Introduction to MATLAB

Sai Janani Ganesan

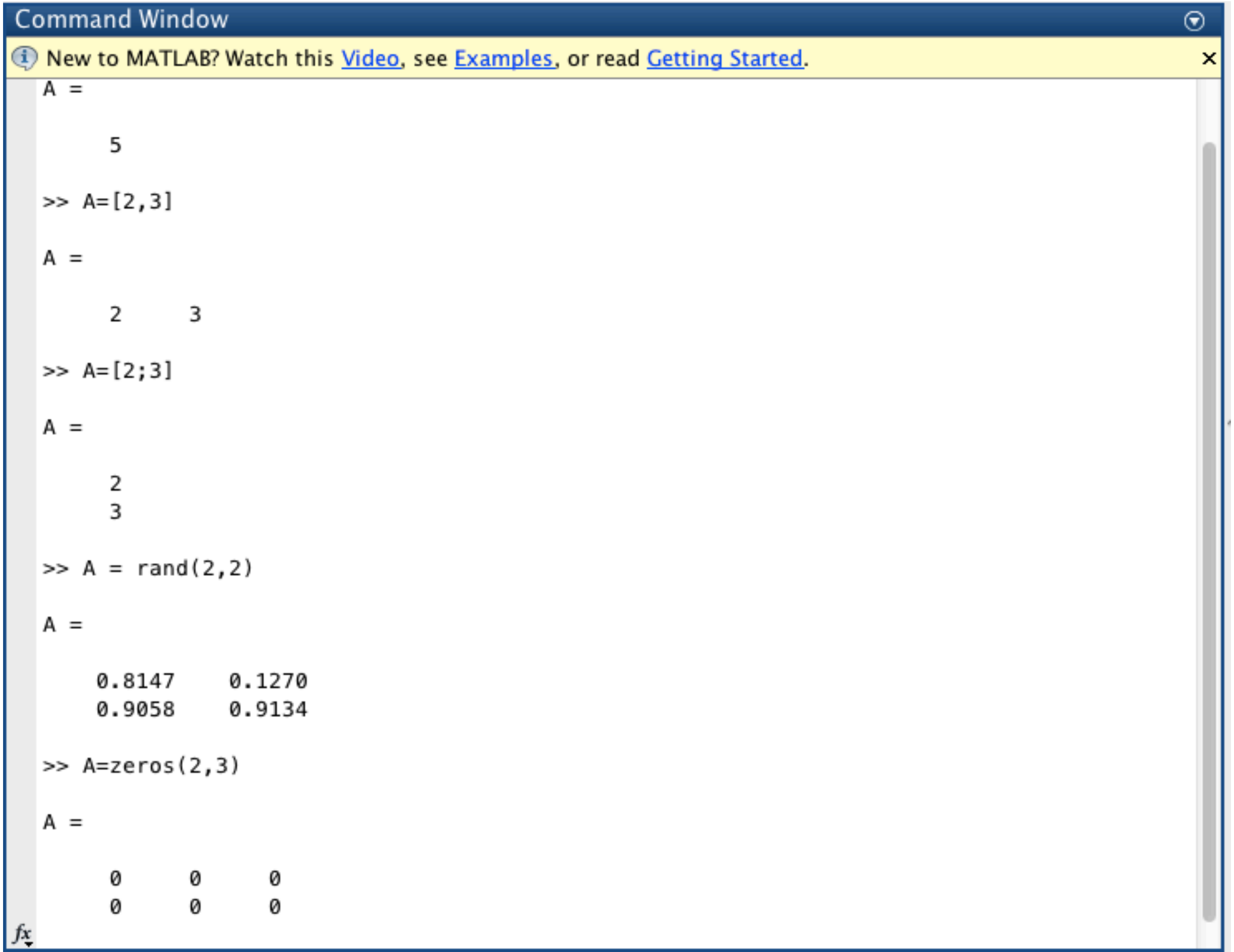
Graduate Student

Matysiak Lab

Outline

- Basics: How to get started, declare variables, basic commands, for loops, if statements
- Functions: What are functions; How to create and execute simple functions
- Solving ODEs: how to solve (a) 1 ODE, (b) system of ODEs
- Useful tips regarding solvers

