DATE : / /

PAGE NO. :

Q Find the solution of differential equations using euler's explicit method.

$$\frac{dy_1}{dx} = -2y_1 + 4e^{-x} \quad \frac{dy_2}{dx} = \frac{-y_1 y_2^2}{3}$$

$$\& \; y_1(0) = 2 \; , \; y_2(0) = 4$$

Sol

Euler Explicit Method:

$$y^{(i+1)} = y^i + h f(x^i; y^i)$$

$$f(x^i; y^i) = \left.\frac{dy}{dx}\right|_{x^i, y^i} !$$

given initial values of x & y we can find further values.

Algorithm:

First we solve for y_1

given $y_1(0) = 2$

$$y_1^{(i+1)} = y_1^i + h f(x^{(i)}, y^{(i)})$$

$$\& \; f(x^{(i)}, y^{(i)}) = \frac{dy_1}{dx} = -2y_1^{(i)} + 4e^{-x^{(i)}}$$

we get y_1 for each x from here

DATE: / /

PAGE NO.:

Now using y_1

we calculate y_2

$$y_2^{(i+1)} = y_2^{(i)} + h f_2(x^{(i)}, y^{(i)})$$

here $f(x^{(i)}, y^{(i)}) = \frac{dy_2}{dx} = \frac{-y_1 y_2^2}{3}$

& given $y_2(0) = 4$

we calculate values of y_2

For converged solution we use

$$\max\left(abs\left(\frac{y_1^{(i+1)} - y_1^{(i)}}{y_1^{(i)}}\right)\right) \;\&\; \max\left(abs\left(\frac{y_2^{(i+1)} - y_2^{(i)}}{y_2^{(i)}}\right)\right)$$

$< \epsilon$

we get converged solution $y_1^{(i+1)}$ & $y_2^{(i+1)}$

The converged solution will be the one with minimum step value h.

Because $f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2} f''(x_0) \ldots$

DATE : / /

PAGE NO. :

$$\left| f(x_0+h) - f(x_0) - h f'(x_0) \right| \approx \frac{h^2}{2} \left| f''(x_0) \right|$$

so to minimize errors we need to minimize the value of h

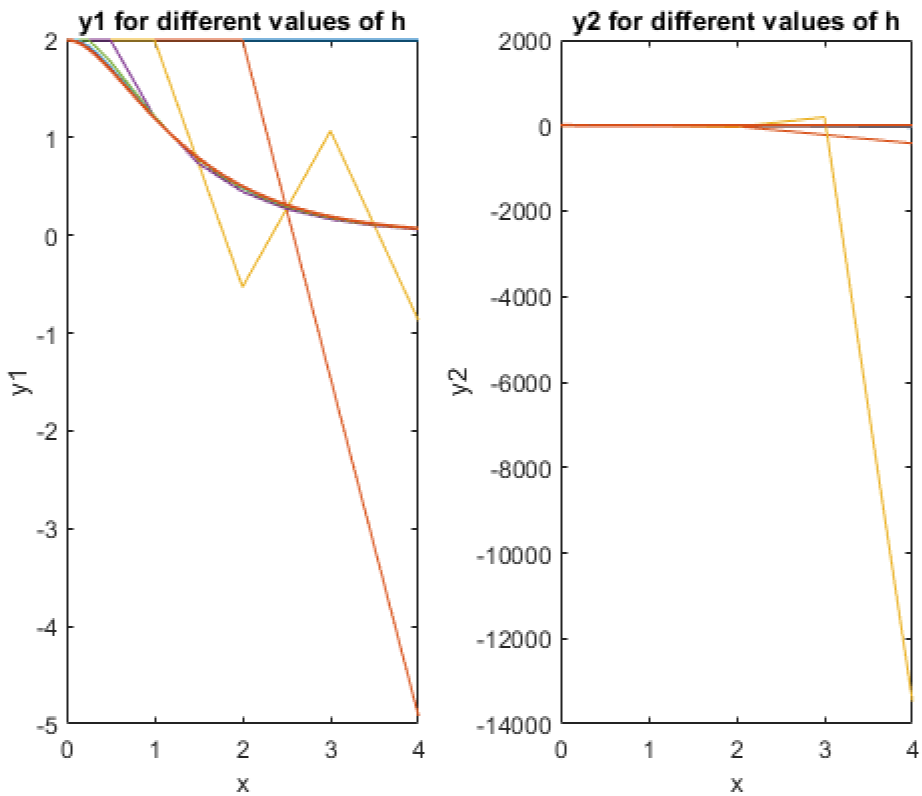
hence for min h solution will be more accurate.

```
%interval of a,b
a = 0;
b= 4;
% Defining the values of h
n = 10;
h = zeros(n,1);
h = [4 2 1 0.5 0.25 0.125 0.0625 0.03125 0.015625];

for i = h
    x = 0:i:4;% defining the values of x depending upon step size
    [y1 y2] = euler(i,a,b,2,4);%calulate functions y1 and y2 for
 stepval = i

%Plotting values of y1 for different h
subplot(1, 2, 1);
if i == min(h)
plot(x,y1,"LineWidth",1.1);% converged solution of y1
else
   plot(x,y1);
end
title("y1 for different values of h");
xlabel("x");
ylabel("y1");
hold on;

%Plotting values of y2 for different h
subplot(1, 2, 2);
if i == min(h)
plot(x,y2,"LineWidth",1.1);% converged solution of y2
else
   plot(x,y2);
end
title("y2 for different values of h");
xlabel("x");
ylabel("y2");
hold on;
end
```



Published with MATLAB® R2021a

```
function [y1 y2] = euler(h,a,b,y1_boundary,y2_boundary)
n = (b-a)/h;%number of intervals
val = zeros(n+1,1);% initialising values of y1
val(1)=y1_boundary; %applying boundary condition on y1
x =a;
%Calculating approx value of function y1 with step size h
for i = 1:n
    val(i+1) = val(i) + h*f1(x,val(i));
    x= x+h;
end
y1 = val;

% Reinitialising values for y2
x = a;
val = zeros(n+1,1);
val(1)= y2_boundary; % Storing boundary condition of y2
%Calculating approx value of function y2 with step size h
for i = 1:n
    val(i+1) = val(i) + h*f2(y1(i),val(i));
end
y2 = val;
return
end
```

```
Not enough input arguments.

Error in euler (line 2)
n = (b-a)/h;%number of intervals
```

Published with MATLAB® R2021a

```
function result = f1(x,y)
% returning the value of dy1/dx
result = -2*y+4*exp(-1*x);

end
```

```
Not enough input arguments.

Error in f1 (line 3)
result = -2*y+4*exp(-1*x);
```

Published with MATLAB® R2021a

```
function result = f2(y1,y2)
% returning the value of dy2/dx
result = (-1/3)*y1*(y2)^2;

end
```

```
Not enough input arguments.

Error in f2 (line 3)
result = (-1/3)*y1*(y2)^2;
```

Published with MATLAB® R2021a