Lecture 38. Conjugate Gradients

The conjugate gradient iteration is the "original" Krylov subspace iteration, the most famous of these methods and one of the mainstays of scientific computing. Discovered by Hestenes and Stiefel in 1952, it solves symmetric positive definite systems of equations amazingly quickly if the eigenvalues are well distributed.

Minimizing the 2-Norm of the Residual

As in the last two lectures, let $A \in \mathbb{R}^{m\times m}$ be real and symmetric, and suppose we wish to solve a nonsingular system of equations $Ax = b$, with exact solution $x_* = A^{-1}b$. Let $\mathcal{K}_n$ denote the nth Krylov subspace (33.5) generated by b,

$$\mathcal{K}_n \;=\; \langle b, Ab, \ldots, A^{n-1}b\rangle. \tag{38.1}$$

One approach based on this Krylov subspace would be to solve the system by GMRES. As described in Lecture 35, this would mean that at step n, x_* is approximated by the vector $x_n \in \mathcal{K}_n$ that minimizes $\|r_n\|_2$, where $r_n = b - Ax_n$. Actually, the usual GMRES algorithm does more work than is necessary for minimizing $\|r_n\|_2$. Since A is symmetric, faster algorithms are available based on three-term instead of $(n+1)$-term recurrences at step n. One of these goes by the names of conjugate residuals or MINRES ("minimal residuals").

These methods, at least when constructed to apply to both definite and indefinite matrices, involve certain complications. Rather than describe them, we turn directly to the simpler and more important positive definite case.

Minimizing the A-Norm of the Error

Assume that A is not only real and symmetric but also *positive definite*. As discussed in Lecture 23, this means that the eigenvalues of A are all positive, or equivalently, that $x^TAx > 0$ for every nonzero $x \in \mathbb{R}^m$. Under this assumption, the function $\|\cdot\|_A$ defined by

$$\|x\|_A = \sqrt{x^TAx} \tag{38.2}$$

is a norm on $\mathbb{R}^m$, as can be verified from the definition (3.1). It is called the *A-norm*. (This is the same as the norm $\|x\|_W$ of (3.3), if W is a Cholesky factor of A or any other matrix satisfying $W^TW = A$.)

The vector whose A-norm will concern us is $e_n = x_* - x_n$, the error at step n. The conjugate gradient iteration can be described as follows. *It is a system of recurrence formulas that generates the unique sequence of iterates $\{x_n \in \mathcal{K}_n\}$ with the property that at step n, $\|e_n\|_A$ is minimized.*

We shall present the formulas for the CG iteration, without motivation at first, and derive some orthogonality properties (Theorem 38.1). From these, the claim about minimality of $\|e_n\|_A$ follows as a corollary (Theorem 38.2), and the motivation appears belatedly as we interpret CG as a nonlinear optimization algorithm.

The Conjugate Gradient Iteration

Here is the iteration that Hestenes and Stiefel made famous.

Algorithm 38.1. Conjugate Gradient (CG) Iteration

```
$x_0 = 0,\ r_0 = b,\ p_0 = r_0$
for $n = 1, 2, 3, \ldots$
    $\alpha_n = (r_{n-1}^T r_{n-1})/(p_{n-1}^T A p_{n-1})$        step length
    $x_n = x_{n-1} + \alpha_n p_{n-1}$                             approximate solution
    $r_n = r_{n-1} - \alpha_n A p_{n-1}$                           residual
    $\beta_n = (r_n^T r_n)/(r_{n-1}^T r_{n-1})$                    improvement this step
    $p_n = r_n + \beta_n p_{n-1}$                                  search direction
```

Before analyzing the mathematical properties of these formulas, let us examine them operationally. First we note that the CG iteration is extraordinarily simple—programmable in a few lines of MATLAB. Since it deals only with m-vectors, not with individual entries of vectors or matrices, it is simpler, for example, than Gaussian elimination with pivoting. The only complication—which we shall not address—is the choice of a convergence criterion.

