Linux and Matlab Basics

Johannes Grassberger, ICTP

Computers at ICTP

- Dual boot computers start up in either Linux or Windows. If the computer does not run the OS you want or need, reboot it
- When you finish using a lab PC, don’t forget to log off!
- In Windows, don’t save your files in the usual “My Documents” etc. folders, but on the network drive O: which is your centralized, backed up storage accessible from all PCs.

Linux distributions

- Red Hat Enterprise Linux (RHEL)
- CentOS
- Fedora
- Ubuntu (with variants, e.g. Kubuntu, Xubuntu, Mint, Ubuntu Studio,…)
- Debian
- OpenSuse

Desktop environments

- KDE
- Gnome
- Cinnamon
- MATE
- Xfce

Package Managers

- Work like app stores
- Software packages contain dependency information, i.e. which other packages are required. Package managers will automatically install all packages that are needed.
- Wealth of free (and open-source) software
- In Ubuntu (and friends): sudo apt install *package-name*

Command line

- To make the most out of Linux, become familiar with UNIX commands.
- To get started, see the tutorial at

http://www.linuxcommand.org/

Scientific Software Tools mathematics

- Mathematica
  (commercial)
  by Wolfram (www.wolfram.com)
- Maple
  (commercial)
  www.maplesoft.com
- Sage
  (free)
  www.sagemath.org

Scientific Software Tools numerical

- Matlab
  (MATrix LABoratory)
  (commercial)
  www.mathworks.com
- GNU Octave
  (free)
  www.gnu.org/software/octave
- SciLab
  (free)
  www.scilab.org

Scientific Software Tools
programming languages

- Classic compiled languages
  Fortran, C, C++, etc.
  Many useful libraries, e.g.: BLAS, Linpack, Lapack, fftw
- Script languages
  Python
  widely used
  Many useful libraries, e.g.: numpy, scipy, matplotlib

  R
  good for statistical computing and graphics

Matlab
basic commands

- Numerical calculations

  ```
  2*3
  5.2^2
  sqrt(7)
  sin(pi/3)
  sqrt(-4)
  ```

- Assignment

  ```
  x=pi/3
  y=sin(x)
  ```

- Silent operation

  ```
  z=exp(2);
  ```

Matlab vectors

- Define individual values

  a = [pi 3/2 -7]

  b = [3; 2; 1]

- Define a sequence of numbers

  c = 1:5

  d = 1:2:5

  e = linspace(1,10,5)

- Transposition

  f = e’

  g = (1:2:5)’

Matlab
vector operations

- Vector with scalar
  d + 0.5
  d * 2
- Vector with vector
  d + [1 2 3]
  d .* d

Matlab matrices

- Defining matrices
  m = [1 2 3; 0 4 5; 0 0 6]
  m2 = rand(3)
  z = zeros(3)
- Matrix multiplation
  m * m2

Matlab
getting help

- Getting documentation on a specific function
  doc rand
  doc zeros

Matlab workspace

- Save all or some variables to a file
  save *filename*
  save *filename variablename*
- Clear the workspace
  clear
- Load all or some variables from file
  load *filename*
  load *filename variablename*

Matlab
elements of arrays

- Specifying single components
  a= [1 2 3; 4 5 6; 7 8 9]
  a23 = a(2,3)
- Getting a row or column
  row2 = a(2,:)
  column1 = a(:,1)
  lastrow = a(end,:)
- Getting part of a vector
  x=linspace(1,5,10)
  y=x(3:5)
  z=x(6:end)
- Setting a value
  x(1)=0

Matlab
logical expressions

- Logical operators work similar to arithmetical ones, with results being 1 for true and 0 for false
```
3 < pi
n = 7
cond = (3 < pi) & (n > 7)
```
- They also work on arrays
```
x = 0:0.1:2*pi
y = sin(x)
y>0.5
```
- Extracting values from an array with certain condition
```
y(y>0.5)
x(y>0.5)
```

Matlab
getting minima and maxima

- Get the maximum or minimum value of a vector
  max(y)
  min(y)
- Also get the index where these values are located
  [ymax, ymaxindex] = max(y)
  [ymin, yminindex] = min(y)

Matlab
creating, plotting, analysing data (1)

- Consider audio signal with frequencies A (440Hz) and A sharp (466Hz), sampled at 2000Hz
- Define the frequencies
  ```
  f1=440
  f2=466
  samplerate=2000
  ```
- Define a sequence of time values from 0 to 0.2 seconds
  ```
  t = 0:1/fs:0.2;
  ```
- Calculate amplitude over time
  ```
  amp=sin(f1*2*pi*t)+sin(f2*2*pi*t);
  ```
- Plot the amplitude against the time
  ```
  plot(t,amp)
  ```
- Do a Fourier transformation
  ```
  spectrum=fft(y);
  ```

Matlab
creating, plotting, analysing data (2)

- Get number of elements of array
  n=numel(amp)
- Calculate the frequencies corresponding to the fft output values
  f=0:fs/n:fs*(n-1)/n;
- Plot the frequency spectrum
  plot(f,abs(spectrum))
- Get maximum frequency value and location
  [maxf, maxind] = max(abs(spectrum))
- Get frequency
  f(maxind)

Thanks!