Revised 11/25/01

LAB 5: Hermitian and Normal Matrices, Positive-Definite Matrices, and the Singular Value Decomposition

In this lab you will use MATLAB to study four topics:

- The eigenvectors/eigenvalues of hermitian and normal matrices
- Tests for positive-definiteness of a real symmetric matrix
- The Cholesky Decomposition of a Positive-definite matrix
- The singular value decomposition of a matrix, which expresses the action of a matrix as a rotation, followed by a stretching in each coordinate direction, followed by another rotation.

Tcodes: For this lab you will need the Teaching Codes

```
house.m, plot2d.m, slu.m
```

Before opening MATLAB to work on the Lab questions you should copy these codes to your disk space. See Lab #2 for further details on how to obtain any of these files from the course home page.

Question 1. Diagonalization of Normal Matrices

For this question first generate random 4×4 complex matrices A and B by

```
A =rand(4) + i*rand(4);   B = (A + A')/2
```

(here $i = \sqrt{-1}$). Note that if A is a complex matrix, then the MATLAB notation `A'` means the *conjugate* transpose A^H. Hence B is hermitian symmetric.

(a) Compute the matrix S of eigenvectors of A and the diagonal eigenvalue matrix L of A by

```
[S, L] = eig(A)
```

Since A is a random matrix, it should have 4 distinct eigenvalues. Verify by MATLAB that $S^{-1}AS = L$. Are the eigenvectors of A mutually orthogonal? Is $S' * A * S = L$? Is $A * A' = A' * A$? Explain the relations among these questions.

(b) Compute the matrix U of eigenvectors of B and the diagonal eigenvalue matrix of B by

```
[U, D] = eig(B)
```

What can you predict about the eigenvalues of B without calculation? Are the eigenvectors of B mutually orthogonal? Confirm your predictions with MATLAB calculations and verify that `U'*B*U = D`.

(c) Generate a random diagonal matrix

```
F= diag(diag(rand(4)+i*rand(4))).
```

Change F into a non-diagonal matrix by setting `C=U*F*U'` (where U is the matrix from part **(b)**). Prove (by matrix algebra without numerical calculation) that $BC = CB$ and $C^H C = CC^H$. Then use MATLAB to check these equations.

(d) Calculate the eigenvalues and eigenvectors of C using

```
[W, E] = eig(C)
```

What is the relation between E and F? Are the eigenvectors of C mutually orthogonal? Are the columns of W eigenvectors for B? Answer these last two questions by general theory, and then confirm by MATLAB calculations.

Question 2. Positive-Definite Matrices

(a) Tests for Positivity: (see Strang, page 331, 6B) Generate a random 3×3 real symmetric matrix A by

```
B = fix(5*rand(3)); A = B + B'
```

Determinant Test: Calculate the three *upper left determinants* of A by

```
D1 = A(1,1), D2 = det(A(1:2,1:2)), D3 = det(A)
```

One test for A to be *positive definite* is

$$D_1 > 0, \quad D_2 > 0, \quad D_3 > 0$$

Does the matrix A pass this test?

Eigenvalue Test: Another test for A to be positive definite is that all the *eigenvalues* of A are positive. Calculate these eigenvalues by the MATLAB command `eig(A)`. Are the results of this test consistent with the Determinant Test?

Pivot Test: A third test for A to be positive definite is that all the *pivots* in the LU decomposition of A are positive. Check this by calculating

```
[L, U] = slu(A)
```

The pivots are the diagonal entries of U (here `slu.m` is the Teaching Code that you used in Lab 2). Are the pivots all positive? Is this consistent with the previous two tests?

If the matrix A you generated failed any of these tests, generate another symmetric matrix until you get one that passes all the tests.

(b) Positivity of Quadratic Form $x^{\mathrm{T}}Ax$: Let A be a matrix from **(a)** that passed all the tests for positivity. Generate several random vectors `x = rvect(3)` and check that `x'*R*x` is positive for each one.

(c) Law of Inertia: Generate a random 3×3 real symmetric matrix A by the method of part **(a)** that is *not* positive definite (you can use one of the matrices that failed the tests in **(a)**). Verify by MATLAB that A has the same number of negative pivots as negative eigenvalues (See Strang, page 342, 6G). (In counting the number of negative pivots, remember that the pivots are D_1, D_2/D_1, D_3/D_2.)