At each step, the CG iteration involves several vector manipulations and one matrix-vector product, the computation of Ap_{n-1} (which appears twice in the listing but need be computed only once). If A is dense and unstructured, this matrix-vector product dominates the operation count, which becomes $\sim 2m^2$ flops for each step. If A is sparse or has other exploitable structure, Ap_{n-1} may be computable in as few as $O(m)$ operations, in which case the operation count may be as low as $O(m)$ flops per step.

From the five lines that define the algorithm, the following properties can be deduced. Like all the theorems in this book that do not explicitly mention rounding errors, this one assumes that the computation is performed in exact arithmetic. If there are rounding errors, these properties fail, and it becomes a subtle matter to explain the still very impressive performance of CG.

Theorem 38.1. *Let the CG iteration (Algorithm 38.1) be applied to a symmetric positive definite matrix problem $Ax = b$. As long as the iteration has not yet converged (i.e., $r_{n-1} \neq 0$), the algorithm proceeds without divisions by zero, and we have the following identities of subspaces:*

$$\begin{aligned} \mathcal{K}_n &= \langle x_1, x_2, \dots, x_n \rangle = \langle p_0, p_1, \dots, p_{n-1} \rangle \\ &= \langle r_0, r_1, \dots, r_{n-1} \rangle = \langle b, Ab, \dots, A^{n-1}b \rangle. \end{aligned} \tag{38.3}$$

Moreover, the residuals are orthogonal,

$$r_n^T r_j = 0 \qquad (j < n), \tag{38.4}$$

and the search directions are "A-conjugate,"

$$p_n^T A p_j = 0 \qquad (j < n). \tag{38.5}$$

Proof. The proof is by induction on n; we sketch it informally. From the initial guess $x_0 = 0$ and the formula $x_n = x_{n-1} + \alpha_n p_{n-1}$, it follows by induction that x_n belongs to $\langle p_0, p_1, \dots, p_{n-1} \rangle$. From $p_n = r_n + \beta_n p_{n-1}$ it follows that this is the same as $\langle r_0, r_1, \dots, r_{n-1} \rangle$. From $r_n = r_{n-1} - \alpha_n A p_{n-1}$, finally, it follows that this is the same as $\langle b, Ab, \dots, A^{n-1}b \rangle$. This establishes (38.3).

To prove (38.4), we apply the formula $r_n = r_{n-1} - \alpha_n A p_{n-1}$ and the identity $(Ap_{n-1})^T = p_{n-1}^T A$ to compute

$$r_n^T r_j = r_{n-1}^T r_j - \alpha_n p_{n-1}^T A r_j.$$

If $j < n-1$, both terms on the right are zero by induction. If $j = n-1$, the difference on the right is zero provided $\alpha_n = (r_{n-1}^T r_{n-1})/(p_{n-1}^T A r_{n-1})$. Now this is the same as the line $\alpha_n = (r_{n-1}^T r_{n-1})/(p_{n-1}^T A p_{n-1})$ of Algorithm 38.1, except that $p_{n-1}^T A p_{n-1}$ has been replaced by $p_{n-1}^T A r_{n-1}$. Since p_{n-1} and r_{n-1} differ by $\beta_{n-1} p_{n-2}$, the effect of this replacement is to change the denominator by $\beta_{n-1} p_{n-1}^T A p_{n-2}$, which is zero by the induction hypothesis.

To prove (38.5), we apply the formula $p_n = r_n + \beta_n p_{n-1}$ to compute

$$p_n^T A p_j = r_n^T A p_j + \beta_n p_{n-1}^T A p_j.$$

If $j < n-1$, both terms on the right are again zero by induction (since (38.4) has now been established for case n). If $j = n-1$, the sum on the right is zero provided $\beta_n = -(r_n^T A p_{n-1})/(p_{n-1}^T A p_{n-1})$, which we can write equivalently in the form $\beta_n = (-\alpha_n r_n^T A p_{n-1})/(\alpha_n p_{n-1}^T A p_{n-1})$. This is the same as the line $\beta_n = (r_n^T r_n)/(r_{n-1}^T r_{n-1})$ of Algorithm 38.1, except that $r_n^T r_n$ has been replaced by $r_n^T(-\alpha_n A p_{n-1})$ and $r_{n-1}^T r_{n-1}$ has been replaced by $p_{n-1}^T(\alpha_n A p_{n-1})$. By the induction hypothesis and lines 3 and 5 of Algorithm 38.1, these replacements can again readily be shown to have zero effect. □

