ME 594 – Numerical Methods – HW08

Viral Panchal | Due Date: 11/16

“I pledge my honor that I have abided by the Stevens Honor system”

Text

Description automatically generated

**Solution:**

Letter

Description automatically generated

**Matlab program:**

* **Explicit Euler function**

% Explicit Euler function

function f = Explicit\_euler(x\_k,y\_k,h)

f = (1-2\*x\_k\*h)\*y\_k;

end

* **Implicit Euler function**

% Implicit Euler function

function f = Implicit\_euler(x\_k\_p1,y\_k,h)

f = y\_k/(1+2\*x\_k\_p1\*h);

end

* **Driver for Q1**

% Q1 driver  
% This driver runs both Explicit and Implcit Euler method outputs plot for  
% both individually.  
clear all  
close all  
clc  
% for Explicit Euler method  
  
x\_0 = 0;  
y\_0 = 1;  
x\_n = 6;  
  
for i = 0:3  
 h = 0.4/2^i;  
 n=(x\_n-x\_0)/h;  
 x = zeros(1,n+1);  
 y = zeros(1,n+1);  
  
 x = x\_0:h:x\_n;  
 y(1) = y\_0;  
 for k = 1:n  
 y(k+1) = feval('Explicit\_euler',x(k),y(k),h);  
 end  
  
 plot(x,y,'color',rand(1,3))  
  
 hold on  
  
end  
  
y\_exact = zeros(1,n+1);  
y\_exact(1) = y\_0;  
for k = 2:n+1  
 y\_exact(k) = exp(-x(k).^2);  
end  
  
plot(x,y\_exact)  
title('Explicit error')  
axis padded  
grid on  
xlabel('x')  
ylabel('y')  
legend('h = 0.4','h =0.2','h=0.1','h=0.5','Exact')  
  
  
figure  
  
% For Implicit Euler method  
% Redefing the initial variables to override the magnitudes of certain  
% variables from the previous metthod.  
x\_0 = 0;  
y\_0 = 1;  
x\_n = 6;  
  
for i = 0:3  
 h = 0.4/2^i;  
 n=(x\_n-x\_0)/h;  
 x = zeros(1,n+1);  
 y = zeros(1,n+1);  
  
 x = x\_0:h:x\_n;  
 y(1) = y\_0;  
 for k = 1:n  
 y(k+1) = feval('Implicit\_euler',x(k+1),y(k),h);  
 end  
  
 plot(x,y,'color',rand(1,3))  
  
 hold on  
  
end  
  
y\_exact = zeros(1,n+1);  
y\_exact(1) = y\_0;  
for k = 2:n+1  
 y\_exact(k) = exp(-x(k).^2);  
end  
  
plot(x,y\_exact)  
title('Implicit error')  
axis padded  
grid on  
xlabel('x')  
ylabel('y')  
legend('h = 0.4','h =0.2','h=0.1','h=0.5','Exact')

**Matlab output:**

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

[*Published with MATLAB® R2021a*](https://www.mathworks.com/products/matlab)

**Text

Description automatically generated**

**Solution:**

**A piece of paper with writing on it

Description automatically generated with medium confidence**

**Matlab program:**

* ***Script for G as shown in manual work***

% G for newton solver

function f = G\_NS(z,h,y\_k,x\_k\_p1)

f = h\*x\_k\_p1\*z^3+z-y\_k;

end

* ***Script for GZ as shown in the above work***

% GP for Newton solver

function f = GZ\_NS(z,x\_k\_p1,h)

f = 3\*x\_k\_p1\*z^2+1;

end

* ***Newton function***

function z = Newton\_z(h,y\_k,x\_k\_p1)

tolerance = h^2/4;

z = y\_k;

max\_1 = 1000;

for j = 1:max\_1

p = z-feval('G\_NS',z,h,y\_k,x\_k\_p1)/feval('GZ\_NS',z,x\_k\_p1,h);

error = abs(p-z);

rel\_error = 2\*error/(abs(p)+abs(z));

f = feval('G\_NS',p,h,y\_k,x\_k\_p1);

z = p;

if(error<tolerance)||(rel\_error<tolerance)||(abs(f)<tolerance)

break

end

end

* ***Driver for Q2***

% Q2 Driver  
clear all  
close all  
clc  
  
x\_0 = 1;  
y\_0 = 1;  
x\_n = 20;  
  
for i = 1:3  
 h = 1/2^i;  
 n = (x\_n-x\_0)/h;  
  
 x = zeros(1,n+1);  
 y = zeros(1,n+1);  
  
 x = x\_0:h:x\_n;  
 y(1) = y\_0;  
 for k = 1:n  
 z = y(k);  
 y(k+1) = Newton\_z(h,z,x(k+1));  
 end  
 plot(x,y,'color',rand(1,3))  
 hold on  
end  
  
y\_exact = zeros(1,n+1);  
y\_exact(1) = y\_0;  
for k = 2:n+1  
 y\_exact(k) = (1+x(k).^2).^(-.5);  
end  
  
plot(x,y\_exact)  
title('Implicit Euler')  
axis padded  
grid on  
xlabel('x')  
ylabel('y')  
legend('h=0.5','h=0.25','h=0.125','Exact')

