to form } w = U_t z = V_t^* \| b \| e_1 \]  
\[ w_{t+1} = -s_t w_t \]  
\[ w_t = c_t w_t \]  
\[ m_t = r_{t,t}^{-1} (q_t - U_{t-1,t} m_{t-1} - U_{t-2,t} m_{t-2}) \]  
\[ x^{(t)} = x^{(t-1)} + w_t m_t \]  
\[ \text{end for} \]

**Figure 1. Minres.** To keep the pseudo-code simple, we use the convention that vectors and matrix/vector elements with nonpositive indices are zeros. By this convention, \( x^{(0)} = q_0 = m_0 = m_{-1} = 0 \), and so on. In an actual code, this convention can be implemented either using explicit zero vectors or using conditionals.

and the error. But there are several good reasons not to worry. First, we use the 2-norm of the residual as a stopping criterion, to stop the iterations only when the 2-norm is small enough, not when the \( A^{-1} \)-norm is small. Second, even when the stopping criterion is based on the 2-norm of the residual, Conjugate Gradients usually converges only
3. CONVERGENCE-RATE BOUNDS

slightly slower than MINRES. Still, the minimization of the 2-norm of the residual in MINRES is more elegant.

Why do people use Conjugate Gradients if MINRES is more theoretically appealing? The main reason that the matrices $M_t = Q_t R_t^{-1}$ and $R_t$ that are used in MINRES to form $x^{(t)}$ can be ill conditioned. This means that in floating-point arithmetic, the computed $x^{(t)}$ are not always accurate minimizers. In the Conjugate Gradients method, the columns of the basis matrix (the equivalent of $M_t$) are $A$-conjugate, not arbitrary. This reduces the inaccuracies in the computation of the approximate solution in each iterations, so the method is numerically more stable than MINRES.

We shall not derive the details of the Conjugate Gradient method here. The derivation is similar to that of MINRES (there are also other ways to derive CG), and is presented in many textbooks.

3. Convergence-Rate Bounds

The appeal of Krylov-subspace iterations stems to some extent from the fact that their convergence is easy to understand and to bound.

The crucial step in the analysis is the expression of the residual $b - Ax^{(t)}$ as an application of a univariate polynomial $\tilde{p}$ to $A$ and a multiplication of the resulting matrix $\tilde{p}(A)$ by $b$. Since $x^{(t)} \in K_t$, $x^{(t)} = K_t y$ for some $y$. That is, $x^{(t)} = y_1 b + y_2 A b + \cdots + y_t A^{t-1} b$. Therefore,

$$b - Ax^{(t)} = b - y_1 A b - y_2 A^2 b - \cdots - y_t A^t b .$$

If we denote $p(z) = 1 - y_1 z - y_2 z^2 - \cdots - y_t z^t = 1 - z\tilde{p}(z)$, we obtain $b - Ax^{(t)} = p(A)b$.

**Definition 3.1.** Let $x^{(t)} \in K_t$ be an approximate solution of $Ax = b$ and let $r^{(t)} = b - Ax^{(t)}$ be the corresponding residual. The polynomial $\tilde{p}_t$ such that $x^{(t)} = \tilde{p}_t(A)b$ is called the *solution polynomial* of the iteration and the polynomial $p_t(z) = 1 - z\tilde{p}(z)$ is called the *residual polynomial* of the iteration.

Figure 2 shows several MINRES residual polynomials.

We now express $p(A)$ in terms of the eigendecomposition of $A$. Let $A = V\Lambda V^*$ be an eigendecomposition of $A$. Since $A$ is Hermitian, $\Lambda$ is real and $V$ is unitary. We have

$$p(A)b = p(V\Lambda V^*)b = Vp(\Lambda)V^*b ,$$

so

$$\|b - Ax^{(t)}\|_2 = \|p(A)b\|_2 = \|Vp(\Lambda)V^*b\|_2 = \|p(\Lambda)V^*b\|_2 .$$

(2)
Figure 2. MINRES residual polynomials of four linear problems. The order \( n \) of the matrices is shown in each plot, as well as the degree of the polynomials. The eigenvalues of the matrices are shown using gray tick marks on the \( x \) axis. In all but the bottom-right plot the solution vector is a random vector; in the bottom-right plot, the solution vector (and hence also the right-hand side) is a random combination of only 100 eigenvectors of \( A \), those associated with the 100 smallest eigenvalues.

