Revised 10/17/2021

LAB 3: *LU* Decomposition and Determinants

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

- The LU decomposition of an invertible square matrix A.
- How to use the LU decomposition to solve the system of linear equations $A\mathbf{x} = \mathbf{b}$.
- Comparison of the computation time to solve $A\mathbf{x} = \mathbf{b}$ by Gaussian elimination vs. solution by LU decomposition of A.
- The determinant of a square matrix, how it changes under row operations and matrix multiplication, and how it can be calculated efficiently by the LU decomposition.
- The geometric properties of special types of matrices (rotations, dilations, shears).

Preliminaries

Reading from Textbook: Before beginning the Lab, read through Sections 2.6, 3.1 and 3.2 of the text and work the suggested problems for these sections.

Tcodes: In this course you will use some instructional MATLAB *m-files* called *Tcodes.* For this lab you will need the Teaching Codes

`cofactor.m, splu.m plot2d.m, house.m`

You can download them from the Canvas site (used for Matlab assignments). Before opening MATLAB to work on the Lab questions you should copy these codes to your directory by the method described above.

Script files: You will need the MATLAB script files `rvect.m` and `rmat.m` from Lab 2 (if you didn't do Lab 2, get a copy of that assignment and follow the directions there to create these m-files). Be sure that you have set the path in MATLAB so that MATLAB can find your own m-files and the Teaching Codes.

Lab Write-up: You should open a diary file at the beginning of each MATLAB session (see Lab 1 for details). Begin the diary file with the following comment lines, filling in your information where appropriate.

```
% [Name]
% [Last 4 digits of RUID]
% Section [Section number]
% Math 250 MATLAB Lab Assignment #3
```

Type `format compact` so that your diary file will not have unnecessary spaces. Put labels to mark the beginning of your work on each part of each question. For example,

```
% Question 1 (a) ...
      ⋮
% Question 1 (b) ...
```

and so on.

Be sure to answer all the questions in the lab assignment. Insert comments in your diary file as you work through the assignment.

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.

5 Question 1. Row Operations and LU Factorization

In this problem you will use MATLAB to carry out elementary row operations and to obtain the matrix factorization $A = LU$ for a square 3×3 matrix A.

Random Seed: 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.

(a) Generate a random 3×3 matrix A and calculate the three *principal minors* of A.

```
A = rand(3), A(1,1), det(A(1:2,1:2)), det(A)
```

The factorization $A = LU$ is only possible when all the principal minors are nonzero. Since A is a random matrix, this condition is almost certainly satisfied. If any of the numbers after the matrix A is zero (this is very unlikely to happen), repeat this step until you generate a matrix A with all three numbers after A nonzero. INCLUDE ALL THE MATRICES THAT YOU GENERATE THIS WAY IN YOUR LAB REPORT.
When you have a matrix A for which all three principal minors are nonzero, you can transform A into an upper-triangular matrix U using only one type of row operation: adding a multiple of one row to a row below. You will choose multipliers to put zeros below the diagonal elements. At the end of the LU algorithm U will be upper triangular.
Start by entering the initial value `U = A` in MATLAB. Now use the MATLAB editor to create an m-file called `col1.m` with the following MATLAB commands:

```
L1 = eye(3);
L1(2,:)  = L1(2,:)  - (U(2,1)/U(1,1))*L1(1,:);
L1(3,:)  = L1(3,:)  - (U(3,1)/U(1,1))*L1(1,:);
L1
```

(notice the use of ; to suppress screen output of the intermediate results). This m-file requires a 3×3 matrix `U` to be already defined in your workspace. Execute this file by typing `col1` at the MATLAB prompt. The matrix `L1` should be *unit lower triangular* with nonzero entries only on the diagonal and in column 1.

(1) Set `U = L1*U` using MATLAB. Remember that the command `X = Y` in MATLAB means to replace the current value of the variable `X` by the current value of the variable `Y`). The new matrix `U` should have zeros in the first column below the main diagonal.