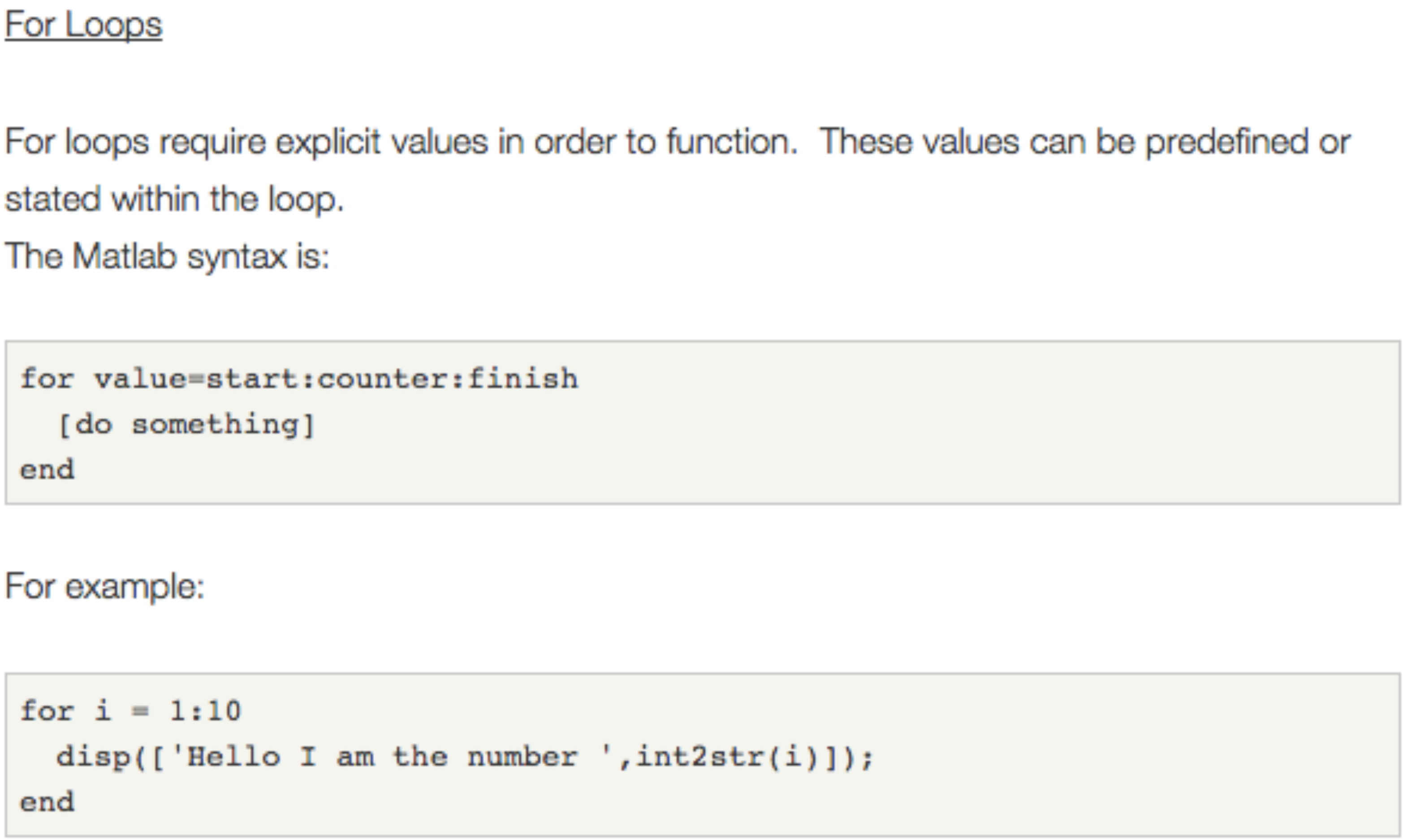
Basics of MATLAB



Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
fx

```
A =

     5

>> A=[2,3]

A =

     2     3

>> A=[2;3]

A =

     2
     3

>> A = rand(2,2)

A =

    0.8147    0.1270
    0.9058    0.9134

>> A=zeros(2,3)

A =

     0     0     0
     0     0     0
```

For Loops



For Loops

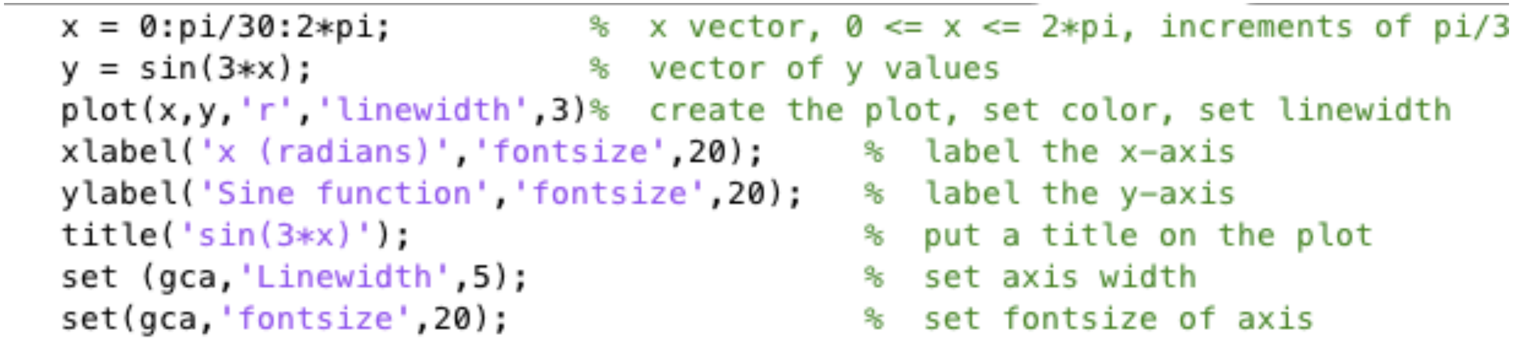
For loops require explicit values in order to function. These values can be predefined or stated within the loop.
The Matlab syntax is:

```
for value=start:counter:finish
  [do something]
end
```

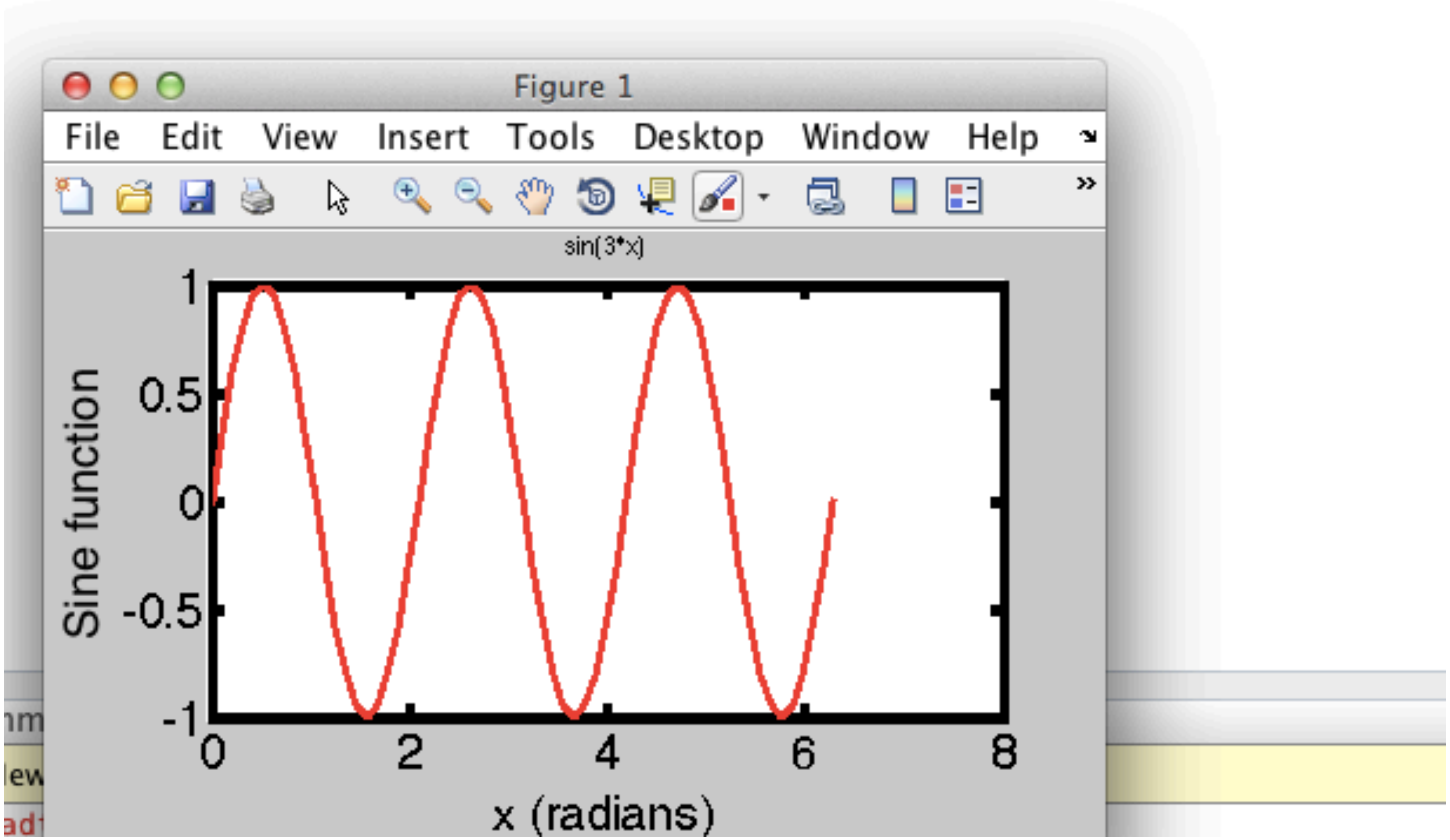
For example:

```
for i = 1:10
  disp(['Hello I am the number ',int2str(i)]);
end
```

