Question 1

1) a) let m_k be the column matrix that represents the multipliers used to annihilate column k in k^{th} stage of elimination. Given $m_k = \begin{bmatrix} -m_{k,1} \\ -m_{k,2} \\ \vdots \\ -m_{k,n} \end{bmatrix}$ $\forall m_{k,i} \in m_k$, $m_{k,i} = 0$ when $i \leq k$.
$\forall m_{k,i} \in m_k$, $m_{k,i}$ is multiplier for ith row when $i > k$.

Given an m_k as described above we see

$$m_k e_k^T = \begin{bmatrix} & 0 & \\ 0 & \begin{matrix} -m_{k,k+1} \\ -m_{k,k+2} \\ \vdots \\ -m_{k,n} \end{matrix} & 0 \end{bmatrix}$$

k^{th} column

First k rows all 0 since $m_{k,i} = 0$ when $i \leq k$ by definition of m_k

next, since e_k is k^{th} row of identity matrix, it has value 1 at only column k, so everywhere besides column k will be 0
AT column k, after row k we get $m_{k,i}$ where $i > k$, meaning each $m_{i,k}$ is multiplier used for ith row on k-th step.

With this we see $I - m_k e_k^T = \begin{bmatrix} 1 & 0 & 0 & \cdots & & 0 \\ & 1 & & & 0 & 0 \\ & & \ddots & & & \vdots \\ & 0 & & 1 & & \\ & & & m_{k,k+1} & \ddots & \\ & & & \vdots & & \\ & & & m_{k,n} & 0 & 1 \end{bmatrix}$

k^{th} column

Where $m_{k,i}$ is multiplier used for row i in step k.

We see this is exactly L_k.
$\therefore L_k = I - m_k e_k^T$

b) Define $m_k' = -m_k = \begin{bmatrix} -m_{k,1} \\ \vdots \\ -m_{k,n} \end{bmatrix}$ where $\forall m_{k,i} \in m_k$ $m_{k,i} = 0$ when $i \leq k$
where $\forall m_{k,i} \in m_k$ $m_{k,i}$ is multiplier for row i in step k

Now we see: $m_k' e_k^T = \begin{bmatrix} & 0 & \\ 0 & \begin{matrix} m_{k,k+1} \\ \vdots \\ m_{k,n} \end{matrix} & 0 \end{bmatrix}$

1^{st} k rows all 0 since $m_{k,i} = 0$ $\forall i \leq k$

→ Since e_k is k^{th} row of I, it has value 1 only at column k. So every other column besides k is 0.
→ At column k, after row k, we get $m_{k,i}$ where $i > k$ meaning each $m_{k,i}$ is multiplier used for ith row in k^{th} step

So we see: $I - m_k' e_k^T = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ & 1 & & 0 & 0 \\ & & 1 & & \vdots \\ 0 & & -m_{k,k+1} & \ddots & \\ & & -m_{k,n} & 0 & 1 \end{bmatrix}$

we see we get a gauss transform with -1 times its multipliers. So this is exactly L_k^{-1}

$\therefore L_k^{-1} = I - m_k' e_k^T$ where $m_k' = -m_k$

c) $(L_k L_j)^{-1} = L_j^{-1} L_k^{-1}$

$$\begin{aligned}
&= (I + m_j e_j^T)(I + m_k e_k^T) \\
&= I + m_k e_k^T + m_j e_j^T + m_j e_j^T m_k e_k^T \\
&= I + m_k e_k^T + m_j e_j^T \quad \text{(see aside)} \\
&= I - (-m_k e_k^T) - (-m_j e_j^T) + I - I \quad \text{(add \& subtract same thing)} \\
&= (I - (-m_k e_k^T)) + (I - (-m_j e_j^T)) - I
\end{aligned}$$

Aside:

