UNIVERSIDAD
ANA G. MÉNDEZ
UAGM
Recinto de Gurabo
Escuela de Ingeniería
Ing. José Domingo Pérez

ASSIGNMENT: MATLAB WORKSHOP

Comp 411

CRN: 10154

Prof. Yahya Masalmah

1.6 Exercise

OK, now it's the time to play around a bit with MATLAB :

1. Start MATLAB .

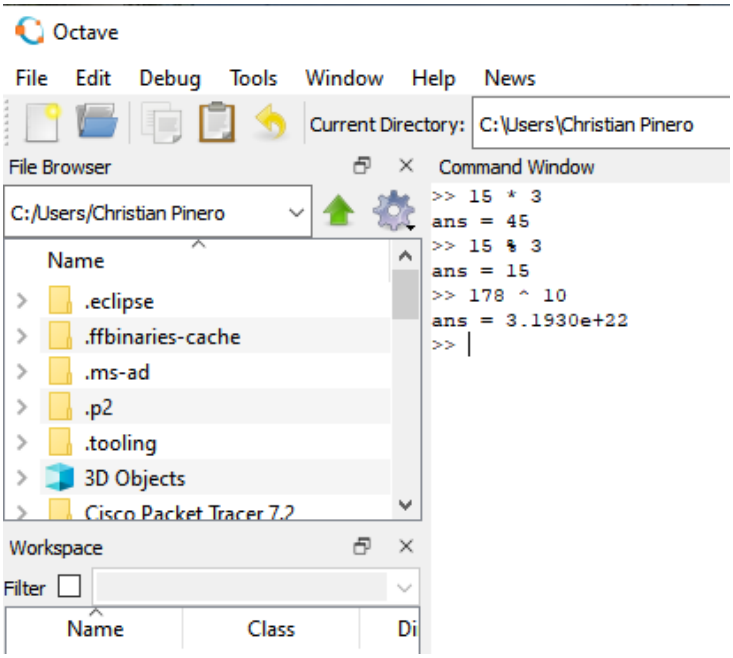
2. Feel free to click around different segments in the MATLAB window, try resizing or closing some of them.

3. Now recover the desktop "default layout", so that your MATLAB window contains the main features shown in Figure 1 again.

4. Change the working directory to "Bioinformatics2010/your name". (Make the directories if needed!)

5. Try out the command window, the easiest way to use command window is to think of it simply as a normal calculator.

6. Done!



2.6 Exercise

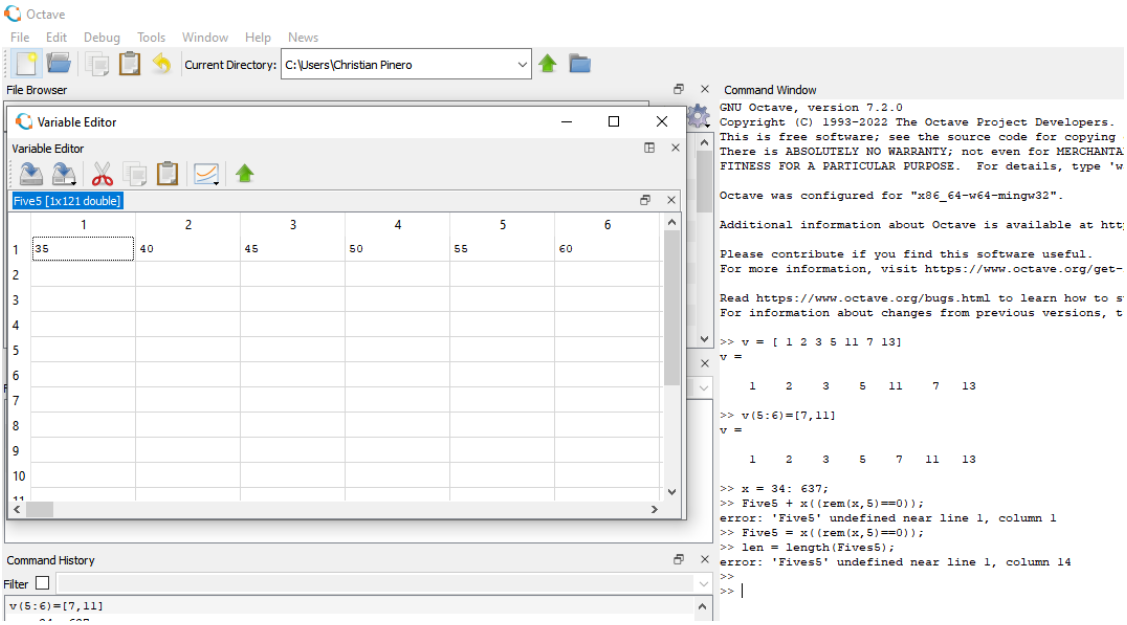
1. Create a row vector **v** with values (1, 2, 3, 5, 11, 7, 13).

