Madhuri Sarda
18CH30030

ASSIGNMENT-2

PROBLEM STATEMENT

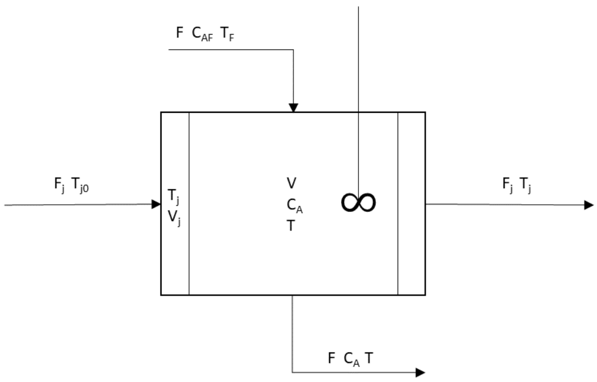
Consider the perfectly mixed CSTR where a first-order exothermic irreversible reaction takes place (r = rate of reaction)

$$A \rightarrow B, \qquad \boldsymbol{r = k_0 e^{-E/RT} C_A}$$

Heat generated by reaction is being removed by the jacket fluid. The reactor volume (V) is constant.

Governing Equations:

(Subscript j indicates parameters related to jacket. Symbols carry their usual significance. Refer to the figure.)



$$\boldsymbol{V\frac{dC_A}{dt} = FC_{Af} - FC_A - rV}$$

$$\boldsymbol{\rho C_p V\frac{dT}{dt} = \rho C_p F\left(T_f - T\right) + (-\Delta H)Vr - UA(T - T_j)}$$

$$\boldsymbol{\rho_j C_j V_j\frac{dT_j}{dt} = \rho_j C_j F_j\left(T_{j0} - T_j\right) + UA(T - T_j)}$$

Model Parameter Values:

Parameter	Value	Parameter	Value
F (m^3/h)	1	C_{Af} (kgmol/m^3)	10
V (m^3)	1	UA (kcal/°C h)	150
k_0 (h^{-1})	36×10^6	T_{j0} (K)	298
(-ΔH) (kcal/kgmol)	6500	$\rho_j C_j$ (kcal/m^3 °C)	600
E (kcal/kgmol)	12000	F_j (m^3/h)	1.25
ρC_p (kcal/m^3 °C)	500	V_j (m^3)	0.25
T_f (K)	298		

Given the above parameter values, there are <u>three steady states</u> for this system. Identify all the steady states by setting LHS of the above ODE to zero and then solving the resulting algebraic equations simultaneously for (*CA*, *T*, *Tj*) by using:

1. Newton's method. Write your own code using any programming language.
2. MATLAB function fsolve.
3. Compare your results (all 3 steady states) obtained by two methods in a Table.

SOLUTION

NEWTON'S METHOD

Code (Using MATLAB)

Initial Guess:

C_{A0} = 5 kgmol/m^3
$T_0 = 300(SS1), 330(SS2), 400(SS3)K;$
$T_{j0} = 500K;$

```
clear all;

global k0 Ea R;
k0 = 36*10^6;
Ea = 12*10^3;
R = 2;
X0 = [5; 400; 500]; %Initial Guess

for i= 1:50
    [f, jacobian] = cstrsteady(X0);
    X1 = X0 - inv(jacobian)*f;  %Newton Raphson method for multivariable

    if(abs(X0-X1)<=10^-6)    %Tolerance Check
        break;
    end
    X0=X1;
end

function [L,jacobian_value] = cstrsteady(X)        %algebraic equations
global k0 Ea R;
F = 1;
rhoCp= 500;   % Model Parameter Values
UA=150;
Tj0=298;
Fj=1.25;
Tf = 298;
rhojCpj=600;
V=1;
Caf=10;
delH = 6500;

Ca = X(1);
T = X(2);
Tj = X(3);

r = k0 * exp(-Ea/(R*T))*Ca ;
L(1, 1) = F*Caf - F*Ca - r*V ;
L(2, 1) = rhoCp*F*(Tf-T) + delH*V*r - UA*(T - Tj);
L(3, 1) = rhojCpj*Fj*(Tj0 - Tj) + UA*(T - Tj);

syms x y z ;                    %Jacobian Evaluation
J = jacobian([10- x- k0 * exp(-Ea/(R*y))*x ; 500*(298-y) + 6500*(k0 * exp(-Ea/(R*y))*x) - 150*(y - z);
600*1.25*(298 - z) + 150*(y - z) ], [x y z]);
jacobian_value = double(subs(J, [x y z], [X(1), X(2), X(3)]));
end
```

fsolve

Code:

Initial Guess:

C_{A0} = 5 kgmol/m^3
$T_0 = 350(SS1), 300(SS2), 400(SS3) K;$
$T_{j0} = 500\ K;$

```
clear all;

global k0 Ea R;
k0 = 36*10^6;
Ea = 12*10^3;
R = 2;
X0 = [5; 350; 500]; %Initial Guess

xout =  fsolve(@cstrsteady,X0);   % fsolve

function [L] = cstrsteady(X)        %algebraic equations
global k0 Ea R;
F = 1;
rhoCp= 500;   % Model Parameter Values
UA=150;
Tj0=298;
Fj=1.25;
Tf = 298;
rhojCpj=600;
V=1;
Caf=10;
delH = 6500;

Ca = X(1);
T = X(2);
Tj = X(3);

r = k0 * exp(-Ea/(R*T))*Ca ;
L(1, 1) = F*Caf - F*Ca - r*V ;
L(2, 1) = rhoCp*F*(Tf-T) + delH*V*r - UA*(T - Tj);
L(3, 1) = rhojCpj*Fj*(Tj0 - Tj) + UA*(T - Tj);
end
```

RESULTS:

Method	Steady State 1			Steady State 2			Steady State 3		
	C_A (kgmol/m3)	$T(K)$	$T_j(K)$	C_A (kgmol/m3)	$T(K)$	$T_j(K)$	C_A (kgmol/m3)	$T(K)$	$T_j(K)$
Newton Raphson	8.49341	313.66	300.611	7.2184	326.927	302.821	1.19437	389.57	313.26
fsolve Method	8.49341	313.66	300.611	7.2184	326.927	302.821	1.19437	389.57	313.26

CONCLUSION:

- For multivariable the equation becomes:

$$F(x^{i+1}) = F(x^{i}) - J(x^{i})^{-1}F(x^{i})$$

Where J(x) is Jacobian which is given as:

$$\begin{vmatrix} f_x & f_y & f_z \\ g_x & g_y & g_z \\ h_x & h_y & h_z \end{vmatrix}$$

- The function must be continuous and differential everywhere so that newton Raphson can be applied.
- If the guess is poor, convergence might take more number of iterations or might diverge.
- For Newton Raphson, we need only one initial guess for calculation. So, it has a very high convergence method when compared to other methods like secant, false position, and bisection methods.
- Both fsolve and Newton Raphson give approximately same results with minimum error.
- There might be error due to computation i.e round off error.