Optimality of CG

In deriving the orthogonality properties (38.4) and (38.5), we have finished the real work. It is now a straightforward matter to confirm that CG minimizes $\|e\|_A$ at each step.

Theorem 38.2. *Let the CG iteration be applied to a symmetric positive definite matrix problem $Ax = b$. If the iteration has not already converged (i.e., $r_{n-1} \neq 0$), then x_n is the unique point in $\mathcal{K}_n$ that minimizes $\|e_n\|_A$. The convergence is monotonic,*

$$\|e_n\|_A \le \|e_{n-1}\|_A, \qquad (38.6)$$

and $e_n = 0$ is achieved for some $n \le m$.

Proof. From Theorem 38.1 we know that x_n belongs to $\mathcal{K}_n$. To show that it is the unique point in $\mathcal{K}_n$ that minimizes $\|e\|_A$, consider an arbitrary point $x = x_n - \Delta x \in \mathcal{K}_n$, with error $e = x_* - x = e_n + \Delta x$. We calculate

$$\begin{aligned}\|e\|_A^2 &= (e_n + \Delta x)^T A (e_n + \Delta x) \\ &= e_n^T A e_n + (\Delta x)^T A (\Delta x) + 2 e_n^T A (\Delta x).\end{aligned}$$

The final term in this equation is $2r_n^T(\Delta x)$, an inner product of r_n with a vector in $\mathcal{K}_n$, and by Theorem 38.1, any such inner product is zero. This is the crucial orthogonality property that makes the CG iteration so powerful. It implies that we have

$$\|e\|_A^2 = e_n^T A e_n + (\Delta x)^T A (\Delta x).$$

Only the second of these terms depends on Δx, and since A is positive definite, that term is ≥ 0, attaining the value 0 if and only if $\Delta x = 0$, i.e., $x_n = x$. Thus $\|e\|_A$ is minimal if and only if $x_n = x$, as claimed.

The remaining statements of the theorem now follow readily. The monotonicity property (38.6) is a consequence of the inclusion $\mathcal{K}_n \subseteq \mathcal{K}_{n+1}$, and since $\mathcal{K}_n$ is a subset of $\mathbb{R}^m$ of dimension n as long as convergence has not yet been achieved, convergence must be achieved in at most m steps. $\square$

The guarantee that the CG iteration converges in at most m steps is void in floating point arithmetic. For arbitrary matrices A on a real computer, no decisive reduction in $\|e_n\|_A$ will necessarily be observed at all when $n = m$. In practice, however, CG is used not for arbitrary matrices but for matrices whose spectra, perhaps thanks to preconditioning, are well-enough behaved that convergence to a desired accuracy is achieved for $n \ll m$ (Lecture 32). The theoretical exact convergence at $n = m$ has no relevance to this use of the CG iteration in scientific computing.

CG as an Optimization Algorithm

We have just shown that the CG iteration has a certain optimality property: it minimizes $\|e_n\|_A$ at step n over all vectors $x \in \mathcal{K}_n$. In fact, as foreshadowed already by the use of such terms as "step length" and "search direction," this iteration can be interpreted as an algorithm of a standard form for minimizing a nonlinear function of $x \in \mathbb{R}^m$. At the heart of the iteration is the formula

$$x_n = x_{n-1} + \alpha_n p_{n-1}.$$

This is a familiar equation in optimization, in which a current approximation x_{n-1} is updated to a new approximation x_n by moving a distance α_n (the step length) in the direction p_{n-1} (the search direction). By a succession of such steps, the CG iteration attempts to find a minimum of a nonlinear function.

