Page No
Date / /

CL249: ASSIGNMENT 7

Himanshu Choudhary
200020059

Problem statement

We have to solve the given ODES

$$\frac{dy_1}{dx} = -2y_1 + 4e^{-x} \qquad y_1(x=0) = 2$$

$$\frac{dy_2}{dx} = \frac{-y_1 y_2^2}{3} \qquad y_2(x=0) = 4$$

using Euler's Explicit Method for initial values given.

And plot y_1 vs x and y_2 vs x for different values of h.

Description of Method

Euler's Explicit method.

We divide ~~into~~ the interval into parts (N) and find the function using the formula.

$$x_{i+1} = x_i + h \qquad h = \frac{b-a}{N}$$

$$y_{i+1} = y_i + h^* \left.\frac{dy}{dx}\right|_{x=x_i}$$

$\frac{dy}{dx} = f'(x,y)$ is the differential eq^n

We are given initial values of ys at $x = 0$.

PSEUDO CODE

main.m

Initialize interval [0, 4)

loop for N

$N = 2^{i-1}$

get x, y_1, y_2 from solver

loop in 1: length (y_{pre})

$error_1 = \left| \frac{y_1(i) - y_{1pre}(i)}{y_i} \right|$

loop in 1 : lpre

$error_2 = \left| \frac{y_2(i) - y_{2pre}(i)}{y_2(i)} \right|$

plot (x, y_1)

plot (x, y_2)

solver.m

get arguments a, b, N

$h = b-a/N$

$y(1) = 2, y2(1) = 4;$

loop from 1 to (N-1)

$x_{i+1} = x_i + h$

$y_1(i+1) = y1(i) + h$ (derivative$_1$)

$y_2(i+1) = y_2(i) + h$ (derivative 2)

return

Page No
Date / /

derivative.m

get arguments x, y_1, y_2

$y_1' = f1(x, y_1)$
$y_2' = f2(y_1, y_2)$

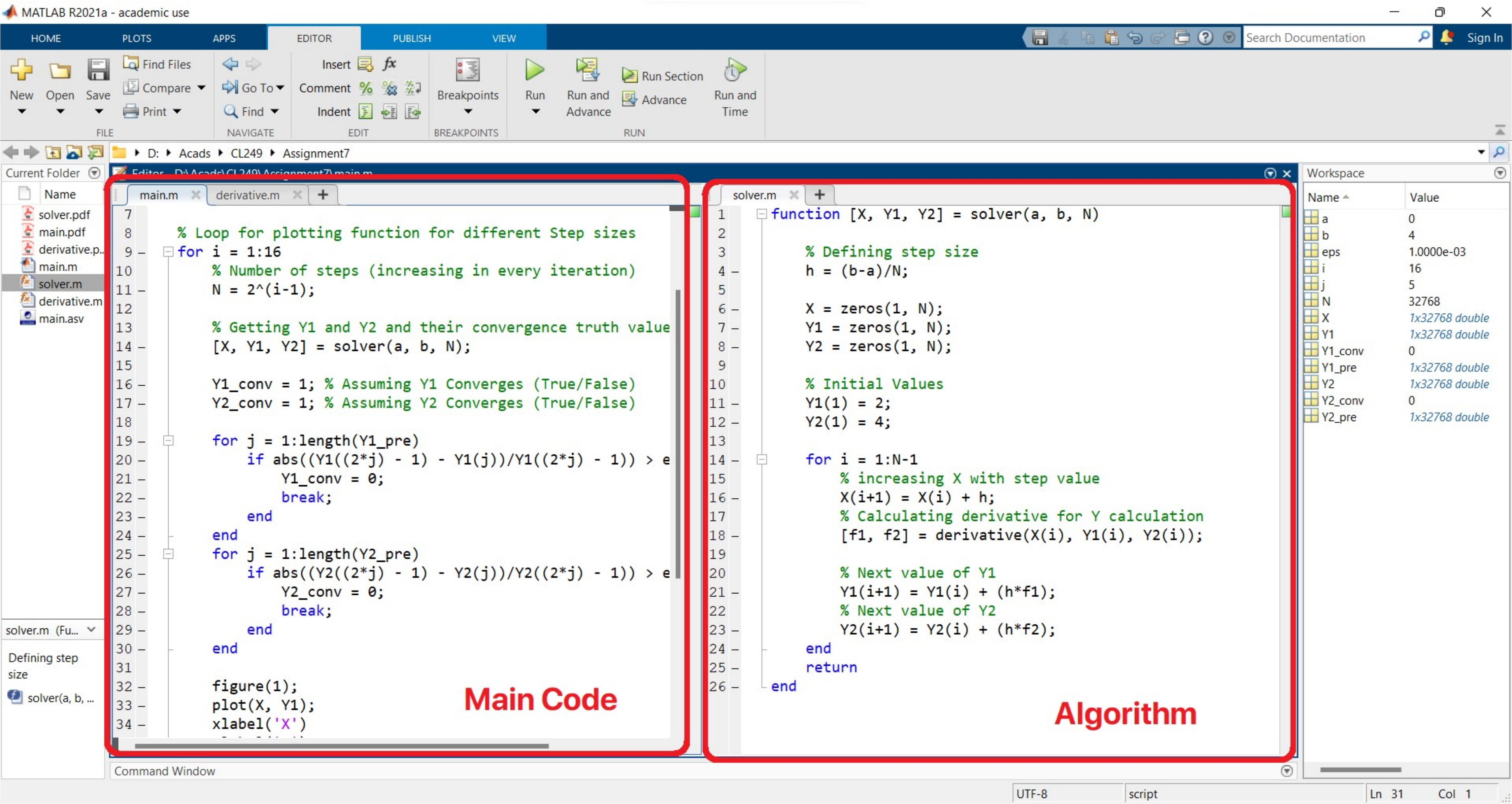
function f1 (x, y)
return $-2y + 4e^{-x}$

