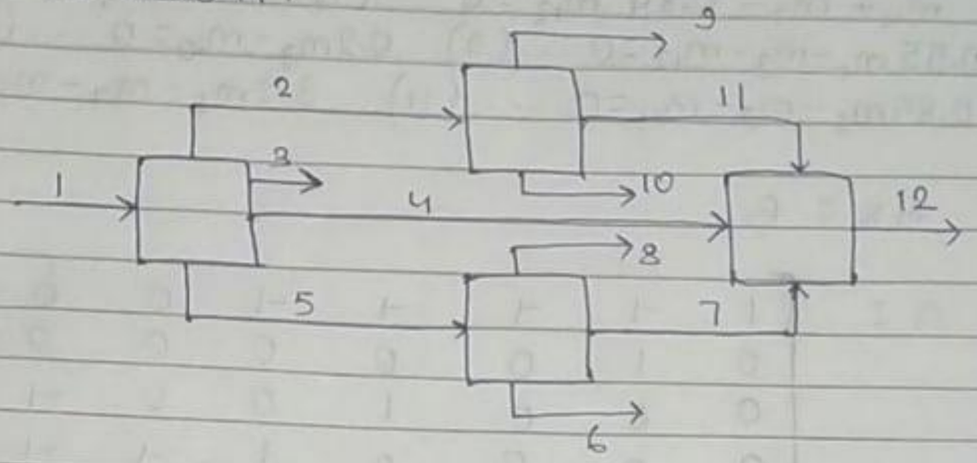
Anshuman Agrawal
18CH10071.



(i) Develop 4 mass balance equations

(ii) $m_1 = 100$; $m_5 = 5 m_8$; $0.84\ m_{12} = m_4 + m_7$
$0.7\ m_1 = m_2 + m_3$; $0.55\ m_1 = m_9 + m_{12}$; $0.2 m_9 = m_{10}$
$0.85\ m_2 = m_9 + m_{11}$; $3.2\ m_6 = m_7 + m_8$.

(iii) solve using a) Matrix inversion method
b) Gauss elimination method

(i) $m_1 = m_2 + m_3 + m_4 + m_5$
$\Rightarrow m_1 - m_2 - m_3 - m_4 - m_5 = 0 \quad \cdots (1)$

$m_2 = m_9 + m_{10} + m_{11}$
$\Rightarrow m_2 - m_9 - m_{10} - m_{11} = 0 \quad \cdots (2)$

$m_4 + m_7 + m_{11} = m_{12}$
$\Rightarrow m_4 - m_7 - m_{11} - m_{12} = 0 \quad \cdots (3)$

$m_5 = m_7 + m_6 + m_8$
$\Rightarrow m_5 - m_6 - m_7 - m_8 = 0 \quad \cdots (4)$

(ii) and $m_1 = 100 \ldots (5)$ $\quad m_5 - 5m_8 = 0 \ldots (6)$

(iii) $m_4 + m_7 - 0.84\, m_{12} = 0 \ldots (7)$ $\quad 0.7m_1 - m_2 - m_3 = 0 \ldots (8)$

$0.55\, m_1 - m_9 - m_{12} = 0 \ldots (9)$ $\quad 0.2m_9 - m_{10} = 0 \ldots (10)$

$0.85m_2 - m_9 - m_{11} = 0 \ldots (11)$ $\quad 3.2m_6 - m_7 - m_8 = 0 \ldots (12)$

MATLAB Code

Matrix Inversion

```
B = zeros (12,1);

B (5) =100;
A = [1 -1 -1 -1 -1 0 0 0 0 0 0 0;
     0 1 0 0 0 0 0 0 -1 -1 -1 0;
     0 0 0 1 0 0 -1 0 0 0 -1 -1;
     0 0 0 0 1 -1 -1 -1 0 0 0 0;
     1 0 0 0 0 0 0 0 0 0 0 0;
     0 0 0 0 1 0 0 -5 0 0 0 0;
     0 0 0 1 0 0 1 0 0 0 0 -0.84;
     0.7 -1 -1 0 0 0 0 0 0 0 0 0;
     0.55 0 0 0 0 0 0 0 -1 0 0 -1;
     0 0 0 0 0 0 0 0 0.2 -1 0 0;
    0 0.85 0 0 0 0 0 0 -1 0 -1 0;
     0 0 0 0 0 3.2 -1 -1 0 0 0 0];

X = A\B;
```

Gauss Elimination

```
B=zeros(12,1);

B(5,1)=100;
C=[1 -1 -1 -1 -1 0 0 0 0 0 0 0;
   0 1 0 0 0 0 0 0 -1 -1 -1 0;
   0 0 0 1 0 0 -1 0 0 0 -1 -1;
   0 0 0 0 1 -1 -1 -1 0 0 0 0;
   1 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 1 0 0 -5 0 0 0 0;
   0 0 0 1 0 0 1 0 0 0 0 -0.84;
   0.7 -1 -1 0 0 0 0 0 0 0 0 0;
   0.55 0 0 0 0 0 0 0 -1 0 0 -1;
   0 0 0 0 0 0 0 0 0.2 -1 0 0;
   0 0.85 0 0 0 0 0 0 -1 0 -1 0;
   0 0 0 0 0 3.2 -1 -1 0 0 0 0];
A = [C B];
```

```
[r,c]=size(A);
% sin -> whether the matrix is singular(sin=1) or non-
singular(sin=0)
singular=false;
for i=1:r
    % finding the i-th pivot:
    % partial pivoting:
    if(i<r)% do partial pivoting only if there are any row
below the current row
        imax=i;     %index of the element with maximum value
        max=A(i,i); %value of that element
        for k=i+1:r
            % finding the max
            if abs(A(k,i))>abs(max)
                max=A(k,i);
                imax=k;
            end
        end
        %swap the rows
        A([i,imax],:)=A([imax,i],:);
    end
    if A(i,i)==0
        % matrix is singular
        singular=true;
    end
    % do for all remaining elements in current row
    for j=i+1:r
        A(j,:)=A(j,:)-A(i,:)*A(j,i)/A(i,i);
        A(j,i)=0; % fill lower triangular matrix with zeros
    end
end
% if matrix is non-singular
if singular==false
    sol=zeros(r,1);% solution array
    % backward susbstitution
    for i=r:-1:1
        s=A(i,c);% s-> it will become the value of x(i)
        for j=r:-1:i+1
            s=s-A(i,j)*sol(j,1);% this value needs to be
removed from s
        end
        sol(i,1)=s/A(i,i);% divide by coeff of x(i)
    end
else
    disp('Matrix is Singular.');
end
```

Results and Answers :-

m_1	100.0000
m_2	20.5230
m_3	49.4770
m_4	37.4653
m_5	-7.4653
m_6	-1.7774
m_7	-4.1948
m_8	-1.4931
m_9	15.3923
m_{10}	3.0785
m_{11}	2.0523
m_{12}	39.6077

Conclusion :-

According to computations in MATLAB, both methods gave the same results for values upto the maximum limit of precision.
However, Gauss-Elimination just uses extra memory by storing the augmented matrix.
By timing both computations, Matrix Inversion seemed to take the lower time as well, chiefly because of the use of the backslash operator and not going with inverting the co-efficient matrix.