LAB 6: Orthonormal Bases, Orthogonal Projections, and Least Squares

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

- Geometric aspects of vectors — norm, dot product, and orthogonal projection onto a line.
- The Gram-Schmidt Algorithm to change an independent set of vectors into an orthonormal set, and the associated \( A = QR \) matrix factorization.
- The orthogonal projection of a vector onto a subspace.
- The best approximate solution to an inconsistent linear system \( Ax = b \).
- The method of least squares for fitting straight lines and parabolas to data points.

Preliminaries

Reading from Textbook: In connection with this Lab, read through Sections 6.1 to 6.6 of the text and work the suggested problems for each section.

Tcodes and Script Files: 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 Teaching Codes directory on the Math Department/Course Materials/Linear Algebra 250 web page to your memory space (see Lab 3 for more details).

You will also need the m-files `rmat.m` and `rvec.m` from Lab 2. These files should already be in your directory. If you didn’t do Lab 2, get a copy of that lab assignment and create these files as indicated there.

Lab Write-up: You should open a diary file at the beginning of each MATLAB session (see Lab 1 for details). Be sure to answer all the questions in the lab assignment.

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

The lab report that you hand in must be your own work. The following problems all use randomly generated matrices and vectors, so the matrices and vectors in your lab report will not be the same as those of other students doing the lab. Sharing of lab report files is not allowed in this course.

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

Generate random vectors \( \mathbf{u}, \mathbf{v} \in \mathbb{R}^2 \) by \( \mathbf{u} = \text{rvec}(2) \), \( \mathbf{v} = \text{rvec}(2) \). Calculate \( \text{rank}(\{\mathbf{u}, \mathbf{v}\}) \) to determine whether they are linearly independent (this is also evident by inspection, since the vectors have integer entries). If the answer is not 2, then generate a new random pair of vectors and calculate the rank. Repeat until rank is 2. Now use these vectors in the following calculations.
(1) (a) State (in words and symbols) the **triangle inequality** relating the norms $\|u\|$, $\|v\|$, and $\|u + v\|$ for a general pair of vectors $u$, $v$. Then use MATLAB to show that your particular vectors $u$, $v$ satisfy this inequality. Note that $\|u\|$ is calculated by the MATLAB command `norm(u)`.

(1) (b) State (in words and symbols) the **Cauchy-Schwarz inequality** relating the dot product $u \cdot v$ and the norms $\|u\|$, $\|v\|$ for a general pair of vectors $u$, $v$. Then use MATLAB to show that your particular vectors $u$, $v$ satisfy this inequality. Note that dot product is calculated in MATLAB by $u'*v$ when $u$ and $v$ are column vectors of the same size. The absolute value $|t|$ of a number $t$ is calculated in MATLAB by `abs(t)`.

(1) (c) The **orthogonal projection** of the vector $u$ onto the line $L$ (one-dimensional subspace) spanned by the vector $v$ is

$$w = \frac{u \cdot v}{v \cdot v} v$$

(see Figure 6.3 on page 366 of the text). Use MATLAB to calculate $w$ for your vectors. Two vectors are orthogonal if their dot product is zero. Verify by MATLAB that the vector $z = u - w$ is orthogonal to $v$. (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.)

(1) Explain why $P$ is a $2 \times 2$ matrix. Calculate by MATLAB that $Pu$ is the vector $w$ for your $u$ and $v$.

**Question 2. Gram-Schmidt Orthogonalization and QR Factorization**

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

$$u_1 = rvec(3), u_2 = rvec(3), u_3 = rvec(3)$$

Check whether they are linearly independent by calculating `rank([u1, u2, u3])`. If the answer is not 3, then generate a new random set of vectors and calculate the rank. Repeat until the rank is 3. Now use these vectors in the following calculations.

(a) Since the vectors $u_1, u_2, u_3$ are chosen at random, it is very unlikely that they are mutually orthogonal. To see this graphically using MATLAB, generate a line plotting parameter $r$ and open a graphics window by the commands

$$r = 0:0.05:1; \text{ hold on}$$

(be careful with the punctuation). Plot the three vectors in the graphics window as red, green, and blue dotted lines by the commands

```matlab
plot3(r*u1(1),r*u1(2),r*u1(3), 'r:')
plot3(r*u2(1),r*u2(2),r*u2(3), 'g:')
plot3(r*u3(1),r*u3(2),r*u3(3), 'b:')
```

(use the up-arrow key ↑ to save typing).

