

Math 6620 Spring 2009 Problem Set 1  
Solutions

1) Consider the Dirichlet problem for the discrete Poisson equation in the unit square  $R = [0, 1] \times [0, 1]$ .

$$-(u_{j,l-1} + u_{j-1,l} - 4u_{j,l} + u_{j+1,l} + u_{j,l+1}) = f_{jl}h^2$$

for  $j = 1, \dots, N$  and  $l = 1, \dots, N$  where  $(N + 1)h = 1$ . Assume that the Dirichlet data is homogeneous, that is,  $u_{jl} = 0$ , for  $j = 0$ ,  $j = N + 1$ ,  $l = 0$ , or  $l = N + 1$ .

Use MATLAB's PCG program (or write your own) to solve this problem using basic CG and preconditioned CG (using MATLAB's CHOLINC or your own incomplete Choleski routine) with different variants of incomplete Choleski factorization as the preconditioner. Use

$$f_{jl} = \sum_{p,q} c_{p,q} \sin(p j \pi h) \cdot \sin(q l \pi h),$$

with

$$c_{p,q} = \frac{1}{p + q}.$$

The corresponding exact solution is

$$u_{jl} = \sum_{p,q} \frac{c_{p,q}}{\lambda^{(p,q)}} \sin(p j \pi h) \cdot \sin(q l \pi h),$$

where  $\lambda^{(p,q)}$  is the eigenvalue of the discrete Laplacian that corresponds to the eigenfunction  $\sin(p j \pi h) \cdot \sin(q l \pi h)$ .

For each iterative method, use  $u_{jl}^{(0)} = 0$  as initial guess. To compute the error, use the *exact* solution  $u_{jl}$  of the *discrete* problem given above. Let  $e_{jl}^{(k)} = u_{jl} - u_{jl}^{(k)}$  denote the iteration error after  $k$  iterations. For each iterative method, how many iterations  $k$  does it take so that  $\|e^{(k)}\|_2 \leq \delta \cdot \|e^{(0)}\|_2$  with  $\delta = 10^{-4}$ ? How does the behavior of CG depend on the preconditioner?

Do all of these experiments with  $h = .05$  and  $h = .025$ .

Compare your results with the ones you obtained earlier using Jacobi, Gauss-Seidel, and SOR. You will have to re-run your Jacobi, Gauss-Seidel, and SOR code for the new  $f$  specified above.

**Solution:**

Computational

2) Trefethen, page 255, problem 33.2

Suppose Algorithm 33.1 is executed for a particular  $A$  and  $\underline{b}$  until at some step  $n$ , an entry  $h_{n+1,n} = 0$  is encountered.

a) Show how 33.13  $AQ_n = Q_{n+1}\tilde{H}_n$  can be simplified in this case. What does this imply about the structure of a full  $m \times m$  Hessenberg reduction  $A = QHQ^*$  of  $A$ ?

b) Show that  $\kappa_n$  is an invariant subspace of  $A$ , i.e.,  $A\kappa_n \subset \kappa_n$ .

c) Show that if the Krylov subspaces of  $A$  generated by  $\underline{b}$  are defined by  $\kappa_n = \langle \underline{b}, A\underline{b}, A^2\underline{b}, \dots, A^{n-1}\underline{b} \rangle$  as in (33.5), then  $\kappa_n = \kappa_{n+1} = \kappa_{n+2} = \dots$

d) Show that each eigenvalue of  $H_n$  is an eigenvalue of  $A$ .

e) Show that if  $A$  is nonsingular, then the solution  $\underline{x}$  to the system of equations  $A\underline{x} = \underline{b}$  lies in  $\kappa_n$ .

**Solution:**

a) If  $h_{n+1,n} = 0$ , then

$$\tilde{H}_n = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,n} \\ h_{2,1} & h_{2,2} & \dots & h_{2,n} \\ & \ddots & \ddots & \vdots \\ & & h_{n,n-1} & h_{n,n} \\ & & & 0 \end{bmatrix}$$

Hence,  $Q_{n+1}\tilde{H}_n = Q_n H_n$ , where  $H_n = \tilde{H}_n(1:n, 1:n)$ , and so  $AQ_n = Q_n H_n$  in this case. The full matrix  $H$  has the form

$$H = \begin{bmatrix} H_n & B_{1,2} \\ 0 & B_{2,2} \end{bmatrix}$$

where  $B_{1,2}$  is  $n \times (m-n)$  and  $B_{2,2}$  is  $(m-n) \times (m-n)$ . That is,  $H$  is block upper-triangular.

