Page No.
Date : / /

HIMANSHU CHOUDHARY (200020059)

CL 249: ASSIGNMENT 8

PROBLEM

We have to solve a set of differential equations using RK4 and RK5 technique, which represent one of the simple models to describe spreading of COVID 19.

$$\frac{dy_1}{dx} = -cy_1y_2 \quad \frac{dy_2}{dx} = cy_1y_2 - dy_2 \quad \frac{dy_3}{dx} = dy_2$$

y_1 = Healthy people
y_2 = Infected people
y_3 = People under quarantine

$y_1(x=0) = 95 \quad y_2(x=0) = 5 \quad y_3(x=0) = 0$

And plot the graphs

Description of Method

RK4 Method

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

and we define   h represents stepsize

$K_1 = f(x_i, y_i)$

$K_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1h)$

$K_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_2h)$

$K_4 = f(x+h, y+K_3h)$

and

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)h$$

where y is the required function

RK5 Method

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

define $\qquad\qquad h =$ stepsize

$$K_1 = f(x_i, y_i)$$

$$K_2 = f\left(x_i + \frac{1}{4}h, \; y_i + \frac{1}{4}K_1 h\right)$$

$$K_3 = f\left(x_i + \frac{h}{4}, \; y_i + \frac{K_1 h}{8} + \frac{K_2 h}{8}\right)$$

$$K_4 = f\left(x_i + \frac{h}{2}, \; y_i - \frac{K_2 h}{2} + K_3 h\right)$$

$$K_5 = f\left(x_i + \frac{3h}{4}, \; y_i + \frac{3K_1 h}{16} + \frac{9K_4 h}{16}\right)$$

$$K_6 = f\left(x_i + h, \; y_i - \frac{3K_1 h}{7} + \frac{2K_2 h}{7} + \frac{12K_3 h}{7} - \frac{12K_4 h}{7} + \frac{8K_5 h}{7}\right)$$

and

$$y_{i+1} = y_i + \frac{1}{90}(7K_1 + 32K_3 + 12K_4 + 32K_5 + 7K_6)h$$

where y is the required f^n

Page No.
Date: / /

PSEUDO CODE

RK4.m to calculate using RK4 method.

let X, Y_1, Y_2, Y_3 be empty array
and $C = 1$ $d = 5$ be given const.

Initial values
$Y_1(1) = 95$; $Y2(1) = 5$;
Iterate through 1 to N-1
$x_{i+1} = x_i + h$
$(K_1y_1; K_2y_2, K_3y_3) =$ derivative $(y1(i), y2(i))$
$(K_2y_1\ K_2y_2\ K_2y_3) =$ der $(y_i + \frac{K_1y_ih}{2})$

$(K_3y_1, K_3y_2, K_3y_3) =$ der $(y_{ij} + \frac{K_2y_ih}{2})$

$(K_4y_1, K_4y_2, K_4y_3) =$ der $(y_i + \frac{K_3yh}{2})$

update values $Y_1\ y_2\ y_3$ as
$y_{i+1} = y_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$

return Y_1, Y_2, Y_3;

derivative.m

get y_1 and y_2 and c, d
$y_1' = -Cy_1y_2$
$y_2' = Cy_1y_2 - dy_2$
$y_3' = dy_2$
return all values.

RK5.m

get all the arrays and constants.
assign initial values to y_1 y_2 y_3

Iterate through 1 to N-1
Increase x_{i+1} as $x_i + h$
~~fr~~
Calculate for y_1 y_2 y_3

(K_1y, K_2y, K_3y) = derivative (x_i, y_i)
~~K_2y K~~
$K_2 = \text{derivative}\left(y_i + \frac{K_1h}{4}\right)$
$K_3 = \text{derivative}\left(y_i + \frac{K_1h}{8} + \frac{K_2h}{8}\right)$

$K_4 = \text{derivative}\left(y_i - \frac{K_2h}{2} + K_3h\right)$

$K_5 = \text{derivative}\left(y_i + \frac{3K_1h}{16} + \frac{9K_4h}{16}\right)$

$K_6 = \text{derivative}\left(y_i - \frac{3K_1h}{7} + \frac{2K_2h}{7} + \frac{12K_3h}{7} - \frac{12K_4h}{7} + \frac{8K_5h}{7}\right)$

update each y as ~~$y_{i+1} = y_i + 4$~~
$y_{i+1} = y_i + \frac{h}{90}(7K_1 + 32K_3 + 12K_4 + 32K_5 + 7K_6)$

