Revised 10/04/03

LAB 3: Least Squares Fit, QR Factorization, and Discrete Fourier Transform

In this lab you will use MATLAB to study the following topics:

- Geometric aspects of vectors: *norm*, *dot product*, and *orthogonal projection* onto a subspace.
- How to transform a linearly independent set of vectors into an orthonormal set (the *Gram-Schmidt Algorithm*). When the given vectors are columns of the matrix A, this gives the $A = QR$ matrix factorization.
- The best approximate solution to an overdetermined system $A\mathbf{x} = \mathbf{b}$ when A has independent columns.
- The method of *least squares* to fit a line to data that have a linear trend and random noise.
- The Discrete Fourier Transform

Tcodes: For this lab you will need the Teaching Codes

```
grams.m,  linefit.m,  lsq.m, partic.m
```

Before beginning work on the Lab questions you should copy these codes from the TCodes directory on the Math Department/Course Materials/Linear Algebra 550A web page to your directory.

Random Seed: When you start your MATLAB session, initialize the random number generator by typing

```
rand('seed', abcd)
```

where *abcd* are the last four digits of your Student ID number. This will ensure that you generate your own particular random vectors and matrices.

BE SURE TO INCLUDE THIS LINE IN YOUR LAB WRITE-UP

Question 1. Norm, Dot Product, and Orthogonal Projection onto a Line

Generate random vectors $\mathbf{u}, \mathbf{v} \in \mathcal{R}^3$ by `u = rvect(3), v = rvect(3)`. Use these vectors in the following.

(a) The *norm* $||\,\mathbf{u}\,||$ of a vector is calculated by the MATLAB command `norm(u)`. The *triangle inequality* asserts that $||\,\mathbf{u}+\mathbf{v}\,|| \le ||\,\mathbf{u}\,|| + ||\,\mathbf{v}\,||$. Verify this by MATLAB for your vectors.

(b) The *dot product* $\mathbf{u} \cdot \mathbf{v}$ is calculated in MATLAB by `u'*v` when $\mathbf{u}$ and $\mathbf{v}$ are column vectors of the same size. The absolute value $|t|$ of a number t is calculated in MATLAB by `abs(t)`. The *Cauchy-Schwarz inequality* asserts that $|\mathbf{u} \cdot \mathbf{v}| \le ||\,\mathbf{u}\,||\,||\,\mathbf{v}\,||$. Verify this by MATLAB for your vectors.

(c) The *orthogonal projection* of the vector $\mathbf{v}$ onto the line $\mathcal{L}$ (one-dimensional subspace) spanned by the vector $\mathbf{u}$ is

$$\mathbf{w} = \frac{\mathbf{u} \cdot \mathbf{v}}{||\,\mathbf{u}\,||^2}\,\mathbf{u}$$

(see Figure 3.5 on page 144 of Strang's text). Use MATLAB to calculate $\mathbf{w}$ for your vectors. Two vectors are *orthogonal* if their dot product is zero. Verify by MATLAB that the vector $\mathbf{z} = \mathbf{v} - \mathbf{w}$ is orthogonal to $\mathbf{u}$. (If the dot product is not exactly zero but is a very small number of size 10^{-13} for example, then the vectors are considered orthogonal for numerical purposes.)

(d) The formula for $\mathbf{w}$ in **(c)** can also be written as a matrix-vector product. Use MATLAB to obtain the matrix

```
P = u*inv(u'*u)*u'
```

(note carefully the punctuation and the order of the factors in this formula). Explain why P is a 3×3 matrix, although $\mathbf{u}$ is a vector. Calculate by MATLAB that $P\mathbf{v}$ is the vector $\mathbf{w}$ for your $\mathbf{u}$ and $\mathbf{v}$. Then write out (by hand) an algebraic justification for the equality $\mathbf{w} = P\mathbf{v}$ in general, using the properties of matrix multiplication.

Question 2. Gram-Schmidt Orthogonalization

Generate three random vectors in $\mathcal{R}^3$ by

```
u1 = rvect(3), u2 = rvect(3), u3 = rvect(3)
```

Use these vectors in the following.

(a) Since the set of vector $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ is random, it should be linearly independent. Use MATLAB to check this by calculating the rank of a suitable matrix. For the same reason, these vectors are almost certainly *not* mutually orthogonal. Check this with MATLAB.