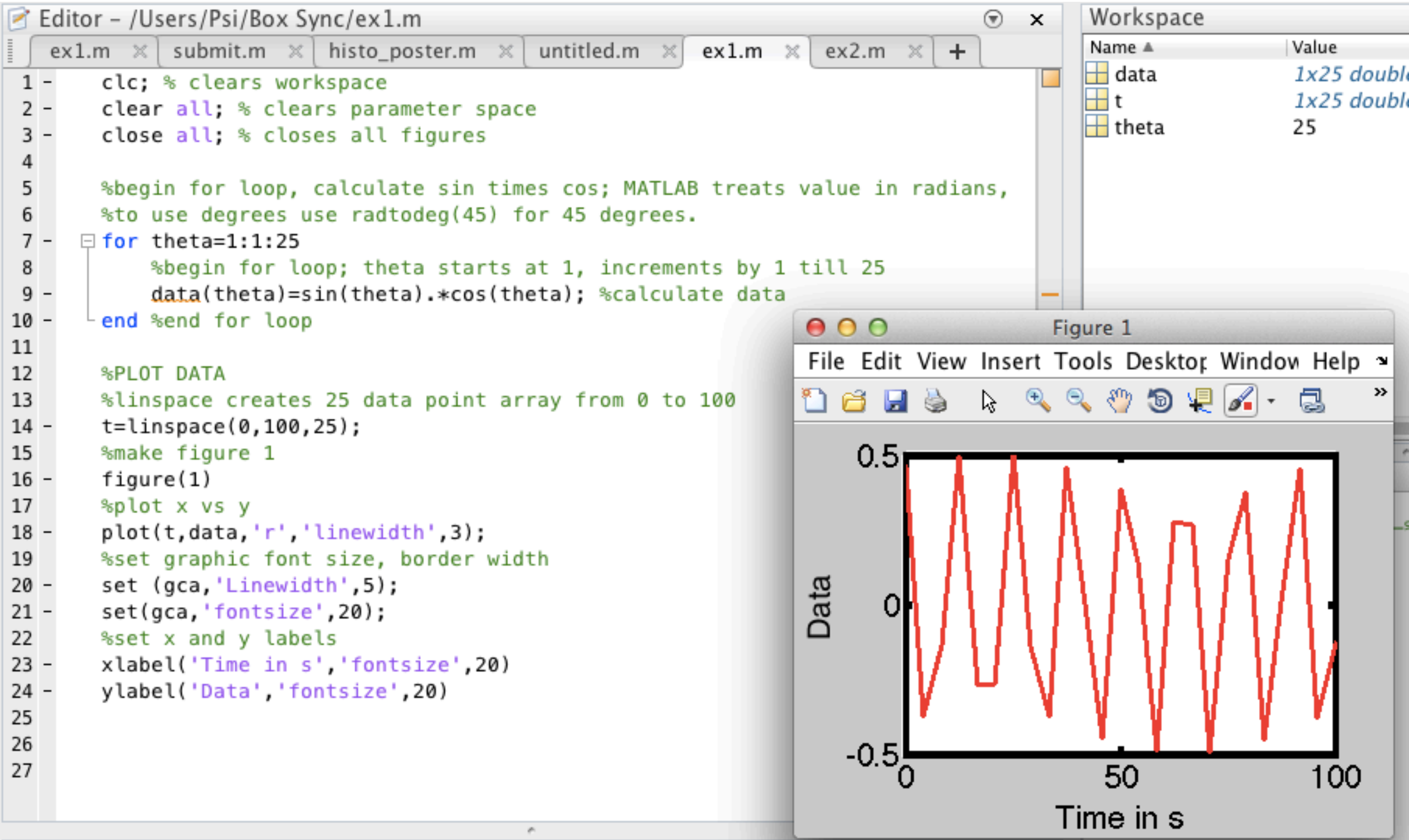
Plotting Data



```
x = 0:pi/30:2*pi;            %  x vector, 0 <= x <= 2*pi, increments of pi/3
y = sin(3*x);                %  vector of y values
plot(x,y,'r','linewidth',3)% create the plot, set color, set linewidth
xlabel('x (radians)','fontsize',20);        %  label the x-axis
ylabel('Sine function','fontsize',20);      %  label the y-axis
title('sin(3*x)');                          %  put a title on the plot
set (gca,'Linewidth',5);                    %  set axis width
set(gca,'fontsize',20);                     %  set fontsize of axis
```



