**PSET 7 – ChE 310 – Assigned on 3.7.2019, Due MIDNIGHT on 3.13.19**

Submit as a single zip file to the course website. Acknowledge collaborations in the comments section.

**Problem 1 -** 11.15 from book – there is a mismatch in the flow rates and concentrations given in the text and in the figure. Let’s use the constants given in the figure to solve the problem. Include a scanned image of part (a) and (b) work or write it all out in the comments section. For part (c) use the LU matrices to find the inverse and check your answer with the build in *inv* function.

**Code:**

% Coded by Nigel F. Reuel on 10.27.16

% This code solves problem 11.15 from the book

%

% See attached image for part (a) and (b)

% Part (c)

A = [132 -22 0;

-5 27 -7;

117 0 -132];

[L,U] = lu(A);

% Part (d) From slide 6 of lecture 19 we see the process of using the LU matrices to

% find the inverse matrix

% Find first column

b = [1; 0; 0];

d = L\b;

x1 = U\d;

% Find second column

b = [0; 1; 0];

d = L\b;

x2 = U\d;

% Find third column

b = [0; 0; 1];

d = L\b;

x3 = U\d;

% Combine for AI matrix

AI = [x1 x2 x3]

% Check with built in function

AI\_check = inv(A)

% Part (e)

b = [1000; 2000; 0]

Cvec = AI\*b;

disp('Part (i) - the steady state concentrations:')

c1 = Cvec(1)

c2 = Cvec(2)

c3 = Cvec(3)

disp('Part (ii) This is solved using the inverse matrix - first row, second column entry.')

DeltaConc = AI(1,2)\*2000;

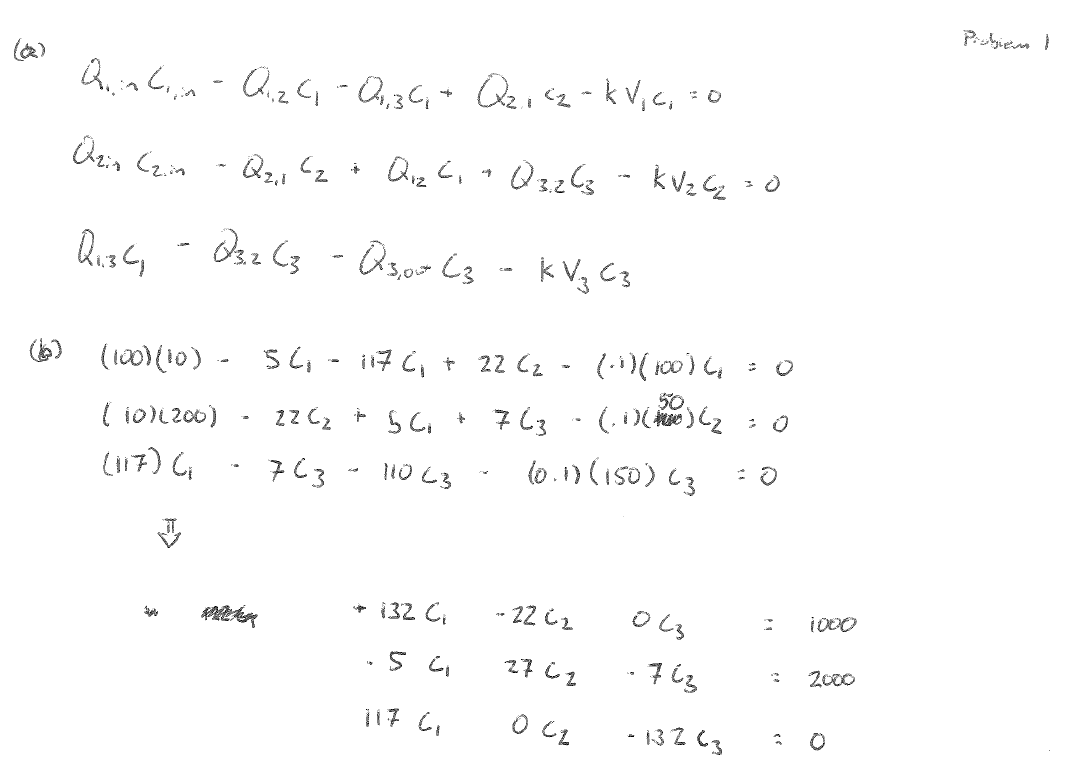
fprintf('Concentration of reactor 1 is reduced by %4.2fg/m^3\n',DeltaConc)

disp('Part (iii) This is solved using the inverse matrix - third row, first and second entries.')