We can obtain several bounds on the norm of the residual from this expression. The most important one is

\[
\| b - Ax^{(t)} \|_2 = \| p(\Lambda) V^* b \|_2 \\
\leq \| p(\Lambda) \|_2 \| V^* b \|_2 = \| p(\Lambda) \|_2 \| b \|_2 \\
= \max_{\lambda_i} \{ |p(\lambda_i)| \} \| b \|_2 .
\]

The last equality follows from the facts that \( p(\Lambda) \) is a diagonal matrix and that the 2-norm of a diagonal matrix is the largest absolute value of an element in it. This proves the following result.
Theorem 3.2. The relative 2-norm of the residual in the tth iteration of minres,
\[ \|b - Ax^{(t)}\|_2 / \|b\|_2 , \]
is bounded by \( \max_{i=1}^{n} \{ |p(\lambda_i)| \} \) for any univariate polynomial \( p \) of degree \( t \) such that \( p(0) = 1 \), where the \( \lambda_i \)'s are the eigenvalues of \( A \).

We can strengthen this result by noting that if \( b \) is a linear combination of only some of the eigenvectors of \( A \), then only the action of \( p \) on corresponding eigenvalues matters (not on all the eigenvalues). An example of this behavior is shown in the bottom-right plot of Figure 2. More formally, from (2) we obtain
\[ \|p(\Lambda) V^* b\|_2 = \left\| \begin{bmatrix} p(\lambda_1)(v_1^* b) \\ p(\lambda_2)(v_2^* b) \\ \vdots \\ p(\lambda_n)(v_n^* b) \end{bmatrix} \right\|_2 . \]

In general, if \( b \) is orthogonal or nearly orthogonal to an eigenvector \( v_j \) of \( A \), then the product \( p(\lambda_j)(v_j^* b) \) can be small even if \( p(\lambda_j) \) is quite large. But since right-hand sides \( b \) with this property are rare in practice, this stronger bound is not useful for us.

Theorem 3.2 states that if there are low-degree polynomials that are low on the eigenvalues of \( A \) and assume the value 1 at 0, the minres converges quickly. Let us examine a few examples. A degree-\( n \) polynomial \( p \) can satisfy \( p(0) = 1 \) and \( p(\lambda_i) = 0 \) simultaneously for any set of \( n \) nonzero eigenvalues \( \lambda_1, \ldots, \lambda_n \). Therefore, in the absence of rounding errors minres must converge after \( n \) iterations to the exact solution. We could also derive this exact-convergence result from the fact that \( K_n = \mathbb{R}^n \), but the argument that we just gave characterizes the minres polynomials at or near convergence: their roots are at or near the eigenvalues of \( A \). If \( A \) has repeated eigenvalues, then it has fewer than \( n \) distinct eigenvalues, so we expect exact convergence after fewer than \( n \) iterations. Even if \( A \) does not have repeated eigenvalues, but it does have only a few tight clusters of eigenvalues, then minres will converge quickly, because a polynomial with one root near every cluster and a bounded derivative at the roots will assume low values at all the roots. On the other hand, a residual polynomial cannot have small values very close to 0, because it must assume the value 1 at 0. These examples lead us to the most important observation about Krylov-subspace solvers:

Symmetric Krylov-subspace iterative methods for solving linear systems of equations \( Ax = b \) converge quickly.
if the eigenvalues of $A$ form a few tight clusters and
if $A$ does not have eigenvalues very close to 0.
Scaling both $A$ and $b$ can cluster the eigenvalues or move them away from zero, but has no effect at all on convergence. Scaling up $A$ and $b$ moves the eigenvalues away from zero, but distributes them on a larger interval; scaling $A$ and $b$ down clusters the eigenvalues around zero, but this brings them closer to zero. Krylov-subspace iterations are invariant to scaling.