b) For  $j = 1:n$ , consider  $A\underline{q}_j$ . Since

$$A = Q \begin{bmatrix} H_n & B_{1,2} \\ 0 & B_{2,2} \end{bmatrix} Q^*,$$

$$A\underline{q}_j = Q \begin{bmatrix} H_n & B_{1,2} \\ 0 & B_{2,2} \end{bmatrix} Q^* \underline{q}_j = Q \begin{bmatrix} H_n & B_{1,2} \\ 0 & B_{2,2} \end{bmatrix} \underline{e}_j = Q \begin{bmatrix} \underline{h} \\ 0 \end{bmatrix}.$$

where  $\underline{h}$  is the  $j^{\text{th}}$  column of  $H_n$  and 0 is a zero vector with  $m-n$  components. So  $A\underline{q}_j = \sum_{k=1}^n h_{k,j} \underline{q}_k \in$

$\kappa_n$ . Hence,  $A\kappa_n \subset \kappa_n$ .

c) Note that  $A^n \underline{b} = A(A^{n-1} \underline{b}) \in \kappa_n$  since  $A^{n-1} \underline{b} \in \kappa_n$  and  $\kappa_n$  is an invariant subspace of  $A$ . So,  $\langle \underline{b}, A\underline{b}, \dots, A^{n-1} \underline{b}, A^n \underline{b} \rangle = \langle \underline{b}, A\underline{b}, \dots, A^{n-1} \underline{b} \rangle$ , that is  $\kappa_{n+1} = \kappa_n$ . Suppose we have shown that  $\kappa_l = \kappa_n$  for all  $l$  which satisfy  $n+1 \leq l \leq L$ . Then to complete an induction proof, we want to show that  $\kappa_{L+1} = \kappa_n$  as well. Now  $\kappa_{L+1} = \langle \underline{b}, A\underline{b}, \dots, A^{L-1} \underline{b}, A^L \underline{b} \rangle$ .  $A^L \underline{b} = A(A^{L-1} \underline{b})$ . Since  $\kappa_L = \kappa_n$ ,  $A^{L-1} \underline{b} \in \kappa_n$ , so  $A(A^{L-1} \underline{b}) \in \kappa_n = \kappa_L$ . Hence,  $\kappa_{L+1} = \kappa_L = \kappa_n$ .

d) Suppose that  $H_n \underline{y} = \lambda \underline{y}$  for  $\underline{y} \in \mathbb{C}^n$ ,  $\underline{y} \neq 0$ . Consider the vector

$$\underline{x} = \begin{bmatrix} \underline{y} \\ 0 \end{bmatrix} \in \mathbb{C}^m.$$

We see that

$$H\underline{x} = \begin{bmatrix} H_n & B_{1,2} \\ 0 & B_{2,2} \end{bmatrix} \begin{bmatrix} \underline{y} \\ 0 \end{bmatrix} = \begin{bmatrix} H_n \underline{y} \\ 0 \end{bmatrix} = \lambda \begin{bmatrix} \underline{y} \\ 0 \end{bmatrix} = \lambda \underline{x}.$$

so  $\lambda$  is an eigenvalue of  $H$ .  $A$  and  $H$  are related by the similarity relation  $A = QHQ^*$ , so  $A$  has the same eigenvalues as  $H$ , and so  $\lambda$  is also an eigenvalue of  $A$ .

e) Consider the least-squares problem,  $\min_{\underline{x} \in \kappa_n} \|A\underline{x} - \underline{b}\|$ . If the solution gives a minimum value of 0, then the solution solves  $A\underline{x} = \underline{b}$  and hence the solution to this linear system is in  $\kappa_n$ . Now

$$\min_{\underline{x} \in \kappa_n} \|A\underline{x} - \underline{b}\| = \min_{\underline{y} \in \mathbb{C}^n} \|AQ_n\underline{y} - \underline{b}\| = \min_{\underline{y} \in \mathbb{C}^n} \|Q_n^*AQ_n\underline{y} - Q_n^*\underline{b}\| = \min_{\underline{y} \in \mathbb{C}^n} \|H_n\underline{y} - Q_n^*\underline{b}\|.$$

$A$  is nonsingular, so it has no zero eigenvalue, so  $H_n$  has no zero eigenvalue, by part (d), and so  $H_n$  is nonsingular. Hence, the equation  $H_n\underline{y} - Q_n^*\underline{b} = 0$  has a solution  $\underline{y}$ , and for  $\underline{x} = Q_n\underline{y}$ , we have that  $A\underline{x} = \underline{b}$ .