Main.m

get all y_1 y_2 y_3 from RK4 and RK5
plot y_1 y_2 y_3 vs x

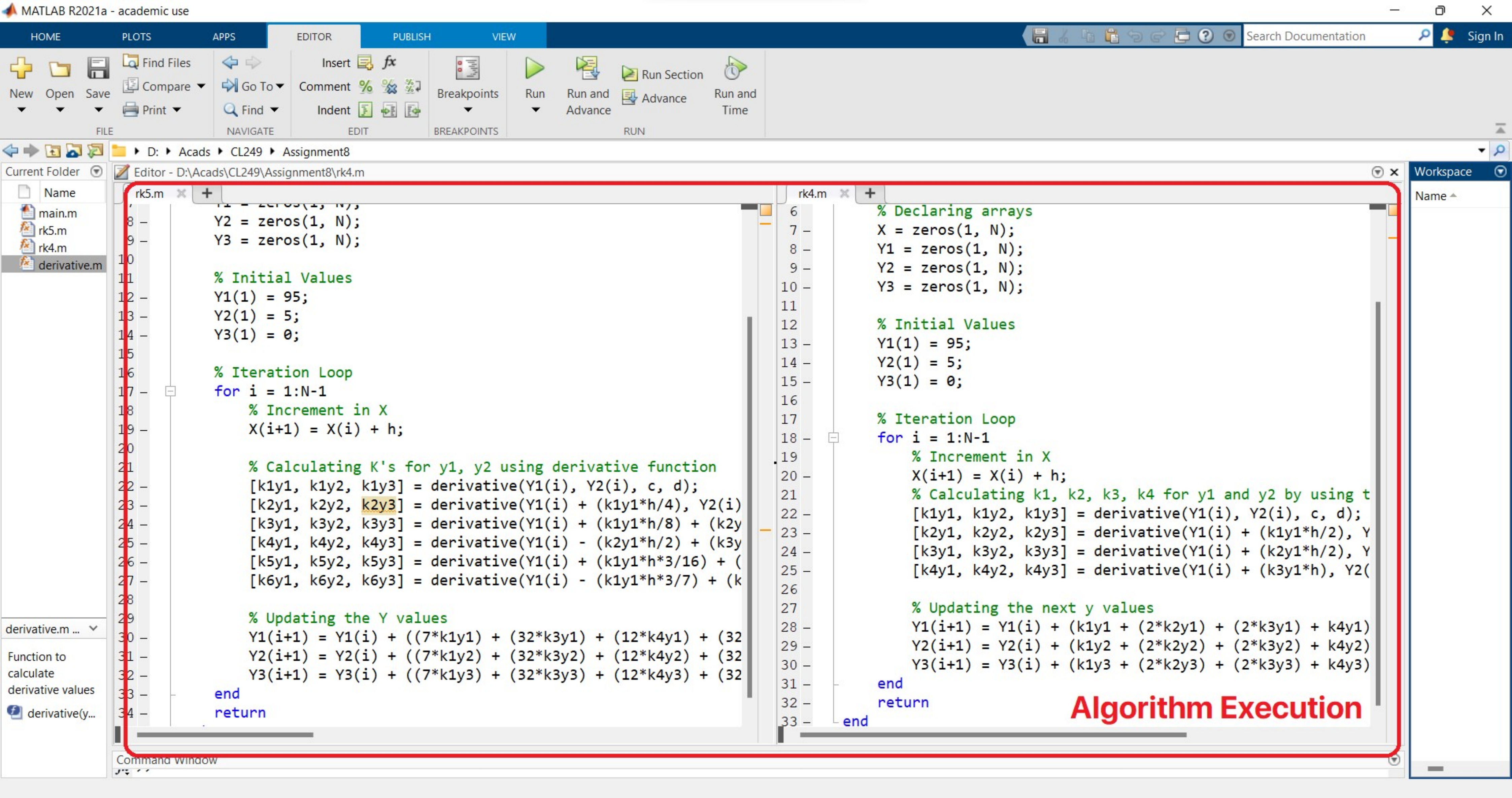
end

```
% Interval of X
a = 0;
b = 4;
% Number of steps
N = 500;
% Step size
h = (b-a)/N;

% Calculating Y1, Y2, Y3, by RK4 Method
[X1, Y1_RK4, Y2_RK4, Y3_RK4] = rk4(a, b, h, N);
% Calculating Y1, Y2, Y3 by RK5 Method
[X2, Y1_RK5, Y2_RK5, Y3_RK5] = rk5(a, b, h, N);

% Plotting the RK4 Method Solution
figure(1)
plot(X1, Y1_RK4, 'DisplayName', 'Y1 (Healthy People)', 'LineWidth', 1.25);
hold on
plot(X1, Y2_RK4,'DisplayName', 'Y2 (Infected Prople)', 'LineWidth', 1.25);
hold on
plot(X1, Y3_RK4, 'DisplayName', 'Y3 (People Under Quarantine)', 'LineWidth',↙
1.25);
title('Solution by RK1 Method')
xlabel('X')
ylabel('Y')
legend

% Plotting the RK5 Method Solution
figure(2)
plot(X2, Y1_RK5, 'DisplayName', 'Y1 (Healthy People)', 'LineWidth', 1.25);
hold on
plot(X2, Y2_RK5,'DisplayName', 'Y2 (Infected Prople)', 'LineWidth', 1.25);
hold on
plot(X2, Y3_RK5, 'DisplayName', 'Y3 (People Under Quarantine)', 'LineWidth',↙
1.25);
title('Solution by RK2 Method')
xlabel('X')
ylabel('Y')
legend
```

```
% Function to calculate solution by RK4 Method
function [X, Y1, Y2, Y3] = rk4(a, b, h, N)
    % Given Constants
    c = 1;
    d = 5;
    % Declaring arrays