This observation leads to two questions, one analytic and one constructive: (1) Exactly how quickly does the iteration converges given some characterization of the spectrum of $A$? (2) How can we alter the spectrum of $A$ in order to accelerate convergence? We shall start with the second question.

4. Preconditioning

Suppose that we have a matrix $B$ that approximates $A$ (in a sense that will become clear shortly), and whose inverse is easier to apply than the inverse of $A$. That is, linear systems of the form $Bz = r$ are much easier to solve for $z$ than linear systems $Ax = b$. Perhaps the sparse Cholesky factorization of $B$ is cheaper to compute than $A$’s, and perhaps there is another inexpensive way to apply $B^{-1}$ to $r$. If $B$ approximates $A$ in the sense that $B^{-1}A$ is close to the identity, then an algorithm like MINRES will converge quickly when applied to the linear system

$$ (B^{-1}A)x = B^{-1}b, $$

because the eigenvalues of the coefficient matrix $B^{-1}A$ are clustered around 1. We will initialize the algorithm by computing the right-hand side $B^{-1}$, and in every iteration we will multiply $q_t$ by $A$ and then apply $B^{-1}$ to the product. This technique is called preconditioning.

The particular form of preconditioning that we used in (3) is called left preconditioning, because the inverse of the preconditioner $B$ is applied to both sides of $Ax = b$ from the left. Left preconditioning is not appropriate to algorithms like MINRES and Conjugate Gradients that exploit the symmetry of the coefficient matrix, because in general $B^{-1}A$ is not symmetric. We could replace MINRES by a Krylov-subspace iterative solver that is applicable to unsymmetric matrices, but this would force us to give up either the efficiency or the optimality of MINRES and Conjugate Gradients.

Fortunately, if $B$ is symmetric positive definite, then there are forms of preconditioning that are appropriate for symmetric Krylov-subspace
4. PRECONDITIONING

4. PRECONDITIONING

Figure 3. MINRES residual polynomials for a 15-by-15 two-dimensional mesh (both graphs). The graph on the left shows polynomial from the application of MINRES directly to the original linear problem \( Ax = b \), and the graph on the right shows a polynomial from the application of MINRES to a preconditioned problem \((L^{-1}AL^{-T})(L^Tx) = (L^{-1}b)\). The scaling of the axes in the two graphs are the same.

Solvers. One form of symmetric preconditioning solves

\[
(B^{-1/2}AB^{-1/2})(B^{1/2}x) = B^{-1/2}b
\]

for \( x \). A clever transformation of Conjugate Gradients yields an iterative algorithm in which the matrix that generates the Krylov subspace is \((B^{-1/2}AB^{-1/2})\), but which only applies \( A \) and \( B^{-1} \) in every iteration. In other words, \( B^{-1/2} \) is never used, not even as a linear operator. We shall not show this transformation here.

We will show how to use a simpler but equally effective form of preconditioning. Let \( B = LL^T \) be the Cholesky factorization of \( A \). We shall solve

\[
(L^{-1}AL^{-T})(L^Tx) = L^{-1}b
\]

for \( y = L^Tx \) using MINRES, and then we will solve \( L^Tx = y \) by substitution. To form the right-hand side \( L^{-1}b \), we solve for it by substitution as well. The coefficient matrix \( L^{-1}AL^{-T} \) is clearly symmetric, so we can indeed apply MINRES to it. To apply the coefficient matrix to \( q_t \) in every iteration, we apply \( L^{-T} \) by substitution, apply \( A \), and apply \( L^{-1} \) by substitution.

Figure 3 shows the spectrum and one MINRES polynomial for two matrices: a two-dimensional mesh and the same mesh preconditioned
as in (5) with a Joshi preconditioner $B$. We can see that this particular preconditioner causes three changes in the spectrum. The most important change is the small eigenvalues of $L^{-1}AL^{-T}$ are much larger than the small eigenvalues of $A$. This allows the minres polynomial to assume much smaller values on the spectrum. The minres polynomials must assume the value 1 at 0, so they tend to have large values on the neighborhood of 0 as well. Therefore, a spectrum with larger smallest eigenvalues leads to faster convergence. Indeed, the preconditioned problem decreased the size of the residual by a factor $10^{-14}$ in 46 iterations, whereas the unpreconditioned problem took 110 iterations to achieve a residual with a similar norm. Two other changes that the preconditioner caused are a large gap in the middle of the spectrum, which is a good thing, and a larger largest eigenvalue, which is not. But these two changes are probably less important than the large increase in the smallest eigenvalues.

