MAD 4401
Project 5
More Numerical Linear Algebra

Revised March 21, 2010

Here is a method for solving the least squares problem using the Gram Schmidt Method. To find the least squares solution for \( Ax = b \), where \( \text{rank}(A) = n \), form the augmented matrix \([A, b]\) and do the Gram Schmidt QR on the augmented matrix, \( QR = [A, b] \) so that \( Q \) is \( m \times (n + 1) \), \( R \) is \((n + 1) \times (n + 1)\) and we write

\[
R = \begin{pmatrix} R_1 & c \\ 0 & d \end{pmatrix}
\]

where \( R_1 \) is an \( n \times n \) upper triangular matrix, \( c \) is an \( n \times 1 \) column vector, and \( d \) is a scalar. Now minimize the residual:

\[
\|Ax - b\|^2 = \|Q^T Ax - Q^T b\|^2 = \| \begin{pmatrix} R_1 \\ 0 \end{pmatrix} x - \begin{pmatrix} c \\ d \end{pmatrix} \|^2 = \|R_1 x - c\|^2 + \|d\|^2
\]

The solution to \( R_1 x = c \) minimizes the residual. This can be found by back substitution. The residual is

\[
\|Ax_{LS} - b\|^2 = \|R_1 x_{LS} - c\|^2 + \|d\|^2 = \|d\|^2.
\]

1. Write a MATLAB function \( x = \text{mgs1sq}(A, b) \) which finds the least squares solution to \( Ax = b \) using the Modified Gram Schmidt method, and \( \text{backsub} \). You should use the method described above. Compare your \( \text{mgs1sq}(A, b) \) with \( A \backslash b \) (which is also a least squares solver) on these systems: Which solution method has the smaller residual?

   (1) \( A = \text{hilb}(6); A = A(:,1:4); b = A * \text{ones}(4,1); \)

   (2) \( A = \text{ones}(6)+\text{eps}*\text{eye}(6); A = A(:,1:4); b = A * \text{ones}(4,1); \)

The QR Decomposition used by MATLAB is not a Gram Schmidt based method, and it can be used to find the least squares solution to \( Ax = b \), where \( \text{rank}(A) = n \). Let \( QR = [A, b] \) where \( Q \) is \( m \times m \) and \( R \) is \( m \times (n + 1) \) and look at the residual

\[
\|Ax - b\|^2 = \|[A, b][x; -1]\|^2 = \|QR[x; -1]\|^2 = \|R[x; -1]\|^2 = \|R_1 x - c\|^2 + \|d\|^2
\]

where \( R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \) and \( R_1 \) is \( n \times n \), \( c \) is \( n \times 1 \) and \( d \) is \( (m - n) \times 1 \). To minimize the residual solve \( R_1 x = c \) using backsubstitution and again note that the minimum residual is \( \|d\| \).

2. Write a MATLAB function \( x = \text{qr1sq}(A, b) \) which computes the least squares solution using MATLAB’s \( \text{qr} \) by the method described above. Test your \( \text{qr1sq} \) and compare with \( \text{mgs1sq} \) and MATLAB’s \( A \backslash b \) on these systems. Which solution method has the smallest residual?

   (1) \( A = \text{hilb}(6); A = A(:,1:4); b = A * \text{ones}(4,1); \)

   (2) \( A = \text{ones}(6)+\text{eps}*\text{eye}(6); A = A(:,1:4); b = A * \text{ones}(4,1); \)
3. Consider the following data:
   \[ b = (1:101)'; \quad a = (b - 1)/100; \]
   Form the matrix
   \[ A = \text{vander}(a); \quad A = A(:,91:101); \]
   (1) Compute the least squares solution to \[ Ax = b \] using your \[ x = \text{linsq}(A,b). \]
   (2) Compute the least squares solution to \[ Ax = b \] using your \[ x = \text{qr}linsq(A,b). \]
   (3) Check how close your solutions are with \text{norm}. Which solution has the smallest residual?
   (4) \text{linsq} is based on solving \[ A^T A x = A^T b. \] What is the condition number for this system?
   (5) \text{qr}linsq is based on solving \[ R_1 x = Q^T b \] where \[ R_1 \] is \[ R(1 : 11, 1 : 11). \] What is the condition number for this system?
   (6) Which solution is more accurate?