By definition of m_j we see:

$$m_j e_j^T = \begin{array}{c} \left[\begin{array}{ccc} & 0 & \\ 0 & \begin{matrix} m_{j,j+1} \\ m_{j,j+2} \\ \vdots \\ m_{j,n} \end{matrix} & 0 \end{array}\right] \end{array} \begin{array}{l} \}\ j \text{ rows} \\ \\ \\ \end{array}$$

*see explanation in part b & a

$$m_k = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ m_{k+1} \\ \vdots \\ m_n \end{bmatrix}$$

*we know $j<k$, so $m_{k,j}=0$ since $j<k$ & by definition of m_k, $m_{k,i}=0$ $\forall\, i \leq k$. $\therefore$ Since all of $m_j e_j^T$ non-zero values are on column j we see $m_j e_j^T e_k = \vec{0}$

$\therefore m_j e_j^T m_k e_k^T = \vec{0}$.

$\therefore$ we see that $(L_k L_j)^{-1} = \overbrace{(I - (-m_k)e_k^T) + (I - (-m_j)e_j^T)}^{①}\overbrace{- I}^{②} = L_k^{-1} + L_j^{-1} - I$.

By part ⓑ, we see that we must flip the signs of all the multipliers (since we have $I - (-m_i)e_i^T$ for both $i=k$ and $i=j$). Then we add the 2 transforms (as highlighted by ①). Finally, we subtract Identity (highlighted by ③).

d)
$$\begin{aligned}
\tilde{L}_k &= P_i L_k P_i \\
&= P_i (I - m_k e_k^T) P_i \\
&= (P_i - P_i m_k e_k^T) P_i \\
&= P_i P_i - P_i m_k e_k^T P_i \\
&= I - P_i m_k e_k^T P_i
\end{aligned}$$

If $i > k$:

→ we see that by pre-multiplying P_i we swap rows i & j. Since $j > i > k$, we are swapping rows that contain multipliers (i.e. all values under a_{kk}), so we are swapping multipliers $m_{k,i}$ and $m_{k,j}$

→ then by post-multiplying by P_i, we swap columns i & j where $j > i > k$ & by def of $m_k e_k^T$, we are simply swapping columns with all 0s (since only column k has non-zero values)

→ After subtraction from I we get $\tilde{L}_k$ which is L_k with multipliers i & j swapped

If $i \leq k$:

→ by premultiplying P_i we swap rows i & j & since $i < k$, we are swapping a $\vec{0}$ row with another row, since by def. of $m_k e_k^T$, all rows $\leq k$ are 0 rows. Hence we are <u>not</u> swapping multiples.

Question 2

$$A = \begin{bmatrix} 1 & & & 4 \\ 2 & 1 & & \\ 0 & 2 & \ddots & \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

$$L_1 = \begin{bmatrix} 1 & \cdots & & \\ -2 & 1 & & 0 \\ 0 & & \ddots & \\ \vdots & 0 & & \\ 0 & & & 1 \end{bmatrix} \qquad L_1 A = \begin{bmatrix} 1 & \cdots & & 4 \\ 0 & 1 & & -8 \\ \vdots & 2 & \ddots & \\ 0 & & 2 & 1 \end{bmatrix}$$

Define $L_k = \begin{cases} a_{ij} = 1 \text{ when } i=j & \text{where } i \in [1,n] \\ a_{ij} = -2 \text{ when } i=k+1 \ \& \ j=k & \text{where } k \in [1, n-1] \\ a_{ij} = 0 \text{ otherwise} \end{cases}$

$L_k^{-1} = \begin{cases} a_{ij} = 1 \text{ where } i=j \\ a_{ij} = 2 \text{ where } i=k+1 \ \& \ j=k \\ a_{ij} = 0 \quad o/w \end{cases}$

