Problem #1

1.
$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 5 & 5 \\ 1 & 5 & 14 \end{pmatrix}$$

(a) Applying Naive Gaussian elimination,

$$\begin{array}{l} \text{pivot} \rightarrow \\ m=1 \\ m=1 \end{array} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & 4 \\ 0 & 4 & 13 \end{pmatrix} \rightarrow \begin{array}{l} \\ \text{pivot} \rightarrow \\ m=1 \rightarrow \end{array} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & 4 \\ 0 & 0 & 9 \end{pmatrix}$$

therefore,

$$A = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & 4 \\ 0 & 0 & 9 \end{pmatrix}}_{U}$$

(b)
$$A = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}}_{L_1} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{pmatrix}}_{D} \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}}_{L_1^T}$$

(c) Letting $L_2 = L_1 D^{1/2}$

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix}$$

Problem #2

(a) Suppose A, B are lower triangular and $C = AB$

Then,

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

- Note that if $a_{ik}\, b_{kj} \neq 0$

$$k \leq i \qquad j \leq k$$

- So when $j > i$, this cannot happen:

$$i \leq k \leq j \quad \text{and} \quad j > i$$

- So all the terms in the sum are 0

$$\Rightarrow c_{ij} = 0 \quad \text{when } j > i$$

(b) Let A, B be unit lower triangular

$$c_{ii} = \sum_{k=1}^{n} a_{ik}\, b_{ki}$$

Again for this to be non-zero

$$k \leq i \text{ and } i \leq k \Rightarrow i = k$$

$$\Rightarrow c_{ii} = a_{ii} \cdot b_{ii} = 1 \cdot 1 = 1$$

(c) Suppose A is unit lower triangular,
Then $B = A^{-1}$ is unit lower triangular.

Proof.

- Note that if $b_i = i^{th}$ column vector of B

$$AB = A \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & & | \\ e_1 & \cdots & e_n \\ | & & | \end{bmatrix}$$

$\Rightarrow A b_n = e_n$

- Now $b_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ k \end{bmatrix}$, otherwise

(d) A, B are upper triangular
$\Rightarrow A^T, B^T$ are lower triangular
As $AB = (B^T A^T)^T$, (a) holds as
$B^T A^T$ is lower $\Delta \Rightarrow AB$ is upper triangular,
(b) holds, (c) hold by exact same reasoning.

Problem 4

$$A = \begin{bmatrix} 4 & 6 & 10 \\ 6 & 25 & 19 \\ 10 & 19 & 62 \end{bmatrix}$$

$$L L^T = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

(a) Row 1: $l_{11}^2 = 4$

$l_{11} l_{21} = 6$

$l_{11} l_{31} = 10$

Row 2: $l_{21} l_{11} = 6$

$(l_{21})^2 + (l_{22})^2 = 25$

$(l_{21})(l_{31}) + (l_{22})(l_{32}) = 19$

Row 3: $(l_{31})(l_{11}) = 10$

$(l_{31})(l_{21}) + (l_{32})(l_{22}) = 19$

$(l_{31})^2 + (l_{32})^2 + (l_{33})^2 = 62$

(b) • $l_{11} = 2$

• $l_{21} = 3$

• $l_{31} = 5$

$l_{21} = 3$

• $l_{22} = \sqrt{25 - 9} = 4$

• $l_{32} = \dfrac{19 - (l_{21})(l_{31})}{l_{22}} = \dfrac{19-(3)(5)}{4} = 1$

• $l_{33} = \sqrt{62 - 1^2 - 5^2} = 6$

$62 - 26$

36

$$LL^T = \begin{bmatrix} 2 & 0 & 0 \\ 3 & 4 & 0 \\ 5 & 1 & 6 \end{bmatrix} \begin{bmatrix} 2 & 3 & 5 \\ 0 & 4 & 1 \\ 0 & 0 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 6 & 10 \\ 6 & 25 & 19 \\ 10 & 19 & 62 \end{bmatrix}$$

✓

Problem 3

Let L be lower triangular, U be upper triangular, and D be diagonal.

(a) Suppose L and U are both unit triangular and $E = LDU$ is diagonal.

1. In the case that D is singular, no. If $D = 0$, for example, then L and U can be any lower and upper triangular matrices.
2. In the case that D is invertible, yes. Not that U must be lower triangular as:

$$U = D^{-1}L^{-1}E$$

And we have proven that inverse of lower triangular matrices are lower triangular. Similarly, L is upper triangular:

$$L = EU^{-1}D^{-1}$$

Therefore U, L must be diagonal.

(b) Suppose E = LDU is nonsingular and diagonal. Then L and U are diagonal.

- Note that L, D, U must be invertible as:

$$det(LDU) = det(L)det(D)det(U)$$

- So we have again,

$$U = D^{-1}L^{-1}E$$

$$L = EU^{-1}D^{-1}$$

Implying by the same reasoning both are diagonal.

(c) Suppose L and U are both unit triangular and if LDU is diagonal, does it follow that L = U = I?