### 3) Trefethen, page 274, problem 35.3

The recurrence  $\underline{x}_{n+1} = \underline{x}_n + \alpha(\underline{b} - A\underline{x}_n)$ , where  $\alpha$  is a scalar constant, is known as *Richardson iteration*.

- What polynomial  $p(A)$  at step  $n$  does this correspond to?
- What choice of  $\alpha$  would you recommend for the matrix  $A$  of Figure 35.2, and what would you expect to be the corresponding convergence rate?
- Same question for the matrix of Figure 35.4.

#### Solution:

a) The solution  $\underline{x}$  to  $A\underline{x} = \underline{b}$  satisfies  $\underline{x} = \underline{x} + \alpha(\underline{b} - A\underline{x})$ , and subtracting the equation that defines Richardson iteration from this, we see that

$$\underline{x} - \underline{x}_{n+1} = \underline{x} - \underline{x}_n - \alpha A(\underline{x} - \underline{x}_n).$$

Letting  $\underline{e}_n = \underline{x} - \underline{x}_n$  denote the error in using the  $n^{\text{th}}$  Richardson iterate  $\underline{x}_n$  to approximate  $\underline{x}$ , we have shown that

$$\underline{e}_{n+1} = (I - \alpha A)\underline{e}_n.$$

It follows that  $\underline{e}_n = (I - \alpha A)^n \underline{e}_0$ . Also,  $\underline{r}_{n+1} = \underline{b} - A\underline{x}_{n+1} = \underline{b} - A(\underline{x}_n + \alpha \underline{r}_n) = (\underline{b} - A\underline{x}_n) - \alpha A\underline{r}_n = (I - \alpha A)\underline{r}_n$ , and so  $\underline{r}_n = (I - \alpha A)^n \underline{r}_0$ . So the error and residual after  $n$  steps are given by

$$\underline{e}_n = p_n(A)\underline{e}_0 \quad \underline{r}_n = p_n(A)\underline{r}_0,$$

where  $p_n(A) = (I - \alpha A)^n$ . Richardson iteration is a version of fixed-point iteration with iteration matrix  $T \equiv I - \alpha A$ . It converges if  $\rho(T) = \rho(I - \alpha A) < 1$ , and converges faster the smaller is  $\rho(I - \alpha A)$ . Let  $\Lambda(B)$  denote the set of eigenvalues of a matrix  $B$ . If  $\lambda \in \Lambda(A)$ , then  $\mu = 1 - \alpha\lambda \in \Lambda(T)$ , so our goal in choosing  $\alpha$  is to minimize  $\max_{\lambda \in \Lambda(A)} |1 - \alpha\lambda|$ . This is equivalent to choosing  $\alpha \neq 0$

to minimize  $\max_{\lambda \in \Lambda(A)} \left| \frac{1}{\alpha} - \lambda \right|$ . Let  $\beta = 1/\alpha$ . Our problem can then be written

$$\min_{\beta \neq 0} \max_{\lambda \in \Lambda(A)} |\beta - \lambda|.$$

b) For the matrix whose eigenvalues are shown in Fig. 35.2, all of the eigenvalues are contained in the disk of radius  $\frac{1}{2}$  and center  $z = 2$  in  $\mathbb{C}$ , and the eigenvalues are pretty much scattered throughout this disk. To minimize the maximum distance between  $\beta$  and any of these eigenvalues,

we choose  $\beta = 2$ . Hence,  $\alpha = \frac{1}{2}$ . Then  $\rho(I - \alpha A) = \rho(I - \frac{1}{2}A) \leq \frac{1}{4}$  because the image of the disk centered at 2 of radius  $\frac{1}{2}$  under the map  $1 - \frac{1}{2}z$  is a disk of radius  $\frac{1}{4}$  centered at the origin. The error should decrease approximately like  $(\frac{1}{4})^n$ .