(b) Now use the vectors $u_1, u_2, u_3$ to obtain an orthogonal basis for $\mathbb{R}^3$, following the Gram-Schmidt algorithm (see the proof of Theorem 6.6 on page 323 of the text). Set $v_1 = u_1$. Define the projection $P_1$ onto the span of $v_1$ as in Question 1(d), and obtain $v_2$ by removing the component of $u_2$ in the direction $v_1$:

$$P_1 = v_1*inv(v_1'*v_1)*v_1', \ v_2 = u_2 - P_1*u_2$$

Calculate the dot product to check that the vectors $v_1$ and $v_2$ are mutually orthogonal (within a negligible numerical error). Also add $v_2$ to your graphics window as a dashed-dotted green line by
plot3(r*v2(1),r*v2(2),r*v2(3), 'g-.')

Using the Rotate 3D command, rotate the frame to see that the red line for \(v_1\) and the green line for \(v_2\) are orthogonal.

Now define \(P_2\) as the projection onto the span of \(v_2\) and obtain \(v_3\) by removing the components of \(u_3\) in the directions of \(v_1\) and \(v_2\):

\[
P_2 = v2*inv(v2'*v2)*v2', \quad v_3 = u3 - P1*u3 - P2*u3
\]

(1)

Calculate dot products by MATLAB to check that \(v_3\) is orthogonal to the vectors \(v_1\) and \(v_2\) (within a negligible numerical error). Add \(v_3\) to your plot as a dashed-dotted blue line by

\[
plot3(r*v3(1),r*v3(2),r*v3(3), 'b-.')
\]

Using the Rotate 3D command, rotate the frame to see that the red line for \(v_1\), the dash-dot green line for \(v_2\), and the dash-dot blue line for \(v_3\) are mutually orthogonal.

(1)

Obtain a good alignment of the graph that shows orthogonality in perspective and then print the graph.

Include the graph in your lab report.

(c) The last step in the Gram-Schmidt algorithm is to rescale the vectors \(v_1, v_2, v_3\) to obtain an orthonormal basis for \(\mathbb{R}^3\):

\[
w_1 = v1/norm(v1), \quad w_2 = v2/norm(v2), \quad w_3 = v3/norm(v3)
\]

Define the matrix \(Q = [w_1, w_2, w_3]\) and give written answers to the following questions.

(1)

(i) Write out a symbolic (not numerical) hand calculation of the entries in the \(3 \times 3\) matrix \(Q^TQ\) in terms of the dot products \(w_i \cdot w_j\). Use this to describe the orthonormal property of \(\{w_1, w_2, w_3\}\) in terms of \(Q^TQ\).

(ii) What is the inverse matrix \(Q^{-1}\)?

Now check your answers to questions (i) and (ii) with MATLAB calculations.

(d) The Gram-Schmidt algorithm gives the QR factorization of a matrix (see the boxed statement on page 382 of the text). To illustrate this using MATLAB, set

\[
A = [u1, u2, u3], \quad R = Q'*A
\]

(1)

Verify by MATLAB that \(A = Q*R\). Then give a symbolic (not numerical) proof of the fact that \(R\) is upper triangular, as follows.

(1)

(iii) Let \(R\) have entries \(r_{ij}\). Use the property \(w_2 \cdot u_1 = 0\) to show that \(r_{21} = 0\). Likewise, use the property \(w_3 \cdot u_1 = w_3 \cdot u_2 = 0\) to show that \(r_{31} = r_{32} = 0\).

4

Question 3. Orthogonal Projection onto a Subspace

Generate three random vectors \(a_1, a_2, a_3 \in \mathbb{R}^5\) and the matrix \(A\) with these vectors as columns:

\[
a1 = rvect(5); \quad a2 = rvect(5); \quad a3 = rvect(5); \quad A = [a1, a2, a3]
\]

Check whether they are linearly independent by calculating \(\text{rank}(A)\). If the answer is not 3, then generate a new random set of vectors and calculate the rank. Repeat until the rank is 3. Now use these vectors and matrix in the following.

(a) Let \(W = \text{Col}(A)\) be the subspace of \(\mathbb{R}^5\) spanned by \(\{a_1, a_2, a_3\}\). The teaching code \text{grams.m} carries out the steps of the Gram-Schmidt algorithm, just as you did step-by-step in Question 2. Calculate

\[
Q = \text{grams}(A); \quad w1 = Q(:,1), \quad w2 = Q(:,2), \quad w3 = Q(:,3)
\]

by MATLAB.

(1)

Calculate \(Q^*Q\) by MATLAB and explain why your answer shows that \(\{w_1, w_2, w_3\}\) is an orthonormal set of vectors. (HINT: Look at Question 2 (c).)

(b) Orthogonal Decomposition \(v = w + z\): The orthogonal projection \(P\) from \(\mathbb{R}^5\) onto the subspace \(W\) is given by the \(5 \times 5\) matrix
\[ P = w_1w_1' + w_2w_2' + w_3w_3' \]

If \( v \in \mathbb{R}^5 \) then \( Pv = (w_1 \cdot v)w_1 + (w_2 \cdot v)w_2 + (w_3 \cdot v)w_3 \) (see the boxed formula on page 376 of the text). Use \text{MATLAB} to calculate \( P \). Then generate a random vector \( v = \text{rvect}(5) \) and calculate

\[ w = P \ast v, \quad z = v - w. \]

If \( z = 0 \) (this is very unlikely) generate another random vector until you get one with \( z \) not zero.

