Problem 4.1.6

Let A be a square matrix with exactly one nonzero entry in each row and column. By Theorem 1 of the section, we can rearrange A in such a way to make it diagonal, since each row and column contains exactly one nonzero entry. For diagonal matrices, their determinants are simply the product of their diagonal. Thus, $\det A = \prod_{i=1}^{n} a_{ii}$, where a_{ii} is the diagonal element of row i and column i. Since each of these is defined to be nonzero, $\det A$ must be nonzero. Thus, by Theorem 4 of the section, since the determinant isn't zero, A must be nonsingular.

Problem 4.1.9

Done in MATLAB:

```
First, we will make the blocks:

a1 = [1 2];
a2 = [1 -1 0 1];
a3 = [-1 1; 0 1; 1 -1; 1 0];
a4 = [1 0 -1 1; -1 1 0 1; 0 0 1 0; 1 2 1 0];
A = [a1 a2; a3 a4];

b1 = [1 0 1; -1 1 2];
b2 = [2 1; 0 1];
b3 = [1 0 1; -1 1 0; 2 1 0; 0 1 1];
b4 = [1 2; 0 1; -2 1; -1 1];
B = [b1 b2; b3 b4];

A*B

ans =

     1     2     7     2     5
    -3     1     3     0     2
    -3     3     2    -2     1
     4     0    -1     0     1
     2     3     2     1     6

Now, we create the whole matricies:

A = [1 2 1 -1 0 1; -1 1 1 0 -1 1; 0 1 -1 1 0 1; 1 -1 0 0 1 0; 1 0 1 2 1 0];
B = [1 0 1 2 1; -1 1 2 0 1; 1 0 1 1 2; -1 1 0 0 1; 2 1 0 -2 1; 0 1 1 -1 1];
A*B
```

```
ans =

     1     2     7     2     5
    -3     1     3     0     2
    -3     3     2    -2     1
     4     0    -1     0     1
     2     3     2     1     6

diary off
```

Problem 4.1.11

Let

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix},$$

where A is an upper triangular matrix that is nonsingular and I is the $n \times n$ identity matrix. Since A is nonsingular, it has an inverse, which we shall call U. Notice that, since $UA = I$, we can form the following linear system for row i of U:

$$\begin{cases} u_{i1}a_{11} = \delta_{i1} \\ u_{i1}a_{12} + u_{i2}a_{22} = \delta_{i2} \\ \vdots \\ u_{i1}a_{1n} + \cdots + u_{in}a_{nn} = \delta_{in} \end{cases}$$

We can find the elements of U by solving the above system:

$$\begin{cases} u_{i1} = \frac{\delta_{i1}}{a_{11}} \\ u_{i2} = \frac{\delta_{i2} - u_{i1}a_{12}}{a_{22}} \\ \vdots \\ u_{in} = \frac{\delta_{in} - \sum_{k=1}^{n-1} u_{ik}a_{kn}}{a_{nn}} \end{cases}$$

From this, it may be seen that whenever the row index of u exceeds its column index, that term will be zero. Thus, the system is upper triangular as well.

Problem 4.1.12

Let A be an $n \times n$ invertible matrix and let u, v be vectors in $\mathbb{R}^n$. We wish to find the necessary and sufficient conditions on u and v so that

$$\begin{bmatrix} A & u \\ v^T & 0 \end{bmatrix}$$

is invertible. The necessary and sufficient condition is that this matrix's rank must be $n+1$. Using Gaussian elimination, the inverse is found to be

$$\begin{bmatrix} A^{-1} - \frac{(A^{-1}u)(v^TA^{-1})}{v^TA^{-1}u} & \frac{A^{-1}u}{v^TA^{-1}u} \\ \frac{v^TA^{-1}}{v^TA^{-1}u} & -\frac{1}{v^TA^{-1}u} \end{bmatrix}.$$

Problem 4.2.1

(a) By Problem 4.1.11, this is true.

(b) We will use an argument similar to the upper triangular case. Let

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix},$$

where A is a lower triangular matrix that is nonsingular and I is the $n \times n$ identity matrix. Since A is nonsingular, it has an inverse which we shall call L. Notice that since $AL = I$, we can form the following linear system for column j of L:

$$\begin{cases} a_{11}l_{1j} = \delta_{1j} \\ a_{21}l_{1j} + a_{22}l_{2j} = \delta_{2j} \\ \vdots \\ a_{n1}l_{1j} + \cdots + a_{nn}l_{nj} = \delta_{nj} \end{cases}$$

We can find the elements of L by solving the above system:

$$\begin{cases} l_{1j} = \frac{\delta_{1j}}{a_{11}} \\ l_{2j} = \frac{\delta_{2j} - a_{21}l_{1j}}{a_{22}} \\ \vdots \\ l_{nj} = \frac{\delta_{nj} - \sum_{k=1}^{n-1} a_{nk}l_{kj}}{a_{nn}} \end{cases}$$

From this, it may be seen that whenever the column index of l exceeds the row index, the term will be zero. Thus, the system is lower triangular as well.

