CH2013
Computational Programming and Simulations Lab
Aug-Dec 2021
Problem Sheet #1

Aug 17 2021

This serves as your introduction to MATLAB. Please note that you need to upload the codes for each question separately in the appropriate location at grader.mathworks.com, pass the pre-tests, and then submit. For this Problem Sheet, I am allowing unlimited submissions/attempts.

1. Make a new script file and do the following. You will be prompted to save the file before you can run the commands. Save it in a (temporary) folder on the desktop, with your roll number as the folder name. The syntax is usually intuitive, and you can google otherwise.

a) Define a variable **a** which has a value of 4.556789, and a variable **b** that has a value of 5.678342

b) Define a new variable **sum1** that contains the sum of a & b (I mean, get MATLAB to add, don't do it yourself ☺)

c) Variable **difference1** is b subtracted from a

d) Variable **product1** is b multiplied by a

e) Variable **power1** is a raised to b.

f) The command **round** in MATLAB is occasionally useful. Look at the help file to understand it. Variables **sum2, difference2, product2 and power2** are same as above, but with only 3 significant digits in each case. Use the round command to calculate these.

g) Now try these commands

```
fprintf('%.3f\n', 1.4986)
fprintf('%.2e\n', 1.4986)
```

& note the results.

2. You can continue to work with the same script file as above, or open a new one. When you submit your code in grader, you can copy paste the lines of code from your matlab window into grader.

a) Define a column vector '**a**' containing the elements 1, 4.5 & 9.6

b) Define a row vector '**b**' containing the elements 1.6, 8, & 7.3

c) Create a (column) vector **c** = [1 2 …n] with n=5 (don't explicitly write out the numbers, use the '**:**' command)

d) Perform the calculation d = transpose(a) + b to get a new vector d

e) Find the second element of vector a and assign the value to variable 'e'

f) Add the third element of vector a to the second element of vector b. Assign this to a variable 'f'

g) Create a matrix $\boldsymbol{G} = \begin{bmatrix} 7 & 2 & -3 \\ 2 & 5 & -3 \\ 1 & -1 & -6 \end{bmatrix}$

h) Find **H** = inverse of G.

i) Find $x = G^{-1}\, z$ where $z = \begin{bmatrix} -12 \\ -20 \\ -26 \end{bmatrix}$

j) If P= $\begin{bmatrix} 2 & 5 & -3 \\ 2 & 5 & -3 \\ 1 & -1 & -6 \end{bmatrix}$, can you find the inverse of P? What is the warning message you get? Do you understand what that means?

k) In MATLAB "element by element" multiplication is achieved using the symbol "**.***" (without quotes). Check that G.*P yields the result you expect. Put this in a new variable **Q**. Perform regular (not element by element) matrix multiplication of G and P, and put this in **R**. Check that Q & R are not the same.

l) Create a 4x4 identity matrix and call if '**S**' (again, don't explicitly type out the 16 numbers, use matlab's short commands – google to find out what the command is)

m) Pick out the diagonal elements of matrix **S** and put it in a vector '**t**'

n) The command to extract the second row of a matrix A is A(2,:) in MATLAB. Find the third COLUMN of the matrix P defined above and assign to the vector '**u**'

3. This question pertains to some simple codes using the for loop function in MATLAB. Again, the syntax is simple, look it up.

a) Create a **for loop** that prints the following statement 10 times [Where ***i*** is an index/integer variable]

```
fprintf("%i\n", i^2)
```

b) The solution of the equation $f(v) = v - e^{-v} = 0$ can be obtained like this – A vector v is created, with each subsequent element containing a better estimate of the solution to the above equation. The initial guess is taken as 0.1. In MATLAB, vector/matrix indices start at 1. So, assign this value to the first element of vector v. Now set up a **for** loop that runs over index **j** from 1 to 6, and executes the following statement: $v(j+1) = e^{-v(j)}$ every time. Report the **final value of** v, after exiting the for loop, as **vfinal**. Check the values and convince yourself that this makes sense.

4. In this question we want to make several plots (graphs). Please make sure you use the help feature in the command window to get the correct syntax. The commands you should read up about include – **linspace, plot, figure, hold on, hold off, semilogx.**

a) The independent variable $x\epsilon[0, \pi]$; and I want you to use 10 equally-spaced intervals to get the discrete values of x.

b) Create a (linear) plot of y vs. x where $\boldsymbol{y} = \mathbf{cos}\ (\boldsymbol{x})$**.** Make sure that the plot is a green coloured, solid line, with "+" signs used as markers.

c) Create a (linear) plot of z vs. x where $\boldsymbol{z} = \mathbf{exp}\ (-\boldsymbol{x})$. Make sure that the plot has red coloured, "+" markers, and that the line is a dotted, cyan-coloured one. I want both the above curves (b & c) to be in the same plot.

d) Create two subplots using the commands below and plot ***y vs. x*** & ***z vs. x*** in the first and second subplot, respectively. This should be a new plot.

e) Create a semilog plot of ***z vs. x.*** This should also be a new plot. *(So totally, there are going to be three 'figures' in the solution to this question)*.

Note that once the figures are plotted (a new window will pop for each figure), you can click on the plot and do some editing. You can change the axis, add some text, change the font, modify the legend etc. Try this out. This is not required for submission, but worthwhile for learning.

--END--