The different forms of preconditioning differ in the algorithmic details of the solver, but they all have the same spectrum.

**Theorem 4.1.** Let $A$ be a symmetric matrix and let $B = LL^T$ be a symmetric positive-definite matrix. A scalar $\lambda$ is either an eigenvalue of all the following eigenvalue problems or of none of them:

\[
\begin{align*}
B^{-1}Ax &= \lambda x \\
B^{-1/2}AB^{-1/2}y &= \lambda y \\
L^{-1}AL^{-T}z &= \lambda z \\
Aw &= \lambda Bw
\end{align*}
\]

**Proof.** The following relations prove the equivalence of the spectra:

\[
\begin{align*}
x &= B^{-1/2}y \\
y &= B^{1/2}x \\
x &= L^{-T}z \\
z &= L^T x \\
x &= w
\end{align*}
\]

We can strengthen this theorem to also include certain semidefinite preconditioners.

**Theorem 4.2.** Let $A$ be a symmetric matrix and let $B = LL^T$ be a symmetric positive-semidefinite matrix such that null($B$) = null($A$). Denote by $X^+$ the pseudo-inverse of a matrix $X$. A scalar $\lambda$ is either
an eigenvalue of all the following eigenvalue problems or of none of them:

\[
\begin{align*}
B^+Ax &= \lambda x \\
(B^+)^{1/2} A (B^+)^{1/2} y &= \lambda y \\
(L^+) A (L^+)^T z &= \lambda z \\
Aw &= \lambda Bw
\end{align*}
\]

**Proof.** We note that \(\text{null}(L+) = \text{null}((L^+)^T) = \text{null}(B^+) = \text{null}(B) = \text{null}(A)\). Therefore, \(\lambda = 0\) is an eigenvalue of all the above problems. If \(\lambda \neq 0\) is an eigenvalue of one of the above problems, then the corresponding eigenvector is not in \(\text{null}(A)\). This implies that the relations defined in the proof of Theorem 4.1, with inverses replaced by pseudo-inverses, define relations between nonzero vectors. Therefore, \(\lambda\) is an eigenvalue of all the eigenvalue problems. \(\square\)

Even though the different forms of preconditioning are equivalent in terms of the spectra of the coefficient matrices, they are different algorithmically. If symmetry is not an issue (e.g., if \(A\) itself is unsymmetric), the form \(B^{-1}A\) is the most general. When \(A\) is symmetric, we usually require that \(B\) is symmetric positive-definite (or semi-definite with the same null space as \(A\)). In this case, the form \(B^{-1/2}AB^{-1/2}\), when coupled with the transformation that allows multiplications only by \(B^{-1}\) (and not by \(B^{-1/2}\), is more widely applicable than the form \(L^{-1}AL^{-T}\), because the latter requires a Cholesky factorization of \(B\), whereas in the former any method of applying \(B^{-1}\) can be used.

One issue that arises with any form of preconditioning is the definition of the residual. If we apply MINRES to \((L^{-1}AL^{-T})(L^T x) = L^{-1}b\), say, it minimizes the 2-norm of the *preconditioned residual*

\[
L^{-1}b - (L^{-1}AL^{-T})(L^T x^{(t)}) = L^{-1}b - L^{-1}Ax^{(t)} = L^{-1}(b - Ax^{(t)})
\]

Thus, the true residual \(b - Ax^{(t)}\) in preconditioned MINRES may not be minimal. This is roughly the same issue as with the norms used in Conjugate Gradients: we minimize the residual in a norm that is related to \(A\).