Which function? According to Theorem 38.2, the answer would appear to be $\|e\|_A$, or equivalently, $\|e\|_A^2$. However, although $\|e\|_A^2$ is indeed a function of x, it is not one we can evaluate without knowing x_*. It would not be very "standard" to interpret CG as an optimization process applied to a function that cannot be evaluated!

On the other hand, given A and b and $x \in \mathbb{R}^m$, the quantity

$$\varphi(x) \;=\; \tfrac{1}{2}x^T A x - x^T b \tag{38.7}$$

can certainly be evaluated. A short computation now reveals

$$\begin{aligned}\|e_n\|_A^2 \;&=\; e_n^T A e_n \;=\; (x_* - x_n)^T A (x_* - x_n)\\ &=\; x_n^T A x_n - 2x_n^T A x_* + x_*^T A x_*\\ &=\; x_n^T A x_n - 2x_n^T b + x_*^T b \;=\; 2\varphi(x_n) + \text{constant}.\end{aligned}$$

Thus $\varphi(x)$ is the same as $\|e\|_A^2$ except for a factor of 2 and the (unknown) constant $x_*^T b$. Like $\|e\|_A^2$, it must achieve its minimum (namely, $-x_*^T b/2$) uniquely at $x = x_*$.

The CG iteration can be interpreted as an iterative process for minimizing the quadratic function $\varphi(x)$ of $x \in \mathbb{R}^m$. At each step, an iterate $x_n = x_{n-1} + \alpha_n p_{n-1}$ is computed that minimizes $\varphi(x)$ over all x in the one-dimensional space $x_{n-1} + \langle p_{n-1}\rangle$. (It is readily confirmed that the formula $\alpha_n = (r_{n-1}^T r_{n-1})/(p_{n-1}^T A p_{n-1})$ ensures that α_n is optimal in this sense among all step lengths α.) What makes the CG iteration remarkable is the choice of the search direction p_{n-1}, which has the special property that minimizing $\varphi(x)$ over $x_{n-1} + \langle p_{n-1}\rangle$ actually minimizes it over all of $\mathcal{K}_n$.

There is a close analogy between the CG iteration for solving $Ax = b$ and the Lanczos iteration for finding eigenvalues. The eigenvalues of A, as discussed in Lecture 27, are the stationary values for $x \in \mathbb{R}^m$ of the Rayleigh quotient, $r(x) = (x^T Ax)/(x^T x)$. As pointed out in Exercise 36.1, the eigenvalue estimates (Ritz values) associated with step n of the Lanczos iteration are the stationary values of the same function $r(x)$ if x is restricted to the Krylov subspace $\mathcal{K}_n$. This is a perfect parallel of what we have shown in the last two pages, that the solution x_* of $Ax = b$ is the minimal point in $\mathbb{R}^m$ of the scalar function $\varphi(x)$, and the CG iterate x_n is the minimal point of the same function $\varphi(x)$ if x is restricted to $\mathcal{K}_n$.

CG and Polynomial Approximation

A theme of the last four lectures has been the connection between Krylov subspace iterations and polynomials of matrices. The Arnoldi and Lanczos iterations solve the Arnoldi/Lanczos approximation problem (34.3), and the GMRES iteration solves the GMRES approximation problem (35.10). For CG, the appropriate approximation problem involves the A-norm of the error.

CG Approximation Problem. Find $p_n \in P_n$ such that

$$\|p_n(A)e_0\|_A = \text{minimum}. \tag{38.8}$$

Here e_0 denotes the initial error, $e_0 = x_* - x_0 = x_*$, and P_n is again defined as in (35.7), the set of polynomials p of degree $\le n$ with $p(0) = 1$. From Theorem 38.2 we may derive the following convergence theorem.