4. Write a MATLAB function \[ [v, r] = \text{powmeth}(A, x, n) \] which locates an eigenvector \( v \) and an eigenvalue \( r \) using the power method. \( n \) is the number of iterations and \( x \) is the seed vector. \( r \) should be found using the Rayleigh Quotient. Test your \text{powmeth} on the following matrices. You can use MATLAB’s \text{eig} to see if you are finding the largest eigenvalue.
   (1) \text{list}(5) This matrix is defined by \[ A = \text{rand}(5); \quad A(:, :) = 1:25 \]
   (2) \text{pascal}(5)
   (3) \text{hilb}(5)
   (4) \text{house}(\text{rand}(5,1))
   (5) \text{magic}(5)

5. Write a MATLAB function \[ [v, r] = \text{invshift}(A, x, s, n) \] which locates an eigenvector \( v \) and an eigenvalue \( r \) of \( A \) using the inverse shift method. \( n \) is the number of iterations and \( x \) is the seed vector. \( r \) should be found using the Rayleigh Quotient. Find as many eigenvalues of \text{hilb}(4) as you can using the functions \text{powmeth} and \text{invshift}.

6. Write a MATLAB function \( r = \text{adaptinvshift}(A, n) \) which adaptively adjusts the shift in the inverse shift method by using the Rayleigh quotient. \( n \) is the number of iterations. Use a random seed vector to start the iteration and take the initial shift to be a random scalar.
   (1) Try your \text{adaptinvshift} on \text{pascal}(5) and \text{hilb}(5).
   (2) If \( r \) and \( s \) are successive values of the shift during the iteration, rewrite your \text{adaptinvshift} to compute and display \text{norm}(r-s) so that you can see the convergence of the eigenvalue. What does the sequence of values of \text{abs}(r-s) look like with 8 iterations on \( A=\text{magic}(7)? \)
   (3) If you should get the error message that the matrix is ill-conditioned or singular to working precision. To what matrix does this refer? Explain why this is good news.

7. Write a MATLAB function \[ [B, r] = \text{QueArr}(A, n) \] which finds the upper triangular matrix (if it converges) \( B \) and the eigenvalues \( r \), as a vector, by the QR iteration applied to \( A \). \( n \) is the number of iterations. Run your \text{QueArr} on \( A=\text{hilb}(4) \) with \( n=5 \). What happens when you run your \text{QueArr} on these matrices:
   (1) \( A = \text{pascal}(5) \)
   (2) \( A = \text{magic}(5) \)
   (3) \( A = \text{rand}(5) \)
   (4) \( A = [0 \ 1; 1 \ 0] \)
   \( A = \text{rand}(5) \) will usually have some complex eigenvalues. If yours does not, try \( A = \text{rand}(5) \) again until you get one with complex eigenvalues. Now run 10 iterations of \text{QueArr} on it. You should see one or more 2 \times 2 blocks along the diagonal of \( B \). Carve out the 2 \times 2 block(s) and use \text{eig} to find the eigenvalues of the 2 \times 2 block(s). How do they fit with the eigenvalues of \( A \)?

**Image Compression with the Singular Value Decomposition**

Install the image \text{lena.tif} from the website into the directory C:\MATLAB7\work or another directory in your MATLAB path. You can set the path from the file pull down menu in MATLAB. An image in JPG or TIF format can be loaded into MATLAB with the command, imread.
lena = imread('lena', 'tif');
If you did not place the semi-colon there this time, you will the next time.

This image is stored in uint8 format. MATLAB will only compute in double precision floating point, so we need to convert the format with the command
lena = double(lena);

Now we can display the image. First we must set the color, since our image is in gray scale, we will set the color interpreter to gray.
colormap(gray)
To display the image, type
imagesc(lena)

8. The image lena is 512 × 512, but we can easily do a singular value decomposition on it.
[U,S,V]=svd(lena);
Write a MATLAB function Ap=rankapprox(U,S,V,k) which produces the rank k approximation to a matrix with the singular value decomposition given by U, S, and V.

(1) Find the rank 100 approximation of lena and graph it. Print this graph.
(2) Find the absolute error and relative error in this approximation
max(max(abs(lena-Ap)));
max(max(abs(lena-Ap)/abs(lena)));

9. You can get a better idea about why the approximation looks so good when the error measurements are so poor by looking at the histogram. Prepare the data by turning it into a a single, very long vector as follows:
L=abs((lena(:)-Ap(:))/lena(:));
Examine these graphs and read about hist in the MATLAB help pages.
hist(L)
hist(L,100)
Print out the 100 bucket histogram. What percentage of pixels have a relative error greater than .15?

10. A rank k compression data structure is the collection of the 2k vectors, u_1, ..., u_k, v_1, ..., v_k and k scalars σ_1, ..., σ_k which are used in the rank k approximation. Let us call the address space, the number floating point scalars needed to store an image. The address space of lena is 512^2.

(1) What is the address space for the rank k compression data structure for lena?
The compression ratio is the ratio
(address space of the rank k compression data structure)/512^2

(2) What is the compression ratio for the rank k compression data structure for lena? In particular, what is the compression ratio for the rank 100 compression data structure for lena?