- In the case that D is singular, no, as D = 0, leaving L and U unconstrained unit triangular matrices
- In the case D in nonsingular, we know L and U are diagonal, and since they are unit triangular, that diagonal must be of 1's. So $U = L = I$

-

<u>Problem 5</u>

Code for Cholesky:

Cholesky.m

```
% Goal: Find Cholesky factorization A = LL^T
% (0) Set up matrices
n = 4;

A = [4 3 2 1;
    3 3 2 1;
    2 2 2 1;
    1 1 1 1;]

L = zeros(4);

% (1) Calculate cholesky factorization
for k = 1:n

    %(1.1) Diagonal entry
    sum = A(k,k);
    for s = 1:k-1
        sum = sum - L(k,s)^2;
    end
    sum = sum^(1/2);
    L(k,k) = sum;


    % (1.2) Remaining column entries
    for i = k+1:n
        sum = A(i,k);
        for s = 1:k-1
            sum = sum - L(i,s)*L(k,s);
        end
        sum = sum / L(k,k);
        L(i,k) = sum;
    end
end

L
```

Output:

```
>> P5_Cholesky

A =

     4     3     2     1
     3     3     2     1
     2     2     2     1
     1     1     1     1


L =

    2.0000         0         0         0
    1.5000    0.8660         0         0
    1.0000    0.5774    0.8165         0
    0.5000    0.2887    0.4082    0.7071
```

Problem 6

(a) Code:

P6_Gauss_Siedel.m

```matlab
% Goal: Implement Gauss-Siedel Algorithm to solve linear system Ax = b

% (0) Parameters
c = 1/3;
n = 5;
eps = 10^(-6);

% (1) Set up A and b
A = eye(n);
A(2,1) = c;
A(n-1, n) = c;

b = ones(n,1);
for i=1:n
    if i == 1 || i == n
        b(i) = b(i) + c;
    else
        b(i) = b(i) + 2*c;
        A(i-1, i) = c;
        A(i+1, i) = c;
    end
end

% (2) Implement Gauss-Siedel Algorithm
kmax = 1000;
xprev = zeros(n,1);
x = zeros(n,1);
x_solution = ones(n,1);

for k = 1:kmax          % For each iteration
    for i = 1:n         % For each row
        sum = b(i);     % Solve for the ith variable
        for j = 1:n
            if j ~= i
                sum = sum - A(i,j)*x(i);
            end
        end
        sum = sum / A(i,i);
        x(i) = sum;
    end

    % Stopping criterion
    if norm (x - xprev) < eps
        fprintf("Solution (iteration #%i):\n",k)
        x
        fprintf("Error: %.2d\n", norm(x - x_solution))
        break
    else
        xprev = x;
    end
end
```

Results:

```
>> P6_Gauss_Siedel
Solution (iteration #38):

x =

    1.0000
    1.0000
    1.0000
    1.0000
    1.0000

Error: 3.52e-07
```

(b) Does Gauss-Siedel converge on this systems for all values of c?

```
>> P6_Gauss_Siedel
Gauss Siedel converged for c = 1.00e-05
Gauss Siedel converged for c = 1.00e-03
Gauss Siedel converged for c = 1.00e-01
Gauss Siedel converged for c = 3.00e-01
Gauss Siedel converged for c = 4.90e-01
Gauss Siedel did not converge for c = 5.00e-01
Gauss Siedel did not converge for c = 01
Gauss Siedel did not converge for c = 10
fx >>
```

Clearly when $c \geq .5$, the matrix is no longer diagonally dominant, and no longer theoretically guaranteed to converge for any initial vector $x^{(0)}$. But it does converge for smaller values of c.

Based on these data points, it seems like it is necessary for this particular matrix to be diagonally dominant for Gauss-Siedel to converge, perhaps because if it is not diagonally dominant, it is not positive definite.

- Note: Varying n did not change the values of c for which this matrix converged

When the Gauss Siedel method does converge (c = 3), increasing n doe not seem to affect number of iterations need to converge dramatically:

```
Solution (iteration #38, n = 5):
Error: 3.52e-07
Solution (iteration #42, n = 50):
Error: 2.78e-07
Solution (iteration #44, n = 500):
Error: 3.99e-07
```

(c) Performing the computation we get that

```
B =

         0    0.5000         0         0         0
         0    0.2500    0.5000         0         0
         0    0.1250    0.2500    0.5000         0
         0    0.0625    0.1250    0.2500    0.5000
         0    0.0312    0.0625    0.1250    0.2500


max_eigenvalue =

    0.7500

Solution (iteration #21, n = 5):
Error: 1.73e+00
```

- Therefore the iterative process (Gauss-Siedel) converges as the spectral radius of B is less than 1.

On the other hand when c=2, the spectral radius is computed to be 12, and so the Gauss-Siedel method does not converge.

```
B =

     0    -2     0     0     0
     0     4    -2     0     0
     0    -8     4    -2     0
     0    16    -8     4    -2
     0   -32    16    -8     4


max_eigenvalue =

   12.0000

Gauss Siedel did not converge for c = 02
```