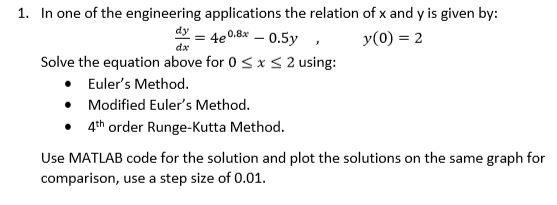
Assignment #1

ME135-02L-Spring2018

Feb/5/2018

By Yeash Patel

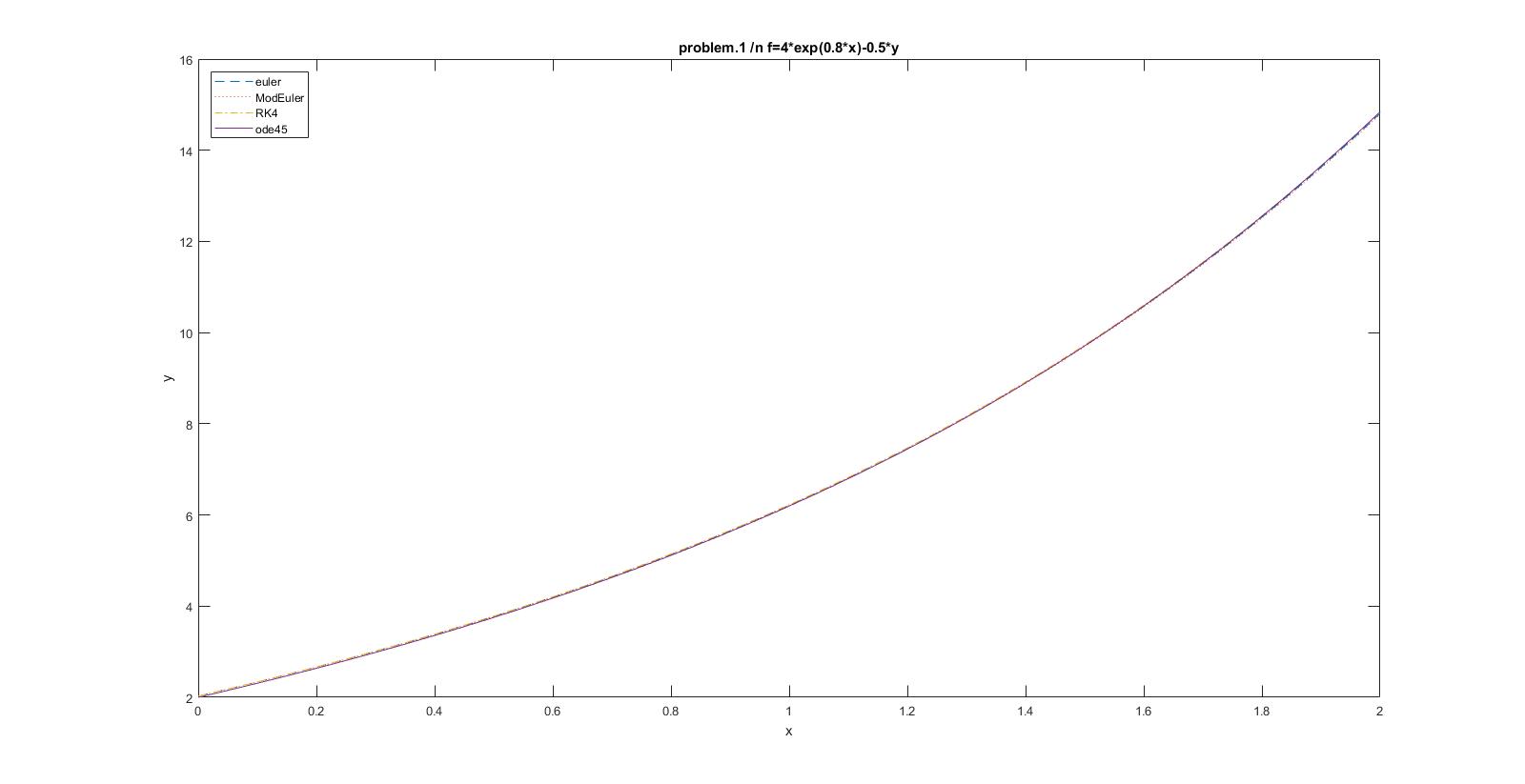
**Problem 1.**

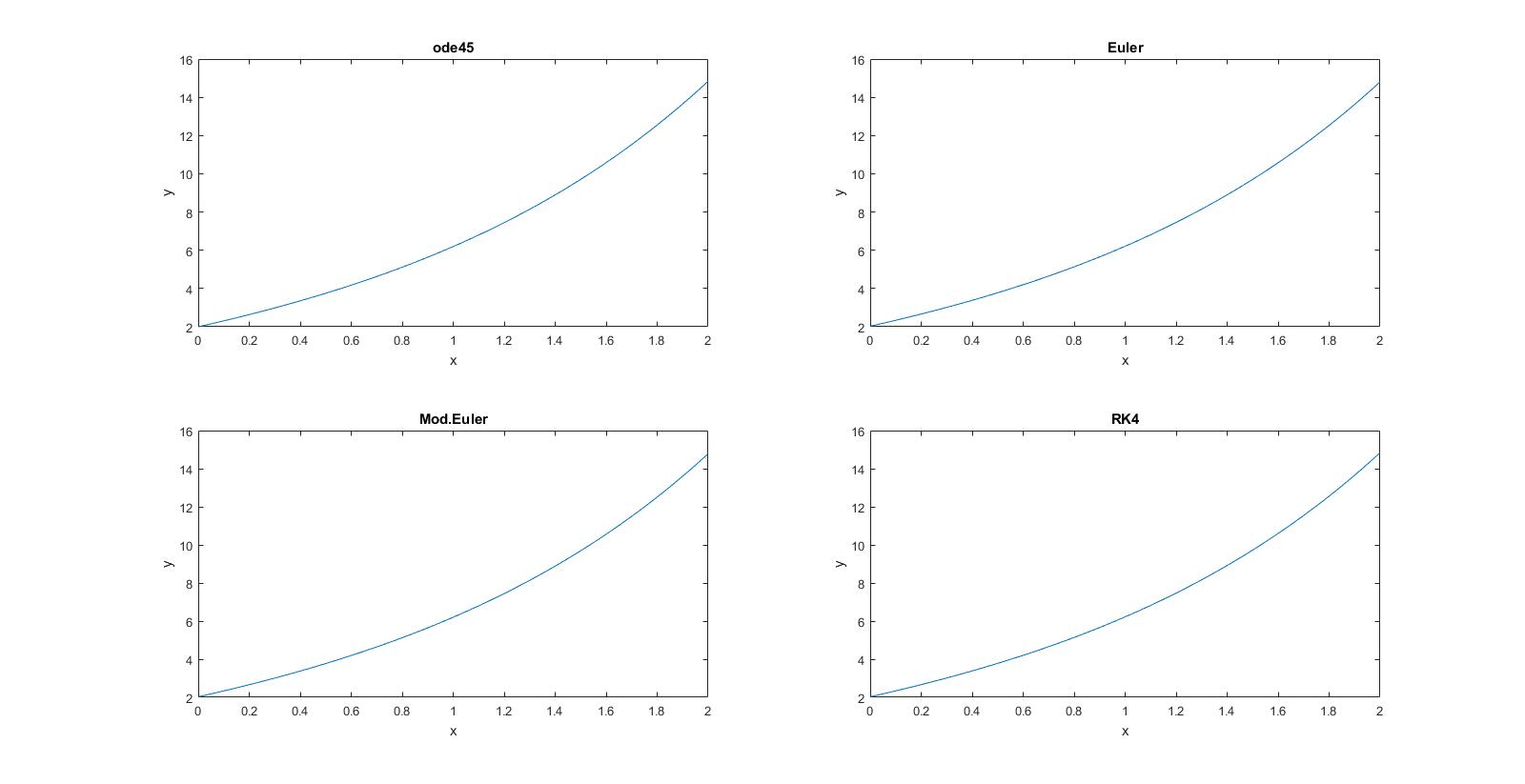
****

To solve this problem Euler’s, Modified Euler’s, 4th order Runge-Kutta method, and ode45 were used. To get a better comparison between the methods I created two graphs one with all the methods compiled on a single graph and another with each separate. The code used for this problem is available in the appendix A and on git hub [https://github.com/Yeash96/Eng135.FEA/tree/master/Assigment\_1].