(1) Describe in words the row operations that change the old value of `U` into the new value of `U`. Use symbolic reference to the entries in `U` as in the `col1.m` code. Don't use the specific decimal entries in `U`.

(b) The next step in the LU factorization is to put a zero below the main diagonal in column 2 of `U`. Use the MATLAB editor to create an m-file called `col2.m` with the commands

```
L2 = eye(3);
L2(3,:)  = L2(3,:)  - (U(3,2)/U(2,2))*L2(2,:);
L2
```

This will be used with the matrix `U` modified as in (a). Execute this file by typing `col2` at the MATLAB prompt. The matrix `L2` should be *unit lower triangular* with nonzero entries only on the diagonal and in column 2.

(1) Now set `U = L2*U` using MATLAB. The new matrix `U` should have all zeros below the main diagonal.

(1) Describe in words the row operations that change the old value of `U` into the new value of `U`. Use symbolic reference to the entries in `U` as in the `col2.m` code. Don't use the specific decimal entries in `U`.

Use MATLAB to verify that `U = L2*L1*A`.

(1) **(c)** To complete the $A = LU$ factorization, calculate

```
inv(L1), inv(L2), L = inv(L1)*inv(L2)
```

Notice that column 1 of `L` is the same as column 1 of `inv(L1)` and column 2 of L is the same as column 2 of `inv(L2)`. (See page 155 of the text.) Check by MATLAB that `A = L*U`.

$\boxed{3}$ Question 2. Using LU Factorization to Solve $A\mathbf{x} = \mathbf{b}$

(1) **(a) Inverting L and U:** Let `L` and `U` be the matrices from Question #1(c). Give a formula for the inverse matrix `inv(L)` in terms of the matrices `L1` and `L2`. Then calculate `inv(L)` and `inv(U)` using MATLAB. Notice that both of these matrices are in *triangular form.*

(2) **(b) Solving $A\mathbf{x} = \mathbf{b}$ using L^{-1} and U^{-1}** (See Example 4 on page 158 of the text): Use the m-file `rvect.m` from Lab 2 to generate a random integer vector `b = rvect(3)`. Calculate the solution

```
c = inv(L)*b
```

to the lower triangular system $L\mathbf{c} = \mathbf{b}$. Then calculate the solution

```
x = inv(U)*c
```

to the upper triangular system $U\mathbf{x} = \mathbf{c}$. Finally, calculate $A\mathbf{x}$ and check that it is $\mathbf{b}$ (since the entries in $\mathbf{b}$ are integers, this should be obvious by inspection).

$\boxed{3}$ Question 3. Speed comparison of LU versus rref for solving $A\mathbf{x} = \mathbf{b}$

In this question you will compare the speed of two methods of solving the equation $A\mathbf{x} = \mathbf{b}$ when A is an invertible square matrix. You will use the MATLAB `tic` and `toc` commands to measure the computation times.

> **Important:** Be sure to use the semicolon ; after each command as indicated below so that the matrices and vector in this question are *not* displayed or included in your diary file. In case you forget to do this, delete the resulting mess from your diary file. Do not include these large matrices and vectors in your lab write-up.

Generate a random 500×500 matrix A, a vector $\mathbf{b} \in \mathcal{R}^{500}$, and calculate the LU decomposition of A by

```
A = rand(500) ;    b = rand(500,1); [L U] = lu(A);
```

(1) **(a)** Solve $A\mathbf{x} = \mathbf{b}$ by using the reduced row echelon form (Gaussian elimination). The last column $\mathbf{y}$ of the augmented matrix $R = \text{rref}([A\ \mathbf{b}])$ satisfies $A\mathbf{y} = \mathbf{b}$ because `rref(A)` is the identity matrix if A is a random square matrix.

```
tic; R = rref([A b]); y = R(:,501); toc
```

Define the number `rreftime` to be the `elapsed_time` given by the MATLAB output in this case.

