1. For the iterative Golub-Kahan orthogonal bidiagonalization with $\beta_1 u_1 = b$, we have
\[
AV_k = U_{k+1}B_k = U_k L_k + \beta_{k+1} v_{k+1} e_k^T,
\]
\[
A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_k^T = V_{k+1} L_{k+1},
\]
and with exact arithmetic either $\beta_{\ell+1} = 0$ or $\alpha_{\ell+1} = 0$ for some $k = \ell$. Prove the following:

(a) $B_k$ has full column rank $k$ for all $k \leq \ell$.

(b) If $\beta_{\ell+1} = 0$, we have $AV_\ell = U_L L_\ell$ with $L_\ell$ nonsingular and $b \in \text{range}(A)$.

(c) If $\beta_{\ell+1} > 0$ but $\alpha_{\ell+1} = 0$, then $b \not\in \text{range}(A)$.

2. From http://stanford.edu/class/msande318/matlab/, download the files bcsstm34.mat and CGtest7.m.\(^1\) Script CGtest7 shows how to load an $n \times n$ matrix $A$ into MATLAB ($n = 588$). The eigenvalues of $A$ range from $\lambda_1 = -2.6830$ to $\lambda_n = 6.8331$, and $\text{cond}(A) \approx 1.1 \times 10^7$.

The matrix $B = A + \sigma_1 I$ with $\sigma_1 = 2.7$ is positive definite, and $\text{cond}(B) \approx 2400$. Define column vector $x = [1/j] (j = 1:n)$ and $b = Bx$.

(a) Solve $Bx = b$ using some of the iterative solvers provided in MATLAB: pcg, minres, and symmlq. Use the parameters $\text{tol} = 1e-8$ and $\text{maxit} = 100$. For each method, plot the residuals $\|r_k\|$ for each iteration $k$. Script CGtest7 already does this in producing figure(1). Add suitable $\{\text{xlabel, ylabel, legend}\}$ and give a table of results.

(b) The residuals for each solver are somewhat different. 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).

---
\[^1\]These questions use Boeing/bcsstm34, a symmetric indefinite sparse matrix $A$ of size $n = 588$ from the SuiteSparse Matrix Collection maintained by Tim Davis: http://faculty.cse.tamu.edu/davis/welcome.html.
3. The matrix \( C = A + \sigma^2 I \) with \( \sigma^2 = 0.5 \) is indefinite, and \( \text{cond}(C) \approx 83,000 \). With the same \( x = [1/j] \), define \( b = Cx \).

   (a) Solve \( Cx = b \) using pcg, symmlq, and minres. Solve \( C^2x = Cb \) using pcg and minres, and \( Cx = b \) using lsqr. Script CGtest7 already does this in producing figure(2). Again, add suitable \{xlabel, ylabel, legend\} and give a table of results.

   (b) Note that pcg on \( Cx = b \) terminates early. What happened?

   (c) When \( C \) is reasonably well-conditioned, lsqr performs much the same as pcg and minres on \( C^2x = Cb \) in terms of iterations. But what are the plots for each solver \{pcg, minres, lsqr\} really showing?

4. The script plots the eigenvalues \( \lambda(C) \) 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 \( Cx = b \)?

To check your intuition, find the eigensystem of \( C \) from

\[
[V, D] = \text{eig(full(C))};
\]

and find \( c \) such \( b = Vc \). (What to we know about \( V \)? How does this help us find \( c \)?) Then do

\[
\text{figure(5); plot} (\text{log10}(\text{sort(abs(c)}, 1, \text{'descend'}))))
\]

and describe what you see.
% CGtest7.m is a script for comparing {cg, symmlq, minres}
% on sparse matrix Boeing/bcsstm34, n=588, nnz=24270).
% See http://faculty.cse.tamu.edu/davis/welcome.html (Tim Davis).
% The matrix A is from a structural problem.
% It is symmetric indefinite with lambda_min = -2.6830.
%
% 09 Apr 2017: Problem Boeing/bcsstm34 used for Homework 3.
%----------------------------------------------------------

load bcsstm34.mat; % lambda(min) = -2.6830, lambda(max) = 6.8331
A   = Problem.A; % Save original matrix A
[n,n] = size(A);
x   = 1./(1:n)';