5. Chebyshev Polynomials and Convergence Bounds

The link between Krylov-subspace iterations and polynomials suggests another idea. Given some information on the spectrum of \(A\), we can try to analytically define a sequence \(\tilde{p}_n\) of solution polynomials such that for any \(b\) the vector \(x^{(t)} \equiv \tilde{p}_n(A)b \in K_t\) is a good approximation to \(x\), the exact solution of \(Ax = b\). More specifically, we can try to
define \( \tilde{p}_t \) such that the residuals \( b - Ax^{(t)} \) are small. We have seen that the residual can be expressed as \( p_t(A)b \), where \( p_t(z) = 1 - z\tilde{p}_t(z) \) is the residual polynomial. Therefore, if \( \tilde{p}_t \) is such that \( p_t \) assumes low values on the eigenvalues of \( A \) and satisfies \( p_t(0) = 1 \), then \( x^{(t)} \) is a good approximate solution. This idea can be used both to construct iterative solvers and to prove bounds on the convergence of methods like MINRES and Conjugate Gradients.

One obvious problem is that we do not know what the eigenvalues of \( A \) are. Finding the eigenvalues is more difficult than solving \( Ax = b \). However, in some cases we can use the structure of \( A \) (even with preconditioning) to derive bounds on the smallest and largest eigenvalues of a positive-definite matrix \( A \), denoted \( \lambda_{\text{min}} \) and \( \lambda_{\text{max}} \).

Suppose that we somehow obtained bounds on the extreme eigenvalues of \( A \),

\[
0 < \rho_{\text{min}} \leq \lambda_{\text{min}} \leq \lambda_i \leq \lambda_{\text{max}} \leq \rho_{\text{max}} .
\]

We shall not discuss here how we might obtain \( \rho_{\text{min}} \) and \( \rho_{\text{max}} \); this is the topic of much of the rest of the book. It turns out that we can build a sequence \( p_t \) of polynomials such that

1. \( p_t(0) = 1 \), and
2. \( \max_{z \in [\rho_{\text{min}}, \rho_{\text{max}}]} |p_t(z)| \) is as small as possible for a degree \( t \) polynomial with value 1 at 0.

The polynomials that solve this optimization problem are derived from Chebyshev polynomials, which can be defined using the recurrence

\[
c_0(z) = 1 \\
c_1(z) = z \\
c_t(z) = 2zc_{t-1}(z) - c_{t-2}(z).
\]

The polynomials that reduce the residual are

\[
p_t(z) = \frac{1}{c_t \left( \frac{\rho_{\text{max}} + \rho_{\text{min}}}{\rho_{\text{max}} - \rho_{\text{min}}} \right)} c_t \left( \frac{\rho_{\text{max}} + \rho_{\text{min}} - 2z}{\rho_{\text{max}} - \rho_{\text{min}}} \right) .
\]

**An Iterative Linear Solver based on Chebyshev Polynomials.** Our first application of Chebyshev polynomials is an iterative Krylov-subspace solver based on them. We will refer to this solver as the *Krylov-Chebyshev* solver\(^1\). The polynomials \( p_t \) implicitly define polynomials \( \tilde{p}_t \) that we can use to construct approximate solutions. The residual for an approximate solution \( x^{(t)} \) is \( r^{(t)} = b - Ax^{(t)} \). If we define \( x^{(t)} = \tilde{p}_t(A)b \), we have \( r^{(t)} = p_t(A)b = b - A\tilde{p}_t(A)b \).