(b) Now use these vectors to obtain an orthogonal basis for $\mathcal{R}^3$, following the Gram-Schmidt algorithm:

```
v1 = u1, v2 = u2 - ((v1'*u2)/(v1'*v1))*v1
```

Note that the denominator in this formula has been written as a dot product instead of $||\mathbf{v}_1||^2$, since this avoids calculating a square root. The vector subtracted from $\mathbf{u}_2$ to obtain $\mathbf{v}_2$ is the projection of $\mathbf{u}_2$ onto the line spanned by $\mathbf{v}_1$, as in Question 1. Check that the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are mutually orthogonal (within a negligible numerical error). Now set

```
v3 = u3 - ((v1'*u3)/(v1'*v1))*v1 - ((v2'*u3)/(v2'*v2))*v2
```

(use the up-arrow key to edit the previous command). The vectors subtracted from $\mathbf{u}_3$ to obtain $\mathbf{v}_3$ are the projections of $\mathbf{u}_3$ onto the lines spanned by $\mathbf{v}_1$ and by $\mathbf{v}_2$, as in Question 1. Check that the vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ mutually orthogonal (within a negligible numerical error).

(c) Finally rescale the vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ to obtain an *orthonormal basis* for $\mathcal{R}^3$:

```
w1 = v1/norm(v1), w2 = v2/norm(v2), w3 = v3/norm(v3)
```

Check (by MATLAB) that $||\mathbf{w}_i|| = 1$ for $i = 1, 2, 3$. What property of the dot product guarantees that the vectors $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ are mutually orthogonal?

(d) $A = QR$ **Factorization:** Set

```
A = [u1, u2, u3], Q = [w1, w2, w3], R = Q'*A
```

Which entries of R do you know are zero for every choice of vectors? Verify by MATLAB that $A = Q * R$. Calculate

```
R(1,1)*w1
R(1,2)*w1 + R(2,2)*w2
R(1,3)*w1 + R(2,3)*w2 + R(3,3)*w3
```

How are these three vectors related to the original vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$?

Question 3. Orthogonal Projections

Generate three random vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \in \mathcal{R}^5$ and the matrix A with these vectors as columns:

```
u1 = rvect(5); u2 = rvect(5); u3 = rvect(5); A = [u1, u2, u3]
```

Use these vectors and this matrix in the following.

(a) Let $W = \text{Col}(A)$ be the subspace of $\mathcal{R}^5$ spanned by $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$. What is $\dim(W)$? Justify your answer by an appropriate rank calculation using MATLAB.

The teaching code `grams.m` carries out the steps of the Gram-Schmidt algorithm, just as you did step-by-step in Question 2. Calculate

```
Q = grams(A); w1 = Q(:,1), w2 = Q(:,2), w3 = Q(:,3)
```

by MATLAB. Calculate `Q'*Q`. Why does the answer tell you that $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ is an orthonormal set?

(b) The orthogonal projection P from $\mathbf{R}^5$ onto the subspace W is given by the 5×5 matrix

P = w1*w1' + w2*w2' + w3*w3'

(note that `w1` is a 5×1 column vector; hence `w1'` is a 1×5 row vector and `w1*w1'` is a 5×5 matrix). Verify by MATLAB that $P - P^{\mathrm{T}} = 0$ and $P^2 - P = 0$ (these are the properties that characterize an orthogonal projection matrix). If $\mathbf{v} \in \mathcal{R}^5$ then

$$P\mathbf{v} = (\mathbf{w}_1 \cdot \mathbf{v})\mathbf{w}_1 + (\mathbf{w}_2 \cdot \mathbf{v})\mathbf{w}_2 + (\mathbf{w}_3 \cdot \mathbf{v})\mathbf{w}_3$$

(c) Orthogonal Decomposition $\mathbf{v} = \mathbf{w} + \mathbf{z}$: Generate another random vector `v = rvect(5)`. Set `w = P*v, z = v - w`. Verify by MATLAB that `w'*z = 0`. This shows that $\mathbf{w}$ is in the subspace W and that $\mathbf{z}$ is the component of $\mathbf{v}$ perpendicular W (see Figure 3.8 on page 155 of Strang's text).

(d) The projection matrix P onto the subspace W can be calculated directly from the matrix A, without first orthogonalizing the columns of A, as in Question 2. Define

```
PW = A*inv(A'*A)*A'
```

(see formula (3) on page 156 of Strang's text). Check by MATLAB that PW = P (up to negligible numerical error).

Question 4. Approximate Solution to Inconsistent Linear System

For this question use the 5×3 matrix A and the vector $\mathbf{v} \in \mathcal{R}^5$ from Question 3.

