1. Let $A$ be the $2 \times 2$ matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. By hand, use Gaussian elimination with back substitution to obtain $A^{-1}$ by solving the two systems $Ax_1 = e_1$ and $Ax_2 = e_2$, where $e_1$ and $e_2$ are the columns of the $2 \times 2$ identity matrix. Note that you can perform both at the same time by considering the augmented system $[A|I]$. Prove that $A^{-1}$ exists if and only if $\det(A) \neq 0$.

**ANS:** Note that $\det(A) = ad - bc$ and form the augmented matrix

$$\begin{pmatrix} a & b & | & 1 & 0 \\ c & d & | & 0 & 1 \end{pmatrix}.$$ 

Then the multiplier $m_{2,1} = c/a$ and perform the row operation $-m_{2,1}R_1 + R_2 \rightarrow R_2$ gives

$$\begin{pmatrix} a & b & | & 1 & 0 \\ 0 & d - bc/a & | & -c/a & 1 \end{pmatrix}.$$

First solving for $x_1$, back substitution gives $(x_1)_2 = (-c/a)/(d - bc/a) = -c/(ad - bc)$. Then $(x_2)_1 = (1-b(x_1)_2)/a = d/(ad - bc)$. Solving for $x_2$ now, $(x_2)_2 = 1/(d - bc/a) = a/(ad - bc)$ and $(x_2)_1 = -b(x_2)_2/a = -d/(ad - bc)$. So we see that

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$ 

From this we see that $A^{-1}$ exists if and only if $ad - bc \neq 0$. But we are not done! Note in the row operation that $a$ was in the denominator of the multiplier. What if $a = 0$? Well, we would have to switch rows then, and in this case we must have that $c \neq 0$ since otherwise $A$ would be singular. So interchanging the rows gives

$$\begin{pmatrix} c & d & | & 0 & 1 \\ 0 & b & | & 1 & 0 \end{pmatrix}.$$ 

Since $c \neq 0$, the matrix is nonsingular if and only if $b \neq 0$, so we must have that $cb \neq 0$, but $ad = 0d = 0$ so this is equivalent to $ad - bc \neq 0$, and again we see that $A^{-1}$ exists if and only if $ad - bc \neq 0$. 


2. We used a number of results concerning matrices in class in the discussion of LU decompositions. These problems are intended as a verification of those results, at least in the case of a $3 \times 3$ matrix. Let

$$E_1 = \begin{pmatrix} 1 & 0 & 0 \\ -m_{2,1} & 1 & 0 \\ -m_{3,1} & 0 & 1 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{3,2} & 1 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. $$

(a) Show that

$$E_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ m_{2,1} & 1 & 0 \\ m_{3,1} & 0 & 1 \end{pmatrix}. $$

(b) Show that

$$E_1^{-1}E_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ m_{2,1} & 1 & 0 \\ m_{3,1} & m_{3,2} & 1 \end{pmatrix}. $$

(c) Show that $P_1^{-1} = P_1$.

**ANS:** For (a) it is easily checked that $E_1 * E_1^{-1} = I$. For (b), following the pattern in (a) to determine $E_2^{-1}$ we see that

$$E_1^{-1}E_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ m_{2,1} & 1 & 0 \\ m_{3,1} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{3,2} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ m_{2,1} & 1 & 0 \\ m_{3,1} & m_{3,2} & 1 \end{pmatrix}, $$

and for (c) note that $P_1^2 = I$ ($P_1A$ swaps rows 1 and 2 of $A$, so applying $P_1$ again just swaps them back!) thus $P_1^{-1} = P_1$. 


3. Find the LU factorization of $A$ by hand without partial pivoting and, showing all steps, use it to solve $Ax = b$.

$$A = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ 6 \\ 6 \\ 5 \end{pmatrix}.$$ 

Check that your answer is correct. Would the use of partial pivoting have made any difference here? Discuss.

