Date / /
Page No.

CL 249 Computational Methods Lab

Assignment 7: Differential Equation (IVP)

Submitted By:

AYUSHMAN CHOUDHARY
(200020039)

Q. Solve ODE'S

$$\frac{dy}{dx} = -2y + 4e^{-x} \quad \text{and} \quad \frac{dz}{dx} = -\frac{yz^2}{3}$$

where $y = 2$ and $z = 4$ at $x = 0$

Solve using Euler's explicit technique to obtain values of y and z from $x=0$ to $x=4$. Also, find the converged/correct solution after plotting

(i) y vs x (ii) z vs x for various values of h.

METHOD

Euler's Explicit technique.

Consider a differential eq^n

$$\frac{dy}{dx} = f(x,y)$$

For a given point x_0, let $y = y_0$

~~Using~~ we can write, $x_i = x_0 + ih$

$y_i = y(x = x_i)$

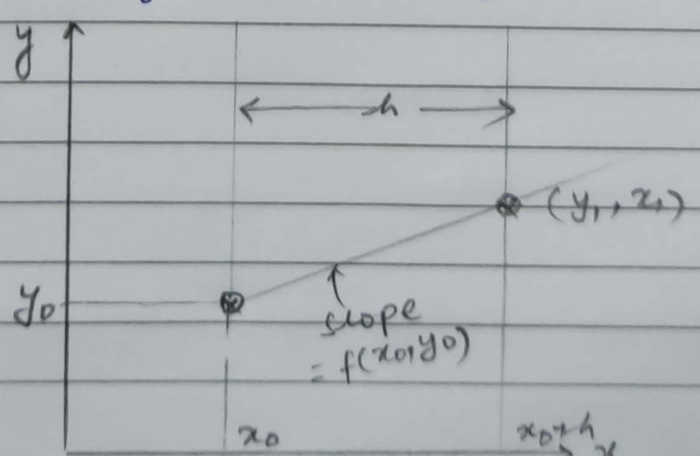
for a given step-size h.

Now, using Taylor's expansion,

$$y(x_0+h) = y(x_0) + hy'(x_0) + \underbrace{\frac{h^2}{2!}y''(x_0) + \cdots}_{\text{, error}}$$

$$\Rightarrow y_{i+1} = y_i + h\,y'(x_i)$$

$$\Rightarrow y_{i+1} = y_i + hf(x_i, y_i)$$



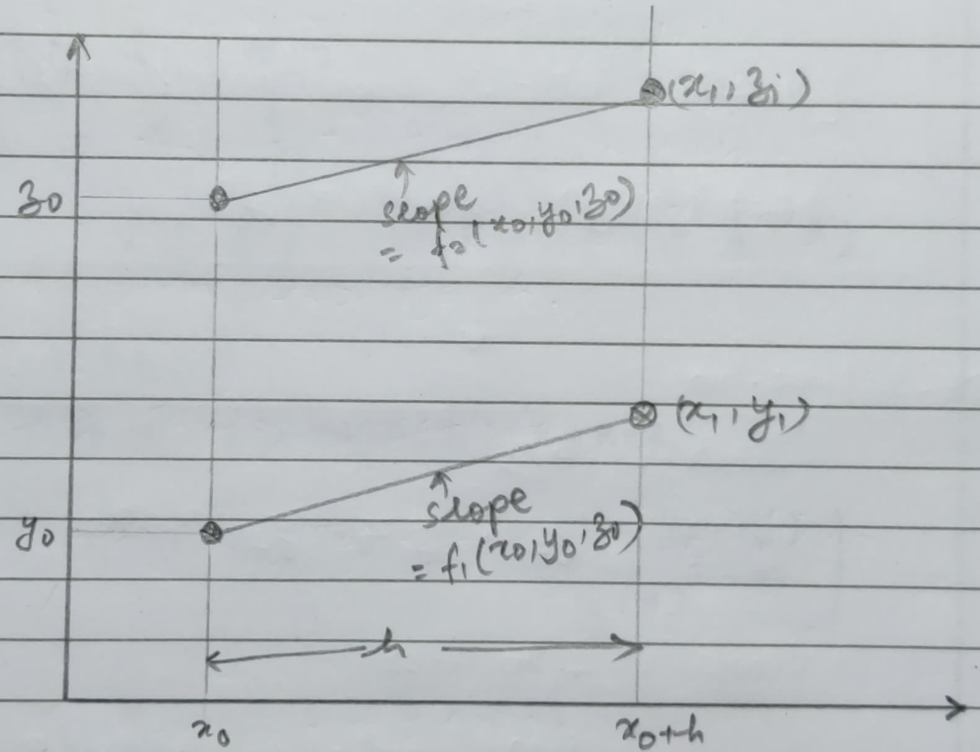
Date / /
Page No.