For Loops



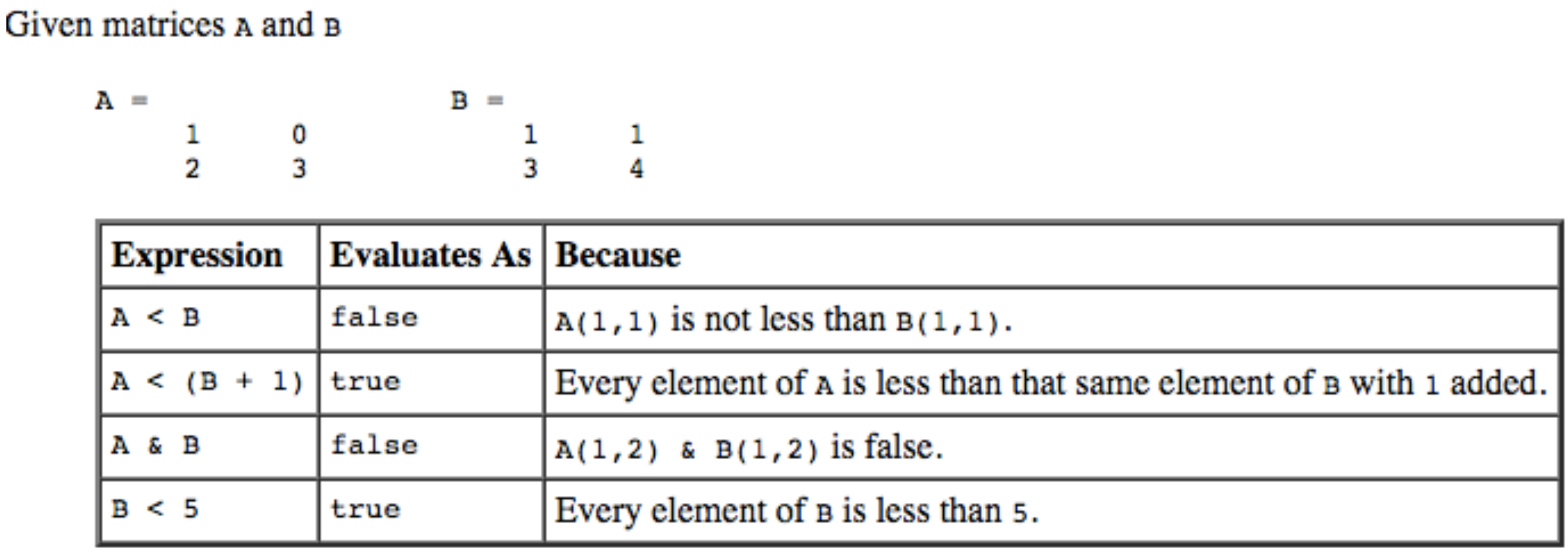
```
clc; % clears workspace
clear all; % clears parameter space
close all; % closes all figures

%begin for loop, calculate sin times cos; MATLAB treats value in radians,
%to use degrees use radtodeg(45) for 45 degrees.
for theta=1:1:25
    %begin for loop; theta starts at 1, increments by 1 till 25
    data(theta)=sin(theta).*cos(theta); %calculate data
end %end for loop

%PLOT DATA
%linspace creates 25 data point array from 0 to 100
t=linspace(0,100,25);
%make figure 1
figure(1)
%plot x vs y
plot(t,data,'r','linewidth',3);
%set graphic font size, border width
set (gca,'Linewidth',5);
set(gca,'fontsize',20);
%set x and y labels
xlabel('Time in s','fontsize',20)
ylabel('Data','fontsize',20)
```

Name ▲	Value
data	1x25 doubl
t	1x25 doubl
theta	25

If Statements

```
if expression
   statements
end
```

```
if expression1
   statements1
elseif expression2
   statements2
else
   statements3
end
```

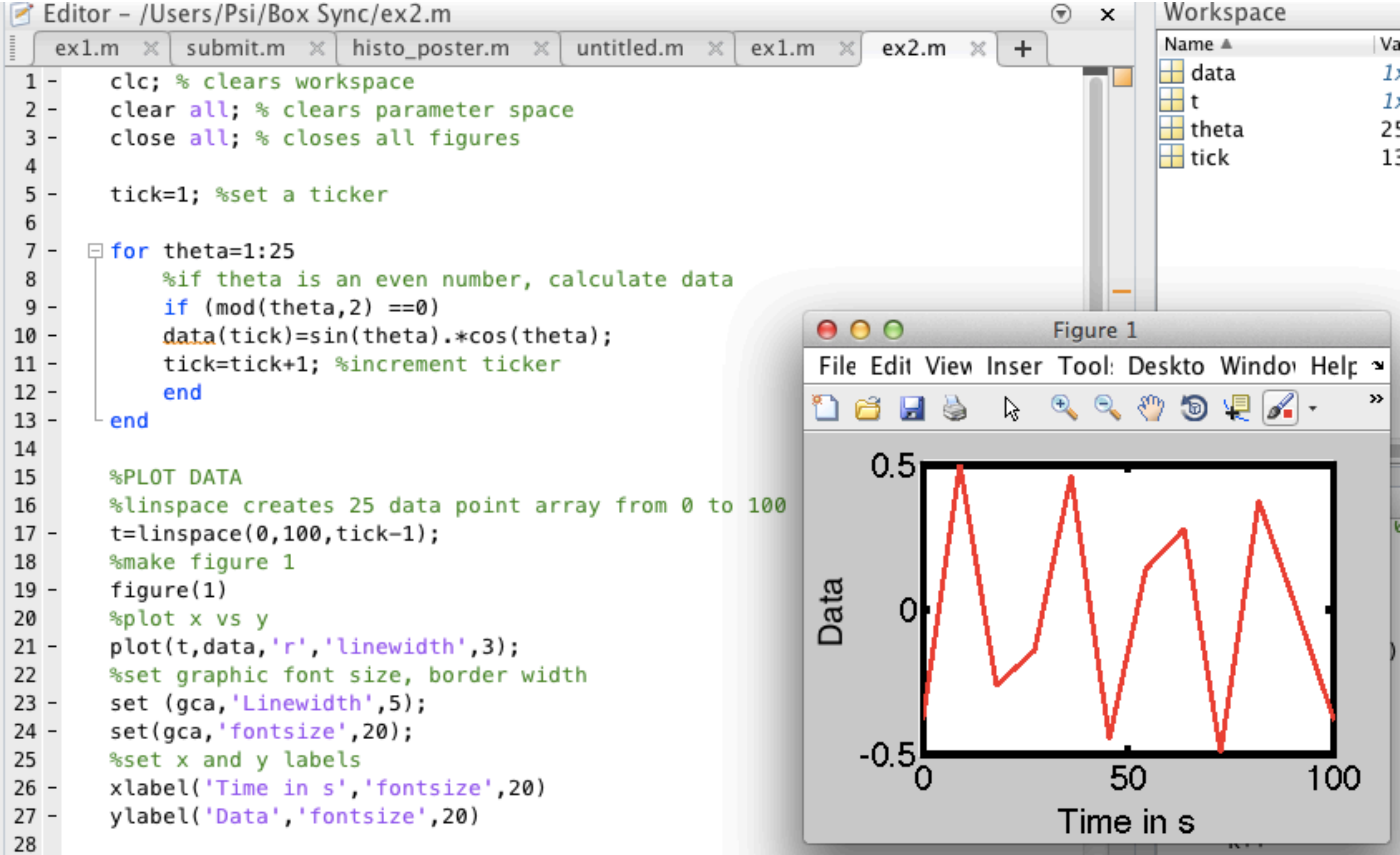


Given matrices A and B

```
A =
     1     0
     2     3

B =
     1     1
     3     4
```