Theorem 38.3. *If the CG iteration has not already converged before step n (i.e., $r_{n-1} \neq 0$), then* (38.8) *has a unique solution $p_n \in P_n$, and the iterate x_n has error $e_n = p_n(A)e_0$ for this same polynomial p_n. Consequently we have*

$$\frac{\|e_n\|_A}{\|e_0\|_A} = \inf_{p\in P_n} \frac{\|p(A)e_0\|_A}{\|e_0\|_A} \le \inf_{p\in P_n} \max_{\lambda\in\Lambda(A)} |p(\lambda)|, \tag{38.9}$$

where $\Lambda(A)$ denotes the spectrum of A.

Proof. From Theorem 38.1 it follows that $e_n = p(A)e_0$ for some $p \in P_n$. The equality in (38.9) is a consequence of this and Theorem 38.2. As for the inequality in (38.9), if $e_0 = \sum_{j=1}^m a_j v_j$ is an expansion of e_0 in orthonormal eigenvectors of A, then we have $p(A)e_0 = \sum_{j=1}^m a_j p(\lambda_j) v_j$ and thus

$$\|e_0\|_A^2 = \sum_{j=1}^m a_j^2 \lambda_j, \qquad \|p(A)e_0\|_A^2 = \sum_{j=1}^m a_j^2 \lambda_j (p(\lambda_j))^2.$$

These identities imply $\|p(A)e_0\|_A^2 / \|e_0\|_A^2 \le \max_{\lambda \in \Lambda(A)} |p(\lambda)|^2$, which implies the inequality in question. □

Rate of Convergence

Theorem 38.3 establishes that the rate of convergence of the CG iteration is determined by the location of the spectrum of A. A good spectrum is one on which polynomials $p_n \in P_n$ can be very small, with size decreasing rapidly with n. Roughly speaking, this may happen for either or both of two reasons: the eigenvalues may be grouped in small clusters, or they may lie well separated in a relative sense from the origin. The two best-known corollaries of Theorem 38.3 address these two ideas in their extreme forms.

First, we suppose that the eigenvalues are perfectly clustered but assume nothing about the locations of these clusters.

Theorem 38.4. *If A has only n distinct eigenvalues, then the CG iteration converges in at most n steps.*

Proof. This is a corollary of (38.9), since a polynomial $p(x) = \prod_{j=1}^n (1 - x/\lambda_j) \in P_n$ exists that is zero at any specified set of n points $\{\lambda_j\}$. □

At the other extreme, suppose we know nothing about any clustering of the eigenvalues but only that their distances from the origin vary by at most a factor $\kappa \ge 1$. In other words, suppose we know only the 2-norm condition number $\kappa = \lambda_{\max}/\lambda_{\min}$, where $\lambda_{\max}$ and $\lambda_{\min}$ are the extreme eigenvalues of A.

Theorem 38.5. *Let the CG iteration be applied to a symmetric positive definite matrix problem $Ax = b$, where A has 2-norm condition number κ. Then the A-norms of the errors satisfy*

$$\frac{\|e_n\|_A}{\|e_0\|_A} \le 2 \Bigg/ \left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)^n + \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)^{-n} \right] \le 2 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^n. \tag{38.10}$$

Proof. By Theorem 38.3, it is enough to find a polynomial $p \in P_n$ whose maximum value for $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ is the middle expression of (38.10). The polynomial we choose is the scaled and shifted Chebyshev polynomial $p(x) = T_n(\gamma - 2x/(\lambda_{\max} - \lambda_{\min}))/T_n(\gamma)$, where T_n is the usual Chebyshev polynomial of degree n and γ takes the special value $\gamma = (\lambda_{\max} + \lambda_{\min})/(\lambda_{\max} - \lambda_{\min}) = (\kappa+1)/(\kappa-1)$. For $x \in [\lambda_{\min}, \lambda_{\max}]$, the argument of T_n in the numerator of $p(x)$ lies in $[-1, 1]$, which means the magnitude of that numerator is ≤ 1. Therefore, to prove the theorem, it will suffice to show

$$T_n(\gamma) \;=\; T_n\left(\frac{\kappa+1}{\kappa-1}\right) \;=\; \frac{1}{2}\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^n + \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^{-n}\right]. \qquad (38.11)$$

