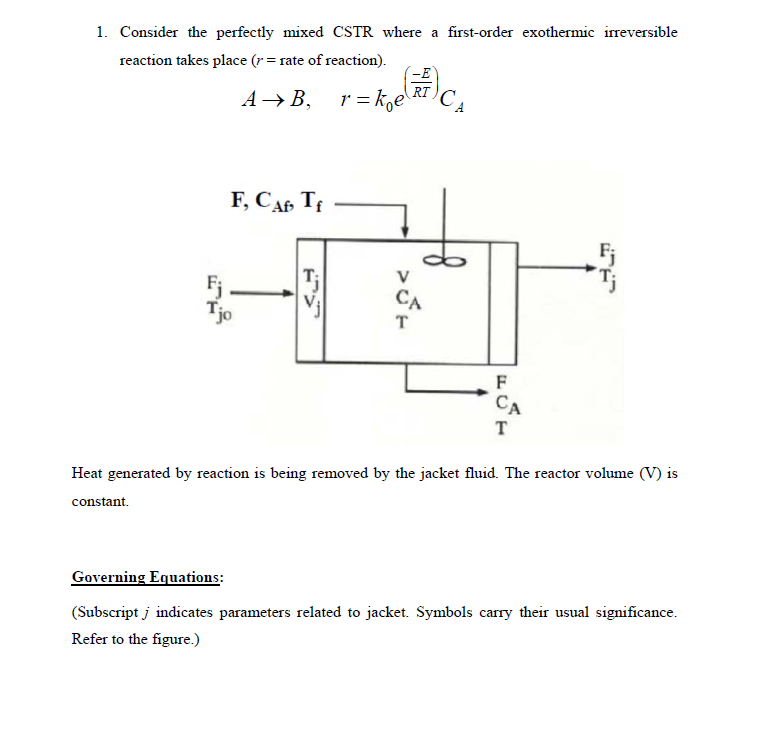
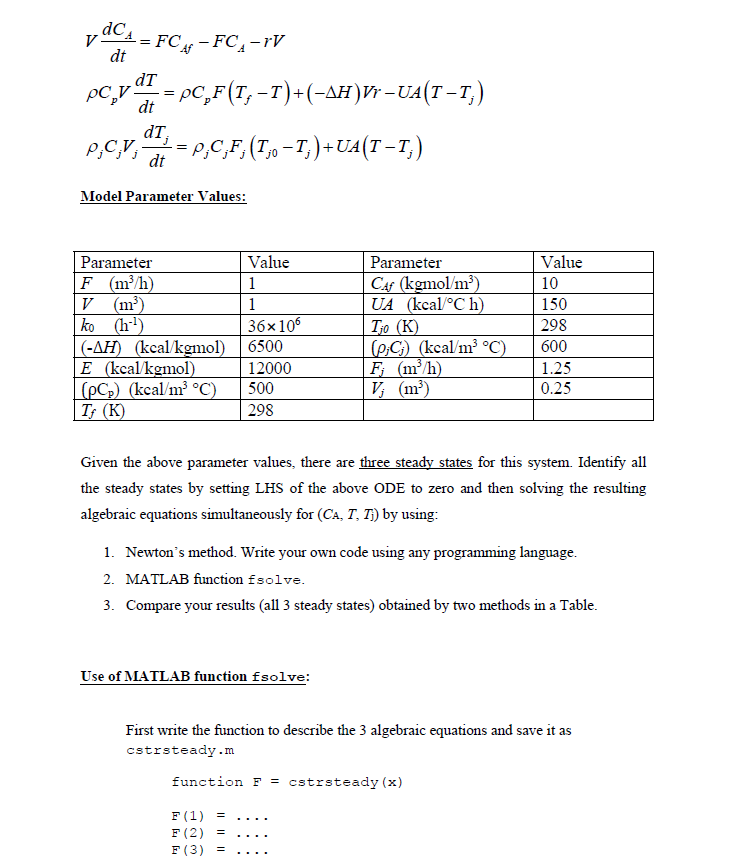
**CAPE Laboratory Assignment-2**