DeltaConc2 = AI(3,1)\*1000-AI(3,2)\*1000;

fprintf('Concentration of reactor 3 is increased by %4.2fg/m^3\n',DeltaConc2)

**Answer:**



**(C)**

Part (i) - the steady state concentrations:

c1 = 21.4017

c2 = 82.9554

c3 = 18.9697

Part (ii) This is solved using the inverse matrix - first row, second column entry.

Concentration of reactor 1 is reduced by 13.26g/m^3

Part (iii) This is solved using the inverse matrix - third row, first and second entries.

Concentration of reactor 3 is increased by 1.34g/m^3

**Problem 2 -** 11.18 from book – Part (a) show work by including a scanned/photo image of your paper work. Check your answer for part c by making the proposed change to your system of equations and then solving again. Feel free to use built in Matlab functions to find inverted matrix for this problem.

**Code:**

% Coded by Nigel F. Reuel on 10.27.16

% This code solves 11.18 from the book.

%

% Part (a) done by hand, see attached scan image

% Part (b)

A = [200 0 -50 0;

-50 150 -50 0;

150 150 -390 90;

0 0 -240 240];

AI = inv(A);

b = [150; 2000; 0; 5000];

x = AI\*b;

disp('Steady state concentrations in each of the rooms (units = mg/m^3):')

C1 = x(1)

C2 = x(2)

C3 = x(3)

C4 = x(4)

%Part (c)

disp('Use the second row of the inverse matrix (room 2) and fourth column (load from room four) to solve:')

D = C2-20;

DeltaLoad = D/AI(2,4);

fprintf('You must reduce the room 4 load by %4.2fmg/hr to maintain 20mg/m^3 in room 2.',DeltaLoad)

% Let's check this answer

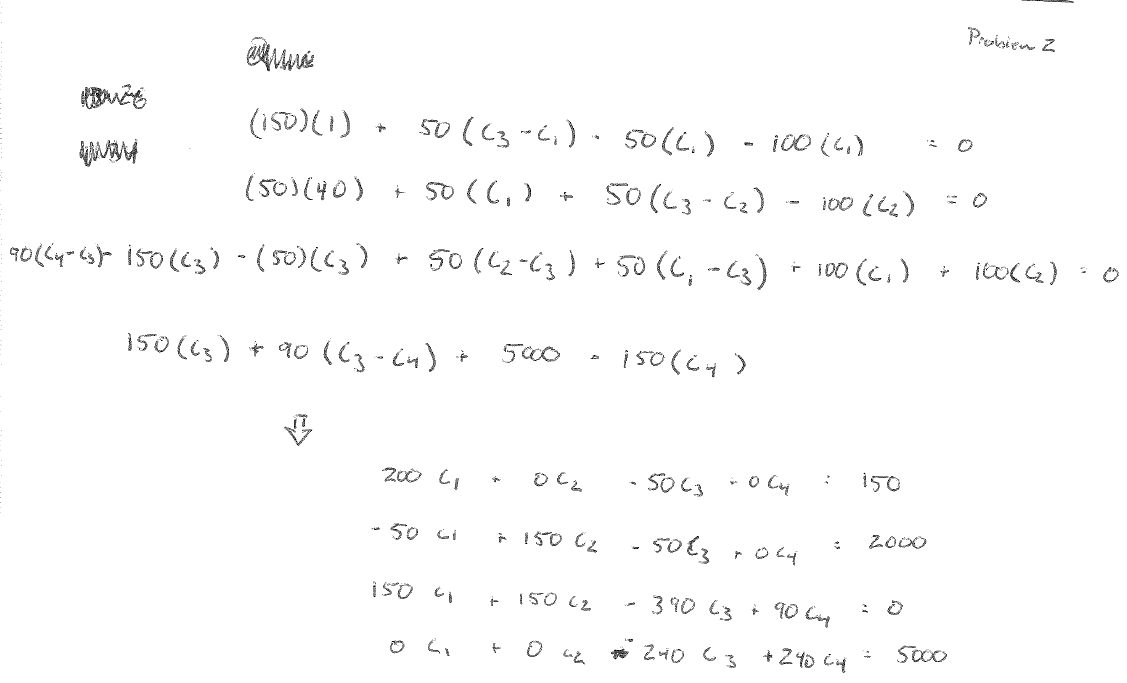
b2 = [150; 2000; 0; 5000-DeltaLoad];