c) For the matrix  $A$  whose eigenvalues are shown in Figure 35.4, we have that the eigenvalues are scattered roughly around an arc-like curve that goes through the points  $(0, 1)$ ,  $(2, 0)$ , and  $(0, -1)$ . Hence the eigenvalues of the matrix  $I - \alpha A$ , are scattered about the image of this curve under the map  $1 - \alpha z$ , which is an arc-like curve that passes through the points  $(1, \alpha)$ ,  $(1 - 2\alpha, 0)$  and  $(1, -\alpha)$ . Our best chance of minimizing the maximum distance of an eigenvalue from the origin is if  $1 - 2\alpha < 0$  and so  $(1, \alpha)$  and  $(1, -\alpha)$  are to the right of the  $y$ -axis. Then we can estimate the best  $\alpha$  by solving the problem

$$\min_{\alpha} \max \left\{ |1 - 2\alpha|, \sqrt{1 + \alpha^2} \right\}.$$

Fig.1 shows the curves  $|1 - 2\alpha|$ ,  $\sqrt{1 + \alpha^2}$ , and  $\max \left\{ |1 - 2\alpha|, \sqrt{1 + \alpha^2} \right\}$  and we see that there is no value of  $\alpha$  for which the minimum is less than 1. This strongly suggests that there is no value of  $\alpha$  for which Richardson iteration will converge for the matrix  $A$  considered here. One can test this for a matrix of the form used to generate Figure 35.4 as in the code `t35p3c.m` at the end of these solutions. This shows that the spectral radius of  $I - \alpha A$  is indeed no less than 1 and so Richardson iteration fails for this matrix.

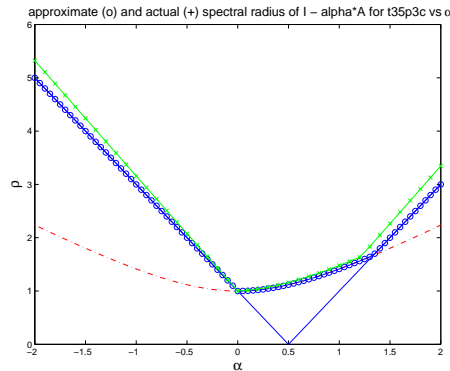


Figure 1: Approximate and computed spectral radius for matrices  $I - \alpha A$  where  $A$  is like the matrix whose eigenvalues are shown in Trefethen Figure 35.4.

4) Trefethen, page 274, problem 35.4

a) Describe a  $O(n^2)$  algorithm based on the QR factorization  $\tilde{H}_n = Q_n R_n$  by Givens rotations for solving the least squares problem of Algorithm 35.1.

b) Show how the operation count can be improved to  $O(n)$  as mentioned on p. 268, if the problem for step  $n - 1$  has already been solved.

**Solution:**

a) In this problem we have a least squares problem  $\min_{\underline{y}} \|\tilde{H}_n \underline{y} - \underline{b}\|_{\ell_1}$ . To solve this, we want to compute the QR factorization of the matrix

$$\tilde{H}_n = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,n} \\ h_{2,1} & h_{2,2} & \dots & h_{2,n} \\ & \ddots & \ddots & \\ & & h_{n,n-1} & h_{n,n} \\ & & & h_{n+1,n} \end{bmatrix}.$$

We have only one below-diagonal nonzero element per column, so we use a sequence of Givens rotations to perform the QR factorization as follows: Let

$$G_1 = \begin{bmatrix} [\tilde{G}_1] & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix},$$

where  $[\tilde{G}_1]$  is a  $2 \times 2$  Givens rotation which applied to the two nonzero entries of column 1 of  $\tilde{H}_n$ , zeros the 2nd of these elements. Hence, applying the matrix  $G_1$  to  $\tilde{H}_n$  zeros the below diagonal entry in column 1. Applying  $G_1$  to  $\tilde{H}_n$  requires applying it to the first two entries in each column of  $\tilde{H}_n$  and this costs  $6n$  floating point operations. Now let

$$G_2 = \begin{bmatrix} 1 & & & \\ & [\tilde{G}_2] & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix},$$

where  $[\tilde{G}_2]$  is a Givens rotation that rotates the 2nd and 3rd elements of column 2 of  $G_1 \tilde{H}_n$  to zero the 3rd element. Applying  $G_2$  to  $G_1 \tilde{H}_n$  zeros the second column below the diagonal.  $[\tilde{G}_2]$  is applied to  $n - 1$  2-vectors, and this costs  $6(n - 1)$  operations. Proceeding similarly with columns 3, 4, ..., n, we obtain an upper triangular  $(n + 1) \times n$  matrix  $R_n$ , and the total cost of the  $n$  Givens rotations is

$$6 \sum_{k=0}^{n-1} (n - k) = 6 \sum_{k'=1}^n k' = 6 \frac{(n + 1)n}{2} \approx 3n^2.$$