%----------------------------------------------------------

sigma1 = 2.7;
B = A + sigma1*speye(n); condB = condest(B);
b = B*x;
tol = 1e-8; % Not highly accurate
maxit = 1000;

[xC,flagC,relresC,iterC,resvecC] = pcg(B,b,tol,maxit);
[xL,flagL,relresL,iterL,resvecL] = symmlq(B,b,tol,maxit);
[xM,flagM,relresM,iterM,resvecM] = minres(B,b,tol,maxit);
	errC = norm(xC-x,inf); % The inf-norm is best for large vectors
errL = norm(xL-x,inf);
errM = norm(xM-x,inf);

fprintf('
POS-DEFINITE B = A + sigma1*I,')
fprintf(' sigma1 =%5.2f, condest(B) = %8.1e

', sigma1, condB)
fprintf(' flag iter relres error
CG Bx = b%4g %5g %8.1e %8.1e b
', flagC,iterC,relresC,errC)
fprintf(' SYMMLQ Bx = b%4g %5g %8.1e %8.1e r
', flagL,iterL,relresL,errL)
fprintf(' MINRES Bx = b%4g %5g %8.1e %8.1e g
', flagM,iterM,relresM,errM)

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

%----------------------------------------------------------

sigma2 = 0.5;
C   = A + sigma2*speye(n); condC = condest(C);
b   = C*x; Cfun = @(x) C*x; % Treat C as a function
b2  = C*b; Cfun2 = @(x) C*(C*x); % Treat C*C as a function

[xC,flagC,relresC,iterC,resvecC] = pcg(Cfun ,b ,tol,maxit);
[xL,flagL,relresL,iterL,resvecL] = symmlq(Cfun ,b ,tol,maxit);
[xM,flagM,relresM,iterM,resvecM] = minres(Cfun ,b ,tol,maxit);
[xN,flagN,relresN,iterN,resvecN] = pcg(Cfun2,b2,tol,maxit);
[xR,flagR,relresR,iterR,resvecR] = minres(Cfun2,b2,tol,maxit);
[xS,flagS,relresS,iterS,resvecS,lsvec] = lsqr(C ,b ,tol,maxit);
errC = norm(xC-x,inf);
errL = norm(xL-x,inf);
errM = norm(xM-x,inf);
errN = norm(xN-x,inf);
errR = norm(xR-x,inf);
errS = norm(xS-x,inf);

fprintf('INDEFINITE C = A + sigma2*I,\n');
sigma2 = 5.2f, condest(C) = 8.1e, sigma2, condC

fprintf('flag iter relres error\n');
CG Cx = b\n', flagC, iterC, relresC, errC)
SYMMLQ Cx = b\n', flagL, iterL, relresL, errL)
MINRES Cx = b\n', flagM, iterM, relresM, errM)
CG C^2x =Cb\n', flagN, iterN, relresN, errN)
MINRES C^2x =Cb\n', flagR, iterR, relresR, errR)
LSQR Cx = b\n', flagS, iterS, relresS, errS)

figure(2);
hold off; plot(log10(resvecC), 'b-')
hold on; plot(log10(resvecL), 'r-')
hold on; plot(log10(resvecM), 'g-')
hold on; plot(log10(resvecN), 'k-')
hold on; plot(log10(resvecR), 'c-')
hold on; plot(log10(lsvec), 'm-')

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

% Show if the eigenvalues are clustered.
figure(4);
hold off; plot(lambda, 250*ones(n,1), 'b.')
hold on

y1 = -3; yn = 7;
step = 0.25; nbar = (yn - y1)/step + 1;
y = zeros(nbar,1);
vlam = zeros(nbar,1);

for i = 1:nbar
  y2 = y1 + step;
  nlam(i) = length( find(lambda> y1 & lambda<=y2) );
  y(i) = y1 + 0.5*step;
  y1 = y2;
end

bar( y, nlam )
xlabel('\lambda(C)'); ylabel('No. of \lambda(C)');
title('Distribution of eigenvalues of C');