Expression	Evaluates As	Because
A < B	false	A(1,1) is not less than B(1,1).
A < (B + 1)	true	Every element of A is less than that same element of B with 1 added.
A & B	false	A(1,2) & B(1,2) is false.
B < 5	true	Every element of B is less than 5.

If statements



```
clc; % clears workspace
clear all; % clears parameter space
close all; % closes all figures

tick=1; %set a ticker

for theta=1:25
    %if theta is an even number, calculate data
    if (mod(theta,2) ==0)
    data(tick)=sin(theta).*cos(theta);
    tick=tick+1; %increment ticker
    end
end

%PLOT DATA
%linspace creates 25 data point array from 0 to 100
t=linspace(0,100,tick-1);
%make figure 1
figure(1)
%plot x vs y
plot(t,data,'r','linewidth',3);
%set graphic font size, border width
set (gca,'Linewidth',5);
set(gca,'fontsize',20);
%set x and y labels
xlabel('Time in s','fontsize',20)
ylabel('Data','fontsize',20)
```

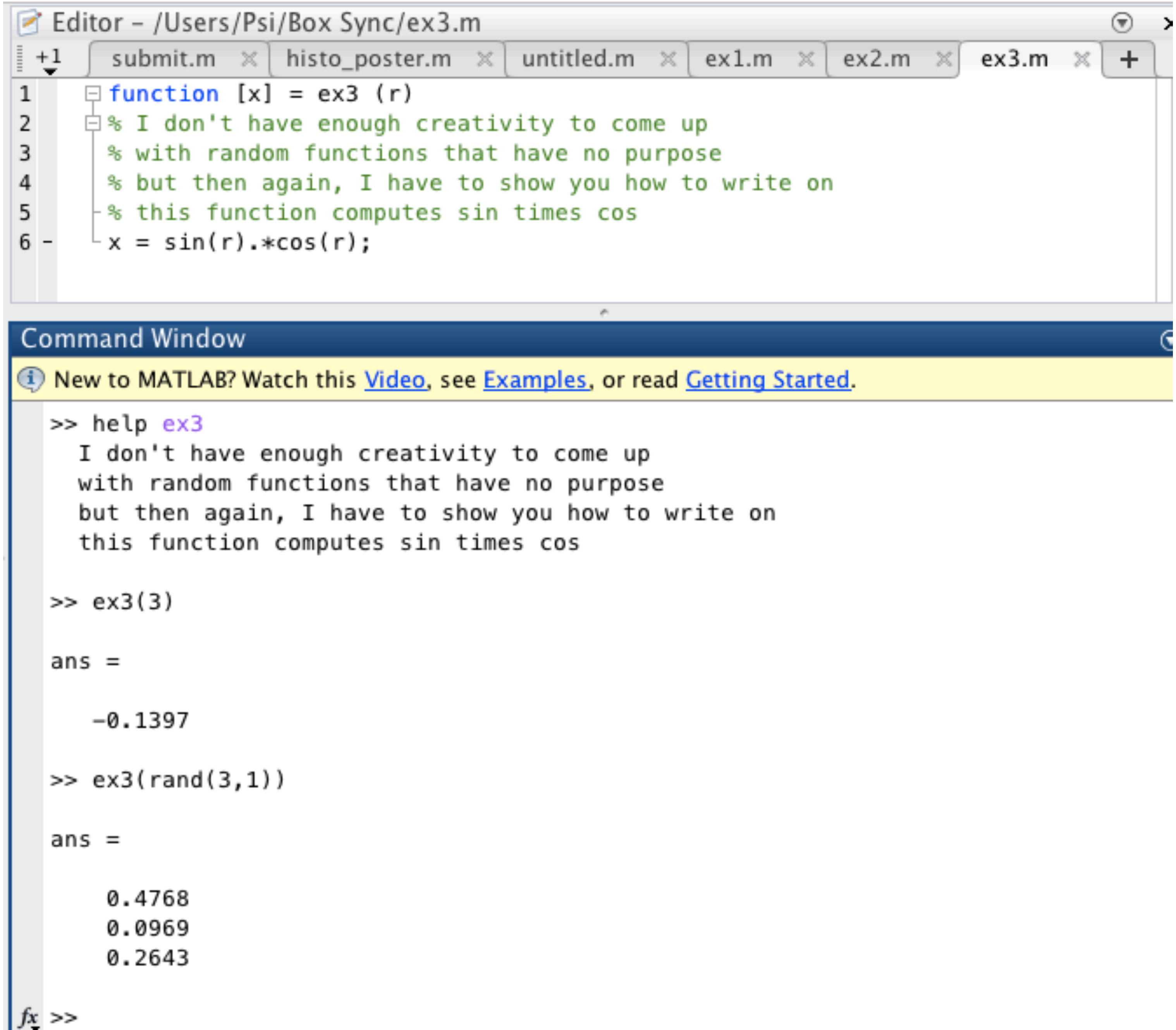
Functions

- Format:

function [outputs] = function_name (inputs)

- Functions make programming easy, your code reusable
- A function need not always output data
- A function can call another function and so on