(1) **(b)** Next, solve $A\mathbf{x} = \mathbf{b}$ by using the LU decomposition of A:

```
tic; c = inv(L)*b; x = inv(U)*c; toc
```

Define the number `lutime` to be the `elapsed_time` given by the MATLAB output in this case.

Check that the solutions from **(a)** and **(b)** are the same (up to round-off error) by calculating `norm(x - y)` (the norm function gives the length of the vector $\mathbf{x} - \mathbf{y}$).

(1) **(c)** According to the table on page 163 of the text, the computation time for Gaussian elimination is approximately $2cn^3/3$, while the time for the LU method (after the L, U factors are already calculated) is approximately $2cn^2$. Here c is a constant depending on the processing speed of the arithmetic processor in the computer and n is the number of equations.

(i) What is the theoretical ratio `rreftime/lutime` when $n = 500$?

(ii) Calculate the observed ratio `rreftime/lutime` using the results of parts **(a)** and **(b)**.

Comment: The basic arithmetic operations in computers are now so fast that a large proportion of the elapsed computing time consists of data transfer. Thus your answer to *(ii)* will probably not agree exactly with *(i)*. However, for matrices of this size (with 250,000 nonzero entries), the LU method (with L and U already calculated) will still be significantly faster than Gaussian elimination.

8 Question 4. The Determinant Function

(a) Cofactor Expansion: The Teaching Code m-file `cofactor.m` calculates the matrix of cofactors of a square matrix. Generate a random 4×4 integer matrix `a = rmat(4,4)`. Then use MATLAB to calculate the cofactor matrix `c = cofactor(a)`. Now use MATLAB to calculate the four sums

```
a(1,1)*c(1,1) + a(1,2)*c(1,2) + a(1,3)*c(1,3) + a(1,4)*c(1,4)
a(2,1)*c(2,1) + a(2,2)*c(2,2) + a(2,3)*c(2,3) + a(2,4)*c(2,4)
a(1,3)*c(1,3) + a(2,3)*c(2,3) + a(3,3)*c(3,3) + a(4,3)*c(4,3)
a(1,4)*c(1,4) + a(2,4)*c(2,4) + a(3,4)*c(3,4) + a(4,4)*c(4,4)
```

(use the up-arrow key ↑ and edit the line to save retyping).

(1) Use Theorem 3.1 (page 203) and Theorem 3.4 (page 214) to explain why all sums give the same number.

Check by using MATLAB to calculate $\det(a)$.

(b) Determinants of Triangular Matrices: Generate a random 5×5 matrix A and its upper triangular part U by

```
A = rmat(5,5), U = triu(A)
```

Calculate the product `A(1,1)*A(2,2)*A(3,3)*A(4,4)*A(5,5)` of the diagonal entries of A and the corresponding product `U(1,1)*U(2,2)*U(3,3)*U(4,4)*U(5,5)` for the matrix U.

(1) Can you obtain $\det(A)$ from this calculation? Can you obtain $\det(U)$ from this calculation? Explain.

Confirm your answers by a MATLAB calculation.

(c) Row Operations: Generate a 5×5 random integer matrix `A = rmat(5,5)`. Then swap the first and second row of A to get the matrix B using the following commands:

```
B = A; B(2,:)  = A(1,:); B(1,:)  = A(2,:)
```

Use properties of the determinant function to answer the following:

(1) *(i)* What is the relation between $\det(A)$ and $\det(B)$?

Check your answer by calculating `det(A)` and `det(B)` using MATLAB. Next, let C be the matrix obtained from A by multiplying the first row of A by 10 and adding to the second row of A using the following commands:

```
C = A; C(2,:)  = A(2,:)  + 10*A(1,:)
```

Use properties of the determinant function to answer the following:

(1) *(ii)* What is the relation between $\det(A)$ and $\det(C)$?

Check your answer by MATLAB. Finally, let D be the matrix obtained from A by multiplying the first row of A by 10:

```
D = A; D(1,:)  = 10*A(1,:)
```

Use properties of the determinant function to answer the following:

(1) *(iii)* What is the relation between $\det(A)$, $\det(D)$, and $\det(10 * A)$?