    X = zeros(1, N);
    Y1 = zeros(1, N);
    Y2 = zeros(1, N);
    Y3 = zeros(1, N);

    % Initial Values
    Y1(1) = 95;
    Y2(1) = 5;
    Y3(1) = 0;

    % Iteration Loop
    for i = 1:N-1
        % Increment in X
        X(i+1) = X(i) + h;
        % Calculating k1, k2, k3, k4 for y1 and y2 by using the derivative
        [k1y1, k1y2, k1y3] = derivative(Y1(i), Y2(i), c, d);
        [k2y1, k2y2, k2y3] = derivative(Y1(i) + (k1y1*h/2), Y2(i) +
(k1y2*h/2), c, d);
        [k3y1, k3y2, k3y3] = derivative(Y1(i) + (k2y1*h/2), Y2(i) +
(k2y2*h/2), c, d);
        [k4y1, k4y2, k4y3] = derivative(Y1(i) + (k3y1*h), Y2(i) + (k3y2*h),
c, d);

        % Updating the next y values
        Y1(i+1) = Y1(i) + (k1y1 + (2*k2y1) + (2*k3y1) + k4y1)*h/6;
        Y2(i+1) = Y2(i) + (k1y2 + (2*k2y2) + (2*k3y2) + k4y2)*h/6;
        Y3(i+1) = Y3(i) + (k1y3 + (2*k2y3) + (2*k3y3) + k4y3)*h/6;
    end
    return
end
```

```
function [X, Y1, Y2, Y3] = rk5(a, b, h, N)
    % Given Constants
    c = 1;
    d = 5;
    % Declaring arrays
    X = zeros(1, N);
    Y1 = zeros(1, N);
    Y2 = zeros(1, N);
    Y3 = zeros(1, N);

    % Initial Values
    Y1(1) = 95;
    Y2(1) = 5;
    Y3(1) = 0;

    % Iteration Loop
    for i = 1:N-1
        % Increment in X
        X(i+1) = X(i) + h;