Function: example 1

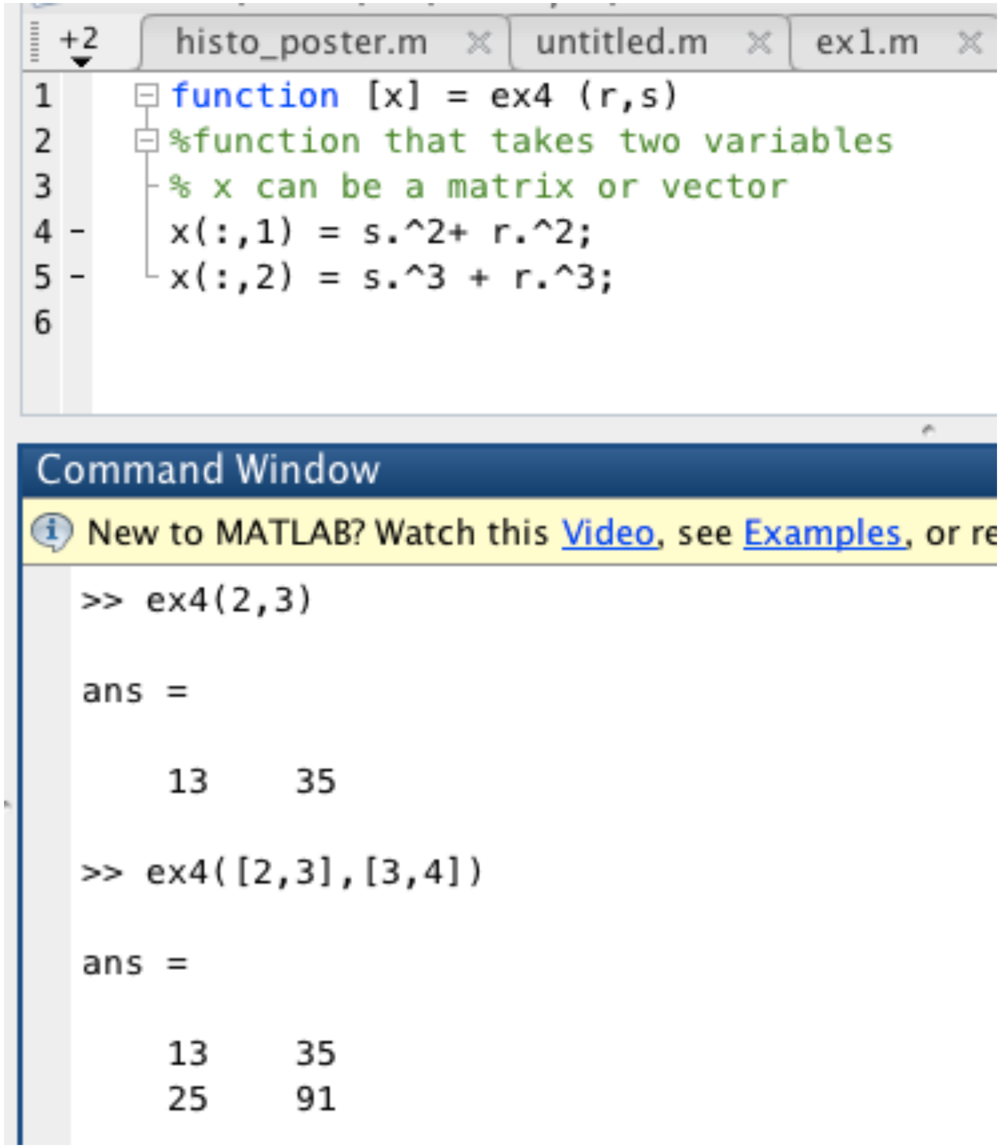


```
function [x] = ex3 (r)
% I don't have enough creativity to come up
% with random functions that have no purpose
% but then again, I have to show you how to write on
% this function computes sin times cos
x = sin(r).*cos(r);
```

New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> help ex3
  I don't have enough creativity to come up
  with random functions that have no purpose
  but then again, I have to show you how to write on
  this function computes sin times cos

>> ex3(3)

ans =

   -0.1397

>> ex3(rand(3,1))

ans =

    0.4768
    0.0969
    0.2643

fx >>
```

Function: example 2

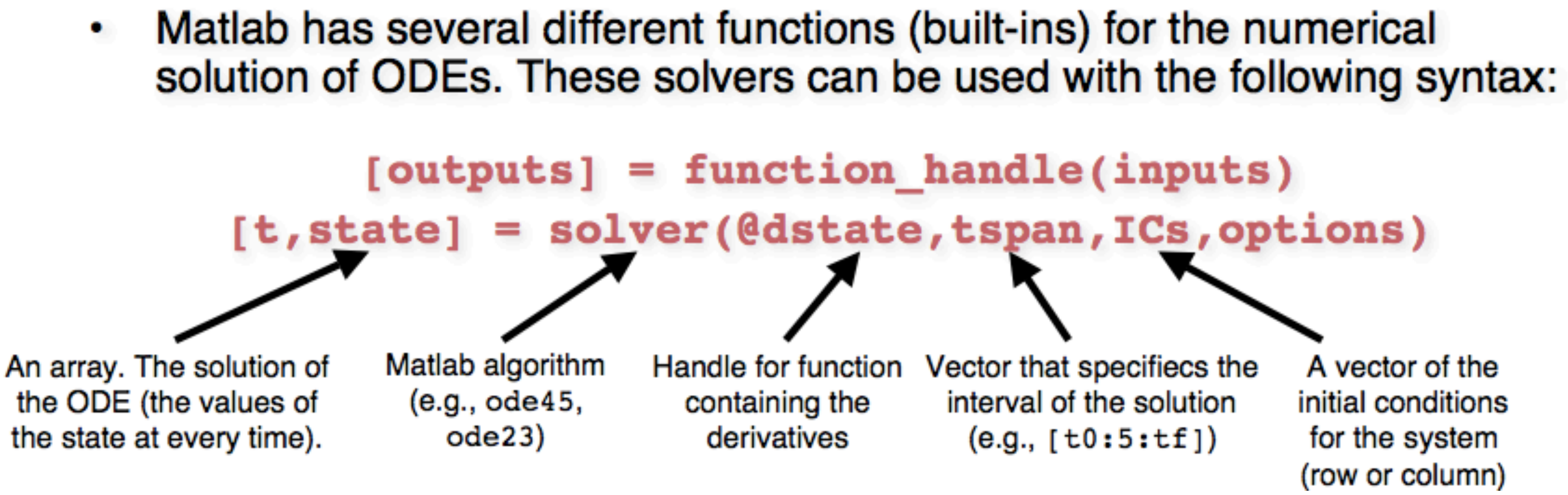


```
function [x] = ex4 (r,s)
%function that takes two variables
% x can be a matrix or vector
x(:,1) = s.^2+ r.^2;
x(:,2) = s.^3 + r.^3;
```

New to MATLAB? Watch this Video, see Examples, or re

