clear;clc

format compact

%% Read or Input any square Matrix

% Parameters

%Parameter to define the range in space and time

L=1; %Length of wire

%Parameter needed to solve the equation within the fully implicit method

m=10; %Number of time steps

n=30; %Number of space steps

D = 0.0013; %Diffusion constant per day

p = 0.012; %growth constant per day

g=0.024;

dx=L/n; %delta x

dt=1/m; %delta t

alpha = ((p-g)\*L^2/D)\*dt;

beta = dt/dx^2;

% Making the Matrix

a = 1+2\*beta/1+alpha; %Diagonal values

b = -(beta/1+alpha); %Off-diagonal values

%Sparce Matrix

A = sparse(n-1,n-1);

A(1,1)=a-b;

for i=2:n

A(i,i-1) = b;

A(i,i) = a;

A(i-1,i)=b;

A(30,30)=a-b;

end

T=full(A);

% coefficients matrix

C=[1:1:30];

C=C';

C(:,:)=4000; % constants vector

n = length(C);

X = zeros(n,1);

Error\_eval = ones(n,1);

lambda=1.9;

%% Start the Iterative method

for k=0:1:m

iteration = 0;

while max(Error\_eval) > 0.000001

iteration = iteration + 1;

Z = X; % save current values to calculate error later

for i = 1:n

I = [1:i-1,i+1:n];

Xtemp = X; % copy the unknows to a new variable

Xtemp(i) = []; % eliminate the unknown under question from the set of values

X(i) = (1-lambda)\*X(i)+lambda/A(i,i)\*(C(i) - sum(A(i,I) \* Xtemp));

end

Xsolution(:,iteration) = X;

Error\_eval = sqrt((X - Z).^2);

end

C(:,1)=Xsolution(:,end);

Error\_eval = ones(n,1);

end

time = cputime;

%% Display Results

MaTrIx =full([Xsolution]);

plot(Xsolution)