Check your answers by MATLAB.

(d) Multiplicative Property: Generate a random 5×5 integer matrix `A = rmat(5,5)`. Then modify A by setting `A(1,1)=0; A(2,1) = 0`. The reduction of the (modified) matrix A to row echelon form can be expressed in terms of a matrix factorization as $PA = LU$. Here P is a *permutation matrix* that expresses the row interchanges that are needed to apply Gaussian elimination to A, and L and U give the LU decomposition of PA (read pages 159-161 of the text). Calculate the $PA = LU$ factorization by using the T-code `splu.m`:

```
[P, L, U, sign] = splu(A)
```

Here `sign` gives $\det(P)$, which is $+1$ for an even number of row interchanges to transform P into the identity matrix, and -1 for an odd number of row interchanges. Check (by MATLAB) that `P*A = L*U`. Then write comments to answer the following.

(1) *(i)* What is $\det(P)$? Why? Compare your answer with the value of `sign` that MATLAB has calculated.

(1) *(ii)* What is $\det(L)$? Why? Check your answer by MATLAB.

(1) *(iii)* What is the relation between $\det(A)$ and $\det(U)$? Why? Check your answer by MATLAB.

$\boxed{6}$ Question 5. Geometry and Matrices

This question uses MATLAB to illustrate the geometric meaning of some special types of matrices. At the MATLAB prompt type

```
H = house; plot2d(H), hold on
```

A graphics window should open and display a crude drawing of a house. The matrix `H` contains the coordinates of the endpoints of the line segments making up the drawing.

(2) **(a) Rotations:** Generate a matrix Q by

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

Let Q act on the house by `plot2d(Q*H)` .

(i) Describe in words how the house has changed.

(ii) Calculate $\det(Q)$. What does this tell you about the area inside the transformed house? (see pages 206-208 of the text).

Repeat this process with `t = - pi/3` (use ↑ to save typing) and answer *(i)* and *(ii)* in this case. Save the graph with the three house images on the same figure as a JPG file, name it house1.jpg.

(2) **(b) Dilations:** Clear the graphics window and generate a new plot of the house as above. Generate a matrix D by

```
r = .9; D = [r, 0; 0, 1/r ]
```

Let D act on the house by `plot2d(D*H)`.

(i) Describe in words how the house has changed.

(ii) Calculate $\det(D)$. What does this tell you about the area inside the transformed house?

Repeat this process with `r = .8` and answer *(i)* and *(ii)* in this case. Save the graph with the three house images on the same figure as a JPG file, name it house2.jpg.

(2) **(c) Shearing Transformations:** Clear the graphics window and generate a new plot of the house as above. Generate a matrix T by

```
t = 1/2; T = [1, t; 0, 1]
```

Now let T act on the house by `plot2d(T*H)`.

(i) Describe in words how the house has changed.

(ii) Calculate $\det(T)$. What does this tell you about the area inside the transformed house?

Repeat this process with `t = - 1/2` and answer *(i)* and *(ii)* in this case. Save the graph with the three house images on the same figure as a JPG file, name it house3.jpg.

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 typing errors. Your write-up should only contain the input commands that you typed and the output results generated by* MATLAB*, together with your answers to the questions.* In particular, remove from your diary file any of the results generated by the `load, save, clear, format, help` commands that you used. Preview the document before submitting and remove unnecessary page breaks and blank space.

Lab write-up submission guidelines: You will submit your diary file `lab3.txt` online via Sakai. In your web browser, go to Sakai and select your tab for this course. You will find the application `Assignments 2`. Open this application, and submit the file `lab1.txt` there. *Lab write-ups will only be accepted if they are in plain text format or PDF format.* **A lab writeup in any other format will not be graded!** In future assignments, you will submit images along with your diary file. Image files will only be graded if they are submitted in `.jpg` or `.png` format!