***Matlab output:***

Chart

Description automatically generated with medium confidence

[*Published with MATLAB® R2021a*](https://www.mathworks.com/products/matlab)

**Text

Description automatically generated**

**Solution:**

**A sheet of music

Description automatically generated with low confidence**

**MATLAB program:**

* ***Euler function script***

% Euler function

function f = Euler(x,y)

f = x-(x\*y)/2;

end

* ***Trapezoid method function script***

% Trapezoid function

function f = Trapezoid(x\_k,x\_k\_p1,y\_k,h)

f = (y\_k+h/2\*(x\_k+x\_k\_p1-x\_k\*y\_k/2))/(1+h/4\*x\_k\_p1);

end

* ***Driver for Q3***

% Driver Q3  
% This driver includes the program for both the Trapezoid and Euler method.  
% The first output figure is for the trapezoid method followed by the  
% modifiied euler method.  
clear all  
close all  
clc  
  
% For trapezoid method  
x\_0 = 1;  
y\_0 = 1;  
x\_n = 5;  
  
for i = 0:2  
  
 h = 0.8/2^i;  
 n = round((x\_n-x\_0)/h);  
 x = zeros(1,n+1);  
 y = zeros(1,n+1);  
  
 x = x\_0:h:x\_n;  
 y(1) = y\_0;  
 for k = 1:n  
 y(k+1) = feval('Trapezoid',x(k),x(k+1),y(k),h);  
 end  
 plot(x,y,'color',rand(1,3))  
 hold on  
end  
  
y\_exact = zeros(1,n+1);  
y\_exact(1) = y\_0;  
for k = 2:n+1  
 y\_exact(k) = 2-exp(((1-x(k).^2)/4));  
end  
  
plot(x,y\_exact)  
title('Trapezoid')  
axis padded  
xlabel('x')  
ylabel('y')  
legend('h=0.8','h=0.4','h=0.2','Exact')  
  
figure  
  
  
% For Euler method  
% Need to redefine the initial values to overide magnitudes from previous  
% method  
x\_0 = 1;  
y\_0 = 1;  
x\_n = 5;  
  
for i = 0:2  
  
 h = 0.8/2^i;  
 n = round((x\_n-x\_0)/h);  
 x = zeros(1,n+1);  
 y = zeros(1,n+1);  
  
 x = x\_0:h:x\_n;  
 y(1) = y\_0;  
 for k = 1:n  
 y\_star = y(k) + h\*(x(k) - x(k)\*y(k)/2);  
 y(k+1) = y(k) + 0.5 \* h\* (feval('Euler',x(k),y(k))+feval('Euler',x(k+1),y\_star));  
 end  
 plot(x,y,'color',rand(1,3))  
 hold on  
end  
  
y\_exact = zeros(1,n+1);  
y\_exact(1) = y\_0;  
for k = 2:n+1  
 y\_exact(k) = 2-exp((1-x(k).^2)/4);  
end  
  
plot(x,y\_exact)  
title('Euler')  
xlabel('x')  
ylabel('y')  
axis padded  
legend('h=0.8','h=0.4','h=0.2','Exact');

Chart

Description automatically generated

Chart

Description automatically generated

[*Published with MATLAB® R2021a*](https://www.mathworks.com/products/matlab)

**Text

Description automatically generated**

**Solution:**

* ***Chemical concentration***

function f = chem(t,y)

k = 0.01;

a = 70;

b = 50;

f = k\*(a-y)\*(b-y);

end

* ***Runge-Kutta function***

% Runge-Kutta function

function [x,y] = RK4(ode,a,b,h,y\_ini)

n = round((b-a)/h)+1;

y = zeros(1,n);

x = linspace(a,b,n);

y(1) = y\_ini;

for i = 1:n-1

k(1:4) = 0;

k(1) = feval(ode,x(i),y(i));

k(2) = feval(ode,x(i)+0.5\*h,y(i)+0.5\*k(1)\*h);

k(3) = feval(ode,x(i)+0.5\*h,y(i)+0.5\*k(2)\*h);

k(4) = feval(ode,x(i)+h,y(i)+k(3)\*h);

y(i+1) = y(i) + 1/6\*(k(1)+2\*k(2)+2\*k(3)+k(4))\*h;

end

* ***Exact solution***

% Script for exact solution

function f = Exact(t)

f = 350\*(1-exp(-0.2\*t))/(7-5\*exp(-0.2\*t));

end

* ***Driver for Q4***

% Q4 driver  
clear all  
close all  
clc  
  
a = 0;  
b = 20;  
h = 0.5;  
y\_ini = 0;  
  
[x,y] = RK4('chem',a,b,h,y\_ini);  
  
n = round((b-a)/h)+1;  
x\_2 = linspace(a,b,n);  
for i = 1:n  
 y\_2(i) = feval('Exact',x\_2(i));  
end  
  
plot(x,y,'o',x\_2,y\_2)  
xlabel('t(s)')  
ylabel('y(millimoles/liter)')  
axis padded  
grid on  
legend('RK4','Exact');

***Matlab output:***

Chart, line chart

Description automatically generated

[*Published with MATLAB® R2021a*](https://www.mathworks.com/products/matlab)