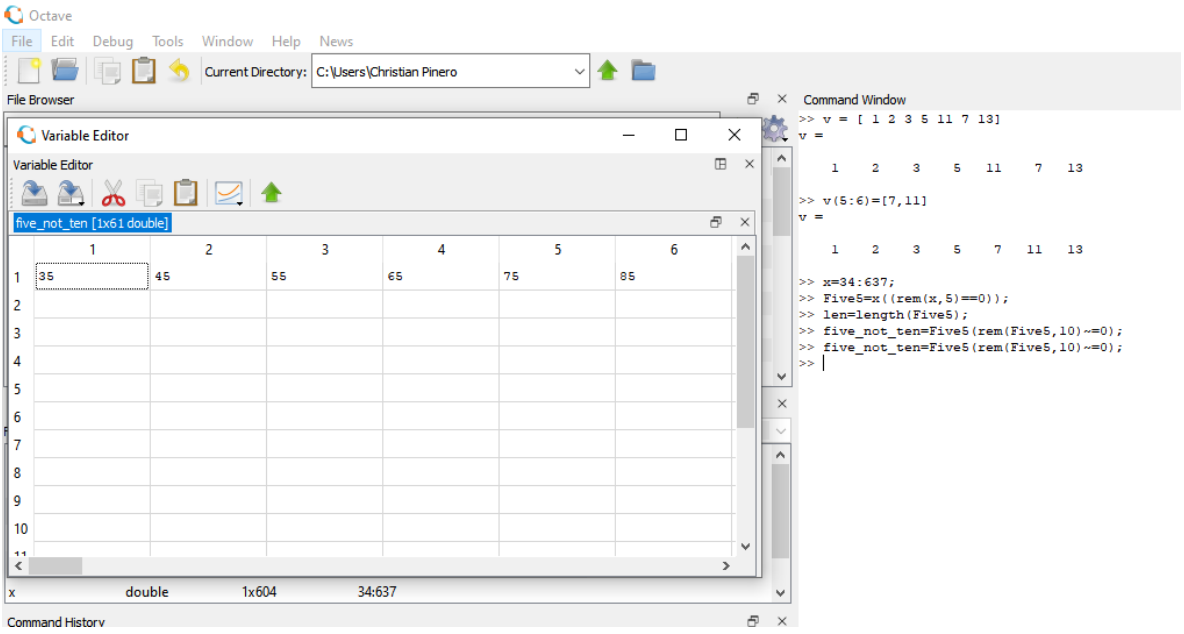
2. Change the value of the 5*th* and the 6*th* element of the **v** to 7 and 11 respectively. Try doing this with only one command as well.

3. Create a vector *Five5* out of all multiples of 5 that fall between 34 and 637. By the way, how many are they? (Tip: look up the command ”length” in help.)