$\therefore$ for simultaneous differential eq^n, like

$$\frac{dy}{dx} = f_1(x, y, z) \quad \& \quad \frac{dz}{dx} = f_2(x, y, z)$$

$$y_{i+1} = y_i + h\, f_1(x_i, y_i, z_i)$$

$$z_{i+1} = z_i + h\, f_2(x_i, y_i, z_i)$$



Date / /
Page No.

PSEUDOCODE

main.m

```
h = 4 , x0 = 0 , y1 = 2 , y2 = 4
h1 = 0   h2 = 0
[Yi , Zi] = ODE_solver(x0, y1, y2, h);

plot (x, yi)   plot(x, zi)

loop i = 1 to n
    h = h/2

    [Yj , Zj] = ODE_solver(x0, y1, y2, h)

    err1 = 0
    loop k = 1 to length(yj)  k = k+2

        err_temp1 = | (yj(k) - yi((k+1)/2)) / yj(k) |

        if (err_temp1 > err1)
            err1 = err_temp1.
    end loop
    err2=0
    loop l = 1 to length (zj)  l = l+2

        err_temp2 = | (zj(l) - zi((l+1)/2)) / zj(l) |

        if (err_temp2 > err2)
            err2 = err_temp2

    end loop
```

```
if(err1 > 1e-4)
    plot(x, yj), h1 = h
if (err2 > 1e-4)
    plot (x, zj), h2 = h

    yi = yj , zi = zj
end loop


h1 = h1/2 , h2 = h2/2

h = min(h1, h2)

[y, z] = ODE-solver(x0, y1, y2, h)

plot(x, y), plot(x, z)
```