Question 3. Cholesky Factorization of Positive-Definite Matrices

(a) If B has independent columns, then the matrix $A = B^{\mathrm{T}}B$ is always positive definite. Check this by generating a random 4×3 matrix `B = rmat(4,3)`. Calculate `rank(B)` to determine if B

has independent columns. If this is the case, set `A = B'*B` (otherwise, generate another matrix B and repeat). Next, calculate `eig(A)` to see that A is positive definite. Generate several random vectors `x = rvect(3)` and check that `x'*A*x` is positive for each one.

(b) Use MATLAB to calculate an orthonormal basis of eigenvectors for A by the command

```
[Q, D] = eig(A).
```

Use MATLAB to verify that Q is an orthogonal matrix and that `Q'*A*Q = D`.

(c) The positive-definite matrix A has a *Cholesky factorization* $A = R^T R$, where R is upper triangular with positive diagonal entries. One way to obtain this factorization is to start with the LU factorization. Compute

```
[L, U] = slu(A),   H = diag((diag(U))) .
```

Show by hand calculation why $A = LHL^T$, and then verify this by MATLAB. The matrices A and H are congruent. Do they have the same eigenvalues? Explain. Now compute `R = sqrt(H)*L'`. Show by hand calculation why $A = R^T * R$. Then verify this by MATLAB. Finally, calculate the Cholesky decomposition of A by the MATLAB m-file `chol(A)` and verify that you get the same matrix R. (The built-in MATLAB function takes advantage of the positive-definiteness of A and is considerably faster than the LU factorization method.)

Question 4. Singular Value Decomposition

(a) Graphic Demo of SVD: Generate a random 2×2 matrix `A = rand(2,2)`. Then type `eigshow(A)` at the MATLAB prompt. A graphics window should open. Click on the `svd` button on the right side of the window. Your matrix A should appear (in MATLAB notation) in the menu bar above the graph. Underneath the graph the statement

```
Make A*x perpendicular to A*y
```

should appear (if it does not, then click on the `svd` button again). The graph shows a pair of orthogonal unit vectors $\mathbf{x}$ and $\mathbf{y}$, together with the transformed vectors $A\mathbf{x}$ and $A\mathbf{y}$.

Move the pointer onto the vector $\mathbf{x}$, and then make the pair of vectors $\mathbf{x}$, $\mathbf{y}$ go around a circle. The transformed vectors $A\mathbf{x}$ and $A\mathbf{y}$ then move around an ellipse, but generally $A\mathbf{x}$ is not perpendicular to $A\mathbf{y}$. Search for the position of $\mathbf{x}$ and $\mathbf{y}$ so that $A\mathbf{x}$ is perpendicular to $A\mathbf{y}$. When this happens, then the *singular values* σ_1 and σ_2 of A are the lengths of the vectors $A\mathbf{x}$ and $A\mathbf{y}$. Give a rough estimate of these lengths from the graph (remember that $\mathbf{x}$ and $\mathbf{y}$ have length one).

(b) Calculation of SVD: Let A be the random matrix you generated in part (a). Use MATLAB to calculate the singular value decomposition of A by

```
[U, S, V] = svd(A)
```

Verify that $A = U * S * V'$ (up to numerical roundoff error).

The diagonal entries of S are the *singular values* σ_1, σ_2 of A. Compare the calculated singular values with your graphical estimates for σ_1, σ_2 from part (a).

The singular values are the square roots of the eigenvalues of $A^T A$. Check this by calculating

```
sqrt(eig(A'*A))
```

(c) Geometric Meaning of SVD: At the MATLAB prompt type

```
H = house; plot2d(H)
```

A graphics window should open and display a crude drawing of a house (the matrix `H` contains the coordinates of the line segments of the figure).

Generate an orthogonal matrix V by

```
t = pi/6; V = [cos(t), -sin(t); sin(t), cos(t)]
```

Let V^T act on the house by `plot2d(V'*H)` (be sure to use the transpose matrix `V'`). How has the house been changed?

Next generate a diagonal matrix S by

```
S = [5/4, 0; 0, 3/4 ]
```

Let S act on the rotated house by `plot2d(S*V'*H)` . How does this change the previous picture?

Now generate another orthogonal matrix U by

```
t = pi/4; U = [cos(t), -sin(t); sin(t), cos(t)]
```

Let U act on the rotated and stretched house by `plot2d(U*S*V'*H)` . How does this change the previous picture? Print this figure and include it in your lab write up.

Set `A = U*S*V'`. Use MATLAB to calculate the singular value decomposition of A by

```
[U1, S1, V1] = svd(A)
```

Verify that `U = U1, S = S1, V = V1` (up to numerical roundoff error).