function f2 (y, y2)
return $-\frac{y_1 y_2^2}{3}$

```matlab
% Defining Interval
a = 0;
b = 4;
Y1_pre = 0;
Y2_pre = 0;
eps = 10^(-3);

% Loop for plotting function for different Step sizes
for i = 1:16
    % Number of steps (increasing in every iteration)
    N = 2^(i-1);

    % Getting Y1 and Y2 and their convergence truth values from solver
    [X, Y1, Y2] = solver(a, b, N);

    Y1_err = 0; % Assuming Y1 Converges (True/False)
    Y2_err = 0; % Assuming Y2 Converges (True/False)

    for j = 1:length(Y1_pre)
        temp = abs((Y1((2*j) - 1) - Y1(j))/Y1((2*j) - 1));
        if temp > Y1_err
            Y1_err = temp;
        end
    end
    for j = 1:length(Y2_pre)
        temp = abs((Y2((2*j) - 1) - Y2(j))/Y2((2*j) - 1));
        if temp > Y2_err
            Y2_err = temp;
            break;
        end
    end

    figure(1);
    plot(X, Y1);
    xlabel('X')
    ylabel('Y')
    title('Function Y1')
    hold on;

    figure(2);
    plot(X,Y2);
    xlabel('X')
    ylabel('Y2')
    title('Function Y2')
    hold on;

    Y1_pre = Y1;
    Y2_pre = Y2;
end
```

```
function [X, Y1, Y2] = solver(a, b, N)

    % Defining step size
    h = (b-a)/N;

    X = zeros(1, N);
    Y1 = zeros(1, N);
    Y2 = zeros(1, N);

    % Initial Values
    Y1(1) = 2;
    Y2(1) = 4;

    for i = 1:N-1
        % increasing X with step value
        X(i+1) = X(i) + h;
        % Calculating derivative for Y calculation
        [f1, f2] = derivative(X(i), Y1(i), Y2(i));

        % Next value of Y1
        Y1(i+1) = Y1(i) + (h*f1);
        % Next value of Y2
        Y2(i+1) = Y2(i) + (h*f2);
    end
    return
end
```

```
% Function to calculate derivatives of Y1 and Y2
function [y1_dash, y2_dash] = derivative(x, y1, y2)
    y1_dash = f1(x, y1);
    y2_dash = f2(y1, y2);
    return
end

% Given function Y1 slope
function t = f1(x, y)
    t = -(2*y) + (4*exp(-1*x));
    return
end

% Given function Y2 slope
function t = f2(y1, y2)
    t = (-1)*y1*(y2^2)/3;
    return
end
```


MATLAB R2021a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
D: ▸ Acads ▸ CL249 ▸ Assignment7
Current Folder
Name
solver.pdf
main.pdf
derivative.p...
main.m
solver.m
derivative.m
main.asv
solver.m (Fu...
Defining step size
solver(a, b, ...
main.m
derivative.m
% Loop for plotting function for different Step sizes
for i = 1:16
% Number of steps (increasing in every iteration)
N = 2^(i-1);
% Getting Y1 and Y2 and their convergence truth value
[X, Y1, Y2] = solver(a, b, N);
Y1_conv = 1; % Assuming Y1 Converges (True/False)
Y2_conv = 1; % Assuming Y2 Converges (True/False)
for j = 1:length(Y1_pre)
if abs((Y1((2*j) - 1) - Y1(j))/Y1((2*j) - 1)) > e
Y1_conv = 0;
break;
end
end
for j = 1:length(Y2_pre)
if abs((Y2((2*j) - 1) - Y2(j))/Y2((2*j) - 1)) > e
Y2_conv = 0;
break;
end
end
figure(1);
plot(X, Y1);
xlabel('X')
Main Code
solver.m
function [X, Y1, Y2] = solver(a, b, N)
% Defining step size
h = (b-a)/N;
X = zeros(1, N);
Y1 = zeros(1, N);
Y2 = zeros(1, N);
% Initial Values
Y1(1) = 2;
Y2(1) = 4;
for i = 1:N-1
% increasing X with step value
X(i+1) = X(i) + h;
% Calculating derivative for Y calculation
[f1, f2] = derivative(X(i), Y1(i), Y2(i));
% Next value of Y1
Y1(i+1) = Y1(i) + (h*f1);
% Next value of Y2
Y2(i+1) = Y2(i) + (h*f2);
end
return
end
Algorithm
Workspace
Name
Value
a 0
b 4
eps 1.0000e-03
i 16
j 5
N 32768
X 1x32768 double
Y1 1x32768 double
Y1_conv 0
Y1_pre 1x32768 double
Y2 1x32768 double
Y2_conv 0
Y2_pre 1x32768 double
Command Window
UTF-8
script
Ln 31
Col 1