        % Calculating K's for y1, y2 using derivative function
        [k1y1, k1y2, k1y3] = derivative(Y1(i), Y2(i), c, d);
        [k2y1, k2y2, k2y3] = derivative(Y1(i) + (k1y1*h/4), Y2(i) +↙
(k1y2*h/4), c, d);
        [k3y1, k3y2, k3y3] = derivative(Y1(i) + (k1y1*h/8) + (k2y1*h/8), Y2↙
(i) + (k1y2*h/8) + (k2y2*h/8), c, d);
        [k4y1, k4y2, k4y3] = derivative(Y1(i) - (k2y1*h/2) + (k3y1*h), Y2(i)↙
- (k2y2*h/2) + (k3y2*h), c, d);
        [k5y1, k5y2, k5y3] = derivative(Y1(i) + (k1y1*h*3/16) +↙
(k4y1*h*9/16), Y2(i) + (k1y2*h*3/16) + (k4y2*h*9/16), c, d);
        [k6y1, k6y2, k6y3] = derivative(Y1(i) - (k1y1*h*3/7) + (k2y1*h*2/7)↙
+ (k3y1*h*12/7) - (k4y1*h*12/7) + (k5y1*h*8/7), Y2(i) - (k1y2*h*3/7) +↙
(k2y2*h*2/7) + (k3y2*h*12/7) - (k4y2*h*12/7) + (k5y2*h*8/7) , c, d);

        % Updating the Y values
        Y1(i+1) = Y1(i) + ((7*k1y1) + (32*k3y1) + (12*k4y1) + (32*k5y1) +↙
(7*k6y1))*h/90;
        Y2(i+1) = Y2(i) + ((7*k1y2) + (32*k3y2) + (12*k4y2) + (32*k5y2) +↙
(7*k6y2))*h/90;
        Y3(i+1) = Y3(i) + ((7*k1y3) + (32*k3y3) + (12*k4y3) + (32*k5y3) +↙
(7*k6y3))*h/90;
    end
    return
end
```

```
% Function to calculate derivative values
function [Y1_dash, Y2_dash, Y3_dash] = derivative(y1, y2, c, d)
    Y1_dash = -c*y1*y2;