Note that while it is useful to think of applying the sequence of  $(n+1) \times (n+1)$  matrices  $G_1, G_2, \dots, G_n$  to  $\tilde{H}_n$  as above, in practice we do not need the matrix  $Q_n$ , we just need  $R_n$  and  $Q_n^* \underline{b}$  in order to solve the LS problem. We get what we need if we think about applying  $G_1, G_2, \dots,$

$G_n$  successively to the expression  $\tilde{H}_n \underline{y} - \|\underline{b}\| \underline{e}_1$ .  $G_n G_{n-1} \dots G_2 G_1 \tilde{H}_n \underline{y} - \|\underline{b}\| G_n G_{n-1} \dots G_2 G_1 \underline{e}_1 = R_n \underline{y} - \|\underline{b}\| G_n G_{n-1} \dots G_2 G_1 \underline{e}_1$ . To apply each Givens rotation to  $\underline{e}_1$  means just rotating two entries, so applying all  $n$  Givens rotations to  $\underline{e}_1$  takes  $O(n)$  operations.

b) The matrices  $\tilde{H}_n$  and  $\tilde{H}_{n-1}$  are related by

$$\tilde{H}_n = \begin{bmatrix} \tilde{H}_{n-1} & \underline{h}_n \\ 0 & h_{n+1,n} \end{bmatrix},$$

where  $\underline{h}_n$  holds the first  $n$  entries of the  $nl^{th}$  column of  $\tilde{H}_n$ . Now suppose that  $G_{n-1}^{(n-1)} \dots G_2^{(n-1)} G_1^{(n-1)} \tilde{H}_{n-1} = R_{n-1}$ . Here the superscript on the  $G$ 's indicates that these are the  $n \times n$  matrices that performed the Givens rotations to reduce  $\tilde{H}_{n-1}$  to  $R_{n-1}$ . Let

$$G_j^{(n)} = \begin{bmatrix} G_j^{(n-1)} & 0 \\ 0 & 1 \end{bmatrix}$$

for  $j = 1, \dots, n-1$ . If we imagine applying  $G_1^{(n)}, G_2^{(n)}, \dots, G_{n-1}^{(n)}$  on the left to  $\tilde{H}_n$ , then what we get in the first  $n-1$  columns of the result is just  $R_{n-1}$  with an extra row of zeros appended at the bottom. For the  $nl^{th}$  column, we rotate 2 entries each time we apply one of the  $G_j^{(n)}$  to it, this takes  $O(n)$  operations, and gives us

$$G_{n-1}^{(n)} \dots G_2^{(n)} G_1^{(n)} \tilde{H}_n = \begin{bmatrix} R_{n-1} & \underline{w} \\ 0 & h_{n+1,n} \end{bmatrix},$$

where  $\underline{w}$  is the result of applying  $G_{n-1}^{(n)} \dots G_2^{(n)} G_1^{(n)}$  to the vector made up of the first  $n$  entries of column  $n$  of  $\tilde{H}_n$ . We apply one more new Givens rotation  $G_n^{(n)}$  to the last two rows of  $G_{n-1}^{(n)} \dots G_2^{(n)} G_1^{(n)} \tilde{H}_n$  to zero  $h_{n+1,n}$  and give us

$$G_n^{(n)} G_{n-1}^{(n)} \dots G_2^{(n)} G_1^{(n)} \tilde{H}_n = \begin{bmatrix} R_{n-1} & \underline{v} \\ 0 & 0 \end{bmatrix} = R_n$$

This takes  $O(1)$  additional operations. We apply the same new Givens rotation to the vector  $\|\underline{b}\| G_{n-1}^{(n)} \dots G_2^{(n)} G_1^{(n)} \underline{e}_1$  after adding a final entry of 0 to make this vector have  $n+1$  components. This also takes  $O(1)$  additional operations, and gives us both the upper triangular matrix  $R_n$  and the vector  $\|\underline{b}\| G_n^{(n)} G_{n-1}^{(n)} \dots G_2^{(n)} G_1^{(n)} \underline{e}_1$  needed to solve the LS problem in the  $nl^{th}$  step of GMRES.