4. Double click on the variable *Five5* in the *Workspace* to inspect your results in the *Variable Editor*.

5. Can you use the colon operator to extract the numbers in the vector *Five5* that are multiples of 5 only but not 10 ?





3.4 Exercise

1. Initialize the random seed with command:

>> S= RandStream(’mrg32k3a’);

2. Create a random vector **u** of 1, 000 uniformly distributed numbers. (Tip: use the *rand* command.)

3. Create a random vector **n** of 1, 000 normally distributed numbers. (Tip: use the *randn* com- mand.)

4. Create a new plot window using the command *Figure*.

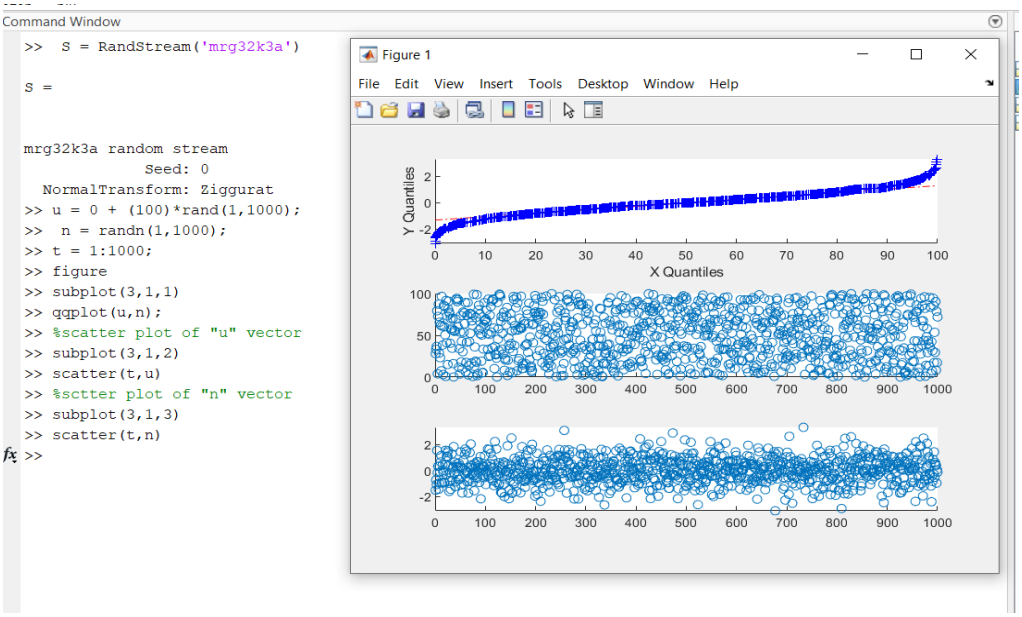
5. Plot the distributions and scatter plot of the elements in **u** and **n** in the figure window. (Three subplots in total.)

6. Make yourself familiar with the interactive plot editor, accessible from ”View Property Editor” in the figure window menu bar. Give fancy labels to your axes and figures, so that they look scientific enough!

