

Stanford University, Dept of Management Science and Engineering
MS&E 318 (CME 338) Large-Scale Numerical Optimization

Instructor: Michael Saunders Spring 2011

Homework 2, Due Monday April 18

<http://www.stanford.edu/class/msande318/homework.html>

1. For the symmetric Lanczos process, prove by induction that with exact arithmetic, the matrix $V_k = (v_1 \ v_2 \ \dots \ v_k)$ has orthonormal columns: $V_k^T V_k = I$. (Hence, V_k is an orthonormal basis for the Krylov subspace $\mathcal{K}_k(A, b)$.)
2. **Extra credit:** For the final Lanczos iteration, prove the following:
 T_ℓ is nonsingular iff $b \in \text{range}(A)$.
3. From <http://www.stanford.edu/class/msande318/matlab/>, download files `CGtest3.m` and `Si2.mat`.¹ Script `CGtest3` shows how to load an $n \times n$ matrix A into MATLAB ($n = 769$). The first eigenvalue of A is negative ($\lambda_1 = -0.38441$). The remaining eigenvalues range from $\lambda_2 = 0.24221$ to $\lambda_n = 41.3813$, so $\text{cond}(A) \approx 170$.
The matrix $P = A + \sigma I$ with $\sigma = 0.4$ is positive definite, and $\text{cond}(P) \approx 2679$. Define $x_j = 1/j$ ($j = 1:n$) and $b = Px$.
 - (a) Solve $Px = b$ using some of the iterative solvers provided in MATLAB: `pcg`, `minres`, and `symmlq`. Use the parameters `tol = 1e-12` and `maxit = 600`. For each method, plot the residuals $\|r_k\|$ for each iteration k . Script `CGtest3` already does this in producing figure(1). Add suitable `{xlabel, ylabel, legend}` and give a table of results.
 - (b) The residuals for each solver are significantly different at around iteration 33. Did all solvers terminate at much the same point? Should we expect them to be more different?
 - (c) `type pcg` allows you to see MATLAB's implementation of CG. For each solver, find which stopping rule was used for the results shown in figure(1).
4. The matrix A is indefinite. With the same $x = [1/j]$, define $b = Ax$.
 - (a) Solve $Ax = b$ using `pcg`, `minres`, and `symmlq`. Solve $A^2x = Ab$ using `pcg`, and $Ax = b$ using `lsqr`. Script `CGtest3` already does this in producing figure(2). Again, add suitable `{xlabel, ylabel, legend}` and give a table of results.
 - (b) Note that `pcg` on $Ax = b$ terminates early. What happened?
 - (c) As we see, when A is reasonably well-conditioned, `lsqr` performs much the same as `pcg` on $A^2x = Ab$. But what is the plot for `pcg` really showing? (It's not $\|r_k\| = \|b - Ax_k\|$.)
5. The script plots the eigenvalues $\lambda(A)$ in figure(3). Does there seem to be any clustering of the eigenvalues? A better picture is given in figure(4). Briefly describe what the script is showing in figure(4). Does it explain why the symmetric solvers needed significantly fewer than n iterations to solve $Ax = b$?

