GNU/Qt Octave

Rohin Kumar Y,

Dept. of Physics & Astro Physics,

University of Delhi

What is Octave?

- Octave is an interactive high-level programming language specifically suited for vectorizable numerical calculations.
- The syntax of Octave resembles that of Matlab.
- An Octave program usually runs unmodified on Matlab.
- Matlab, being commercial software, has a larger function set, and so the reverse does not always work (esp. when the program makes use of specialized add-on toolboxes for Matlab)
- GNU or Qt version?
- It is FREE!!!
- Happy Ugadi! :)

Intro

- The Most important command of octave

"HELP" :)

- All commands can be typed in at the prompt or read from a script.
- Scripts are plain text files with file suffix .m. They are imported by calling the file name without the suffix and behave as if their content was typed in line by line.
- ';' separates several commands within a line. A command terminated by ';' does not display its result on-screen.
- ',' separates two commands without suppressing on-screen output.
- '…' at the end of the line denotes that an expression continues into the next line.
- Comments are preceded by %.
- Octave is case sensitive. And don't use filenames same as or similar to available functions

Variables and Standard operations

- varname = expression assigns the result of expression to varname.
- Octave has all the usual mathematical functions +, -, *, /, ^, sin, cos, exp, acos, abs, etc.
- It understands basic constants like pi, e, i etc.
- The operators of elementary logic are:
- < smaller <= smaller or equal & and
- > greater >= greater or equal | or
- == equal ~= not equal ~ not
- When debugging user-defined objects, the following commands are useful:
- 'whos' shows all user-defined variables and functions
- 'clear name' clears name from memory; if no name is given, all variables and functions will be cleared. (clear all & clc)
- 'type name' displays information about the variable or function name on the screen.

Input and output

- 'save data var1 [var2 ...]' saves the values of variables var1 etc. into the file data.
- 'load data' reads the file data, restoring the values of the previously saved variables.
- blah=input(”Some Message”);
- 'fprintf(string[,var1,...])' resembles C syntax for formatting output(even printf (”blah\n”) works!
- 'format [long|short]' enlarges or reduces the number of decimals displayed. Calling format without argument restores the default.
- ' pause' Suspends evaluation until a key is pressed.
- 'history n' Shows last n commands
- 'what', 'pwd', 'cd', 'ls' work as usual
- 'echo', 'diary file',
- Data Structures work in a really simple way and functions can return structures

Vectors & Matrices

- Row Vector, Column Vector

A=[1,2,3] B=[1;2;3]

- General Matrices A=[1,2;3,4]
- ones(m,n)
- zeros(m,n)
- rand(m,n)
- eye(m,n)
- diag(V,k)
- linspace(x1,x2,N)
- logspace(x1,x2,N)

Basic Matrix Addressing & Operations

- *,^,.^,.*,+,-,'
- A(m,n),A(m,:),A(:,n),v(k),v(m:n),length(v),[r,c] =size(A)
- inv(A), A\b, lu(A), eig(A), [v, d]=eig(A), find(A), any(A), rank(A), roots(v)
- abs(z), imag(z), real(z), angle(z),conj(z)
- Lsode for 1st order DE

[X, ISTATE, MSG] = lsode (FCN, X_0, T, T_CRIT)

- Any thing else?

Control Structures

- Loops, Branching
- While, for, if-else, switch
- Break and continue work as usual
- Global Variables
- Functions, Functions of Functions

Functions

- function y = f (x)

  y = cos(x/2)+x;

end

- function [out1,out2] = dolittle (x)

  out1 = x^2;

  out2 = out1*x;

end

Functions of Functions

- function y = gauss(x)

```
y = exp(-x.^2/2);
end
```

- function S = mpr(fun,a,b,N)

```
h = (b-a)/N;
S = h*sum(feval(fun,[a+h/2:h:b]));
end
```

- mpr('gauss',0,5,500)

File Management

- fid=fopen(”filename”,”r”)
- fprintf,fscanf and others can be used as usual
- Save, load commands with file names to save files
- fclose(fid)
- fgets, fputs work in the specified format as their counterparts of C

Graphics!

- 'plot(x,y[,fmt])' plots a line through the points x,y. With the string fmt you can select line style and color; see help plot for details.
- 'semilogx(x,y[,fmt])' like plot with a logarithmic scale for the X-axis.
- 'semilogy(x,y[,fmt])' like plot with a logarithmic scale for the Y-axis.
- 'loglog(x,y[,fmt])' like plot with a logarithmic scale for both axes.
- 'title(string)' writes string as title for the graphics.
- 'xlabel(string)' labels the X-axis with string.
- 'ylabel(string)' labels the Y-axis with string.
- 'axis(v)' set axes limits for the plot. v is a vector of form v = (xmin, xmax, ymin, ymax[, zmin zmax]).
- 'hold [on|off]' controls whether the next graphics output should or not clear the previous graphics.
- 'clg' clears the graphics window. 'subplot(m,n)'
- figure(n) creates a new window with no. n

The End