x2 = AI\*b2;

CheckAns = x2(2)

disp('Looks, like the concentration is maintained at 20mg/m^3 with this change.')

**Answer:**



**(b)**

Steady state concentrations in each of the rooms (units = mg/m^3):

C1 = 5.7813

C2 = 21.9688

C3 = 20.1250

C4 = 40.9583

Use the second row of the inverse matrix (room 2) and fourth column (load from room four) to solve:

You must reduce the room 4 load by 2520.00mg/hr to maintain 20mg/m^3 in room 2.

CheckAns = 20

Looks, like the concentration is maintained at 20mg/m^3 with this change.

**Problem 3 –** Practice iterative methods

**Part 1 -** 12.5 from book – compare your Gauss Seidel solution to another method we have learned for systems of linear equations.

**Part 2 -** 12.9 from book.

**Code Part 1:**

% Coded by Nigel F. Reuel on 10.27.2016

% This code solves problem 12.5 from the textbook

%

% NOTE: the equation is in diagonal dominant form, so no need to rearrange

% the equations.

A = [15 -3 -1;

-3 18 -6;

-4 -1 12];

b = [3800; 1200; 2350];

es = 0.05

x = GaussSeidel(A,b,es);

disp('The concentrations using the Gauss-Seidel method are (in g/m^3):')

c1 = x(1)

c2 = x(2)

c3 = x(3)

% Verify using '\'

x = A\b;

disp('The concentrations using \ function are (in g/m^3):')

c1 = x(1)

c2 = x(2)

c3 = x(3)

function x = GaussSeidel(A,b,es,maxit)

% GaussSeidel: Gauss Seidel method

% x = GaussSeidel(A,b): Gauss Seidel without relaxation

% input:

% A = coefficient matrix

% b = right hand side vector

% es = stop criterion (default = 0.00001%)

% maxit = max iterations (default = 50)

% output:

% x = solution vector

if nargin<2,error('at least 2 input arguments required'),end

if nargin<4|isempty(maxit),maxit=50;end

if nargin<3|isempty(es),es=0.00001;end

[m,n] = size(A);

if m~=n, error('Matrix A must be square'); end

C = A;

for i = 1:n

C(i,i) = 0;

x(i) = 0;

end

x = x';

for i = 1:n

C(i,1:n) = C(i,1:n)/A(i,i);

end

for i = 1:n

d(i) = b(i)/A(i,i);

end

iter = 0;

while (1)

xold = x;

for i = 1:n

x(i) = d(i)-C(i,:)\*x;

if x(i) ~= 0

ea(i) = abs((x(i) - xold(i))/x(i)) \* 100;

end

end

iter = iter+1;

if max(ea)<=es | iter >= maxit, break, end

end

end

**Answer Part 1:**

The concentrations using the Gauss-Seidel method are (in g/m^3):

c1 = 320.2010

c2 = 227.1967

c3 = 321.5001

The concentrations using \ function are (in g/m^3):

c1 = 320.2073

c2 = 227.2021

c3 = 321.5026

**Code Part 2:**

% Coded by Nigel F. Reuel on 10.27.16

% This code solves for problem 12.9 from the textbook

%

% Part (a) graphical solution

F1 = @(x) (5-x^2)^(1/2);

F2 = @(x) x^2-1;