7. Save your figures with ”.eps” extension in your folder, and close the figure.

8. Save your workspace to a file called ”My Workspace” in your folder using *save* command.

9. Close the MATLAB , done!



4.5 Exercises

1. Create a function which returns the average of given input vector.

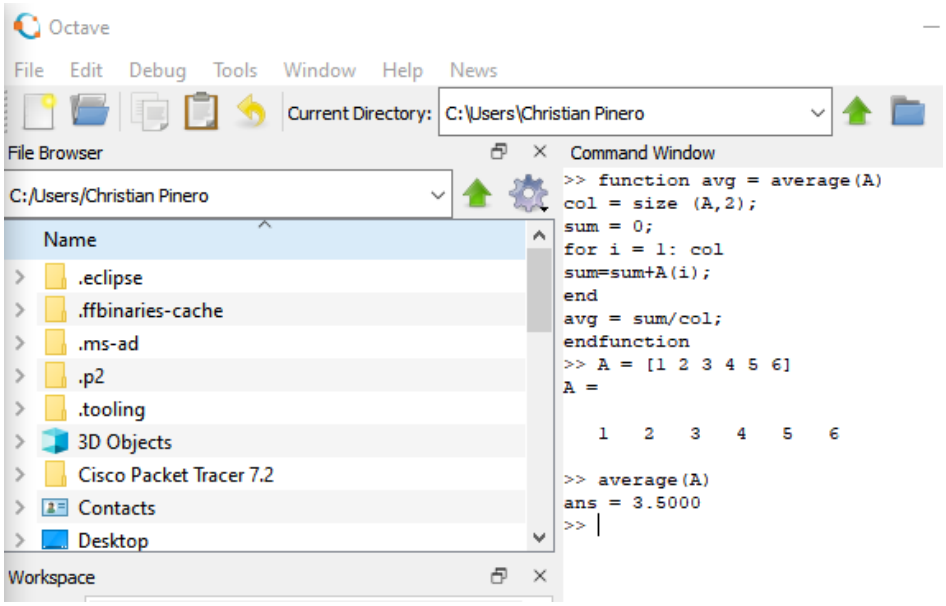
2. Create a function which computes the binomial coefficient $_n = _{n!}$.

3. Create a function which plots sin(ωx) for a given input frequency ω in the range $0 \leq x \leq 2\pi$.

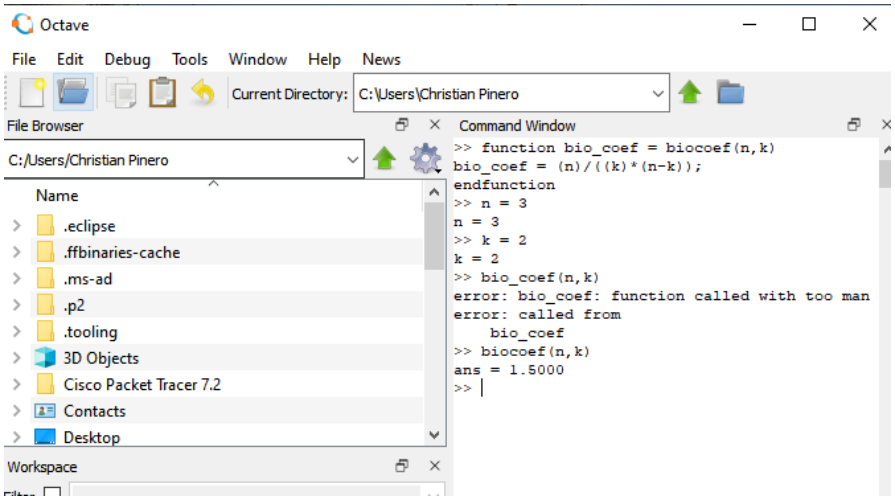
4. Create a function which plots the surface defined by sin($\omega_1 x$) cos($\omega_2 y$) in the range 0 x, y 2π. You can use the meshgrid function to create the x and y matrices.

.