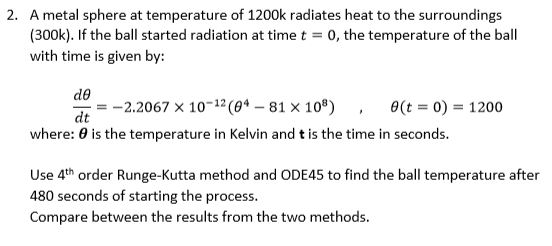
**Matlab input and output**

>>Pr1





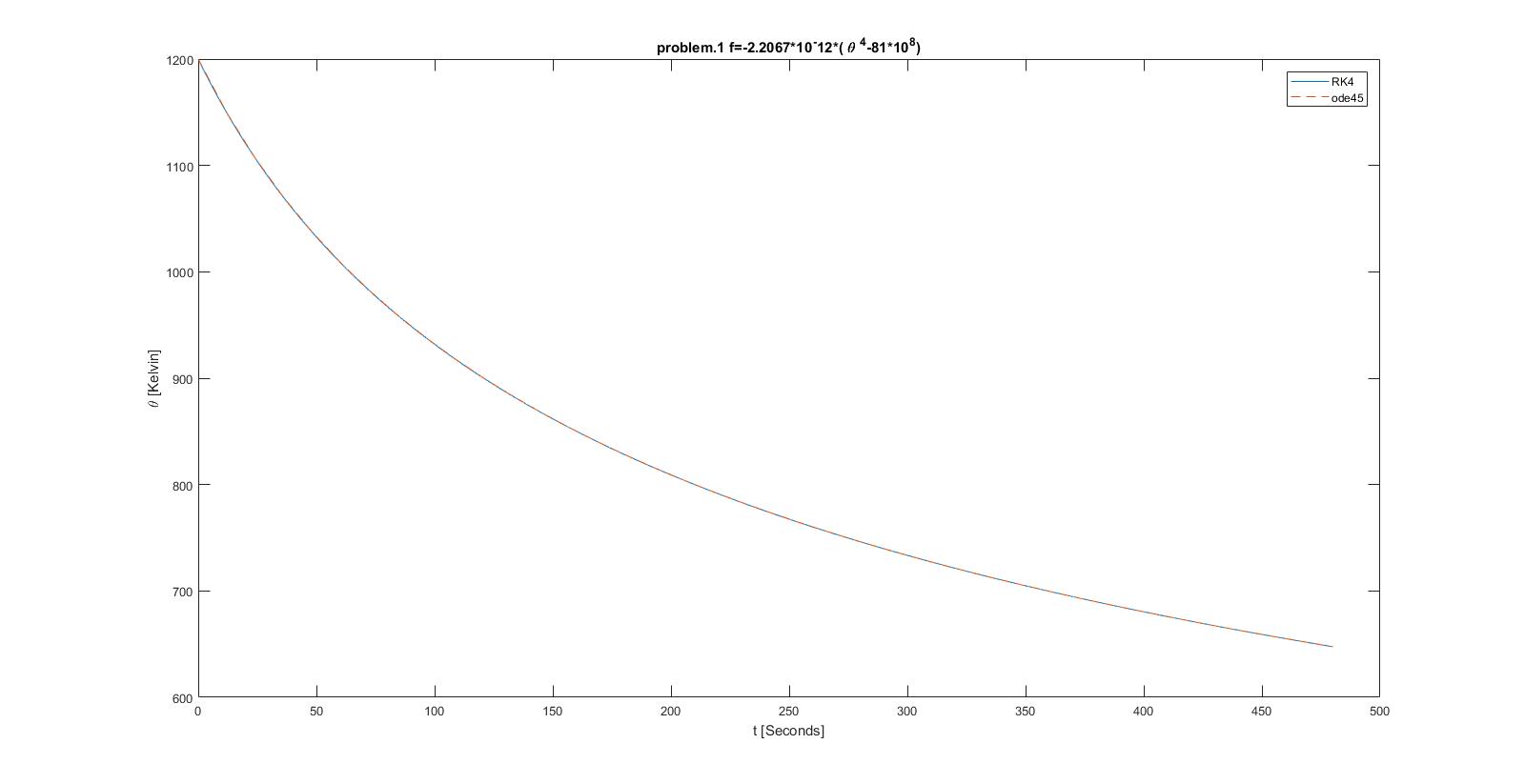
**Problem 2.**

****

To solve this problem 4th order Runge-Kutta (RK4) method and ode45 were used. By comparing the two we find that they the RK4 method to be very accurate. I did used 48000 iterations with a step size of 0.001 which must have greatly aided in its accuracy. The code used for this problem is available in the appendix A and on git hub [https://github.com/Yeash96/Eng135.FEA/tree/master/Assigment\_1].