$A = L_n^{-1} \cdots L_1^{-1} U$

w. $U = \begin{cases} a_{ij} = 1 \text{ when } i=j \\ a_{ij} = (-2)^{i+1} \text{ when } j=n \text{ and } i \in [1, n-1] \\ a_{ij} = (-2)^{i+1}+1 \text{ when } j=n \text{ and } i=n \\ a_{ij} = 0 \quad o/w \end{cases}$

$L = L_n^{-1} \cdots L_1^{-1}$

$= L_n + L_{n-1} + \cdots + L_1 - (n-1) I$

$= \begin{cases} a_{ij} = 1 \text{ if } i=j & \text{where } i \in [1,n] \\ a_{ij} = -2 \text{ if } i=j+1 & \text{where } j \in [1, n-1] \\ a_{ij} = 0 \quad o/w \end{cases}$

b) The risk of over flow comes from the compounding of 4 throughout the last column.

Since $fl(x) \in \mathbb{R}_{10}(16,2)$ the largest # we can store is 9999999999999999 $\times 10^{99}$ so when the compounding of 4 (as defined as $(-1)^{i+1} 2^{i+1}$ in our definition of U), passes that point we will get overflow

$|(-1)^{i+1} 2^{i+1}| \leq 10^{100} - 1$ must be true to store a float.

$2^{i+1} = 10^{100}$

$i+1 = \log_2(10^{100})$

$i+1 = 332.19$

$i = 331.19$

$\therefore$ we see that the largest order matrix in this form that can be factored in $\mathbb{R}_{10}(16,2)$ before overflow occurs is order 331

c) No the factorization is not stable, as it will fail when the order of the matrix is greater than 331 due to the overflow error.

Question 3

a) $$A=\begin{bmatrix} 1 & & & 4 \\ 2 & 1 & & \\ & 2 & \ddots & \\ 0 & & 2 & 1 \end{bmatrix}$$

Let P_{ij} be a permutation matrix that swaps rows i & j where $i<j$

$$L_1=\begin{bmatrix} 1 & & 0 \\ -\frac{1}{2} & \ddots & \\ 0 \quad 0 & & 1 \end{bmatrix} \qquad L_1 P_{12} A=\begin{bmatrix} 2 & 1 & & 1 \\ & -\frac{1}{2} & & 4 \\ 0 & 2 & 1 & \\ & & \ddots & 1 \end{bmatrix}$$

$$L_2=\begin{bmatrix} 1 & & & 0 \\ & 1 & \ddots & \\ 0 & \frac{1}{4} & & \\ & 0 \quad 0 & & 1 \end{bmatrix} \qquad L_2 P_{23} L_1 P_{12} A=\begin{bmatrix} 2 & 1 & \cdots & 0 \\ 0 & 2 & 1 \cdots & 0 \\ 0 & 0 & \frac{1}{4} \cdots & 4 \\ & 0 & 2 \ddots & 0 \\ & & & 1 \end{bmatrix}$$

$\therefore$ we see $$U=\begin{cases} a_{ij}=2 & \text{when } i=j \ \forall i\in[1,n-1] \\ a_{ij}=1 & \text{when } i=j-1 \ \forall j\in[1,n] \\ a_{ij}=4+(-1)^{i+1}(2)^{1-i} & \text{when } i=n \text{ and } k=n \\ a_{ij}=0 & \text{o/w} \end{cases}$$

$$L_k=\begin{cases} a_{ij}=1 & \text{when } i=j \ \forall i\in[1,n] \\ a_{ij}=(-2)^{-k} & \text{when } i=k+1 \text{ and } j=k \\ a_{ij}=0 & \text{o/w} \end{cases}$$

$$L_k^{-1}=\begin{cases} a_{ij}=1 & \text{when } i=j \ \forall i\in[1,n] \\ a_{ij}=(-1)^{k+1}\cdot 2^{-k} & \text{when } i=k+1 \text{ and } j=k \\ a_{ij}=0 & \text{o/w} \end{cases}$$