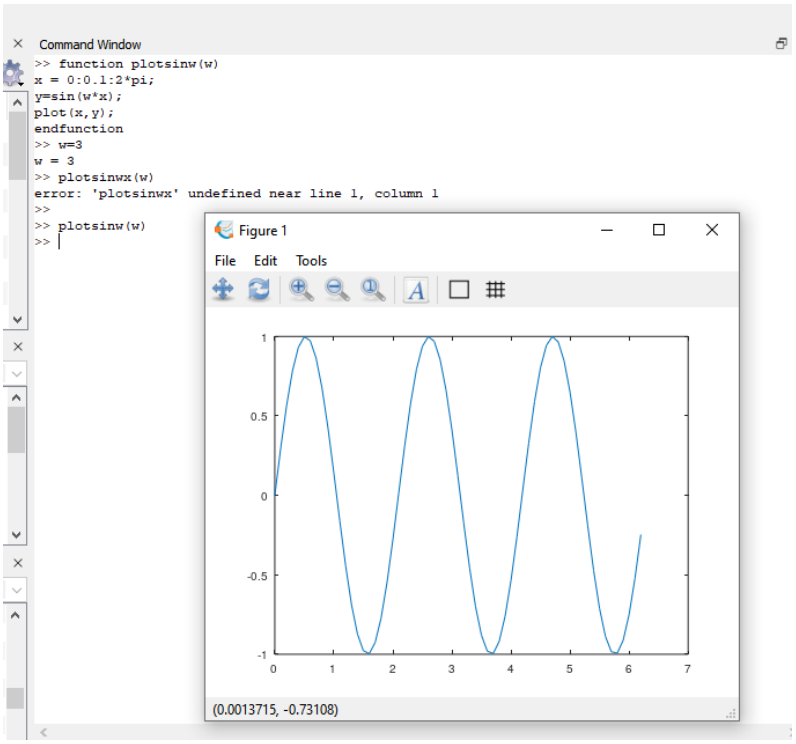
1.



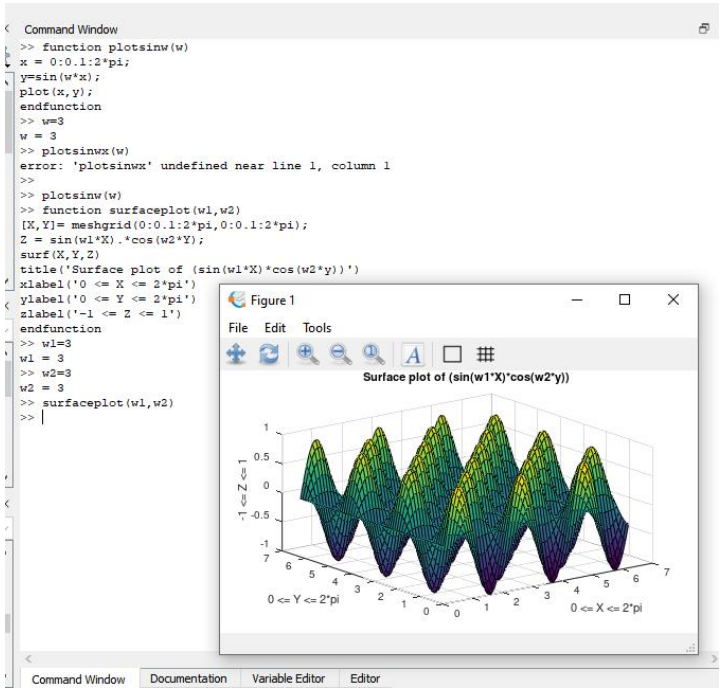
2.



3.


Command Window
>> function plotsinw(w)
x = 0:0.1:2*pi;
y=sin(w*x);
plot(x,y);
endfunction
>> w=3
w = 3
>> plotsinwx(w)
error: 'plotsinwx' undefined near line 1, column 1
>>
>> plotsinw(w)
>>
Figure 1
File Edit Tools
(0.0013715, -0.73108)

4.


Command Window
>> function plotsinw(w)
x = 0:0.1:2*pi;
y=sin(w*x);
plot(x,y);
endfunction
>> w=3
w = 3
>> plotsinwx(w)
error: 'plotsinwx' undefined near line 1, column 1
>>
>> plotsinw(w)
>> function surfaceplot(w1,w2)
[X,Y]= meshgrid(0:0.1:2*pi,0:0.1:2*pi);
Z = sin(w1*X).*cos(w2*Y);
surf(X,Y,Z)
title('Surface plot of (sin(w1*X)*cos(w2*y))')
xlabel('0 <= X <= 2*pi')
ylabel('0 <= Y <= 2*pi')
zlabel('-1 <= Z <= 1')
endfunction
>> w1=3
w1 = 3
>> w2=3
w2 = 3
>> surfaceplot(w1,w2)
>>
Figure 1
File Edit Tools
Surface plot of (sin(w1*X)*cos(w2*y))
-1 <= Z <= 1
0 <= Y <= 2*pi
0 <= X <= 2*pi
Command Window
Documentation
Variable Editor
Editor