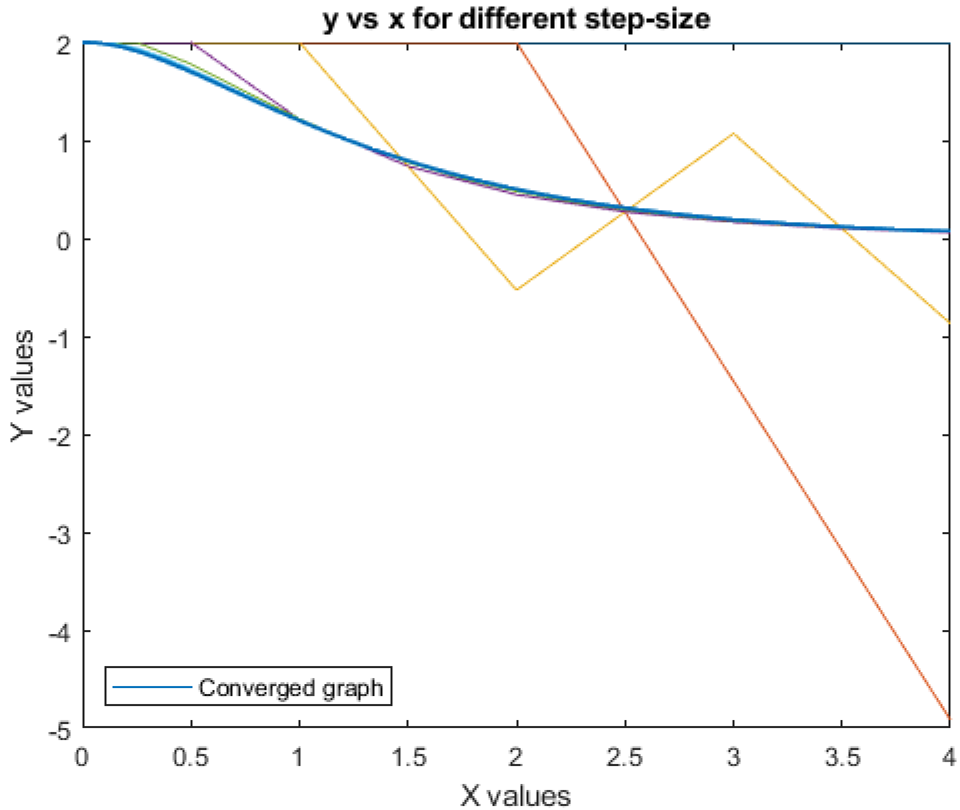
ODE-Solver.m

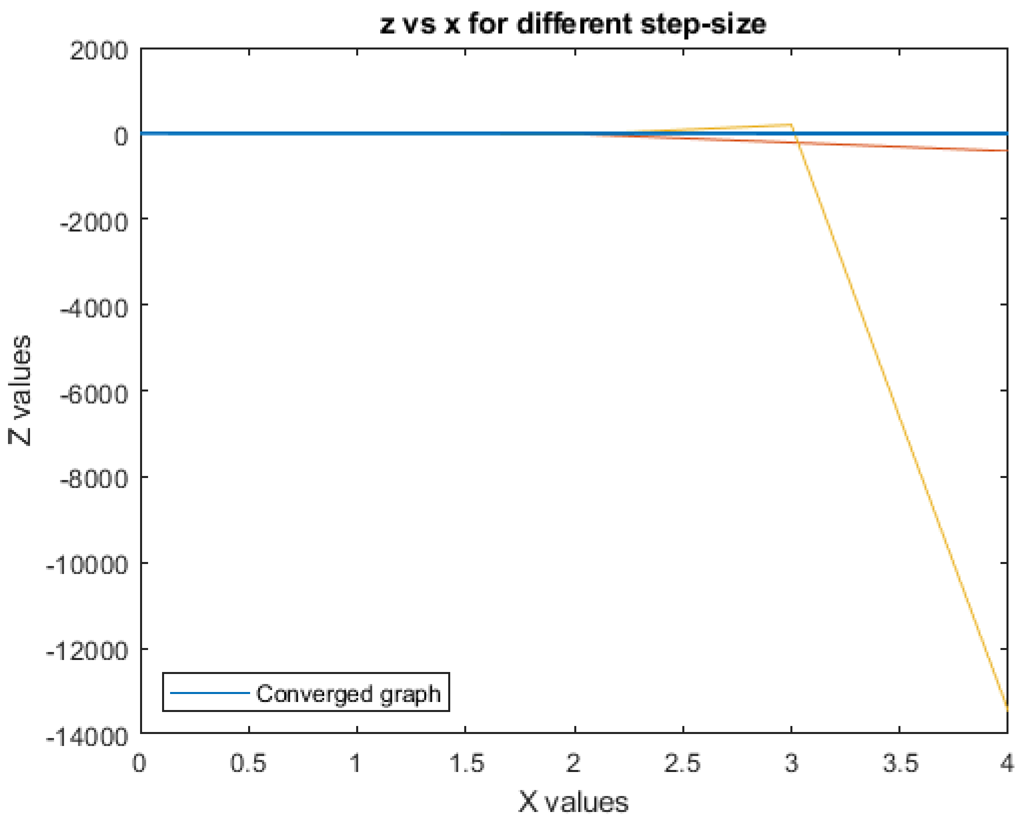
```
iter = 4/h ,
y(1) = y1 , z1 = y2 , x = x0

loop i = 1 to iter

    y(i+1) = y(i) + h* dy(x, y(i))
    z(i+1) = z(i) + h* dz(y(i), z(i))

    x = x+h

end loop
```