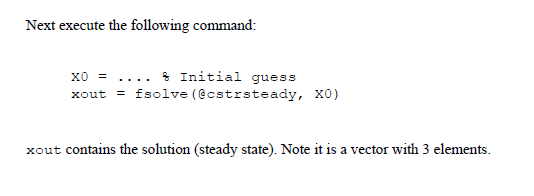
Instructor: Dr Debashis Sarkar

*18CH10071, Anshuman Agrawal*

**Problem Statement**







**MATLAB Code**

**Multivariable Newton Raphson**

clear all;

global k0 Ea R;

k0 = 36\*10^6;

Ea = 12000;

R = 2;

Xi = [5; 300; 400]; %Initial Guess

for i= 1:50

[f, jacobian] = cstr(Xi);

Xsol = Xi - inv(jacobian)\*f; %Multivariable Newton Raphson Method statement

if(abs(Xi-Xsol)<=10^-4) %Tolerance Check

break;

end

Xi=Xsol;

end

function [Func,j\_val] = cstr(X) %Equations

global k0 Ea R;

F = 1;

Rho\_Cp= 500; % Model Parameter Values

UA=150;

Tj0=298;

Fj=1.25;

Tf = 298;

Rhoj\_Cpj=600;

V=1;

CAf=10;

deltaH = 6500;

CA = X(1);

T = X(2);

Tj = X(3);

r = k0 \* exp(-Ea/(R\*T))\*CA ;

Func(1, 1) = F\*CAf - F\*CA - r\*V ;

Func(2, 1) = Rho\_Cp\*F\*(Tf-T) + deltaH\*V\*r - UA\*(T - Tj);

Func(3, 1) = Rhoj\_Cpj\*Fj\*(Tj0 - Tj) + UA\*(T - Tj);

syms x y z ; %Jacobian calculate

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]);

j\_val = double(subs(J, [x y z], [X(1), X(2), X(3)]));

end

**Using *fsolve* function**

clear all;

global k0 Ea R;

k0 = 36\*10^6;

Ea = 12\*10^3;

R = 2;

Xi = [5; 350; 500]; %Initial Guess

xout = fsolve(@cstrsteady,Xi); % fsolve statement

function [Func] = cstrsteady(X) %Equations definition

global k0 Ea R;

F = 1;

Rho\_Cp= 500; % Model Parameter Values

UA=150;

Tj0=298;

Fj=1.25;

Tf = 298;

Rhoj\_Cpj=600;

V=1;

CAf=10;

deltaH = 6500;

CA = X(1);

T = X(2);

Tj = X(3);

r = k0 \* exp(-Ea/(R\*T))\*CA ;

Func(1, 1) = F\*CAf - F\*CA - r\*V ;

Func(2, 1) = Rho\_Cp\*F\*(Tf-T) + deltaH\*V\*r - UA\*(T - Tj);

Func(3, 1) = Rhoj\_Cpj\*Fj\*(Tj0 - Tj) + UA\*(T - Tj);

end

**Results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Method** | **Multivariable Newton Raphson** | | | ***fsolve* function** | | |
| **Variables** | **CA (kgmol/ m3)** | **T (K)** | **Tj (K)** | **CA (kgmol/ m3)** | **T (K)** | **Tj (K)** |
| **Steady State 1** | 8.4934 | 313.6684 | 300.6114 | 8.4934 | 313.6684 | 300.6114 |
| **Steady State 2** | 7.2185 | 326.9277 | 302.8213 | 7.2185 | 326.9277 | 302.8213 |
| **Steady State 3** | 1.1944 | 389.5785 | 313.2631 | 1.1944 | 389.5785 | 313.2631 |

**Conclusion**

For same initial guesses, *fsolve* function is seen to take less computational time as compared to the multivariable Newton-Raphson method when compared on the MATLAB profiler. However, accuracy wise both methods are observed to give same results. Care needs to be taken while defining the initial guesses because sometimes they can lead the method to diverge or even converge after a very large number of iterations, and both situations are unfavourable for us. It is to be noted that the multivariable Newton Raphson method will work only if our functions are continuous and differentiable everywhere. Since all the equations are related to one another due to presence of multiple variables, obtaining 3 steady states seems to be logical since, the system can attain steady state in multiple ways too.