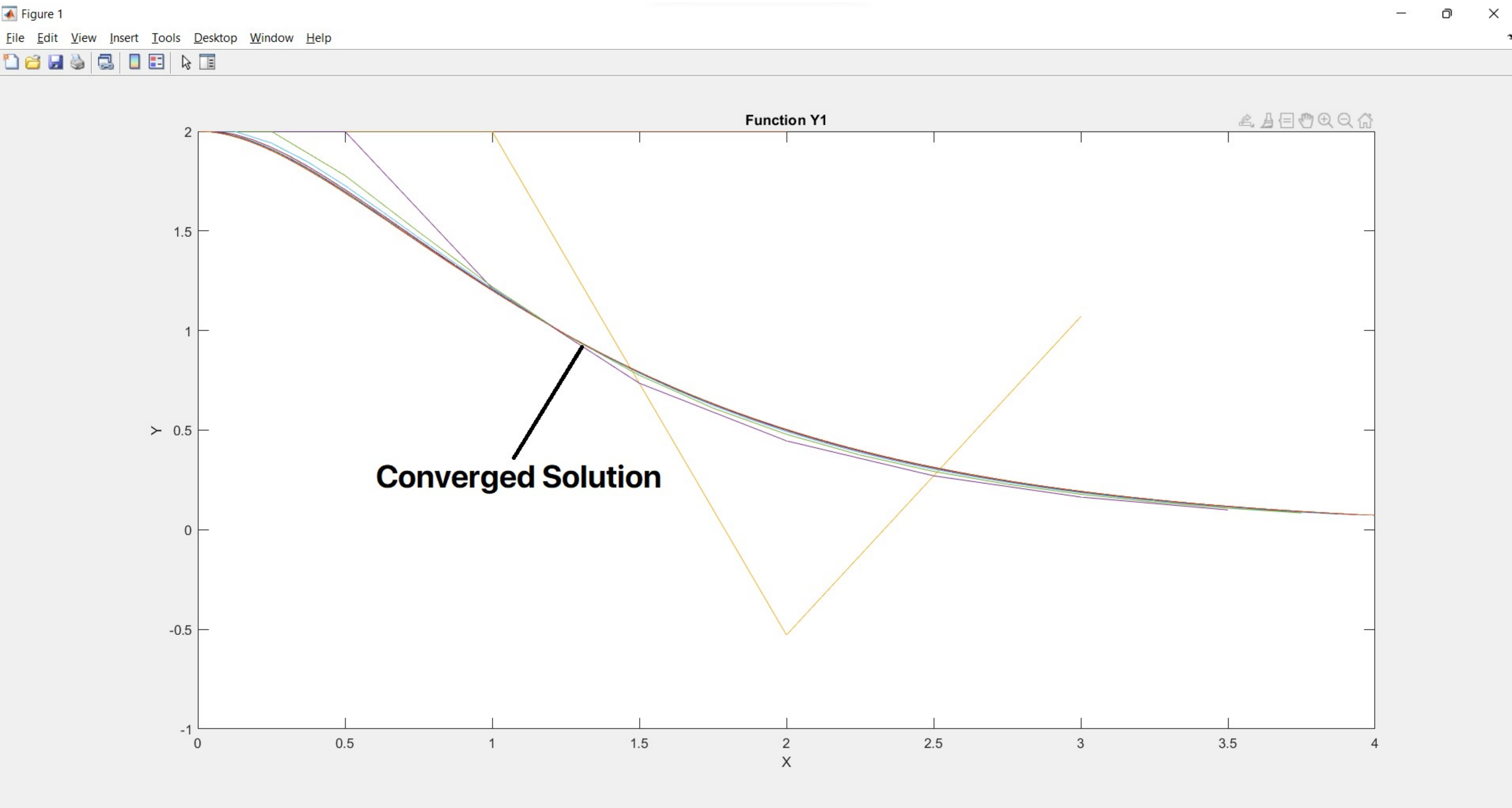


Figure 1
File Edit View Insert Tools Desktop Window Help
Function Y1
Y
X
Converged Solution
2
1.5
1
0.5
0
-0.5
-1
0
0.5
1
1.5
2
2.5
3
3.5
4

