# MAT 239 Lab 1

## Introduction to MATLAB

### Adapted from materials developed by Stefania Tracogna, SoMSS, ASU.

## 1 Instructions

1) Do the problems in "Introductory problems."

2) Download **lab_template.m** and **fancyname.m** from the "MATLAB Labs" page of our Bb Learn course and open **lab_template.m** in MATLAB and update the file in **lab_template.m** to reflect the problems you do in "Problems to submit," saving the file as **yourLastName_lab_1.m**.

3) Only include the positively numbered problems in your final, published report, and be sure to include written statements that identify what your final answer is and your interpretation of it.

4) Click "Publish" to view your report, modify your script so that intermediate steps and long vectors are **not** shown (suppress output using **;**), erroneous lines of code are removed, and problems aren't split across multiple pages unless necessary. You can Click "Publish" to view your report after any changes are made.

5) Print your report to bring to class on the due date and send me an email with your **yourLastName_lab_1.m** file attached.

## 2 Introductory problems

**Problem -25.** *Download MATLAB by navigating here and following the instructions.*

**Problem -24.** *Install MATLAB by running the file that downloaded and following the instructions.*

**Problem -23.** *Open MATLAB, type* **demo** *next to >>, press "Enter", and click "MATLAB" to the left under "Category". In MATLAB, the act of typing something, called a* command *and displayed in upright bold font, next to >> and pressing enter is called* running.

**Problem -22.** *Make a note of the sub-categories to the left under "MATLAB" to use as resources when needed.*

**Problem -21.** *Watch the videos in the YouTube playlist linked in the "MATLAB Labs" page of our Bb Learn course, following along in your MATLAB session.*

**Problem -20.** *Run each of the |||-separated commands below separately and document your observations.*

**a=2** ||| **b=2;** ||| **b** ||| **c=3;** ||| **a+pi** ||| **a-b** ||| **a*b** ||| **a/b** ||| **c=4; d=10; c/4, d*2**

**Problem -19.** *Run* **format loose; f=4, g=8**, **format compact; f=4, g=8**, *then either* **format loose** *or* **format compact** *according to your preference.*

**Problem -18.** *Run* **theta=pi/3; cos(theta)^2+sin(theta)^2** *and document your observations.*

**Problem -17.** *Run* **help cosh**, **help elfun**, *and* **help** *separately; document your observations.*

**Problem -16.** *Run* **class=239;** *and* **Class***; document your observations.*

**Problem -15.** *Run* **avo=6.0221e23** *and* **emass=9.1093e-31***; document your observations.*

**Problem -14.** *Run* **format short; 4/3***,* **format long; 4/3***,* **format short e; 4/3***,* **format long e; 4/3***, then* **format short***,* **format long***,* **format short e***, or* **format long e** *according to your preference.*

**Problem -13.** *Run* **1/0** *and* **0/0***; document your observations.*

**Problem -12.** *Run* **v=3:0.3:6** *and* **ls=linspace(1,5,47)***; document your observations. What kinds of objects are* **v** *and* **ls***?*

**Problem -11.** *Run* **y=(v.^2-sin(pi.*v)+exp(v))./(v-4)** *and document your observations. Notice, in particular, that some of the operators here are preceded by periods. This is necessary to apply the operators element-wise to* **v***.*

**Problem -10.** *Run* **plot(v,y)** *and document your observations.*

**Problem -9.** *Run* **v=3:0.01:6;***, press the up arrow on your keyboard, click on the command that defined* **y***, press "Enter," then run* **plot(v,y)***; document your obervations.*

**Problem -8.** *Run* **plot(v,y,'r-')** *and document your observations. Use the* **help** *command to develop a command that will create a similar plot with green plus symbols instead of red line segments.*

**Problem -7.** *Run* **axis([2,7,-1000,1000]); grid on;***,* **title('f(v)=(v^2-sin(\pi v)+e^v)/(v-4)');***, and* **xlabel('v'); ylabel('y');***. What do each of these commands do?*

**Problem -6.** *Click "New" then click "Script." In the* script M-file*, called a script, that appears, write a few commands, a line containing only* %%*, then a few more commands. Click above or below the line containing* %% *and click "Run Section." Document your observations. In the directory bar above the "Editor" window, change the directory to somewhere you'd like to save files for this class. Click "Save" then "Save" or "Save As..." and give the script a name of your choice that begins with a letter and contains only letters, numbers, and underscores. It's good practice to run* **exist('the_file_name_you_want')** *before you save a script or function file, giving the file that name only if the previous command prints* **0***. Close the script.*