5) Trefethen, page 274, problem 35.5

Our statement of the GMRES algorithm (Algorithm 35.1) begins with the initial guess  $\underline{x}_0 = 0$ ,  $\underline{r}_0 = 0$ . Show that if one wishes to start with an arbitrary initial guess  $\underline{x}_0$ , this can be accomplished by an easy modification of the right-hand side  $\underline{b}$ .

**Solution:**

Suppose we want to use an initial guess  $\underline{x}_0 \neq 0$ . Let  $\underline{x} = \underline{x}_0 + \underline{y}$ . Then  $\underline{b} = A\underline{x} = A(\underline{x}_0 + \underline{y}) = A\underline{x}_0 + A\underline{y}$ . Hence,  $\underline{y}$  solves the problem  $A\underline{y} = (\underline{b} - A\underline{x}_0) \equiv \underline{b}_0$ . Apply GMRES to  $A\underline{y} = \underline{b}_0$  with initial guess  $\underline{y}_0 = 0$  to find  $\underline{y}$ . Then, let  $\underline{x} = \underline{x}_0 + \underline{y}$  to find a solution to the original problem.

Code for exploring spectral radius of Richardson iteration matrix  $I - \alpha A$  for matrix of Problem 35.3c.

```
% Code to generate matrix like that whose eigenvalues are shown
% in Trefethen Figure 35.4
%
% First define matrix A like that whose eigenvalues
% are shown in Figure 35.2
m=200;
A = 2*eye(m) + 0.5*randn(m)/sqrt(m);
evA = eig(A);
xA = real(evA);
yA = imag(evA);
% plot A's eigenvalues
plot(xA,yA,'ro')
hold on;
%
% modify A to B by adding a diagonal matrix as described on page 274
ang = pi/(m-1);
k = [0:1:m-1];
d = (-2 + 2*sin(k*ang)) + i*cos(k*ang);
B = A + diag(d);
% calculate the eigenvalues of this matrix
evB = eig(B);
xB = real(evB);
yB = imag(evB);
plot(xB,yB,'bx')
% plot axes
plot([-0.5 3],[0 0],'g-')
plot([0 0],[-1.5 1.5],'g-')
%
% compute approximate spectral radius for I - alpha B as function of alpha
figure
alpha = [-2:.05:2];
z1 = abs(1-2*alpha);
z2 = sqrt(1 + alpha.*alpha);
plot(alpha,z1,'b-')
```

```

hold on
plot(alpha,z2,'r-.')
zm = max(z1,z2);
plot(alpha,zm,'bo-')
xlabel('\alpha','FontSize',16)
ylabel('\rho','FontSize',16)
%
% for each matrix C = I - alpha B, compute spectral radius
%
alphas = [-2:.1:2];
specr = zeros(length(alphas),1);
for j=1:length(alphas),
    C = eye(m) - alphas(j)*B;
    ev = eig(C);
    specr(j) = max(abs(ev));
end;
%
% plot spectral radius of C as function of alpha
plot(alphas,specr,'g-x')
title(' approximate (o) and actual (+) spectral radius of
      I - alpha*A for t35p3c vs \alpha','FontSize',14)

```

6) Trefethen, page 302, problem 38.4

$A$  is a dense symmetric positive definite 1000-by-1000 matrix and  $\kappa(A) = 100$ . Estimate roughly how many flops are required to solve  $A\mathbf{x} = \mathbf{b}$  to 10-digit accuracy by a) Cholesky, b) Richardson iteration with optimal  $\alpha$ , and c) Conjugate Gradient.

**Solution:**

a) Choleski factorization requires approximately  $1/3m^3$  flops for an  $m \times m$  matrix so the flop count here is  $1/3 \cdot 10^9$ .

b) The convergence rate for Richardson iteration is determined by the spectral radius of the iteration matrix  $T$  for this iterative scheme. From homework last semester, we know that the optimal value of  $\alpha$  is  $\alpha = 2/(\lambda_{\max} + \lambda_{\min})$  where  $\lambda_{\max}$  and  $\lambda_{\min}$  are the largest and smallest of  $A$ 's eigenvalues. So

$$T = I - \frac{2}{(\lambda_{\max} + \lambda_{\min})}A$$

and

$$\begin{aligned} \rho(T) &= \max \left\{ \left| 1 - \frac{2\lambda_{\min}}{(\lambda_{\max} + \lambda_{\min})} \right|, \left| 1 - \frac{2\lambda_{\max}}{(\lambda_{\max} + \lambda_{\min})} \right| \right\} \\ &= \left| \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right|. \end{aligned}$$