5.3 Exercises

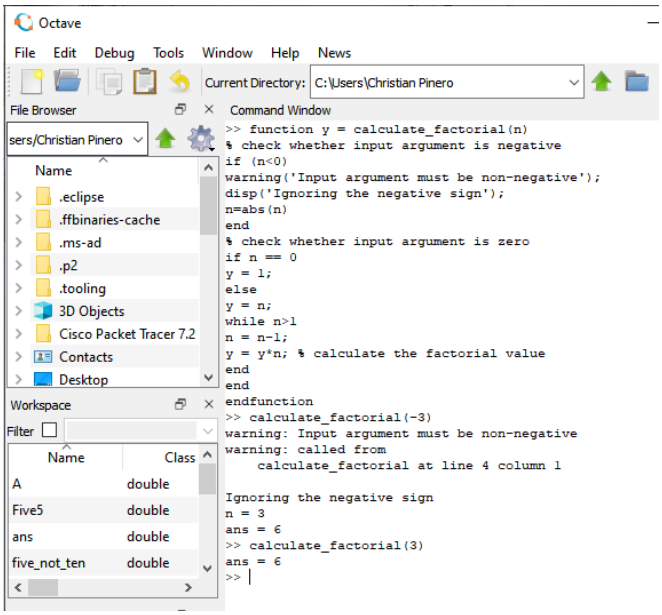
1. Extend the factorial function defined above, such that it checks for a positive input argument.

2. Write a function that calculates a random walk trajectory based on the iteration $x_{i+1} = x_i + E$, where E is a uniformly distributed random variable between -1 and 1. Plot the resulting trajectory.

3. Change the sorting algorithm given above such that it sorts the elements in descending order. Compare the speed of your function with the built-in MATLAB function sort.

4. Approximation of π. The area of a circle is given by $A = \pi r^2$ where r is the radius. Assume a circle with $r = 0.5$ embedded in a unit square. A point in the unit square is defined by $p = (x, y)$ where $0 \leq x, y \leq 1$. If we draw n_{tot} times two uniformly distributed random numbers (x, y) and count how often the corresponding points falls into the circle (n_{circ}), we get an approximation of the circle area by $_n circ$ and by that of the number π. Write a function which approximates π by the described method. How many draws do you need to get the first three digits right?

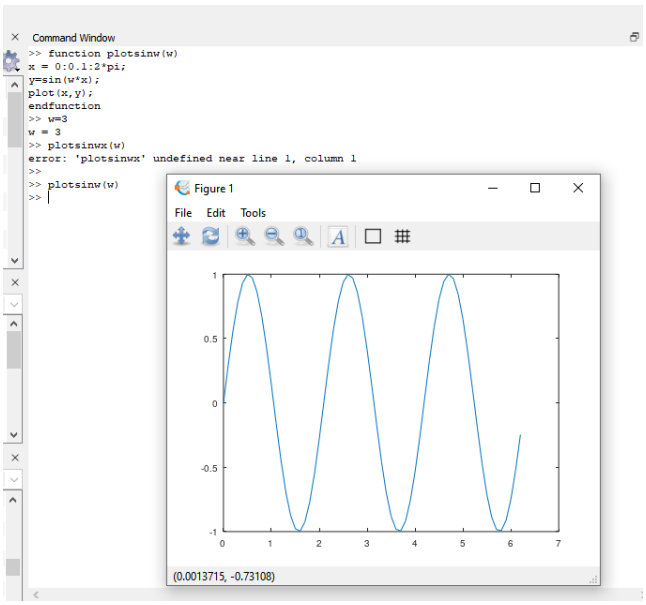
1.