    Y2_dash = (c*y1*y2) - (d*y2);
    Y3_dash = d*y2;
    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 ▸ Assignment8
Current Folder
Name
main.m
rk5.m
rk4.m
derivative.m
derivative.m ...
Function to calculate derivative values
derivative(y...
Editor - D:\Acads\CL249\Assignment8\rk4.m
rk5.m
Y2 = zeros(1, N);
Y3 = zeros(1, N);
% Initial Values
Y1(1) = 95;
Y2(1) = 5;
Y3(1) = 0;
% Iteration Loop
for i = 1:N-1
% Increment in X
X(i+1) = X(i) + h;
% Calculating K's for y1, y2 using derivative function
[k1y1, k1y2, k1y3] = derivative(Y1(i), Y2(i), c, d);
[k2y1, k2y2, k2y3] = derivative(Y1(i) + (k1y1*h/4), Y2(i)
[k3y1, k3y2, k3y3] = derivative(Y1(i) + (k1y1*h/8) + (k2y
[k4y1, k4y2, k4y3] = derivative(Y1(i) - (k2y1*h/2) + (k3y
[k5y1, k5y2, k5y3] = derivative(Y1(i) + (k1y1*h*3/16) + (
[k6y1, k6y2, k6y3] = derivative(Y1(i) - (k1y1*h*3/7) + (k
% Updating the Y values
Y1(i+1) = Y1(i) + ((7*k1y1) + (32*k3y1) + (12*k4y1) + (32
Y2(i+1) = Y2(i) + ((7*k1y2) + (32*k3y2) + (12*k4y2) + (32
Y3(i+1) = Y3(i) + ((7*k1y3) + (32*k3y3) + (12*k4y3) + (32
end
return
rk4.m
% Declaring arrays
X = zeros(1, N);
Y1 = zeros(1, N);
Y2 = zeros(1, N);
Y3 = zeros(1, N);
% Initial Values
Y1(1) = 95;
Y2(1) = 5;
Y3(1) = 0;
% Iteration Loop
for i = 1:N-1
% Increment in X