So

$$\rho(T) = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{\kappa - 1}{\kappa + 1} = \frac{99}{101}.$$

To reduce an  $O(1)$  initial error by a factor of  $10^{-10}$  requires  $n$  iterations where  $(99/101)^n \leq 10^{-10}$ . This implies that  $n \geq 1152$ . The main work in a Richardson iteration is a matrix-vector product which takes about  $2m^2 = 2(10)^6$  flops. The total work therefore is about  $2.3(10)^9$  flops.

c) After  $n$  iterations of CG, the error satisfies

$$\frac{\|e^n\|}{\|e^0\|} \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n.$$

We want to choose  $n$  so that the factor on the right-hand-side is less than  $10^{-10}$ . So we need

$$2 \left( \frac{9}{11} \right)^n \leq 10^{-10}$$

which implies that  $n \geq 119$ . The main work in a CG iteration is a matrix-vector product, which here costs about  $2(10)^6$  flops, so the total cost of CG is about  $2.4(10)^8$  flops.

7) Trefethen, page 302, problem 38.5.

Consider

$$\phi(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{b}.$$

- Derive the formula  $\nabla \phi(\mathbf{x}) = -\mathbf{r}$ .
- Derive the formula for the optimal step length  $\alpha$  of the steepest descent algorithm.
- Write down the full steepest descent algorithm.

**Solution:**

a) Written in expanded form

$$\phi(\mathbf{x}) = \frac{1}{2} \sum_{k=1}^m \sum_{j=1}^m x_k a_{kj} x_j - \sum_{k=1}^m x_k b_k.$$

Therefore,

$$\begin{aligned} \frac{\partial \phi}{\partial x_i} &= \frac{1}{2} \left( \sum_{k=1}^m \sum_{j=1}^m x_k a_{kj} \delta_{ji} + \sum_{k=1}^m \sum_{j=1}^m \delta_{ki} a_{kj} x_j \right) - b_i = \frac{1}{2} \left( \sum_{k=1}^m x_k a_{ki} + \sum_{j=1}^m a_{ij} x_j \right) - b_i \\ &= \sum_{j=1}^m a_{ij} x_j - b_i \\ &= -r_i. \end{aligned}$$

b) We seek the value of  $\alpha^{(k)}$  that minimizes  $\phi$  along the line  $\mathbf{x} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{r}^{(k)}$ . For clarity, write  $\mathbf{x} = \mathbf{x}^{(k)}$ ,  $\alpha = \alpha^{(k)}$ , and  $\mathbf{r} = \mathbf{r}^{(k)}$ . Let

$$\begin{aligned} f(\alpha) &\equiv \phi(\mathbf{x}^{(k+1)}) = \phi(\mathbf{x} + \alpha \mathbf{r}) \\ &= \frac{1}{2} (\mathbf{x} + \alpha \mathbf{r})^T A (\mathbf{x} + \alpha \mathbf{r}) - (\mathbf{x} + \alpha \mathbf{r})^T \mathbf{b} \\ &= \frac{1}{2} \mathbf{r}^T A \mathbf{r} \alpha^2 + (\mathbf{r}^T A \mathbf{x} - \mathbf{r}^T \mathbf{b}) \alpha + \frac{1}{2} \mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{b}. \end{aligned}$$

Then

$$f'(\alpha) = (\mathbf{r}^T A \mathbf{r}) \alpha - \mathbf{r}^T (\mathbf{b} - A \mathbf{x}) = (\mathbf{r}^T A \mathbf{r}) \alpha - \mathbf{r}^T \mathbf{r}.$$

So  $f'(\alpha) = 0$  when

$$\alpha = \frac{\mathbf{r}^T \mathbf{r}}{\mathbf{r}^T A \mathbf{r}}.$$

c) The entire steepest descent algorithm is

Guess  $\mathbf{x}^{(0)}$

For  $k = 0, 1, 2, \dots$

$$\mathbf{r}^{(k)} = \mathbf{b} - A \mathbf{x}^{(k)}$$

$$\alpha^{(k)} = \frac{(\mathbf{r}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{r}^{(k)})^T A \mathbf{r}^{(k)}}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{r}^{(k)}$$

End.

8) Trefethen, page 302, problem 38.6

**Solution:**

Computational.