We can do this by making the change of variables $x = \frac{1}{2}(z + z^{-1})$, $T_n(x) = \frac{1}{2}(z^n + z^{-n})$, standard in the study of Chebyshev polynomials. If $(\kappa+1)/(\kappa-1) = \frac{1}{2}(z+z^{-1})$, that is, $\frac{1}{2}z^2 - (\kappa+1)/(\kappa-1)z + \frac{1}{2} = 0$, then we have a quadratic equation with solution

$$\begin{aligned} z \;&=\; \left(\frac{\kappa+1}{\kappa-1}\right) + \sqrt{\left(\frac{\kappa+1}{\kappa-1}\right)^2 - 1} \;=\; \frac{\kappa+1+\sqrt{(\kappa+1)^2-(\kappa-1)^2}}{\kappa-1} \\ &=\; \frac{\kappa+1+\sqrt{4\kappa}}{\kappa-1} \;=\; \frac{(\sqrt{\kappa}+1)^2}{(\sqrt{\kappa}+1)(\sqrt{\kappa}-1)} \;=\; \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}. \end{aligned}$$

Thus $T_n(\gamma) = \frac{1}{2}(z^n + z^{-n})$ for this value of z, which is (38.11), as claimed. □

Theorem 38.5 is the most famous result about convergence of the CG iteration. Since

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \sim 1 - \frac{2}{\sqrt{\kappa}}$$

as $\kappa \to \infty$, it implies that if κ is large but not too large, convergence to a specified tolerance can be expected in $O(\sqrt{\kappa})$ iterations. One must remember that this is only an upper bound. Convergence may be faster for special right-hand sides (not so common) or if the spectrum is clustered (more common).

Example

For an example of the convergence of CG, consider a 500×500 sparse matrix A constructed as follows. First we put 1 at each diagonal position and a random number from the uniform distribution on $[-1, 1]$ at each off-diagonal position (maintaining the symmetry $A = A^T$). Then we replace each off-diagonal entry with $|a_{ij}| > \tau$ by zero, where τ is a parameter. For τ close to zero, the result is a well-conditioned positive definite matrix whose density of nonzero entries is approximately τ. As τ increases, both the condition number and the sparsity deteriorate.

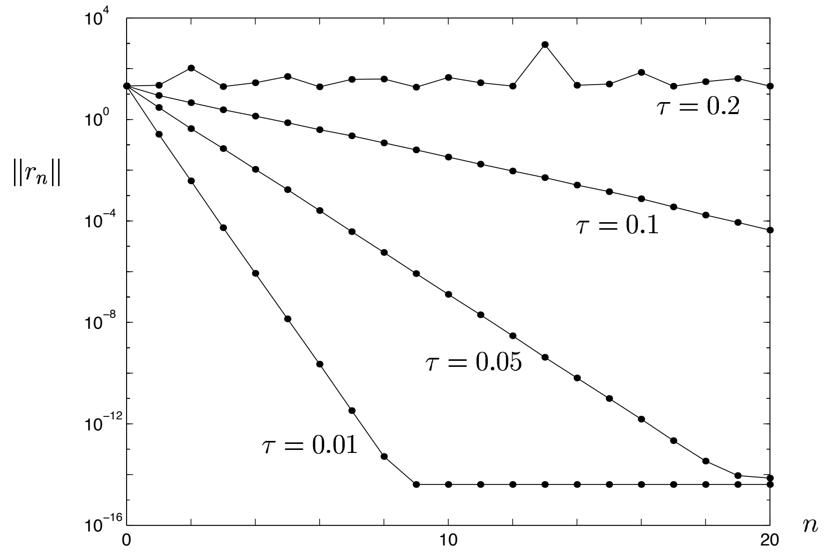

Figure 38.1. *CG convergence curves for the* 500×500 *sparse matrices* A *described in the text. For* $\tau = 0.01$*, the system is solved about* 700 *times faster by CG than by Cholesky factorization. For* $\tau = 0.2$*, the matrix is not positive definite and there is no convergence.*

