

Octave

Tutorial

Daniel Lamprecht

Graz University of Technology

March 21, 2011

Introduction

What is Octave?

- GNU Octave is “a high-level interactive language for numerical computations” [eaton:2002]
- mostly compatible to MATLAB
www.gnu.org/software/octave/FAQ.html#MATLAB-compatibility

Why use Octave?

- easy to use
- free software

Installing Octave

How to get Octave?

- part of most Linux repositories
- <http://www.gnu.org/software/octave/>
- be sure to use the right version 3.2.x
- assignments will be tested with 3.2.4

Libraries:

- Gnuplot: www.gnuplot.info

Introduction

- Octave does not come with its own IDE
- but there are some available on the net

- [OctaveNB](#) NetBeans IDE Integration
- [QtOctave](#) Qt based IDE front-end
- [octavede](#) GTK based IDE front-end
- [Kalkulus](#), [Xoctave](#)

Or just use a text editor and a command shell!

Operators and Variables

Numbers:

```
n = 25           % Numeric
n = 25.5        % Numeric
```

- double precision floating point number
- no type declaration necessary

Strings:

```
s = "Hello world!" % String
s = 'Hello world!' % String
```

Operators and Variables

Basic Operations:

```
6 * 5  
# ans = 30
```

- results will be stored in the variable `ans`, if not otherwise specified

```
number = 25 + pi  
# number = 28.142  
number = number + 5;
```

- predefined constants `pi`, `e`, `i` ...
- use semicolon to suppress output

Structs

Defining a Struct:

```
coordinates.x = 5;  
coordinates.y = 8;
```

```
coordinates  
# coordinates =  
# {  
#   x = 5  
#   y = 8  
# }
```

Accessing a Single Member of a Struct:

```
coordinates.x  
# ans = 5
```

Comparators

Comparing Values:

```
25 ~= 25  
# ans = 0
```

- 0, *false* are false
- 1, *true* and all other numeric values are true

Comparing Strings:

```
"hello" == "hello"  
# ans = 1 1 1 1 1
```


Vectors and Matrices

Defining a Vector:

```
vector = [1 2 3];
```

- lines are separated through spaces or commas

Defining a Matrix:

```
matrix = [1 2; 3 4];
```

- rows are separated through semicolons

```
matrix(2,2)  
# ans = 4
```

- note that indexing starts with 1

Vectors and Matrices

Matrix Operations:

```
matrix_a = [1 2; 3 4];  
matrix_b = [5 6; 7 8];  
matrix_a + matrix_b  
# ans =  
#      6      8  
#     10     12
```

```
matrix_a * matrix_b  
# ans =  
#     19     22  
#     43     50
```

```
% multiplication element by element  
matrix_a .* matrix_b
```

Vectors and Matrices

Transpose of a Matrix:

```
matrix = [1 2 3; 4 5 6; 7 8 9];  
matrix'  
#ans =  
#    1    4    7  
#    2    5    8  
#    3    6    9
```

Inverse of a Matrix:

```
inv(matrix)  
#ans =  
#   -4.5036e+15    9.0072e+15   -4.5036e+15  
#    9.0072e+15   -1.8014e+16    9.0072e+15  
#   -4.5036e+15    9.0072e+15   -4.5036e+15
```

Control Structures

If Statement:

```
if length(vector) < 4
    vector(4) = 0;
else
    vector(4)
end
```

Loops:

```
for i = 1:10      % loop from 1 to 10
    i
endfor

while i <= 10    % loop from 1 to 10
    i++
endwhile
```

Performance Improving

- Octave works well with array operations and can be slow when using loops.
(Matlab = Matrix Laboratory)

Vectorizing

```
i = 0;
for t = 0:.01:10
    i = i + 1;
    y(i) = sin(t);
end
```

A vectorized version of the same code is

```
t = 0:.01:10;
y = sin(t);
```

Built-In Functions

Useful Built-In Functions:

- the length of a vector or matrix:

```
matrix = [1 2 3; 4 5 6; 7 8 9];  
length(matrix)  
# ans = 3
```

- the size of a matrix:

```
size(matrix)  
# ans =  
#      3      3
```

- the diagonal of a matrix:

```
diag(matrix)'  
# ans = 1      5      9
```

Built-In Functions

- the sum:

```
matrix = [1 2 3; 4 5 6; 7 8 9];  
sum(matrix)  
# ans = 12    15    18
```

- the minimum and maximum:

```
min(max(matrix))  
# ans = 7
```

- identity matrix:

```
eye(3)  
# ans =  
#    1    0    0  
#    0    1    0  
#    0    0    1
```

Built-In Functions

Other Useful Functions:

- `abs`: the absolute value of a number
- `ones`: a matrix containing only ones
- `zeros`: a matrix containing only zeros
- `repmat`: returns a matrix composed of multiple other matrices
- `factorial`: the faculty of a number
- `any`: detecting rows containing only zeros in matrices
- many functions have overloaded signatures
- use the help system to get the one you need

```
help functionname    % short description
doc functionname     % documentation
```


Scripts

script.m:

```
printf(" Octave Tutorial Test Script\n");  
number = 3;  
printf(" Number = %d\n", number);
```

```
script                % call script  
# Octave Tutorial Test Script  
# Number = 3
```

- multiple commands can be stored in a script-file
- call the script without the .m extension
- file must be placed in Octave paths or in the current directory
- use `addpath` to add a folder to Octave paths

Functions

Faculty Function:

```
function result = faculty(value)
result = 1;
for i = 1:value
    result *= i;
endfor
endfunction
```

- function result-values = function-name(parameters)
- if stored in a file function name and file name have to be equal

```
faculty(4)
# ans = 24
```

Functions

Multiple Return Values:

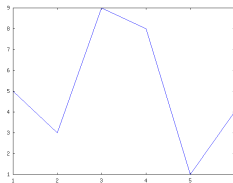
```
function res_a res_b = swap(val_a , val_b)
res_a = val_b;
res_b = val_a;
endfunction
```

```
[val_c , val_d] = swap(5,8)
# val_c = 8
# val_d = 5
```

- if a function has more than one return value you have to store them in different variables
- otherwise only the first return value will be stored

Two-Dimensional Plotting

```
vector = [5 3 9 8 1 4];  
plot(vector)
```



Saving a Plot to a File:

```
print("my_plot.png");
```

Two-Dimensional Plotting

Additional Commands:

```
% create new figure and set a property
f = figure('visible', 'off');





xlabel("x");           % naming the x-axis
ylabel("y");          % naming the y-axis
title("MyPlot");      % set the plot title
```

Two-Dimensional Plotting

- all plotting functions use `gnuplot`
- download from www.gnuplot.info
- included in the installer for MS Windows
- `plot` could be used with many different input parameters
- use the help system

```
help plot      % short description of plot  
doc plot      % plot documentation
```

Further reading

-  Octave Documentation
(www.gnu.org/software/octave/doc/interpreter)
-  A Short Octave Tutorial (German)
(www.christianherta.de/octaveMatlabTutorial.html)
-  Octave Tutorial at the New York University
-  Octave Programming Tutorial at Wikibooks
(http://en.wikibooks.org/wiki/Octave_Programming_Tutorial)

based upon the tutorial by Ingo Holzmann in 2010



John W. Eaton.

GNU Octave Manual.

Network Theory Limited, 2002.