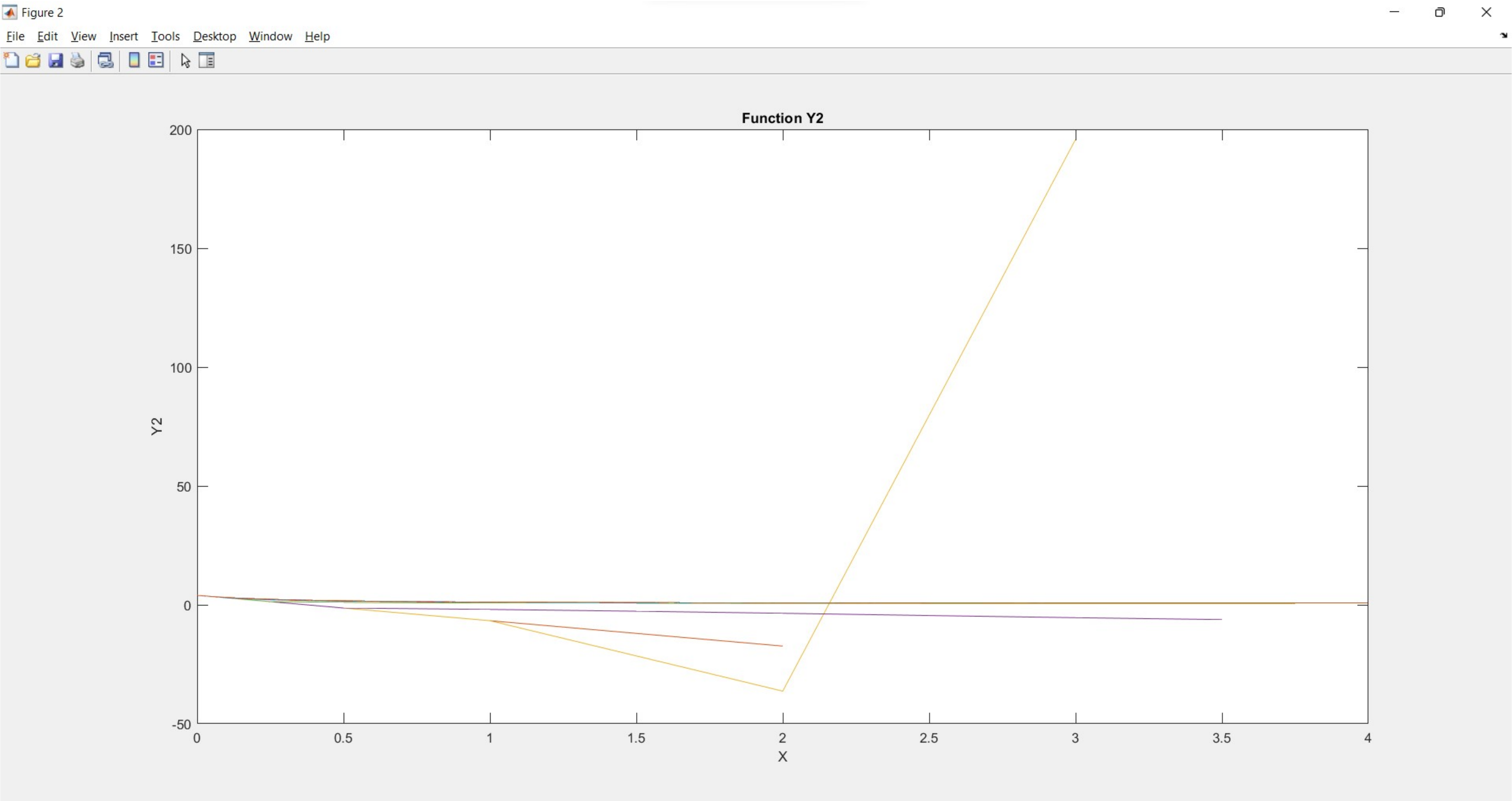


Figure 2
File Edit View Insert Tools Desktop Window Help
Function Y2
200
150
100
50
0
-50
Y2
0
0.5
1
1.5
2
2.5
3
3.5
4
X

Convergence

The two plots y_1 and y_2, the graphs are getting close as we are decreasing step size.

- $h = h/2$ is operating in each iteration as $n = 2n$

- we have also found the max. error between y_1 and y_1 from previous iteration.