**ANS:** With $A^{(1)} = A$, $m_{2,1} = 1/4$, so performing $-m_{2,1}R_1 + R_2 \rightarrow R_2$ and storing the multiplier gives

$$A^{(2)} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1/4 & 15/4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{pmatrix}.$$ 

Next, $m_{3,2} = 1/(15/4) = 4/15$, storing this and performing $-m_{3,2}R_2 + R_3 \rightarrow R_3$ gives

$$A^{(3)} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1/4 & 15/4 & 1 & 0 \\ 0 & 4/15 & 56/15 & 1 \\ 0 & 0 & 1 & 4 \end{pmatrix}.$$ 

Finally, $m_{4,3} = 1/(56/15) = 15/56$, storing this and performing $-m_{4,3}R_3 + R_4 \rightarrow R_4$ gives

$$A^{(4)} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1/4 & 15/4 & 1 & 0 \\ 0 & 4/15 & 56/15 & 1 \\ 0 & 0 & 15/56 & 209/56 \end{pmatrix}.$$ 

Writing $L$ and $U$ explicitly gives

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 1 & 0 & 0 \\ 0 & 4/15 & 1 & 0 \\ 0 & 0 & 15/56 & 1 \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 0 & 15/4 & 1 & 0 \\ 0 & 0 & 56/15 & 1 \\ 0 & 0 & 0 & 209/56 \end{pmatrix}.$$ 

Next, to solve $Ax = LUx = b$ let $z = Ux$ and first solve $Lz = b$ by forward substitution:

$$z_1 = b_1 = 5$$
$$z_2 = b_2 - (1/4)z_1 = 6 - 5/4 = 19/4$$
$$z_3 = b_3 - (4/15)z_2 = 6 - 19/15 = 71/15$$
$$z_4 = b_4 - (15/56)z_3 = 5 - 71/56 = 209/56$$

And finally, we solve $Ux = z$ by back substitution:

$$(209/56)x_4 = z_4 = 209/56 \implies x_4 = 1$$
$$(56/15)x_3 = z_3 - x_4 = 71/15 - 1 = 15/56 \implies x_3 = 1$$
$$(15/4)x_2 = z_2 - x_3 = 19/4 - 1 = 15/4 \implies x_2 = 1$$
$$4x_1 = z_1 - x_2 = 5 - 1 = 4 \implies x_1 = 1$$

So $x = [1 \ 1 \ 1 \ 1]^T$ is easily seen to be the solution of $Ax = b$ since $Ax$ is then just the row sums of $A$, which is $b$. 

What about partial pivoting? Notice that at each stage of the elimination the entry of maximum absolute value on or below the diagonal of the column being eliminated was already on the diagonal. The matrix $A$ is symmetric and positive definite, and we'll prove later in the course that this is always the case for such matrices. Thus partial pivoting is not needed!
4. Write a MATLAB m-file (function) to find the LU decomposition of a given $n \times n$ matrix $A$ using partial pivoting. The routine should return the updated matrix $A$ and the pivot vector $p$. Name the file mylu.m, the first few lines of which should be as follows:

```matlab
function [a,p]=mylu(a)
    %
    [n n]=size(a);
    p=(1:n)’;
    (your code here!)
```

The code above sets $n$ equal to the dimension of the matrix and initializes the pivot vector $p$. Make sure to store the multipliers $m_{ij}$ in the proper matrix entries. For more help on function m-files see pages 10 – 13 of the MATLAB primer available from the course webpage. You should experiment with a few small matrices to make sure your code is correct. As a test of your code, in MATLAB execute the statements (exactly as they appear)

```matlab
>>diary mylu.txt
>>format short e
>>type mylu.m
>>a=[2 2 -3;3 1 -2;6 8 1];
>>[a,p]=mylu(a)
>>diary off
```

Recall we discussed the use of the `diary` command in class last term. If need be, check the online MATLAB help or the Mathworks webpage for help using `diary` to save your output. Print and hand-in the text file mylu.txt.