**Matlab input and output**

>>Pr2



Appendix A

Pr1.m

clc

clear all

close all

%given stuff

f=@(x,y)4\*exp(0.8\*x)-0.5\*y; %function

alpha=2;%initial value

a=0;%start point

b=2;%end point

N=200;%number of iteration N= (b-a)/h h is the step size

h=0.01; %step value

x=linspace(0,2,200);%x values,

Eurz=Euler(f,a,b,alpha,N,h); %fuction returns 1x200 array using euler method

MEZ=ModEuler(f,a,b,alpha,N,h); %fuction returns 1x200 array using mod.euler method

RK4z=RK4(f,a,b,alpha,N,h); %fuction returns 1x200 array using runga-kutta 4 method

[t,y]=ode45(f,[0,2],2); %matlabs ode function solver returns t=45x1, y=45x1 array

%graphing stuff

figure(1)

plot(x,Eurz,'--',x,MEZ,':',x,RK4z,'-.',t,y);

legend('euler','ModEuler','RK4','ode45','Location','NorthWest')

xlabel('x')

ylabel('y')

title('problem.1 /n f=4\*exp(0.8\*x)-0.5\*y')

figure(2)

subplot(2,2,1)

plot(t,y)

title('ode45')

xlabel('x')

ylabel('y')

subplot(2,2,2)

plot(x,Eurz)

title('Euler')

xlabel('x')

ylabel('y')

subplot(2,2,3)

plot(x,MEZ)

title('Mod.Euler')

xlabel('x')

ylabel('y')

subplot(2,2,4)

plot(x,RK4z)

title('RK4')

xlabel('x')

ylabel('y')

Pr2.m

clc

clear all

close all

%given stuff

f=@(t,theta)-2.2067\*10^(-12)\*(theta^4-81\*10^8);%function

alpha=1200; % initial value

a=0; %start point

b=480; % end point

N=48000; % number of iteration N= (b-a)/h

h=0.01; % step size

x=linspace(0,480,48000); % x values

RK4z=RK4(f,a,b,alpha,N,h); % Runga-Kutta 4 method returns 1x48000

[t,y]=ode45(f,[0,480],1200); %matlabs ode function solver returns t=41x1, y=41x1 array

% graph stuff

plot(x,RK4z,t,y,'--');

legend('RK4','ode45')

xlabel('t [Seconds]')

ylabel('\theta [Kelvin]')

title('problem.1 f=-2.2067\*10^-12\*( \theta ^4-81\*10^8)')

Euler.m

function [E] = Euler( f,a,b,alpha,N,h )

t=a; % start point labeled as t

w=alpha; % initial value is now w

for i=(1:N)

w=w+h\*f(t,w); %eulers method formula

E(i)=w; %records values to return

t=a+i\*h; % next step

end

end

ModEuler

function [ E ] = ModEuler(f,a,b,alpha,N,h)

t=a; % start point labeled as t

w=alpha; % initial value is now w

for i=(1:N)

z=w+h\*f(t,w); % inital eulers method to find y\*i+1

y=w+h\*((f(t,w)+f(t,z))/2);% final solution to find real yi+1

E(i)=y;% records solutions to return

w=y; % re assign vaules for next loop

t=a+i\*h; % step foward

end

end

RK4.m

function [ E ] = RK4(f,a,b,alpha,N,h)

t=a; % start point labeled as t

w=alpha; % initial value is now w

for i=(1:N);

%runga kutta 4 method

k1=h\*f(t,w); % solving k(n) constants

k2=h\*f(t+h/2,w+k1/2);

k3=h\*f(t+h/2,w+k2/2);

k4=h\*f(t+h,w+k3);

w= w+1/6\*(k1+2\*k2+2\*k3+k4); % putting it all together

E(i)=w; % record solution to return

t=a+i\*h; % step foward

end

end