$$L=\begin{cases} a_{ij}=1 & \text{when } i=j \ \forall i\in[1,n] \\ a_{ij}=(-1)^{k+1}2^{-k} & \text{when } i=j+1 \ \forall j\in[1,n-1] \\ a_{ij}=0 & \text{o/w} \end{cases}$$

$$P=\begin{cases} a_{ij}=1 & \text{when } i=j-1 \ \forall j\in[1,n] \\ a_{ij}=1 & \text{when } i=n \text{ and } j=1 \\ a_{ij}=0 & \text{o/w} \end{cases}$$

b) There is no risk of overflow when using pivoting, as our definition of U shows that we never deal with any large numbers.

c) Yes the factorization is stable since there is no risk for overflow error

Question 4

a) $A = \begin{bmatrix} 2 & 5 & 10 \\ 8 & 32 & 8 \\ 1 & 8 & 13 \end{bmatrix}$ $\quad b = \begin{bmatrix} 7 \\ -16 \\ 6 \end{bmatrix}$

$P_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\quad L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1/4 & 1 & 0 \\ -1/8 & 0 & 1 \end{bmatrix}$ $\quad L_1P_1A = \begin{bmatrix} 8 & 32 & 8 \\ 0 & -3 & 8 \\ 0 & 4 & 12 \end{bmatrix}$

$P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ $\quad L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3/4 & 1 \end{bmatrix}$ $\quad L_2P_2L_1P_1A = \begin{bmatrix} 8 & 32 & 8 \\ 0 & 4 & 12 \\ 0 & 0 & 17 \end{bmatrix} = U$

$L_2P_2L_1P_1A = U$

$P_2L_1P_1A = L_2^{-1}U$

$\underbrace{P_2L_1P_2}_{\hat{L}_1}P_2P_1A = L_2^{-1}U$

$P_2P_1A = \hat{L}_1^{-1}L_2^{-1}U$

$\hat{L}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1/8 & 1 & 0 \\ -1/4 & 0 & 1 \end{bmatrix}$ $\quad \hat{L}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1/8 & 1 & 0 \\ 1/4 & 0 & 1 \end{bmatrix}$ by lemma 1

$L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3/4 & 1 \end{bmatrix}$ by lemma 1

$L = \hat{L}_1^{-1}L_2^{-1} = \hat{L}_1^{-1} + L_2^{-1} - I$ by lemma

$= \begin{bmatrix} 1 & 0 & 0 \\ 1/8 & 1 & 0 \\ 1/4 & -3/4 & 1 \end{bmatrix}$

$P = P_2P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$

$\therefore$ we get $PA = LU$ where $P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ $\quad A = \begin{bmatrix} 2 & 5 & 10 \\ 8 & 32 & 8 \\ 1 & 8 & 13 \end{bmatrix}$ $\quad L = \begin{bmatrix} 1 & 0 & 0 \\ 1/8 & 1 & 0 \\ 1/4 & -3/4 & 1 \end{bmatrix}$

and $U = \begin{bmatrix} 8 & 32 & 8 \\ 0 & 4 & 12 \\ 0 & 0 & 17 \end{bmatrix}$

b) $A\vec{x} = \vec{b}$

$PA\vec{x} = P\vec{b}$

$L\underbrace{U\vec{x}}_{\vec{d}} = P\vec{b}$

$L\vec{d} = P\vec{b}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1/8 & 1 & 0 \\ 1/4 & -3/4 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} -16 \\ 6 \\ 7 \end{bmatrix}$$

$d_1 = -16$

$d_2 = 6 - \frac{1}{8}d_1 = 6 - \frac{1}{8}(-16) = 8$

$d_3 = 7 - \frac{1}{4}(d_1) + \frac{3}{4}d_2 = 7 - \frac{1}{4}(-16) + \frac{3}{4}(8) = 7 + 4 + 6 = 17$