\(^1\)The algorithm that we describe below is related to a more well-known Chebyshev linear solver that is used with matrix splittings.
We now derive recurrences for $x^{(t)}$ and $r^{(t)}$. To keep the notation simple, we define $\rho_+ = \rho_{\text{max}} + \rho_{\text{min}}$ and $\rho_- = \rho_{\text{max}} - \rho_{\text{min}}$. For the two base cases we have
\[
 r^{(0)} = p_0(A)b = c_0^{-1} \left( \frac{\rho_+}{\rho_-} \right) c_0 \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) b = Ib = b
\]
\[
 r^{(1)} = p_1(A)b = c_1^{-1} \left( \frac{\rho_+}{\rho_-} \right) c_1 \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) b
 = \left( \frac{\rho_+}{\rho_-} \right)^{-1} \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) b = b - \frac{2}{\rho_+} Ab .
\]
This implies that
\[
 x^{(0)} = 0
\]
\[
 x^{(1)} = \frac{2}{\rho_+} b .
\]
For $t \geq 2$ we have
\[
p_t(A)b = c_t^{-1} \left( \frac{\rho_+}{\rho_-} \right) c_t \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) b
 = c_t^{-1} \left( \frac{\rho_+}{\rho_-} \right) \left( 2 \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) c_{t-1} \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) - c_{t-2} \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) \right) b
 = c_t^{-1} \left( \frac{\rho_+}{\rho_-} \right) \left( 2 \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) c_{t-1} \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) \right) p_{t-1}(A) - c_{t-2} \left( \frac{\rho_+}{\rho_-} \right) p_{t-2}(A) b
 = c_t^{-1} \left( \frac{\rho_+}{\rho_-} \right) \left( 2 \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) c_{t-1} \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) \right) r^{(t-1)} - c_{t-2} \left( \frac{\rho_+}{\rho_-} \right) r^{(t-2)} .
\]
To compute $r^{(t)}$ from this recurrence, we need $r^{(t-1)}$ and $r^{(t-2)}$, three elements of the sequence $c_t(\rho_+/\rho_-)$, and one multiplication of a vector by $A$. Therefore, we can compute $r^{(t)}$ and $c_t(\rho_+/\rho_-)$ concurrently in a loop. From the recurrence for $r^{(t)}$ we can derive a recurrence for $x^{(t)}$,
\[
r^{(t)} = p_t(A)b
 = c_t^{-1} \left( \frac{\rho_+}{\rho_-} \right) \left( \frac{2\rho_+}{\rho_-} c_{t-1} \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) r^{(t-1)} - 4 \rho_- A c_{t-1} \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) r^{(t-1)} - c_{t-2} \left( \frac{\rho_+}{\rho_-} \right) r^{(t-2)} \right)
 = c_t^{-1} \left( \frac{\rho_+}{\rho_-} \right) \left( \frac{2\rho_+}{\rho_-} c_{t-1} \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) \right) \left( b - Ax^{(t-1)} \right)
 - 4 \rho_- A c_{t-1} \left( \frac{\rho_+}{\rho_-} I - \frac{2}{\rho_-} A \right) r^{(t-1)}
 - c_{t-2} \left( \frac{\rho_+}{\rho_-} \right) \left( b - Ax^{(t-2)} \right) .
\]
5. CHEBYSHEV POLYNOMIALS AND CONVERGENCE BOUNDS

Figure 4. Chebyshev residual polynomials for two 200-by-200 matrices with different spectra, where $\rho_{\text{min}} = \lambda_{\text{min}}$ is set at the minimal eigenvalue and $\rho_{\text{max}} = \lambda_{\text{max}}$ is set at the largest eigenvalue. The two matrices have the same extreme eigenvalues but otherwise their spectra is very different. The Chebyshev polynomials depend only on $\rho_{\text{max}}$ and $\rho_{\text{min}}$. Compare to the MINRES polynomials for the similar spectra in Figure 2.

so

$$x^{(t)} = \tilde{p}_t(A) b$$

$$= c_t^{-1} \left( \frac{\rho_+}{\rho_-} \right) \left( \frac{2\rho_+}{\rho_-} c_t^{-1} \left( \frac{\rho_+}{\rho_-} \right) x^{(t-1)} + 4 \frac{\rho_+}{\rho_-} c_t^{-1} \left( \frac{\rho_+}{\rho_-} \right) x^{(t-1)} - c_t^{-2} \left( \frac{\rho_+}{\rho_-} \right) x^{(t-2)} \right).$$

This algorithm converges more slowly than MINRES and Conjugate Gradients. MINRES is guaranteed to minimize the residual over all $x^{(t)}$ in $K_t$. The solution that we obtain from the Krylov-Chebyshev recurrences is in $K_t$, so it cannot yield a smaller residual than the MINRES solution. Its main algorithmic advantage over MINRES and Conjugate Gradients is that the polynomials $\tilde{p}_t$ that it produces depend only on $t$ and on $\rho_{\text{min}}$ and $\rho_{\text{max}}$, but they do not depend on $b$. Therefore, $\tilde{p}_t(A)$ is a fixed linear operator, so it can be used as a preconditioner $B^{-1} = \tilde{p}_t(A)$. In contrast, the polynomials that MINRES and Conjugate Gradients generate depend on $b$, so we cannot use these solvers as preconditioners unless we set very strict convergence bounds, in which
case an application of these solvers is numerically indistinguishable from an application of $A^{-1}$.

