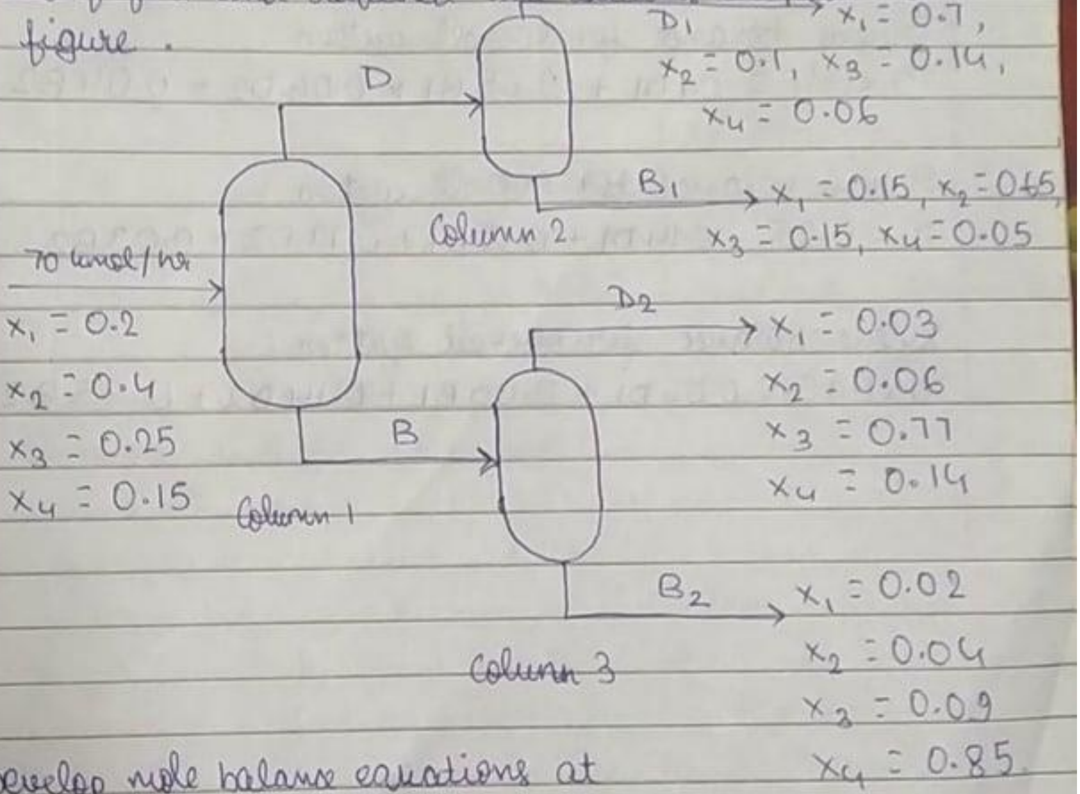
Anshuman Agrawal
18CH10071

System of linear algebraic equations

Separation of benzene (1), toluene (2), styrene (3) and xylene (4) is carried out in a system consisting 3 distillation columns as shown in figure 1. D1, D2, and B1, B2 are the distillate and bottom molar flow rates of columns 2 and 3 respectively. Compositions (mole fractions) of feed and required exit streams are provided in figure.



(1) Develop mole balance equations at steady state.

(2) Solve the equations to find D1, B1, D2, B2 by
a) Gauss Elimination b) Gauss Seidel

Assumptions :-

i) No reflux taking place in either column.

ii) Fully conservative system with no mole losses.

For column 1, $70 = D + B$ ... (i)

For column 2, $D = D1 + B1$ ... (ii)

For column 3, $B = D2 + B2$ ... (iii).

Benzene balance for overall system :

$70 \times 0.2 = 0.7 D1 + 0.15 B1 + 0.03 D2 + 0.02 B2$ ..... (1)

Toluene balance for overall system :

$70 \times 0.4 = 0.1 D1 + 0.65 B1 + 0.06 D2 + 0.04 B2$ ... (2)

Styrene balance for overall system :

$70 \times 0.25 = 0.14 D1 + 0.15 B1 + 0.77 D2 + 0.09 B2$ ... (3)

Xylene balance for overall system :

$70 \times 0.15 = 0.06 D1 + 0.05 B1 + 0.14 D2 + 0.85 B2$ ... (4)

MATLAB Code

Gauss Elimination Method

```
B=zeros(4,1);

B(1)=14;
B(2)=28;
B(3)=17.5;
B(4)=10.5;

C=[0.7 0.15 0.03 0.02;
   0.1 0.65 0.06 0.04;
   0.14 0.15 0.77 0.09;
   0.06 0.05 0.14 0.85];
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
```

Gauss-Seidel Method

```
B=zeros(4,1);

B(1)=14;
B(2)=28;
B(3)=17.5;
B(4)=10.5;

A=[0.7 0.15 0.03 0.02;
   0.1 0.65 0.06 0.04;
   0.14 0.15 0.77 0.09;
   0.06 0.05 0.14 0.85];

X=zeros(4,1);
tol=0.0001; % assumed tolerance since nothing is given
iterations=0;
while true
    temp=X;
    X(1)=(B(1)-A(1,2)*X(2)-A(1,3)*X(3)-
A(1,4)*X(4))/A(1,1);
    X(2)=(B(2)-A(2,1)*X(1)-A(2,3)*X(3)-
A(2,4)*X(4))/A(2,2);
    X(3)=(B(3)-A(3,1)*X(1)-A(3,2)*X(2)-
A(3,4)*X(4))/A(3,3);
    X(4)=(B(4)-A(4,1)*X(1)-A(4,2)*X(2)-
A(4,3)*X(3))/A(4,4);
    iterations=iterations+1;
    fprintf('The iteration vector is: [');
    fprintf('%g ',X);
    fprintf(']\n');

    ctr_false=0;ctr_true=0;
    for i=1:4
        if(abs(X(i)-temp(i))>tol)
            ctr_false=ctr_false+1; % to count number of
times tolerance is not met
        else
            ctr_true=ctr_true+1;
        end
    end

    if(ctr_false>0)
        continue; % continuing iterations when tolerance
is not met
    else
```

```
        break; % stopping iterations when tolerance met
for all variables
    end
end
```

Results and Answers :-

Gauss Elimination :-

D1	10.7308
B1	39.8571
D2	12.1648
B2	7.2473

Gauss Seidel :-

Iteration	D1	B1	D2	B2
0	0	0	0	0
1	20	40	11.2987	6.72727
2	10.7521	39.9658	12.2005	7.23355
3	10.7063	39.8585	12.1706	7.24802
4	10.7302	39.8567	12.1649	7.24738
5	10.7309	39.8571	12.1647	7.24733
6	10.7308	39.8571	12.1648	7.24733

Tolerance : 0.0001 (assumed).

Conclusion :-

The Gauss-Seidel method took slightly less computational time than Gauss-Elimination. However, this can't be concluded in general about the method because depending on the system, the iterations might increase by a huge amount. According to the set tolerance, Gauss-Seidel gave slightly better accuracy for B2 value than Gauss-Elimination. Sometimes the initial guess taken could also influence the computational time for the better or for the worse.