Summer School in Computational Sensory-Motor Neuroscience

The first day of Matlab tutorials is designed to get a feeling where you are and how much variance there is in the course in terms of Matlab expertise. For some of you the exercises will be trivial and for others they will be a challenge. Everything is fine. Please team up with participants who can help you, or team up with participants whom you can help.

From each of you I would like to get feedback that allows me to assess how much you know about Matlab. Please pencil for each bit of information and each exercise a number indicating how you dealt with it. The code for the numbers is:

- (1) No problem
- (2) I figured it out
- (3) I would figure it out if I only had a bit more time
- (4) I don't have the background and it would take me considerable time
- (5) No chance, entirely lost.

What you need to know about Matlab

A brief summary of matlab function you may find useful:

File Management

- load, save
- textread, textwrite, textscan
- wavread, wavwrite
- dlmread, dlmwrite
- xlsread, xlswrite, csvread, csvwrite

Generic Data/Variable Management

- size, length, numel
- cat, strcat, sum, diff, prod
- mat2str, str2num, num2str, mat2cell, cell2mat, num2cell, cellstr, char, logical, int32, etc
- end (when used as an array index [eg., test_array(2:end)]
- repmat, reshape, squeeze, shiftdim
- sort
- cell, struct, zeros, ones

Logical Operations

- ==
- ~
- ||, &&
- <, >, <=, >=

Simple(r) Mathematical Operations

- trigonometric functions (sin, cos, tan, sind, asin, atan, atan2, etc.)
- logarithmic/exponentiation functions (log, log10, exp, sqrt)
- round, ceil, floor, mod, abs,
- transfer functions (compet, hardlim, hardlims, poslin, purelin, radbas, tansig, etc) [these are in the neural network toolbox, but most are simple to recreate yourself]

Generating Randomness

- rand, randn, random, randperm, normrnd

Statistics

- simple descriptive statistics: mean, mode, median, std, var, min, max, kurtosis, skewness, norminv, zscore
- pdf, cdf, icdf, distribution-specific functions(normpdf, normcdf, expcdf etc.)
- corrcoef, cov, ttest, ttest2, ztest

Figures

- plot, figure
- image, imagesc (imread, imwrite)
- hole on, hold off
- subplot

Flow Control / General scripting

- for
- while
- if... elseif... else...
- switch
- try-catch
- function
- inline

Signal Processing

- ifft, fft
- filter
- 'angle' and 'real' may be useful

Solving linear equation systems, underconstrained and overconstrained: The backslash operator

- \

Help

- help
- which
- helpdesk

Helpful help pages

- "Operator Precedence"
- "Graphics"
- "Cell Arrays"
- "Parametric Distributions: Continuous Distributions (Data)"
- “Using Logicals in Array Indexing”
- “Vectorizing Loops”

Data structures

- scalar
- vector
- array
- multidimensional array
- structure
- cell array
- string

Exercise 1: Matrix operations, relational operators, logical indexing

1. Create the following matrices in your workspace:

$$A=\begin{bmatrix}1 & 21 & 6\\ 5 & 17 & 9\\ 31 & 2 & 7\end{bmatrix} \qquad B=\begin{bmatrix}1 & 64 & 122 & 78 & 38 & 55\end{bmatrix}$$

$$C=\begin{bmatrix}4\\ 22\\ 16\\ 160\end{bmatrix} \qquad D=\begin{bmatrix}8 & 41 & 166 & 42\\ 55 & 28 & 16 & 2\\ 0 & 0 & 1 & 12\\ 25 & 65 & 24 & 19\end{bmatrix}$$

$E=\begin{bmatrix}65\end{bmatrix}$ $\qquad F=$ 3 x 4 matrix of 1's

Remove the 2nd row from matrix D

Add a column of 4's to the end of matrix A

Add 15 to every element in the first row of matrix A

Subtract 5 from every element in matrix B

Remove the last row of matrix C

Create a 3-dimensional matrix G by concatenating matrices D and F

Find a quick way to confirm the number of dimensions of G

Use logical indexing to find the zeros in D and replace with them with -1's

Create the dot product of matrices A and D

Reverse the order: Try to create the dot product of D and A. Why doesn't it work

Create a matrix H that is the dot product of the first 3 columns of matrix A and the first 3 columns of matrix D

Use relational operators and logical indexing to find the elements of A that are greater than 10

Exercise 2: Some important concepts

Functions vs scripts

What is the difference between a function and a script? What are advantages/disadvantages of both?

Are you familiar with the concept of the "cell" in Matlab scripts? Check whether *cell mode* is enabled. For the following exercises, rather than typing commands directly into the command line, use a script, turn cell mode on, and use it to run only the bits you are currently working on.

Plotting

Create vectors x and y in your workspace. x = [3 4 5 6 7 8 9 10], y = [17 23 20 29 32 40 38 45]

Plot y against x as a scatter plot. How can you change symbols and their colour?

Use the *backslash operator* to find intercept and slope of the regression line for this data set

Add the regression line to the scatter plot using *refline* (if you have access to the statistics toolbox)

Do the same but using the *line* command

Control flow

Are you familiar with basic flow control structures?: for, while, if … else

Use loops and/or condition statements to create the following matrix

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \\ 26 & 27 & 28 & 29 & 30 \end{bmatrix}$$

Do the same but just using the *colon operator* and the *reshape* command.

Without using the find command or changing any element manually, use loops and/or condition statements to change every even-numbered element in A to be equal to the product of its row and column index.

For example row 2 column 1 of A should be changed from 6 to 2 i.e, A(2,1) = 6 should be changed to A(2,1) = 2*1 = 2, and so on…

Vectorization

Only using *for* loops, write a function that computes the dot product of two matrices.

Create two matrices B and C, each with 1000 rows and 1000 columns.

Check the result against the matlab built in *dot* function (which is the function behind the * operator).

Use the *profile* function to compare processing speed of the function you wrote with the built in *dot* function

Exercise 3: Fourier analysis

Generate two vectors A and B that each contain 10 seconds worth of "signal". Each is sampled at 1000 Hz. Signal A is a pure sine wave with a frequency of 60 Hz and an amplitude of 12. Signal B is also a pure sine wave, but has a frequency of 220 Hz and an amplitude of 5. Add the two signals together to make the complex signal C. Plot the single signals and the compound signal at a resolution that allows you to visualize them well.

Now, forget how you made signal C. Use Fourier analysis to recover the frequencies of the pure components. Plot the power spectrum. The Matlab function *fft* implements the Fast Fourier Transform.

Try the same but add Gaussian white noise to the signal (use the *randn* function). What happens to the power spectrum?

Given the power spectrum, could you reconstruct the original signal? If so, how would you do that? If not, what is the problem?

Exercise 4: Principle components analysis

I will provide you with motion capture data of a walking person. The data are contained in a 3-dim matrix with the first dimension encoding time, the second the individual markers, and the third their Cartesian coordinates in body centred coordinates.

I will also provide you with a function that allows you to animate the data in terms of a point-light display.

Each sample in time contains a “posture”. With 15 markers and three spatial dimensions each posture is a 45-dim vector.

- Compute the principle components of the set of postures.
- How much variance does each component contribute to the data set?
- How do the loadings of the first few principle components change with time?
- Display individual components. What do they encode?