Since this algorithm does not exploit the symmetry of $A$, it can be used with left preconditioning, not only with symmetric forms of preconditioning.

**Chebyshev-Based Convergence Bounds.** We can also use the Chebyshev iteration to bound the convergence of MINRES and Conjugate Gradients. The key is the following theorem, which we state without a proof.

**Theorem 5.1.** Let $p(z)$ be the Chebyshev polynomials defined above with respect to the interval $[\lambda_{\text{min}}, \lambda_{\text{max}}]$. Denote by $\kappa = \lambda_{\text{max}}/\lambda_{\text{min}}$. For any $z \in [\lambda_{\text{min}}, \lambda_{\text{max}}]$ we have

$$|p_t(z)| \leq 2 \left( \frac{\kappa + 1}{\sqrt{\kappa} - 1} \right)^t + \left( \frac{\kappa + 1}{\sqrt{\kappa} - 1} \right)^{-1} \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^t.$$

In this theorem, $0 < \lambda_{\text{min}} \leq \lambda_{\text{max}}$ are arbitrary positive numbers that denote the end of the interval that defines $p_t$. But when these numbers are the extreme eigenvalues of a symmetric positive definite matrix, the ratio $\kappa = \lambda_{\text{max}}/\lambda_{\text{min}}$ plays an important-enough role in numerical linear algebra to deserve a name.

**Definition 5.2.** Let $A$ be a symmetric semidefinite matrix, and let $\lambda_{\text{min}}$ and $\lambda_{\text{max}}$ be its extreme nonzero eigenvalue. The ratio

$$\kappa = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} = \|A\|_2 \|A^+\|_2$$

is called the *spectral condition number* of $A$. The definition

$$\kappa = \|A\| \|A^+\|$$

generalizes the condition number to any matrix and to any norm.

We can use Theorem 5.1 to bound the residuals in MINRES.

**Theorem 5.3.** Consider the application of MINRES to the linear system $Ax = b$. Let $r(t)$ be the MINRES residual at iteration $t$. Then

$$\frac{\|r(t)\|_2}{\|b\|_2} \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^t.$$

**Proof.** The residual $r(t)$ is the minimal residual for any $x(t) \in K_t$. Let $r(t)$ be the Krylov-Chebyshev residual for $A$ and $b$ and let $p_t$ be the Krylov-Chebyshev residual polynomial. Let $A = V\Lambda V^*$ be
an eigendecomposition of $A$. The theorem follows from the following inequalities.

$$
\| r(t) \|_2 \leq \| r(t) \|_2 = \| p(A)b \|_2 = \| VP(\Lambda)V^*b \|_2 = \| p(\Lambda)V^*b \|_2 \\
\leq \| p(\Lambda)V^* \|_2 \| b \|_2 = \| p(\Lambda) \|_2 \| b \|_2 \\
= \max \{ |p(\lambda_i)| \} \| b \|_2 \\
\leq \max_{z \in [\lambda_{\min}, \lambda_{\max}]} \{ |p(z)| \} \| b \|_2 \\
\leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^t \| b \|_2.
$$

Similar results can be stated for the error and residual of Conjugate Gradients in the $A$ and $A^{-1}$ norms, respectively.

For small $\kappa$, we can expect convergence to a fixed tolerance, say $\| r(t) \|_2 \leq 10^{-12} \| r(t) \|_2$ within a constant number of iterations. As $\kappa$ grows,

$$
\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \to 1 - \frac{2}{\sqrt{\kappa}},
$$

so we are guaranteed convergence to a fixed tolerance within $O(\sqrt{\kappa})$ iterations.