X(i+1) = X(i) + h;
% Calculating k1, k2, k3, k4 for y1 and y2 by using t
[k1y1, k1y2, k1y3] = derivative(Y1(i), Y2(i), c, d);
[k2y1, k2y2, k2y3] = derivative(Y1(i) + (k1y1*h/2), Y
[k3y1, k3y2, k3y3] = derivative(Y1(i) + (k2y1*h/2), Y
[k4y1, k4y2, k4y3] = derivative(Y1(i) + (k3y1*h), Y2(
% Updating the next y values
Y1(i+1) = Y1(i) + (k1y1 + (2*k2y1) + (2*k3y1) + k4y1)
Y2(i+1) = Y2(i) + (k1y2 + (2*k2y2) + (2*k3y2) + k4y2)
Y3(i+1) = Y3(i) + (k1y3 + (2*k2y3) + (2*k3y3) + k4y3)
end
return
end
Algorithm Execution
Workspace
Name
Command Window

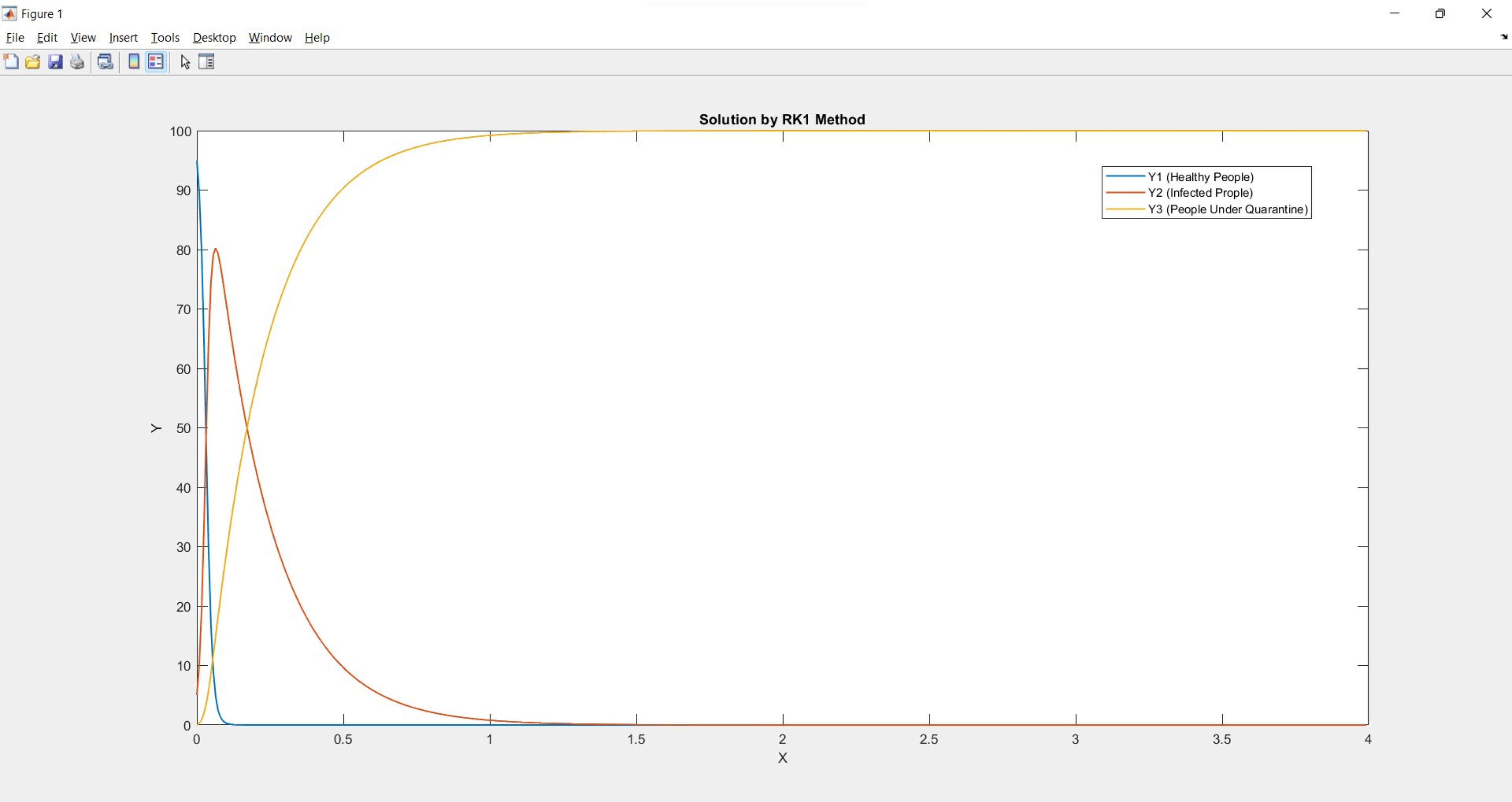


Figure 1
File Edit View Insert Tools Desktop Window Help
Solution by RK1 Method
Y1 (Healthy People)
Y2 (Infected Prople)
Y3 (People Under Quarantine)
Y
X
0
10
20
30
40
50
60
70
80
90
100
0.5
1
1.5
2
2.5
3
3.5
4

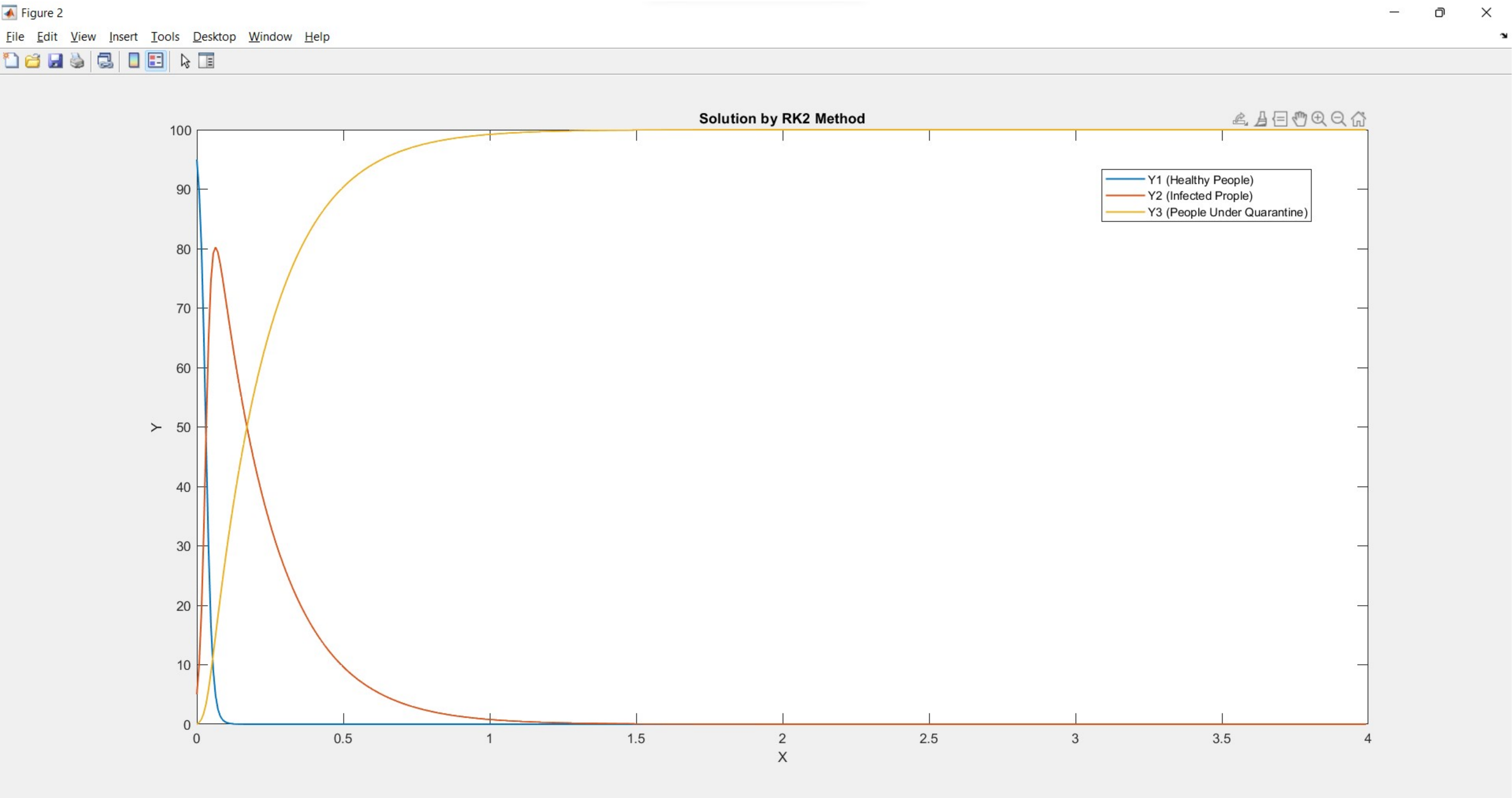


Figure 2
File Edit View Insert Tools Desktop Window Help
Solution by RK2 Method
Y1 (Healthy People)
Y2 (Infected Prople)
Y3 (People Under Quarantine)
Y
X
0
10
20
30
40
50
60
70
80
90
100
0.5
1
1.5
2
2.5
3
3.5
4

Date : / /

Conclusions

- RK4 ~~And~~ method is accurate enough and takes less time (less calculations), so it is more reliable.

- RK5 method is accurate but takes more calculations and hence computational time.