**Octave**

Contents

[Day 1: Friday Mar 3, 2017 - Introduction 2](#_Toc476436666)

[Octave Project 2](#_Toc476436667)

[Things to bear in mind 3](#_Toc476436668)

[Octave Installation 3](#_Toc476436669)

[Getting Started with Octave 4](#_Toc476436670)

[Day 2: Saturday March 4, 2017 – Data Types 9](#_Toc476436671)

[Day 2 Assignment 13](#_Toc476436672)

# Day 1: Friday Mar 3, 2017 - Introduction

Octave is MATLAB clone. Take this project as your contribution to advanced computer science. Do it as hobby or serious work. Both are ok. Spend with a week with system ON and you would become experts. When you are part of great products, you will feel proud. It will give great satisfaction. If you get expertise, you will be automatically sought after, there is no doubt in this. We must be able to learn all these so that we are at the level of NASA. People who learn here will be confident of achieving in any software solution.

## Octave Project

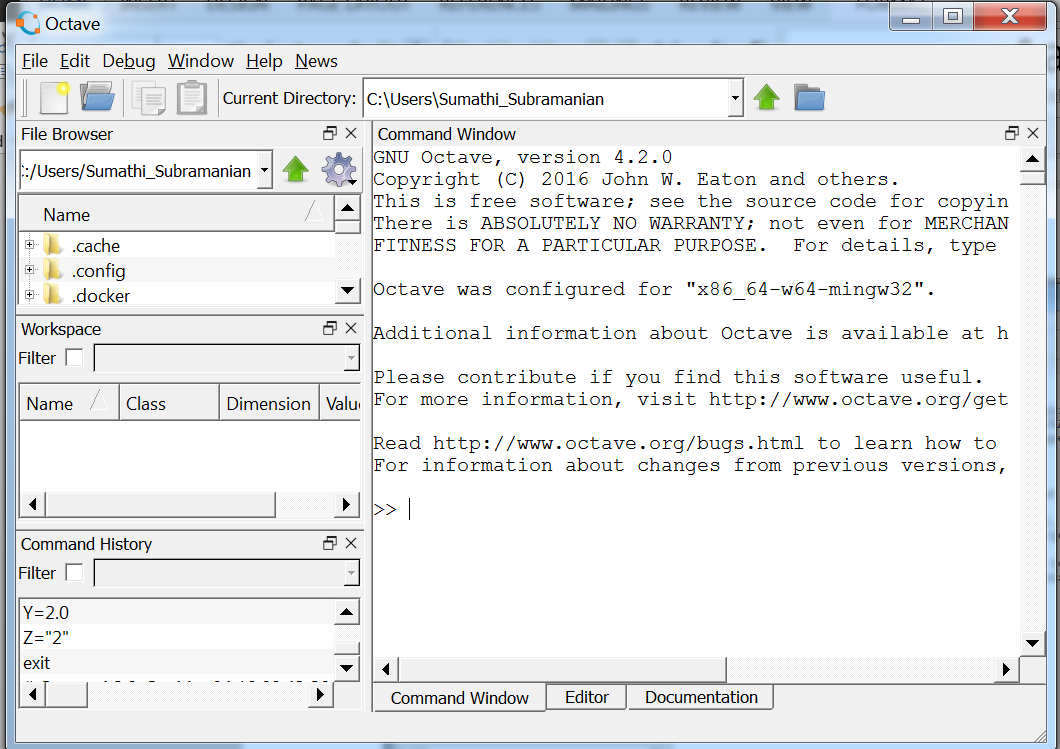
1. **Coders**: They don’t need to not know any mathematics, they just need to code the way it is told, using simple limited functions as instructed
2. **Domain experts:** These individuals will use their expertise in the domain which they are trained, and try to create some objects in that domain, this requires little more skill
3. **Systems programmers:** These individuals need advanced computer science methodologies.
4. **System Engineers:** System programmers can graduate to Systems Engineering and dabble with new concepts of parallel processing
5. **Front end team:** Exclusively concentrates on building front ends to develop BPHS algorithms. Building GUI is fast in Octave
6. **Technical Documentation and Project Management**: Non science background individuals to document, track tasks, create reports, keep track of programs, maintain versions - Soni, Raghuveer, Sumati etc. Project repository in Onedrive, google drive is free
7. **Design Team:** Contribution to design areas and suggest improvements - Ramesh, Hrishikesh, Krishna