2.

```
>> function RWT ()
e = linspace(-1,1,1000);
x(1)=3;
for i = 1:length(e)-1
x(i+1)=x(i)+e(i);
end
u = x;
v=x;
fprintf("Time taken by userdefine sort is : ");
tic
for i =1:length(e)-1
for j =1:length(e)-1-i
if u(j) < u (j+1)
temp=u(j);
u(j)=u(j+1);
u(j+1)=temp;
end
end
end
toc
fprintf("Time taken by system sort is : ");
tic
v=sort(v,'descend');
toc
endfunction
>> RWT
Time taken by userdefine sort is : Elapsed time is 5.54026 seconds.
Time taken by system sort is : Elapsed time is 0.104161 seconds.
>>
```

3.



4.

```
>> function FOR ()
ntot = 0;
while 1
ntot = ntot +1;
if pi == findPI(ntot)
break
end
end
display(ntot);
function res = findPI(ntot)
nc=0;
for i=1:ntot
point = rand (1,2);
if isInCircle(point)==1
nc = nc +1;
end
end
area = nc/ntot;
% i am calculate pi and rounding it off to 3
res = floor((area/0.25)*1000)/1000;
end
function res = isInCircle(point)
diff = [0.5 0.5]-point;
dist=sqrt(sum(diff.^2));
if dist <= 0.5
res = 1;
else
res = 0;
end
endfunction
endfunction
```

6.1 Exercises

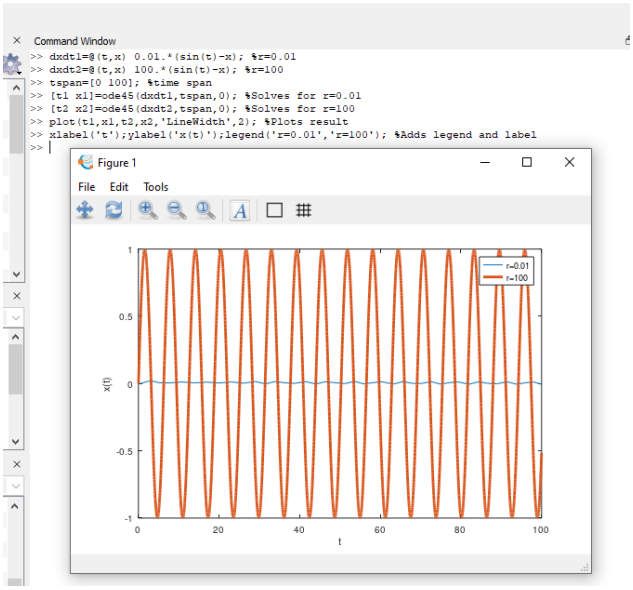
1. Implement the following non-automomous ODE in MATLAB .

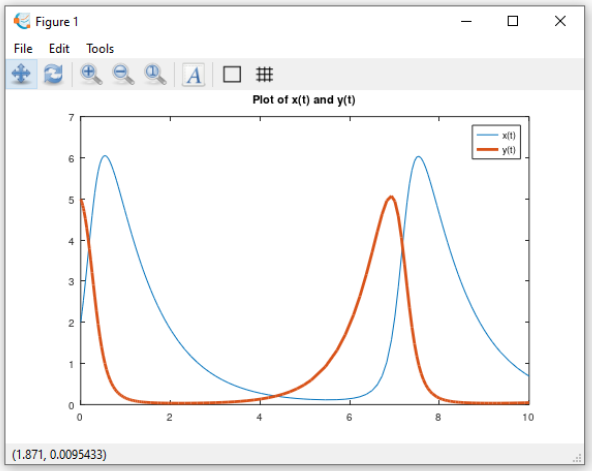
$\dot{x}(t) = r(\sin(t) - x(t))$
Simulate the ODE with $r = 0.01$ and $r = 100$ on the interval $0\ t\ 100$. Plot your solution $x(t)$ together with $\sin(t)$.