where dy and dz calculate diff value $\frac{dy}{dx}$ & $\frac{dz}{dx}$

```
clc
clear
clear all

h=4; h1=0; h2=0;
x0=0; y1=2; y2=4;

[y_i,z_i]= ODE_solver(x0,y1,y2,h); % initial h result
x=0:h:4;

%plot
figure(1)
plot(x,y_i);
hold on;

figure(2)
plot(x,z_i);
hold on

for i=1:6 %looping for decreasing h values
    h= h./2;
    [y_j,z_j]= ODE_solver(x0,y1,y2,h); % finding y and z for new h
    x=0:h:4;

    % Error calculation with prev comparison
    err1=0;
    for k=1:2:length(y_j)
        err_temp1= abs((y_j(k)-y_i((k+1)./2))./y_j(k));
        if(err_temp1 > err1)
            err1= err_temp1;
        end
    end

    err2=0;
    for l=1:2:length(z_j)
        err_temp2= abs((z_j(l)-z_i((l+1)./2))./z_j(l));
        if(err_temp2 > err2)
            err2= err_temp2;
        end
    end

    % convergence check & plot
    if(err1 > 1e-4)
    figure(1)
    plot(x,y_j);
    hold on;
    h1=h;
    end

    if(err2 > 1e-4)
    figure(2)
    plot(x,z_j);
```

```
    hold on;
    h2=h;
    end

    % swapping values for next iteration
    y_i=y_j;
    z_i=z_j;
end

% final converged graph
h1=h1./2; h2= h2./2;
h= min(h1,h2);
[y,z]= ODE_solver(x0,y1,y2,h);
x=0:h:4;
figure(1)
plot(x,y,'Linewidth',1.2);
title('y vs x for different step-size');
legend('Converged graph','Location','southwest');
xlabel('X values');
ylabel('Y values');

figure(2)
plot(x,z,'Linewidth',1.2);
title('z vs x for different step-size');
legend('Converged graph','Location','southwest');
xlabel('X values');
ylabel('Z values');

hold off
```


y vs x for different step-size
Y values
X values
2
1
0
-1
-2
-3
-4
-5
0
0.5
1
1.5
2
2.5
3
3.5
4
Converged graph


z vs x for different step-size
Z values
X values
2000
0
-2000
-4000
-6000
-8000
-10000
-12000
-14000
0
0.5
1
1.5
2
2.5
3
3.5
4
Converged graph

```
function [y,z]= ODE_solver(x0,y1,y2,h)
    iter=4./h;
    y(1)=y1;
    z(1)=y2;
    x=x0;
    for i=1:iter

        % Euler's Explicit rule formula
        y(i+1)= y(i) + h.*dy(x,y(i));
        z(i+1)= z(i) + h.*dz(y(i),z(i));

        x= x + h;
    end
end
```

Published with MATLAB® R2021a

```
function y1=dy(x,y)
    % Returns differential value at particular x,y
    y1= -2.*y + 4.*exp(-x);
end
```

Published with MATLAB® R2021a

```
function y2=dz(y,z)
    % Returns differential value at particular y,z
    y2= (-y.*z.^2)./3;
end
```

Published with MATLAB® R2021a

Comments & Remarks

The error term in Euler's explicit method is given by:

$$\text{error} = \frac{h^2}{2} y''(\xi_p)$$

So, decreasing h, leads to better accuracy.

Plus, if we decrease h by $\frac{1}{2}$, the number of data points increase constantly.

It can be ensured that

$$\max \left| \frac{y(x_i, h_j) - y(x_i, h_{j+1})}{y(x_i, h_{j+1})} \right| < \varepsilon$$

for common points in consecutive Solⁿ.

This ensures converged solution.