**Problem -5.** *In "Command Window," type the name of the script you just saved and press enter; document your observations.*

**Problem -4.** *Click "New" then click "Function." Within the* function definition *(the first line) in the* function M-file*, called a function, that appears, replace* **outputArg1** *with* **y1***,* **outputArg2** *with* **y2***,* **untitled2** *with a function name of your choice that begins with a letter and contains only letters, numbers, and underscores,* **inputArg1** *with* **x1***, and* **inputArg2** *with* **x2***. Within the* function body *(the lines between the first line and the line containing* **end***), replace* **outputArg1** *with* **y1***,* **inputArg1** *with some arithmetic operation on* **x1** *and* **x2***,* **outputArg2** *with* **y2***, and* **inputArg1** *with some different arithmetic operation on* **x1** *and* **x2***. At the end of each of these lines, put a* % *and write a* comment *that details what the line does. Click "Save" then "Save" or "Save As..." and give the file the same name as the function name you chose above. You can give functions like this additional arguments, outputs, or operations by adding variable names within the brackets, variable names within the parentheses, or lines of code within the body, respectively. You can define several functions in the same file by defining them below each other.*

**Problem -3.** *In "Command Window," run* **[z1,z2]=myfn(n1,n2)***, where* **myfn** *is the name of the function you just saved and* **n1** *and* **n2** *are numbers that are in the domain of your function. Document your results.*

**Problem -2.** *Write a script that contains a function definition at its end (copy and paste the template from a new function file) and several lines of code at its beginning, with at least 1 of the lines* calling *the function defined at its end. Save this script-function, run it in "Command Window," then document your results.*

**Problem -1.** *Write a function file containing the function you defined in the previous problem and save it. Write a separate script file that includes the lines of code from the script you wrote in the previous problem and save it. Run the function file then the script file in "Command Window" and document your results.*

**Problem 0.** *Run* **f = @(t,y)(t^2-y); f(4,2)** *and document your results. This is called an anonymous (definition of a) function.*

# 3 Problems to submit

**Problem 1.** *All points of the form* $(x, y) = (r\cos\theta, r\sin\theta)$*, where* $r$ *is a constant, lie on a circle with radius* $r$ *and satisfy the equation Pythagorean identity* $x^2 + y^2 = r^2$*. Create a vector,* **theta***, that has at least 5 unique elements, each representing an angle measurement in radians. Compute vectors* **x** *and* **y** *that contain the* $x$*- and* $y$*-coordinates, respectively, of points on a circle of radius 2 with angles of rotation from* **theta***. Verify that these vectors satisfy the Pythagorean identity.*

**Problem 2.** *The* $i$*th element of a vector, say* $v$*, is denoted* $v_i$*. Create vectors* **x***, with elements 4, 4.1, 4.2, ..., 27, and* **y***, with* $y_i = \exp(x_i/30)\cos(4x_i)/(x_i^2 + 3)$ *for* $i = 1, 2, ..., 231$*. Plot* **x** *and* **y** *with* **y** *along the vertical axis, including a title with an expression for* $y$ *and axial labels. Make a similar plot with small circles for each point instead of a smooth curve. Make a final plot that includes both the small circles and the smooth curve.*

**Problem 3.** *Create a 3-dimensional plot of the helix defined by the parametric functions* $x(t) = 2\sin t$*,* $y(t) = 2\cos t$*, and* $z(t) = 3t$ *using the* **plot3** *function. Use vectors of reasonable length so that the helix looks rather smooth. Add a grid to the plot using* **grid on***.*

**Problem 4.** *On the same plot and with a grid, plot* $y = \cos x$ *in green with a dashed curve and* $T = 1 - x^2/2 + x^4/24$ *in blue with a solid curve for* $-2\pi \le x \le 2\pi$*. Use* **axis tight** *to set the axial limits.*

**Problem 5.** *Solve the differential equation* $y' = 3x^3 - 6x^2 + x - 4$ *analytically. Write a script-function file to plot the solutions to this differential equation that correspond to initial conditions* $y(0) = 10, 4, -5$*. These solutions should be plotted smoothly in different colors or line styles on the same plot, with a titled and legend, for* $-2 \le x \le 4$*. The function you write should be the analytical solution to this differential equation that takes 2 arguments,* $x$ *and the integration constant* $C$*.*

**Problem 6.** *Define the function* $f(a, b) = b^3 - \cos a \exp(b)/(a^2 - b^2)$ *anonymously and evaluate it at the point* $(3, 2)$*. Run* **clear f***. Write a function file for the same function and evaluate the function at the point* $(3, 2)$*.*