(c) Let

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ 0 & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & b_{nn} \end{bmatrix}$$

be $n \times n$ upper triangular matrices. Then, their product AB has element c_{ij} calculated as follows:

$$c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}.$$

For any element of an upper triangular matrix, whenever the row index is larger than the column index, the element is 0. Thus, the only times when c_{ij} can be nonzero are when $j \geq i$, so C is also an upper triangular matrix. A similar argument works for lower triangular matrices.

Problem 4.2.4

Each of these algorithms requires n^2 arithmetic operations.

Problem 4.2.5

Let A be a nonsingular upper triangular square matrix, and suppose for contradiction that one of its diagonal elements is zero. The determinant of A will be of the form

$$\det A = a_{11} \cdot a_{22} \cdot \ldots \cdot a_{nn}$$

However, if one of the diagonal elements, a_{ii} is zero, then $\det A = 0$ and by Theorem 4 of section 4.1, A is nonsingular, which is a contradiction.

Now suppose A is an upper triangular square matrix with at least one zero in its diagonal. Suppose for contradiction that A is nonsingular. Then, $\det A \neq 0$. But, $\det A = a_{11} \cdot a_{22} \cdot \ldots \cdot a_{nn}$, so if A is nonsingular then $a_{ii} \neq 0$ $(1 \leq i \leq n)$, which contradicts the fact that A has at least one zero in its diagonal. Thus, an upper triangular matrix is nonsingular $\iff$ its diagonal elements are all nonzero. A similar argument applies for lower triangular matrices.

Problem 4.2.7

Let

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

We wish to show that A does not have an LU-factorization. Suppose for contradiction that such a factorization exists so

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$$

Thus, we may form the following system of equations:

$$\begin{cases} l_{11}u_{11} = 0 \\ l_{11}u_{12} = 1 \\ l_{21}u_{11} = 1 \\ l_{21}u_{12} + l_{22}u_{22} = 1 \end{cases}$$

Notice, for the first equation to hold true, either $l_{11} \equiv 0$ or $u_{11} \equiv 0$. Suppose the former is true, then our second equation becomes $0 \cdot u_{12} = 1$, which is not possible. Now suppose the latter is true, then our third equation becomes $l_{21} \cdot 0 = 1$, which again can not be possible. Thus, there are no triangular matrices L and U that satisfy $LU = A$.

Problem 4.2.26

Let A be a positive definite matrix and let B be nonsingular. Thus, $\forall y \neq 0$, we know $y^T Ay > 0$. Let $y = B^T x$, which is nonzero for $x \neq 0$, and thus $x^T BAB^T x > 0$, so BAB^T is positive definite.

Now let BAB^T be positive definite. Thus, $\forall x \neq 0$, $x^T BAB^T x > 0$. From this, we know that $Bx^T \neq 0$, otherwise this would contradict $x^T BAB^T x$ being positive definite. Thus, B must be singular. Now,

$$x^T Ax = x^T B^{-1} BAB^T B^{-T} x = y^T BAB^T y > 0,$$

where $y = B^{-T} x \neq 0$. So, A is positive definite and we are done.

Problem 4.2.29

Problem 4.2.30

For this problem, I wrote a MATLAB code called "lufactor.m" and obtained the following results:

```
A = [3 0 1; 0 -1 3; 1 3 0];
[L,U] = lufactor(A)


L =


    3.0000         0         0
         0   -1.0000         0
    1.0000    3.0000    8.6667



U =


    1.0000         0    0.3333
         0    1.0000   -3.0000
         0         0    1.0000


diary off
```

Problem 4.2.31

This problem was done with MATLAB, using the built in "chol" function:

```
A = [1 2; 2 5]


A =


     1     2
```

```
     2     5


chol(A)


ans =


     1     2
     0     1


diary off
```

Code:

```
type lufactor.m


function [L,U] = lufactor(A,q)
%[L,U] = lufactor(A)
%
%This is an LU factorization algorithm written by Alexander Winkles that
%performs Crout factorization.
%
%A : The matrix to be factorized
%q : Determines what specific LU factorization will be returned:
%       q == 1 : Crout factorization
%       q == 2 : UL factorization with L unit lower triangularl

if size(A,1) ~= size(A,2)
    fprintf('This matrix is not square!')
else
    n = size(A,1);

    L = zeros(n,n);
    U = zeros(n,n);
    if q == 1
        for i=1:n
            for j=1:n

                sum = 0;
```

```
                m = i - 1;
                for k=1:m
                    sum = sum + (L(i,k)*U(k,j));
                end;

                if i == j
                    U(i,j) = 1;
                end;

                if i >= j
                    L(i,j) = A(i,j) - sum;
                else
                    U(i,j) = (A(i,j)-sum)/L(i,i);
                end;
            end;
        end;
    end;
    if q == 2 %IN PROGRESS
        for i=n:-1:1
            for j=n:-1:1



                if i == j
                    L(i,j) = 1;
                end;
            end;
        end;
    end;
end;
diary off
```