```
>> ex4(2,3)

ans =

    13    35

>> ex4([2,3],[3,4])

ans =

    13    35
    25    91
```

Solving ODEs



- Matlab has several different functions (built-ins) for the numerical solution of ODEs. These solvers can be used with the following syntax:

```
[outputs] = function_handle(inputs)
[t,state] = solver(@dstate,tspan,ICs,options)
```

An array. The solution of the ODE (the values of the state at every time).

Matlab algorithm (e.g., `ode45`, `ode23`)

Handle for function containing the derivatives

Vector that specifiecs the interval of the solution (e.g., `[t0:5:tf]`)

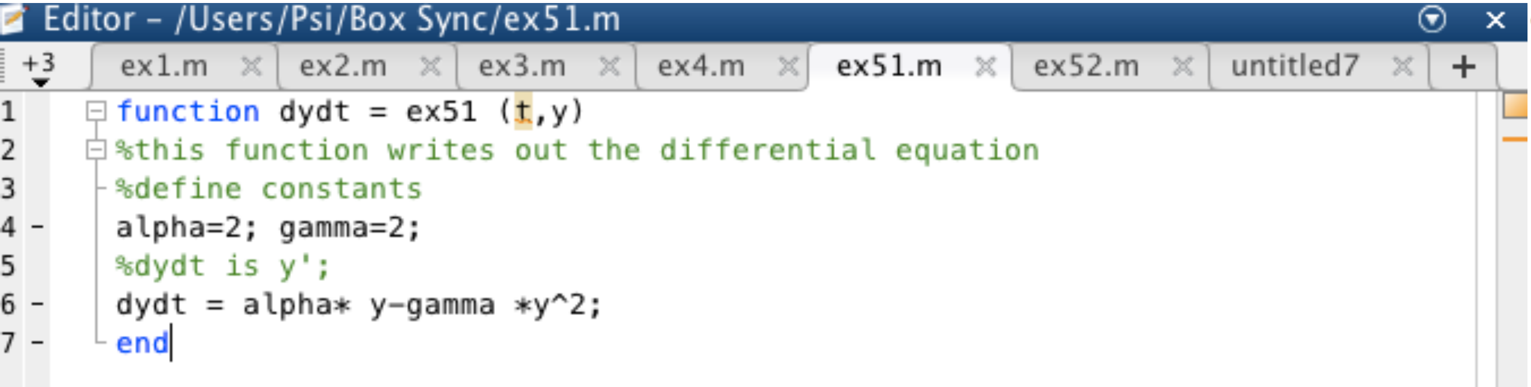
A vector of the initial conditions for the system (row or column)

- Ode45 is a MATLAB function that takes another function (dstate) as input
- You need to write a function dstate, and call ode45 (or other solvers)

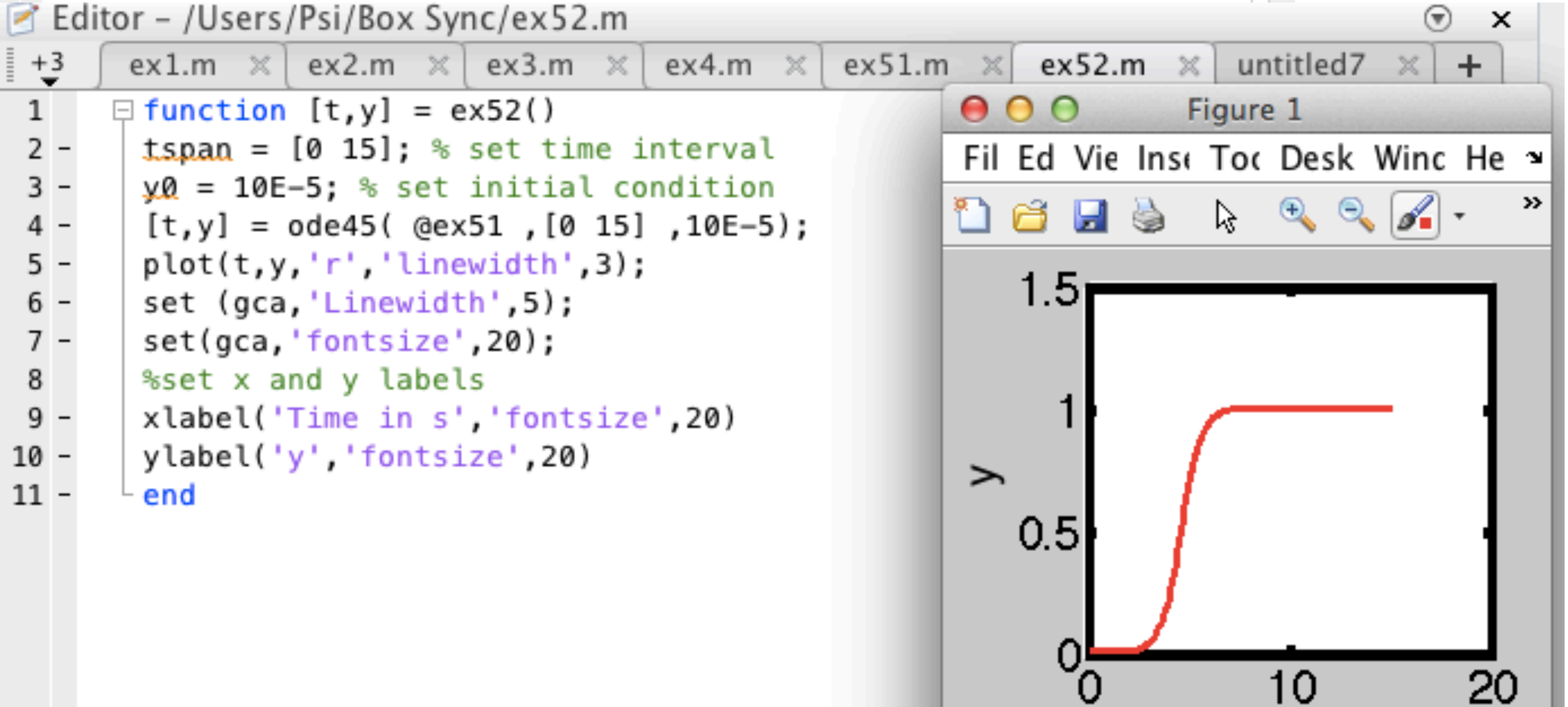
Solving 1 ODE

$$\frac{dy}{dt} = y'(t) = \alpha y(t) - \gamma y(t)^2$$

$$y(0) = 10$$



