

Math 267 MATLAB Practice Lab

For a review of some MATLAB technique, enter the following commands in the MATLAB command window, which has `>>` prompt. You can write a script instead, but we'll do this in the final step.

Basics

- Type `a=3`
Now the variable `a` will have the value 3.
- Type `b=4;`
The colon will suppress the output.
- Type `x = 1:5`
This will make the vector `x = [1 2 3 4 5]`.
- Type `x = 0:0.5:4;`
This will make the vector `x = [0 0.5 1 1.5 2 2.5 3 3.5 4]`. You can create a vector with five components (that is, a matrix with one row and five columns) all of whose entries are zero by using the command `x = zeros(1,5)`
- Type `y(3)`
This returns 1, the third component of the vector `y`.
- Type `y(3) = 0;`
Now the third component of `y` is 0 instead of 1.
- Type `a^2`
This returns the square of the value of `a`.
- Type `x^2`
You should get an error message.
- Type `x.^2`
The period tells MATLAB to square each component of the vector `x`.
- Type `y = sin(pi * x)`
Now each component of `y` is the sin of the product of π with the corresponding component of `x`.
- To clear all your variable values, type `clear all`
Now `a`, `b`, `x`, or `y` will not have any values assigned to them. To start fresh, clearing all variable values and closing all figure windows, type `clear all; close all;`

Symbolic and Numerical Integration

- Type `x = 0:0.01:1 ; y = x.^2 ; approxint = trapz(x,y)`
This gives the trapezoidal rule approximation, with step size 0.01, to the integral $\int_0^1 x^2 dx$. The answer is $\frac{1}{3}$.

If your version of MATLAB has the Symbolic Math Toolbox installed, you can also use MATLAB to do symbolic integration.

- Type `syms k t`
This creates two symbolic variables.
- Type `g = t*sin(k*t); int(g, t)`
This gives the indefinite integral of $t \sin(kt)$ with respect to t . To get the definite integral $\int_0^1 t \sin(kt) dt$, use `int(g, t, 0, 1)` instead.

Plotting

- Type `x = -2*pi:0.1:2*pi ; y = sin(x) ;`
y is a vector whose components are the sine of the corresponding components of the vector x.
- Type `plot(x,y)`
- To label the x and y-axes, and to add a title, type
`xlabel('x-axis')`
`ylabel('y-axis')`
`title('sin(x)')`

- Type
`close all ;`
`x = 0:0.1:1 ;`
`subplot(1,2,1);`
`hold on; plot(x,x); plot(x,x.^2); hold off;`
`subplot(1,2,2);`
`hold on; plot(x,1-x); plot(x,(1-x).^2); hold off;`

This should have produced a figure window containing two separate figures side by side. The command `subplot(1,2,1)` told MATLAB that we wanted a figure window with one row of figures (that's the first argument of subplot) and with two figures in that row (that's the second argument). The third argument informs MATLAB which of the two figures we are working on. The command `hold on` informed MATLAB that we wanted all plots before the `hold off` to appear at the same time. That's why there are two graphs in the left hand figure and two graphs in the right hand figure. Without the hold commands, each successive plot erases the previous one.

- To obtain more information about any function in MATLAB, such as the plot function, type `help plot`

Creating your own function

- Open a new m-file (`File`→`New`→`M-file` or you can use your favourite text editor)
- In this file, type the following:
`function yvalues = myfunction(x)`
`yvalues = x.^2 - 1;`
- Save the file as `myfunction.m` in your MATLAB Current Directory.
NOTE: When saving your function, the filename must always be the same as the function name, otherwise the function will not work.
- Now, in the command window, type `myfunction(1)`
You should get 0. Try entering different values into myfunction.
- Type `x = -2:0.1:2; y = myfunction(x) ; plot(x,y)`
You should see a plot of the parabola $y = x^2 - 1$ on the interval $[-2, 2]$.

Flow control

- Type
`sum = 0`
`for n = 1:5`
`sum = sum + n`
`end`

This computes the sum $1 + 2 + 3 + 4 + 5$.

- Type

```
x = -2:2
for n = 1:5
    if x(n) < 0
        x(n) = -x(n)
    end
end
```

This replaces each component of the vector x with its absolute value. The purpose of this example is just to illustrate the MATLAB `if ... end` syntax. It is easier take the componentwise absolute value of the vector x with the single command `x = abs(x)`

Putting it all together

Although you can do most of your work on the command line, it is often more convenient to write all your commands in a separate file and then run that file in MATLAB. Create a new m-file and call it whatever you want, but include the suffix “.m”. For example `anything.m`. In this file, type the following:

```
clear all; close all ;
x = 0:0.1:1 ;
hold on; plot(x,x); plot(x,x.^2); hold off;
xlabel('x');
ylabel('y=x,x^2');
y = 7
```

Now run the file by typing the name of your file (excluding the .m at the end) in the command window. The last line of the file was put there just to remind you that command outputs are displayed in the MATLAB command window when the line does not end with a semicolon.