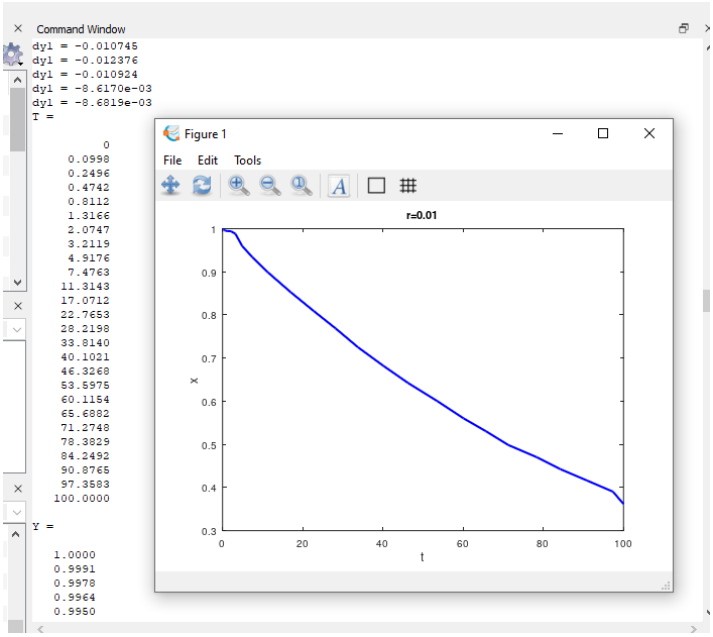
2. The Lotka-Volterra model describes the dynamics of two interacting populations consisting of predator (x) and pray (y).

$\dot{x}(t) = ax(t)y(t) - bx(t)$
$\dot{y}(t) = cy(t) - dx(t)y(t)$
Implement the Lotka-Volterra model in MATLAB and simulate it with ode15s using as parameter values $a = 1$, $b = 1$, $c = 1.5$, $d = 1$ and as initial conditions $x_0 = 2$, $y_0 = 5$. Plot $x(t)$ and $y(t)$ as well as the $y(x)$. Hint: you can plot the phase space $x(t)$ vs. $y(t)$.

1.







Exeercise 8.3

1. Assume the following ODE.

$$\dot{x} = x(2 - \frac{x}{2} - \frac{x}{0.02 + x^2}) \tag{7}$$

Use the `fzero` function to find steady state solutions, i.e., roots of $\dot{x}$. Compare your results by plotting the function in the range $0 \leq x \leq 5$.

2. Find the maximum of the following equation under the given constrain.

$$\arg\min_{x,y} f(x, y) = x + y, \quad x^2 + y^2 = 1 \tag{8}$$

Note: nonlinear constraints have to be code in a separate function:

```
[ineq, eq] = confun(x)
ineq = ... % can be empty if not specified
eq = ...
```

Consult the documentation how to call fmincon in this case.

3. Previously, you have encountered the logistic differential equation. Its analytic solution is:

$$x(t) = \frac{K}{1 + (K/x_0 -) \exp(-rt)}$$

Assume you have the following data set to estimate the unkown parameters K and r.

Time	0	1	2	3	4	5	6	7	8
Data	0.16	0.59	1.75	4.49	7.14	7.78	8.68	8.77	8.67

A common way to find an optimal parameter set is by minimizing is the sum of squared deviations of the solution $x(t_i)$ from the data points D_i.

$$\arg\min_{K,r} S(K, r) = \sum_{i=1}^{n} (x(t_i, K, r) - D_i)^2$$

For simplicity we will assume that $x_0 = D_1$.

- Implement the function to be minimized.
- Call the fminsearch function to find a minimum. Use $K = 1, r = 1$ as starting conditions.
- Plot your final solution together with the data points.
- How stable is your final estimate to different starting conditions for r and K?