## Things to bear in mind

* Usually octave is used to solve differential equations, matrix manipulation sparse matrices, image processing, but here it will be used for artificial intelligence, sometimes it will be made so simple, that you may ask where the artificial intelligence here is.
* Building a GUI is very fast in octave, it is fairly simple
* When programming in octave you must stop thinking in terms of cc++
* Without the knowledge of cc ++ it’s almost impossible to even get started in octave
* For learning follow verbatim, as minimum basics you just need information in c and c++
* MATLAB MAPLE MATHEMATICA are highly advanced, at this stage you don’t need to bother about it
* Most of the solutions can be achieved by Octave itself, if not you can do a lot of batch processing very easily in octave with single line.
* For the beginners it should be kept in mind, learning basics means you only become a coder, and to become a domain expert, it takes more to use these software’s excellently.
* Knowing computers is no big deal, however it’s more important knowing how to apply it to a specialised application. You will learn newer and newer concepts by building a specialised application.
* Every day there will be a challenging task, and for once no wasting time in figuring out how to solve will be allowed. You must code very fast, or guruji will code it faster and you must try to figure out later, this is to develop and learn
* **Working Domain** - Astrology for now, but its higher mathematics and dimensional space science
* Last week guruji, taught Octave to a diploma in electronics, who knew nothing and was out of job for 15 years. Today she is creating databases in Octave.
* Those who are fast will be taught Signal processing
* Intermediate will maintain database, data conditioning
* Weak will handle GUI

## Octave Installation

Install Octave - 4.2.0 from [link](https://www.gnu.org/software/octave/download.html). System requirements for Windows: JRE should be installed before you start installation of Octave. After installation, open Octave GUI, not Octave CLI, and you should see as below



## Getting Started with Octave

Type these in command Window after >>. Press **up or down arrow keys** if you need to re run commands. Perform computation and immediately visualise it in graphics. This philosophy must GO into mind quickly. We are here to design, solve real world problems.

|  |  |
| --- | --- |
| **Command** | **Output** |
| graphics\_toolkit  graphics\_toolkit("gnuplot") | Both fltk, qt, will not give proper results. Fltk gives good results on linux |
| figure; | We should get a new window like below. |
| subplot(2,1,1) | If graph took a longer time to get generated, it is qt effect  In subplot(p,q,r) are column, row and index |
| x=[ 1 2 3 4 5 6 7] | We can concentrate on algorithm development rather than coding initial variable conditions, declaring variables. If we write x=[1 2 3 4] the array is ready |
| y=x.^2 |  |
| plot(x,y) |  |
| z=x.\*10 |  |
| subplot(2,2,1) |  |
| plot(x,z) |  |
| subplot(2,2,2)  plot(x,y) |  |
| x=”hari” | Now x has automatically changed datatype |
| str="RAAMA"  str2="KRISHNA"  str3=[str str2]  str3=[str " " str2] | This concatenates strings. We can have easy string manipulations |
| Look at the left lower window | You can see the list of all commands listed here. |
| See in the window above it | You will see all variables there |
| clear “x” | See there is no variable in variable box |
| Y | You see the squares of x assigned in the 6th step |
| y=[] | y becomes empty |
| * Select commands from command history by pressing shift and pressing arrow. * After selecting from x=[…], y=[..] all commands including subplot * Now look at the bottom |  |
| * Click on Editor * In the editor place cursor and press Ctrl+V * All commands should be pasted * Save the file as .m file in a location * Give it a name say test.m * **Note file names are case sensitive** |  |
| * Set current directory to test, the same location saved above * See upper end white box |  |
| * Running Scripts * Now go back to Command Window and enter   test | You should se all previous commands getting executed. We can save scripts and run from command window just as we run other commands |
| In the Editor click new | You should get a new window    Now in the new window type below commands |
| function[y]=powerr(x)  y=x.^2; | See all function is in blue, f is in small letter, not capital letter. Note the double r in powerr(x). Instead of power, we can also use double\_r |
| Now in the command prompt type  h=[2 4 6 8]  hp=power(h) | Note if you save your .m file by any other name, use that name and not power with case sensitivity consideration. |