$U\vec{x} = \vec{d}$

$$\begin{bmatrix} 8 & 32 & 8 \\ 0 & 4 & 12 \\ 0 & 0 & 17 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -16 \\ 8 \\ 17 \end{bmatrix}$$

$17x_3 = 17$

$x_3 = 1$

$4x_2 + 12x_3 = 8$

$4x_2 + 12(1) = 8$

$x_2 = -1$

$8x_1 + 32x_2 + 8x_3 = -16$

$8x_1 + 32(-1) + 8(1) = -16$

$8x_1 = 8$

$x_1 = 1$

$\therefore$ we get solution $\vec{x} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$

c) Consider the case when you have several linear systems that differ only by the right side (ie. same A but different $\vec{b}$ in $A\vec{x} = \vec{b}$). It is much cheaper to first factor out the matrix (which has 1 time cost of $\frac{n^3}{3} + O(n^2)$) and then use factors to solve each linear system (each time costing $n^2 + O(n)$), rather than doing elimination each time (takes $\frac{n^3}{3} + O(n^2)$ flops each time).

Question 5

a) $z = B^{-1}(2A+I)(C^{-1}+A)x$

$Bz = (2A+I)(C^{-1}+A)x$

$= (2A+I)(C^{-1}x + Ax)$ $(*)$

Steps:

① Since C is invertible, we first get system $C\vec{y} = \vec{x}$ (since $C^{-1}\vec{x} = y$). This will take $\frac{n^3}{3} + O(n^2)$ flops

② Next, we calculate $A\vec{x}$ through matrix multiplication. This will take n^2 flops

③ Next, we add $A\vec{x}$ & $C^{-1}x$ which we calculated in ① & ②, call this resulting $\vec{w} \in \mathbb{R}^n$, this addition takes 0 flop

$$B\vec{z} = (2A+I)\vec{w}$$
$$= 2A\vec{w} + I\vec{w}$$
$$= 2A\vec{w} + \vec{w}$$

④ Next, we find $2A\vec{w}$ using matrix multiplication, which takes n^2 flop

⑤ Next, we add the $2A\vec{w}$ calculated in ④ & $\vec{w}$ calculated in ③, call the result $\vec{g} \in \mathbb{R}^n$ which take 0 flop

$$B\vec{z} = \vec{g}$$

⑥ Finally we solve the resulting system, which take $\frac{n^3}{3} + O(n^2)$ flop

Total flop: $\frac{n^3}{3} + O(n^2) + n^2 + n^2 + \frac{n^3}{3} + O(n^2)$

$= \frac{2n^3}{3} + O(n^2)$

$\therefore$ the complexity of this approach is $\frac{2n^3}{3} + O(n^2)$

b) $y = A^{-6}x$

Steps: ① find PA=LU factorization, which takes $\frac{n^3}{3} + O(n^2)$ flop

② multiply A on both sides.

$$y = A^{-6}x$$
$$Ay = AA^{-1}A^{-5}x$$
$$Ay = A^{-5}x$$

③ repeat step ② 5 more times

$$AAAAAA\vec{y} = \vec{x}$$

④ Next, we use the factorization in step ① to solve 6 linear systems involving A (each taking n^2 flop)

$$\left.\begin{aligned} AAAAA\underbrace{A\vec{x}}_{\vec{d}} &= \vec{x} \\ AAAA\underbrace{A\vec{e}}_{\vec{f}} &= \vec{d} \\ AAA\underbrace{A\vec{g}}_{h} &= \vec{f} \\ AA\underbrace{A\vec{i}}_{j} &= \vec{h} \\ A\underbrace{A\vec{k}}_{\ell} &= j \\ A\vec{m} &= \vec{\ell} \end{aligned}\right\}$$

6 times we must use LU-factorization to do solves

total flop $= \frac{n^3}{3} + 6n^2 + O(n)$

$\therefore$ we see complexity of algorithm is $\frac{n^3}{3} + 6n^2 + O(n)$