\((1)\)

Verify by \text{MATLAB} that \( P \ast w = w \) and \( P \ast z = 0 \). This shows that \( w \) is in the subspace \( W \) and that \( z \) is the component of \( v \) perpendicular \( W \) (see Figure 6.9 on page 388 of the text).

\((1)\)

(c) The projection matrix \( P \) onto the subspace \( W \) can be calculated directly from the matrix \( A \), without first orthogonalizing the columns of \( A \). Use \text{MATLAB} to obtain the matrix

\[ PW = A \ast \text{inv}(A' \ast A) \ast A' \]

(see Theorem 6.8 on page 395 of the text). Check by \text{MATLAB} that \( \text{norm}(PW - P) \) is zero (up to negligible numerical error).

4

**Question 4. Approximate Solution to Inconsistent Linear System**

For this question use the \( 5 \times 3 \) matrix \( A \) and random vector \( v \) from Question 3.

(a) **Approximate Solution to** \( Ax = v \): Let \( v = w + z \) be the orthogonal decomposition from Question 3(c). Show that

\((1)\)

(i) The equation \( Ax = v \) is *inconsistent* (has no solutions).

\((1)\)

(ii) The equation \( Ax = w \) is *consistent*.

(Calculate the reduced row echelon form of the augmented matrix in each case, and explain).

\((1)\)

Now solve the equation in (ii) by

\[ xls = \text{inv}(A' \ast A) \ast A' \ast v \]

(see page 404 of the text). Check by \text{MATLAB} that \( A \ast xls = w \) (up to negligible numerical error). Denote the vector \( xls \) by \( \bar{x} \). It is called the *least squares* approximate solution to the inconsistent equation \( Ax = v \).

\((1)\)

(b) **Closest Vector Property:** Generate a random vector \( y = \text{rvect}(3) \) and verify by \text{MATLAB} that \( P \ast A \ast y = A \ast y \). This shows that \( A \ast y \in W \). Since \( A \ast x = w \), the vector \( A \ast x \) is the vector in \( W \) that is closest to \( v \). Thus the function \( ||A \ast y - v|| \) is *minimized* by choosing \( y = \bar{x} \). This is the *closest vector* property (see page 397 of the text), which is also called the *least squares* property. To illustrate this, use the \text{MATLAB} \( \text{norm} \) function to verify that \( ||A \ast \bar{x} - v|| < ||A \ast y - v|| \).

6

**Question 5. Fitting Curves to Data Points**

Read Section 6.4 of the text and work the suggested exercises for that Section.

(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

\[ y = 4 + t; \ \text{linefit}(t,y) \]

(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 \( 4 + t \).

\((1)\)

Print this graph and 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):

\[ y = 2 + 4 \ast \text{rand}(10,1) + t; \ \text{linefit}(t,y) \]
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.

(1) How many data points are above the best-fitting line? How many data points are below the best-fitting line? The equation of the best line fitting the random data points is displayed above the graph.

(1) Label and print the figure. Include the printed copy in your lab write-up.

(c) Fitting a Parabola to Data: This part refers to Exercise #38 on page 410 of the text.

(1) Use MATLAB to define vectors \( t, y \in \mathbb{R}^6 \) using the data given in Exercise #38. Then define a \( 6 \times 3 \) matrix \( C \) whose first column has all ones, the second column is \( t \), and the third column has the squares of the entries in \( t \). You can get the third column by the command \( t\cdot2 \) (notice the period before the exponent). Use the method of Example 2 on page 405 of the text to calculate the vector \( x \in \mathbb{R}^3 \) whose components \( a, b, c \) are the coefficients in the best-fitting parabola.

(1) Plot the data points and the least-squares parabola fitting these points by the commands

```matlab
figure; plot(t,y,'*'); hold on
s = [0:0.1:30]; u = ones(301,1); A = [u s' (s.^2)'];
plot(s, A*x)
```

You should get a figure with the six data points indicated by * and a smooth parabola passing very close to each data point (as in Figure 6.17 on page 406 of the text). Label and print out the figure and include it in your lab write-up.

(1) Calculate the relative least-squares error \( \frac{\text{norm}(y - C\cdot x)}{\text{norm}(y)} \) This should be less than 0.01 (a one-percent error).

Final Editing of Lab Write-up: After you have worked through all the parts of the lab assignment, you will need to edit your diary file. Remove all errors and other material that is not directly related to the questions. Your write-up should only contain the required MATLAB calculations and the answers to the questions. Preview the document before printing and remove unnecessary page breaks and blank space.