```
clear all; close all; clc;

% Dustin (Ting-Hsuan) Ma
% ME 2055 CFD - Homework 2 1D Heat Conduction
% Dr. Peyman Givi
% Due: Feb 26, 2019

%% Exact Solution
% Problem Statement
LX = 1;         %length of X
T = @(x) x + 1; %Exact solution

% User defined parameters
num = 10;       %number of elements in X direction
Dx = LX/(num-1);  %step size in x
Dt = 0.006;     %step size in t

% Plotting Exact Solution
figure();
x = linspace(0,LX,num);
plot(x,T(x),'ks-','LineWidth',2)
xlabel('X'); ylabel('Temperature'); title('Steady State Solution')

%% Explicit (FTCS)

% Checking the CFL value
CFL = Dt/Dx^2;
fprintf('CFL = %0.3f\n',CFL)

% Memory Allocation
U = zeros(1,num+1);
Unew = zeros(1,num+1);
relU = zeros(1,2);

% Initialization/ applying BC
U = boundaryConditionD(U);

% Solve for Steady State
iter = 0;
eps = 1e-8;
for t = 1:Dt:100
    for i = 2:(numel(U)-1)
        Unew(i) =  (U(i-1)-2*U(i)+U(i+1))*CFL + U(i);
    end

    Unew = boundaryConditionD(Unew); %Applying BC Dirichlet
    Unew = boundaryConditionNE(Unew, Dx); % Applying BC Neumann
    U = Unew;   % Array swapping

    % Calculating residual/ Stopping criteria
    relU(1) = relU(2);
    relU(2) = L2RelNorm(U);
    residual = relU(2) - relU(1);
    if residual <= eps
        break;
    else
        iter=iter+1;
    end
end
err = percentErr(Unew(1:end-1),T(x));

hold on
plot(x,Unew(1:end-1),'ro--','LineWidth',2)
hold off

fprintf('Explicit Method outputs with an overall of %2.2e percent error\n',err)
fprintf('The iteration after %d iterations finished with a residual of %2.2e\n',iter,residual)

%% Implicit (TDMA)

% Memory Allocation
U = zeros(num,1);
Unew = zeros(num,1);
a = zeros(1,num);
b = zeros(1,num);
c = zeros(1,num);
relU = zeros(1,2);

% Initialization/ applying BC
U = boundaryConditionDI(U,Dt,Dx);

A = zeros(num,num);
for i=2:num-1
    A(i,i-1) = -CFL;
    A(i,i) = 1+2*CFL;
    A(i,i+1) = -CFL;
```



```
end
A(1,1) = 1;
A(num,num) = 1;

iter = 0;
for count = 1:Dt:100
    U = boundaryConditionDI(U,Dt,Dx); %Applying BC Dirichlet
    U = boundaryConditionNI(U,Dx);
    Unew = A\U;
    U = Unew;   % Array swapping

    % Calculating residual/ Stopping criteria
    relU(1) = relU(2);
    relU(2) = L2RelNorm(U);
    residual = relU(2) - relU(1);
    if residual <= eps
        break;
    end
    iter = iter + 1;
end
err = percentErr(Unew,T(x));
fprintf('Implicit Method outputs with an overall of %2.2e percent error\n',err)
fprintf('The iteration after %d iterations finished with a residual of %2.2e\n',iter,residual)

hold on
plot(x,Unew,'b*:','LineWidth',2)
hold off
legend('Exact','Explicite','Implicit','Location','NorthWest')

movegui('northeast')

%% Functions

% Explicit Boundaries
function [U] = boundaryConditionD(U)
U(1) = 1;
end
function [U] = boundaryConditionNE(U,Dx)
U(end) = 2*Dx + U(end-2);   % Central Difference
end

% Implicit Boundaries
function [U] = boundaryConditionDI(U,Dt,Dx)
U(1) = 1;
U(end) = U(end)+Dt/Dx;
end
function [U] = boundaryConditionNI(U,Dx)
U(end) = Dx + U(end-1);   % Central Difference
end

% Calculating the L2 of Matrix
function [rel] = L2RelNorm(U)
part = 0;
for i = 1:numel(U)-1
    part = part + U(i).^2;
end
rel = sqrt(part);
end

function [err] = percentErr(numerical,exact)
err = zeros(1,numel(exact));
for i = 1:numel(exact)
    err(i) = (exact(i)-numerical(i))/exact(i) * 100;
end
err = sum(err)/numel(err);
end
```