**ANS:** Here is the output in mylu.txt:

```matlab
format short e
type mylu.m

function [a,p] = mylu(a)
[n n] = size(a);
p = (1:n)’;
for k = 1:n-1
    [val,ind] = max(abs(a(p(k:n),k))); % find pivot in p(k:n)
tind = p(k);
    % interchange rows via pivot
    p(k) = p(ind+k-1); % vector, making sure to
    p(ind+k-1) = tind; % adjust for offset
    if (p(k) == 0) % check for 0 pivot
        disp(’Matrix is singular!’);
        return;
    end
    for i = (k+1):n % row loop for elimination
        m = a(p(i),k)/a(p(k),k); % set multiplier
        a(p(i),k) = m;
        % store multiplier for L
        a(p(i),k+1:n) = a(p(i),k+1:n) ... % loop over columns k+1:n
```

```
\[-m*a(p(k), k+1:n);\]

end

end

a=[2 2 -3;3 1 -2;6 8 1];
[a,p]=mylu(a)

a =

\[
\begin{bmatrix}
3.3333e-01 & 2.2222e-01 & -2.7778e+00 \\
5.0000e-01 & -3.0000e+00 & -2.5000e+00 \\
6.0000e+00 & 8.0000e+00 & 1.0000e+00
\end{bmatrix}
\]

p =

\[
\begin{bmatrix}
3 \\
2 \\
1
\end{bmatrix}
\]
5. Consider the problem $Ax = f$ where $A$ is an $n \times n$ tridiagonal matrix,

$$
A = \begin{bmatrix}
    b_1 & c_1 & 0 & \cdots & 0 \\
    a_2 & b_2 & c_2 & \ddots & \vdots \\
    0 & \ddots & \ddots & \ddots & 0 \\
    \vdots & & a_{n-1} & b_{n-1} & c_{n-1} \\
    0 & \cdots & 0 & a_n & b_n
\end{bmatrix}.
$$

Note that the right-hand side of the system is denoted by the vector $f$ here. Write out pseudo-code to solve the system assuming that partial pivoting is not implemented. What is the operation count for the forward elimination and the back substitution steps of Gaussian elimination in this case? Count add/sub and mult/div operations separately, but then give the overall order (big $O$) of the total operations needed.

**ANS:** I will include a MATLAB m-file in place of pseudo-code:

```matlab
function x = mytrige(a,b,c,f)

n = length(b); % determine system size
x = zeros(n,1); % allocate x
for i = 1:n-1 % forward elimintationsweep GE
    m = a(i+1)/b(i); % compute multiplier
    b(i+1) = b(i+1)-m*c(i); % b (but not c) is affected in row operation
    f(i+1) = f(i+1)-m*f(i); % row op applied to RHS
end

x(n) = f(n)/b(n); % back substitution
for i = n-1:-1:1
    x(i) = (f(i)-c(i+1)*x(i+1))/b(i);
end

x = mytrige(a,b,c,f); % include f
end
```

So only a simple forward elimination is needed to zero the entries of $a$ just below the diagonal, followed by a back substitution. For the elimination, the loop is over $i = 1 : n-1$, with 3 mults/divs and 2 adds/subs for each $i$. Thus

$$
\text{# mults/divs} = \sum_{i=1}^{n-1} 3 = 3n - 3 \quad \text{and} \quad \# \text{adds/subs} = \sum_{i=1}^{n-1} 2 = 2n - 2.
$$

Performing the back substitution, we start with 1 division, and now loop over $i = n-1 : -1 : 1$ performing 1 subtraction and 2 mults/divs. Thus

$$
\text{# mults/divs} = 1 + \sum_{i=1}^{n-1} 2 = 2n - 1 \quad \text{and} \quad \# \text{adds/subs} = \sum_{i=1}^{n-1} 1 = n - 1.
$$

Putting these together we have

$$
\text{# mults/divs} = 5n - 4 \quad \text{and} \quad \# \text{adds/subs} = 3n - 3.
$$

Thus, the overall operation count is $O(8n)$. 