(a) Approximate Solution to $A\mathbf{x} = \mathbf{v}$: Let $\mathbf{v} = \mathbf{w} + \mathbf{z}$ be the orthogonal decomposition from Question 3(c). Show that

(i) The equation $A\mathbf{x} = \mathbf{v}$ is *inconsistent* (has no solutions).

(ii) The equation $A\mathbf{x} = \mathbf{w}$ is *consistent*.

(Calculate the rank of the augmented matrix in each case). Now solve the equation (ii) by

```
xls = inv(A'*A)*A'*v
```

(see equation (2) on page 156 of Strang's text). Check by MATLAB that `A*xls = w` (up to negligible numerical error). Denote the vector `xls` by $\bar{\mathbf{x}}$. It is called the *least squares* approximate solution to the (unsolvable) equation $A\mathbf{x} = \mathbf{v}$.

(b) Closest Vector Property: Generate a random vector `y = rvect(3)` and verify by MATLAB that `P*A*y = A*y`. This shows that $A\mathbf{y} \in W$. Since $A\bar{\mathbf{x}} = \mathbf{w}$, the vector $A\bar{\mathbf{x}}$ is the vector in W that is closest to $\mathbf{v}$. Thus the function $||A\mathbf{y} - \mathbf{v}||$ is *minimized* by choosing $\mathbf{y} = \bar{\mathbf{x}}$. This is the *closest vector* property (see page 155 of Strang's text), which is also called the *least squares* property.

To illustrate the closest vector property, compare `norm(A*y - v)` with `norm(A*xls - v)`. Is the inequality

$$||A\bar{\mathbf{x}} - \mathbf{v}|| < ||A\mathbf{y} - \mathbf{v}||$$

satisfied?

Question 5. Fitting a Line to Data Points

(a) Generating and Plotting Linear Data: Define a column vector of ten equally-spaced t values

```
t = [1:10]'
```

Now generate a column vector of the corresponding values of the linear function $y = 4 + t$ and plot it by

```
y = 4 + t; linefit(t,y)
```

(Note the semicolon that suppresses the display of the y data). MATLAB should open a new window in which this line is plotted. Notice that the line passes exactly through each data point, and the 'best fit' equation is the given function $4 + t$. Print this graph by clicking on `file` in the figure tool bar, and then click on `print`. Include the printed copy in your lab write-up.

(b) Linear Data with Random Noise: Now add some random noise to each y data value in part **(a)**:

```
y = 2 + 4*rand(10,1) + t; linefit(t,y)
```

These random data points don't lie on the line from part **(a)**, even though they show the same general trend. The line that is plotted is the *best fitting* line; it minimizes the *mean square error* between the y coordinates on the line and the data values. Notice how some data points are above the line, and others are below the line. The equation of the best line fitting the random data points is displayed above the graph. Print this graph (following the same procedure as in part **(a)**). Include the printed copy in your lab write-up.

Question 6. Discrete Fourier Transform

In this problem you will use MATLAB to study the discrete Fourier transform (DFT).

(a) Discrete Fourier Matrix: The $n \times n$ discrete Fourier transform matrix $F(n)$ has i, j entry $w^{(i-1)(j-1)}$, where the complex number

$$w = \exp(2\pi\sqrt{-1}/n) = \cos(2\pi/n) + \sqrt{-1}\sin(2\pi/n)$$

is a *primitive* nth root of unity. You can generate $F(n)$ by the following MATLAB function m-file:

```
% Fourier  Matrix of size n
function   y = dft(n)
           F = ones(n);
           w = exp(2*pi*sqrt(-1)/n);
           for i=1:n
             for j=1:n
               F(i,j)=w^((i-1)*(j-1));
             end
           end
           y = F;
```

(Note the use of the semicolon at the ends of lines to suppress displaying the results on the screen.) Create this m-file and save it as `dft.m`. Calculate `F2 = dft(2)` and `F4 = dft(4)` and check that they are the correct matrices (see Strang, page 187).

(b) Unitarity of Fourier Matrix: Generate `F = dft(9);` and a random vector complex vector `y = rand(9,1) + sqrt(-1)*rand(9,1);`. Calculate `c = F*y;` (use ; so these large matrices and vectors don't print to screen or appear in your diary) . Calculate `norm(c)/norm(y)`. Will you get the same ratio for *every* vector $\mathbf{y}$? Explain. For what value of the constant μ is the matrix $U = \mu F$ unitary? Verify that

```
norm(U'*U - eye(9)) = 0
```

for your choice of μ (up to a numerical error ϵ). Note that command `F'` in MATLAB gives the *conjugate transpose* matrix when F has complex entries (this is denoted by F^{H} in Strang's book).