Figure 38.1 shows convergence curves corresponding to 20 steps of the CG iteration for matrices of this kind with $\tau = 0.01, 0.05, 0.1, 0.2$. (The right-hand side b was taken to be a random vector.) For $\tau = 0.01$, A has 3092 nonzero entries and condition number $\kappa \approx 1.06$. Convergence to machine precision takes place in 9 steps, about 6×10^4 flops. For $\tau = 0.05$, there are 13,062 nonzeros with $\kappa \approx 1.83$, and convergence takes 19 steps, about 5×10^5 flops. For $\tau = 0.1$ we have 25,526 nonzeros and $\kappa \approx 10.3$, with only 5 digits of convergence after 20 steps and 10^6 flops. For $\tau = 0.2$, with 50,834 nonzeros, there is no convergence at all. The lowest eigenvalue is now negative, so A is no longer positive definite and the use of the CG iteration is inappropriate. (In fact, the CG iteration often succeeds with indefinite matrices, but in this case the matrix is not only indefinite but ill-conditioned.)

Note how closely the $\tau = 0.01$ curve of Figure 38.1 matches the schematic ideal depicted in Figure 32.1! For this example, the operation count of 6×10^4 flops beats Cholesky factorization (23.4) by a factor of about 700. Unfortunately, not every matrix arising in practice has such a well-behaved spectrum, even after the best efforts to find a good preconditioner.

Exercises

38.1. Based on the condition numbers κ reported in the text, determine the rate of convergence predicted by Theorem 38.5 for the matrices A of Figure 38.1 with $\tau = 0.01, 0.05, 0.1$. Draw lines on a copy of Figure 38.1 indicating how closely these predictions match the actual convergence rates.

38.2. Suppose A is a real symmetric 805×805 matrix with eigenvalues $1.00, 1.01, 1.02, \ldots, 8.98, 8.99, 9.00$ and also $10, 12, 16, 24$. How many steps of the conjugate gradient iteration must you take to be sure of reducing the initial error $\|e_0\|_A$ by a factor of 10^6 ?

38.3. The conjugate gradient is applied to a symmetric positive definite matrix A with the result $\|e_0\|_A = 1$, $\|e_{10}\|_A = 2 \times 2^{-10}$. Based solely on this data,

(a) What bound can you give on $\kappa(A)$?

(b) What bound can you give on $\|e_{20}\|_A$?

38.4. Suppose A is a dense symmetric positive definite 1000×1000 matrix with $\kappa(A) = 100$. Estimate roughly how many flops are required to solve $Ax = b$ to ten-digit accuracy by (a) Cholesky factorization, (b) Richardson iteration with the optimal parameter α (Exercise 35.3), and (c) CG.

38.5. We have described CG as an iterative minimization of the function $\varphi(x)$ of (38.7). Another way to minimize the same function—far slower, in general—is by the method of *steepest descent*.

(a) Derive the formula $\nabla\varphi(x) = -r$ for the gradient of $\varphi(x)$. Thus the steepest descent iteration corresponds to the choice $p_n = r_n$ instead of $p_n = r_n + \beta_n p_{n-1}$ in Algorithm 38.1.

(b) Determine the formula for the optimal step length α_n of the steepest descent iteration.

(c) Write down the full steepest descent iteration. There are three operations inside the main loop.

38.6. Let A be the 100×100 tridiagonal symmetric matrix with $1, 2, \ldots, 100$ on the diagonal and 1 on the sub- and superdiagonals, and set $b = (1, 1, \ldots, 1)^T$. Write a program that takes 100 steps of the CG and also the steepest descent iteration to approximately solve $Ax = b$. Produce a plot with four curves on it: the computed residual norms $\|r_n\|_2$ for CG, the actual residual norms $\|b - Ax_n\|_2$ for CG, the residual norms $\|r_n\|_2$ for steepest descent, and the estimate $2(\sqrt{\kappa} - 1)^n/(\sqrt{\kappa} + 1)^n$ of Theorem 38.5. Comment on your results.