¹These questions use PARSEC/Si2, a symmetric indefinite sparse matrix A of size $n = 769$ from a collection of sparse-matrix data and software maintained by Tim Davis (University of Florida): <http://www.cise.ufl.edu/research/sparse/>.

```

% CGtest3.m is a script for comparing {pcg, minres, symmlq}
% on UF sparse matrix number 1360 (PARSEC/Si2, n=769).
% See http://www.cise.ufl.edu/research/sparse/matrices/ (Tim Davis).
% The matrix A is an example from quantum chemistry.
% It is symmetric indefinite with lambda_min = -0.3844.
%
% 30 Mar 2010: MS&E 318 / CME 338 Homework 2.
%-----

load Si2.mat
A      = Problem.A;
[m,n] = size(A);
x      = 1./(1:n)';

%-----

fprintf('\nPOSITIVE DEFINITE SYSTEM')
fprintf('\n          flag iter  relres  error\n')
P      = A + 0.4*speye(n);
b      = P*x;
tol    = 1e-12;
maxit  = 600;

[xL,flagL,relresL,iterL,resvecL] = symmlq(P,b,tol,maxit);
[xC,flagC,relresC,iterC,resvecC] = pcg  (P,b,tol,maxit);
[xM,flagM,relresM,iterM,resvecM] = minres(P,b,tol,maxit);

errL   = norm(xL-x,inf);
errC   = norm(xC-x,inf);
errM   = norm(xM-x,inf);
fprintf(' SYMMLQ  Px = b%4g %5g %8.1e %8.1e  r\n', flagL,iterL,relresL,errL)
fprintf('  CG      Px = b%4g %5g %8.1e %8.1e  b\n', flagC,iterC,relresC,errC)
fprintf(' MINRES   Px = b%4g %5g %8.1e %8.1e  g\n', flagM,iterM,relresM,errM)

figure(1)
hold off; plot(log10(resvecL),'ro')
hold on;  plot(log10(resvecC),'b-')
hold on;  plot(log10(resvecM),'g.')

%-----

fprintf('\nINDEFINITE SYSTEM')
fprintf('\n          flag iter  relres  error\n')
Afun   = @(x) A*x;      b   = A*x;      % Treat A as a function
Afun2  = @(x) A*(A*x); b2  = A*b;      % Treat A*A as a function

[xL,flagL,relresL,iterL,resvecL] = symmlq(Afun ,b, tol,maxit);
[xC,flagC,relresC,iterC,resvecC] = pcg  (Afun ,b ,tol,maxit);
[xM,flagM,relresM,iterM,resvecM] = minres(Afun ,b ,tol,maxit);
[xN,flagN,relresN,iterN,resvecN] = pcg  (Afun2,b2,tol,maxit);
[xS,flagS,relresS,iterS,resvecS] = lsqr (A ,b ,tol,maxit);

errL   = norm(xL-x,inf);
errC   = norm(xC-x,inf);
errM   = norm(xM-x,inf);

```

```

errN = norm(xN-x,inf);
errS = norm(xS-x,inf);
fprintf(' SYMLQ   Ax = b%4g %5g %8.1e %8.1e  r\n', flagL,iterL,relresL,errL)
fprintf('  CG     Ax = b%4g %5g %8.1e %8.1e  b\n', flagC,iterC,relresC,errC)
fprintf(' MINRES  Ax = b%4g %5g %8.1e %8.1e  g\n', flagM,iterM,relresM,errM)
fprintf('  CG     A^2x =Ab%4g %5g %8.1e %8.1e  k\n', flagN,iterN,relresN,errN)
fprintf(' LSQR    Ax = b%4g %5g %8.1e %8.1e  m\n', flagS,iterS,relresS,errS)

figure(2)
hold off; plot(log10(resvecL),'ro')
hold on; plot(log10(resvecC),'b-')
hold on; plot(log10(resvecM),'g.')
hold on; plot(log10(resvecN),'k.')
hold on; plot(log10(resvecS),'m-')

%return

%-----
% Plot the eigenvalues of A.
%-----
lambda = eig(full(A));
figure(3)
hold off; plot(lambda,'b.')
xlabel('Eigenvalue number')
ylabel('\lambda(A)')
title('Eigenvalues of A')

% Try to show if the eigenvalues are clustered.
figure(4);
hold off; plot(lambda,10*ones(n,1),'b.')
hold on;

y1 = -5;
yn = 44.9;
step = 2;

for y = y1:step:yn
    y2 = y1 + step;
    nlambda = length( find(lambda>y1 & lambda<=y2) );
    bar(y1+0.5*step,nlambda)
    y1 = y2;
end

xlabel('\lambda(A)')
ylabel('No. of \lambda(A)')
title('Distribution of eigenvalues of A')

```