```
function dydt = ex51 (t,y)
%this function writes out the differential equation
%define constants
alpha=2; gamma=2;
%dydt is y';
dydt = alpha* y-gamma *y^2;
end
```


Editor – /Users/Psi/Box Sync/ex52.m
+3
ex1.m
ex2.m
ex3.m
ex4.m
ex51.m
ex52.m
untitled7
+
Figure 1
Fil Ed Vie Ins Toc Desk Winc He
1.5
1
0.5
0
y
0
10
20

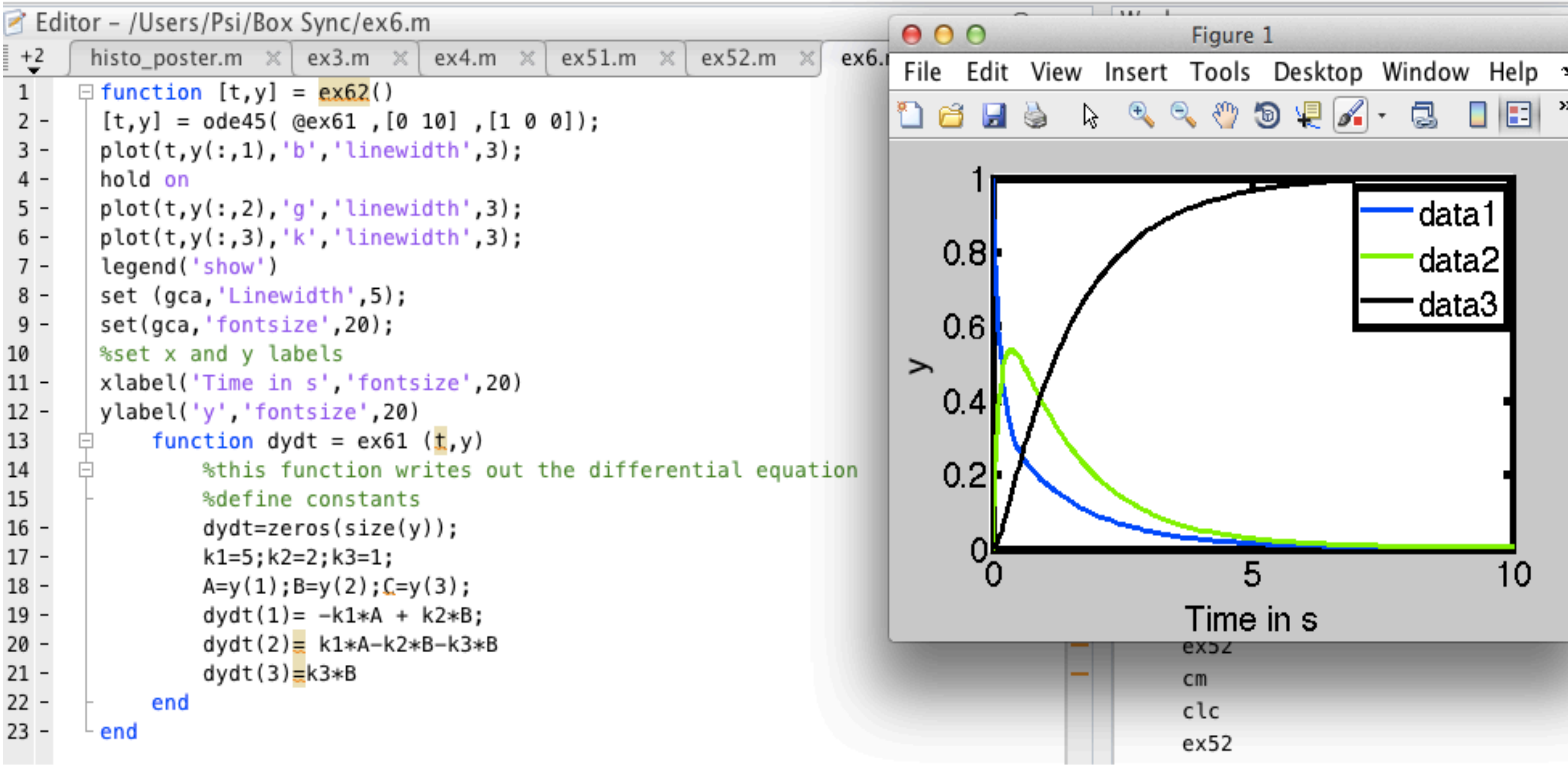
```
function [t,y] = ex52()
tspan = [0 15]; % set time interval
y0 = 10E-5; % set initial condition
[t,y] = ode45( @ex51 ,[0 15] ,10E-5);
plot(t,y,'r','linewidth',3);
set (gca,'Linewidth',5);
set(gca,'fontsize',20);
%set x and y labels
xlabel('Time in s','fontsize',20)
ylabel('y','fontsize',20)
end
```

Solve Rate Equations: system of ODEs

$$A \leftrightarrow B \rightarrow C$$

- Rates: k1,k2 and k3
- Initial value: A0=1; B0=0; C0=0
- Rate values: k1=5; k2=2; k3=1

Solve Rate Equations: system of ODEs



```
function [t,y] = ex62()
[t,y] = ode45( @ex61 ,[0 10] ,[1 0 0]);
plot(t,y(:,1),'b','linewidth',3);
hold on
plot(t,y(:,2),'g','linewidth',3);
plot(t,y(:,3),'k','linewidth',3);
legend('show')
set (gca,'Linewidth',5);
set(gca,'fontsize',20);
%set x and y labels
xlabel('Time in s','fontsize',20)
ylabel('y','fontsize',20)
    function dydt = ex61 (t,y)
        %this function writes out the differential equation
        %define constants
        dydt=zeros(size(y));
        k1=5;k2=2;k3=1;
        A=y(1);B=y(2);C=y(3);
        dydt(1)= -k1*A + k2*B;
        dydt(2)= k1*A-k2*B-k3*B
        dydt(3)=k3*B
    end
end
```

```
ex52
cm
clc
ex52
```

Solver	Accuracy	Description
`ode45`	Medium	This should be the first solver you try
`ode23`	Low	Less accurate than `ode45`
`ode113`	Low to high	For computationally intensive problems
`ode15s`	Low to medium	Use if `ode45` fails because the problem is stiff*

Runge-Kutta (4,5) formula

*No precise definition of stiffness, but the main idea is that the equation includes some terms that can lead to rapid variation in the solution.