# Day 2: Saturday March 4, 2017 – Data Types

|  |  |
| --- | --- |
| **Command** | **Output** |
| Open power.m saved  function[y]=powerr(x)  % this function computes square of an array  y=x.^2; | Adding comments to file |
| In Command window  help power | This lists only the first comment added to the file. Note the function name and file name should be the same. Otherwise compiler will take a long time to resolve name. |
| X=2  Y=2.0  Z=”2” | Check the data types in workspace as double and char |
| X+1  Y+1.2  Z+1  To get 3 from “2”+1  str2num(Z)+1 | Note 51 as “2”+1. So type conversion is automatic in Octave, we should be careful. str2num converts string to numeric |
| * Opening file names programmatically if file names are file1, file2 file3 file 4.txt * Create a variable filep   filep=”file”  I=1  Filep1=[filep num2str(I)]  I=I+1  Filep2=[filep num2str(I)] | Now you have see numbers, strings, arrays |
| Creating a structure  W.name=”Ram” |  |
| W.age=23 |  |
| W.b=[1 2;3 4] |  |
| To create nested structure  W.c={["e";"r";"y"];"$"} | So we see creating structure in Octave is easy. U can also create nested structures. Structure contains mixed data types in one place |
| W(1).c={["r";"d";"a"];"@"} | C can be a structure too |
| W(2).c=W(1).c |  |
| W(2) | * See even though you have assigned only W(2).c octave has automatically initialised W(2) all fields * Name age are empty matrices. Did you get this? * So this means you can dynamically increase the length of records * Suppose you have 100 students * You have created 100 records ie W(100) * A new student takes admission you can dynamically add a new record * Adding W(101) is automatic * Is this understood? |
| You can also dynamically add a field new field  W(2).d=[2 4 5 7 78] |  |
| W(1).d | See you have W(1).d too initialised, Without you having to write code about it  This gives immense power during complex computations |
| Now assign t=W(2).d  t=W(1).d |  |
| t(end) |  |
| To access any member of the array, you need not know how big “t” is  te=t(end-1) |  |
| To know the size of the array  [row col]=size(t) | Now you have W, a structure, which gets lost when you close the session. So save it further research on need basis. |
| save mydata | Note, mydata is saved |
| load mydata |  |
| W |  |
| To save file in matlab  save -v7 mydata.mat W |  |
| You need to know how to take input  Input("enter coordinates of planet sun " )  You can type 12.345  It gets assigned | https://scontent.fdel1-1.fna.fbcdn.net/v/t34.0-12/17125032_10210281854697145_1248651423_n.png?oh=ca41aec47ed2c250419b344b3a8f896f&oe=58BDD7F6 |

## Day 2 Assignment

Create a structure which contains your planetary position of horoscope

Including

* Date of birth
* Time of birth
* Place of birth
* Nine grahas + lagna, also including upagrahas
* Write a function to calculate upagrahas (chapter 3 of BPHS)
* Use object oriented design
* Use nested structures
* Write a function to calculate upagrahas
* You must write separate function for each upagraha
* Upagraha must be in separate structure nested inside main structure
* Planets must be in one nested structure
* Save this structure