%

fplot(F1,[0 2]);

hold on

fplot(F2,[0 2]);

hold off

disp('From inspection of plot, the intersection is at x = 1.6 and y = 1.56')

% Part (b)

% Let's iterate 10 times and see how it does

count = 0

x = 1.5;

%y = 1.5;

while count < 10

y = (5-x^2)^(1/2);

x = (y+1)^(1/2);

count = count + 1;

end

disp('Solution found by substituition method:')

x

y

% Part (c) Newton Raphson Method

x0 = [1.5; 1.5];

[X,~,~,~]=newtmult(@NewtFunc,x0,.0001,100);

disp('Solution from Newton Raphson method:')

x = X(1)

y = X(2)

function [J,f] = NewtFunc(X)

%Define the jacobian and function value vector for this problem

x = X(1);

y = X(2);

J = [-2\*x -2\*y; 2\*x -1];

f = [(5-y^2-x^2);(x^2-y-1)];

end

function [x,f,ea,iter]=newtmult(func,x0,es,maxit,varargin)

% newtmult: Newton-Raphson root zeroes nonlinear systems

% [x,f,ea,iter]=newtmult(func,x0,es,maxit,p1,p2,...):

% uses the Newton-Raphson method to find the roots of

% a system of nonlinear equations

% input:

% func = name of function that returns f and J

% x0 = initial guess

% es = desired percent relative error (default = 0.0001%)

% maxit = maximum allowable iterations (default = 50)

% p1,p2,... = additional parameters used by function

% output:

% x = vector of roots

% f = vector of functions evaluated at roots

% ea = approximate percent relative error (%)

% iter = number of iterations

if nargin<2,error('at least 2 input arguments required'),end

if nargin<3|isempty(es),es=0.0001;end

if nargin<4|isempty(maxit),maxit=50;end

iter = 0;

x=x0;

while (1)

[J,f]=func(x,varargin{:});

dx=J\f;

x=x-dx;

iter = iter + 1;

ea=100\*max(abs(dx./x));

if iter>=maxit|ea<=es, break, end

end

end

**Answer Part 2:**

**(a)**

From inspection of plot, the intersection is at x = 1.6 and y = 1.56



**(b)**

Solution found by substitution method:

x = 1.6005

y = 1.5615

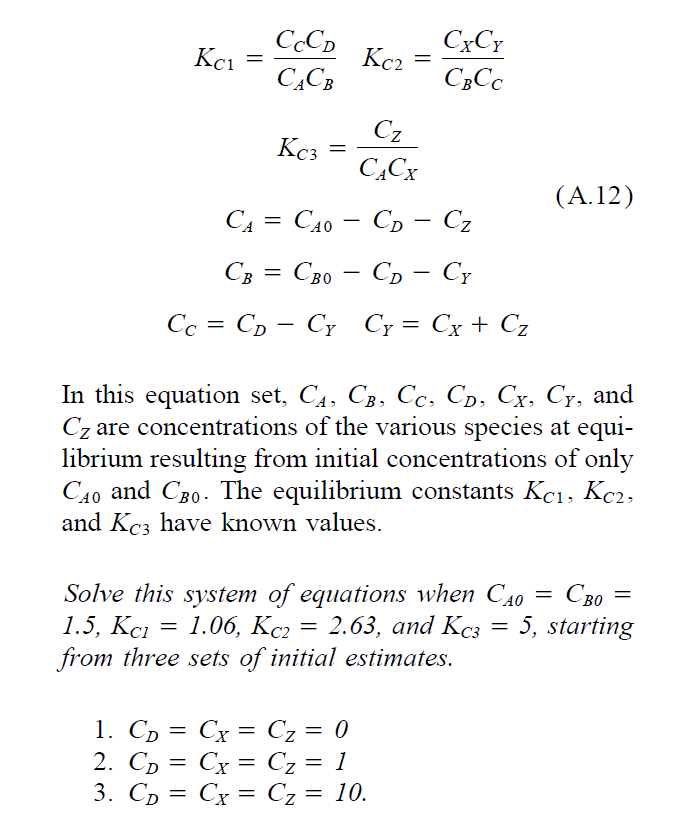
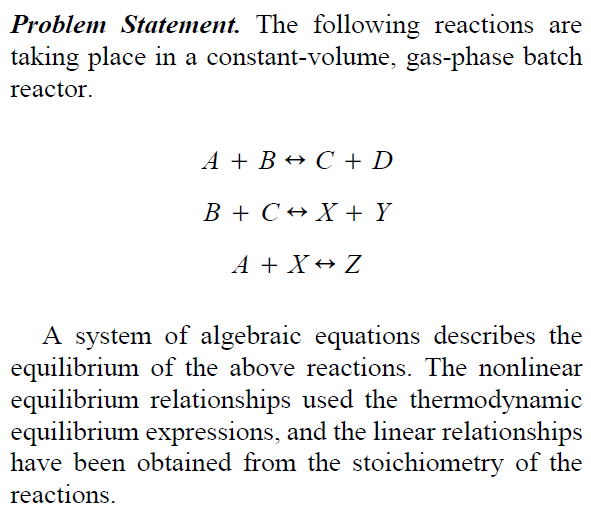
**(c)**

Solution from Newton Raphson method:

x = 1.6005

y = 1.5616

**Problem 4 -** Solve the following system of non-linear equations. Use MATLAB to solve. Also specify which set of initial conditions below gives you realistic answers. (note: I used initial condition of 1 for the other variables that are not specified below).



**Code:**

% Coded by Nigel F. Reuel on 10.14.2016

% This code solves the sixth problem on the homework set

%

% Define inital guess of variables that are to be optimized

Ca\_o = 1;

Cb\_o = 1;

Cc\_o = 1;

Cy\_o = 1;

% These are the initial conditions they want us to vary

Cd\_o = 1;

Cx\_o = 1;

Cz\_o = 1;

% Place the initial conditions in a single vector to work in fminsearch

x0 = [Ca\_o Cb\_o Cc\_o Cy\_o Cd\_o Cx\_o Cz\_o];

% Define parameters that do not change:

CAO = 1.5;

CBO = 1.5;

KC1 = 1.06;

KC2 = 2.63;

KC3 = 5;

% Place parameters into a single vector to pass to function

param = [CAO CBO KC1 KC2 KC3];

% Run fminsearch. Note, that when it is called on a subfunction you need

% to use the @ symbol!

Avec = fminsearch(@GPR,x0,[],param);

% Print solution to the screen

disp('I used the initial guess values of Cd = Cx = Cz = 1. This provided results that are non zero and positive')

disp('The unknown concentrations are as follows:')

Ca = Avec(1)

Cb = Avec(2)

Cc = Avec(3)

Cy = Avec(4)

Cd = Avec(5)

Cx = Avec(6)

Cz = Avec(7)

function OUT = GPR(x0,param)

% Let's pull out the variables from the two vectors that are passed in:

Ca = x0(1);

Cb = x0(2);

Cc = x0(3);

Cy = x0(4);

Cd = x0(5);

Cx = x0(6);

Cz = x0(7);

CAO = param(1);

CBO = param(2);

KC1 = param(3);

KC2 = param(4);

KC3 = param(5);

% Now write the seven linear and non-linear equations, such that the

% equality is trying to be forced to zero

Z1 = Cc\*Cd/(Ca\*Cb)-KC1;

Z2 = Cx\*Cy/(Cb\*Cc)-KC2;

Z3 = Cz/(Ca\*Cx)-KC3;

Z4 = CAO-Cd-Cz-Ca;

Z5 = CBO-Cd-Cy-Cb;

Z6 = Cd-Cy-Cc;

Z7 = Cx+Cz-Cy;

% Now combine the 'errors' as the output of the function (this is what we

% want matlab to minimize!)

OUT = sqrt(Z1^2+Z2^2+Z3^2+Z4^2+Z5^2+Z6^2+Z7^2);

end

**Answer:**

The unknown concentrations are as follows:

Ca = 0.4919

Cb = 0.2062

Cc = 0.1548

Cy = 0.5491

Cd = 0.6898

Cx = 0.1528

Cz = 0.3756

**GROUP POINTS –** Demonstrate collaboration on slack.