

Web-Octave

Submitted for partial fulfilment of the Degree
of
Bachelor of Technology
(Computer Science)



Submitted By:

Deepti Sharma
1410821
145021

Submitted To:

Sukhjit Singh Sehra
Training Coordinator
CSE Department

Department of Computer Science & Engineering
Guru Nanak Dev Engineering College
Ludhiana 141006

Acknowledgement

I, student of Guru Nanak Dev Engineering College, Ludhiana, have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

The author is highly grateful to Dr. M.S. Saini Director, Guru Nanak Dev Engineering College, Ludhiana for providing her with the opportunity to carry out his Six month Training at Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana.

The author would like to whole heartedly thank Dr. H.S. Rai Dean, Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana who is a vast sea of knowledge and without whose constant and never ending support and motivation, it would never have been possible to complete the project and other assignments so efficiently and effectively.

Finally, I would thanks to all whoever have contributed in this report work with Amarjeet Kapoor (D4 CSE) and all other trainees. Without their encouragement, it would not have been possible to complete this project in such an efficient manner.

Deepti Sharma

Abstract

This project discusses mainly about Weboctave. I devised this tool which basically is collection of direct and iterative methods to solve problems of various equations, matrices, graph plotting and ordinary linear equations.

The increasing importance of numerical methods in applied sciences have led to enhanced demand to deal with techniques of numerical analysis. The reason for this is that numerical methods can give the solution to applied problems when ordinary analytical methods fail.

Furthermore, coming back it deals with Weboctave which is web interface to use octave i.e. anyone can use this service and need not have octave installed on their systems and can use this service remotely.

Also, this project is completely open source and the entire code is available to the user as and when required. There is Complete developer's Documentation as well as User manual alongwith it that helps using it a lot easier.

1	Introduction To Organisation	1
1.1	Testing and Consutancy Cell	2
2	Introduction to Project	4
2.1	Overview	4
2.1.1	What is Octave?	4
2.1.2	What Octave is not?	4
2.1.3	Who uses Octave?	4
2.1.4	Why not use a normal highlevel language, e.g. C++ ?	5
2.2	The Existing System	5
2.2.1	Limitations of previous system	5
2.2.2	Advantages of shifting the system from CGI to PHP:	5
2.3	User Requirement Analysis	6
2.4	Feasibility Study	6
2.5	Types of Feasibility	7
2.5.1	Technical Feasibility	7
2.5.2	Economic Feasibility	8
2.5.3	Operational Feasibility	9
2.5.4	Technological Feasibility	9
2.5.5	Behavioral Feasibility	9
2.6	Objective of Project	10
3	Project Design	11
3.1	Software Requirement Analysis	11
3.1.1	Functional Requirements	12
3.1.2	Non functional requirements	12
3.2	Other Specifications	13
3.3	DFDs	14
3.4	Flowchart	15
3.5	UI Flow Diagram	15
3.6	Problem Formulation	16
3.7	Facilities required for proposed work	16
3.7.1	Hardware Requirements	16

3.7.2	Software Requirements	16
3.8	Methodology	16
3.9	Project Work	16
4	Development and Implementation	18
4.1	Introduction to Languages	18
4.1.1	HTML	18
4.1.2	CSS	19
4.1.3	PHP	20
4.1.4	BootStrap	21
4.2	Introduction to Octave	22
4.2.1	Installation of Octave	22
4.3	Ubuntu: An open source OS	24
4.4	Working with Experimental Server	25
4.5	Introduction to Doxygen	27
4.5.1	Features of Doxygen	27
4.5.2	Installation of Doxygen	28
4.6	Introduction to Reveal-js & Reveal-md	29
4.6.1	Installation of reveal-md	29
4.7	Introduction to L ^A T _E X	30
4.7.1	Typesetting	31
4.8	Introduction to Github	32
4.8.1	What is Git?	33
4.8.2	Installation of Git	34
4.8.3	Various Git Commands	34
4.8.3.1	Create Repositories	34
4.8.3.2	Make Changes	34
4.8.3.3	Group Changes	35
4.8.3.4	Synchronize Changes	35
5	Experimental Results and Comparison	36
5.1	Experimental Results	36
5.1.1	Testing	38
6	Conclusion,Summary and Future Scope	39
6.1	Future Scope	39
6.2	Technical and Managerial Lesson Learnt	39

LIST OF FIGURES

1.1	Guru Nanak Dev Engineering College	1
1.2	Testing and Consultancy Cell	2
3.1	Data flow LEVEL 0	14
3.2	Data Flow LEVEL 1	14
3.3	UI Flow diagram	15
4.1	HTML5 logo	18
4.2	CSS logo	19
4.3	PHP logo	20
4.4	BootStrap logo	21
4.5	GNU Octave	22
4.6	Ubuntu	24
4.7	Server Communication	25
4.8	Doxygen logo	27
4.9	MD & JS	29
4.10	Donald Knuth, Inventor Of T _E X typesetting system	30
4.11	Github Logo	32
4.12	Git Logo	33
5.1	Graph Plotted	36
5.2	Code	37
5.3	Output	37

LIST OF TABLES

5.1	Octave Release	38
5.2	Computational Analysis	38
5.3	Test case (general)	38

CHAPTER 1

INTRODUCTION TO ORGANISATION



Figure 1.1: Guru Nanak Dev Engineering College

I had my Six Weeks Training at TCC-Testing And Consultancy Cell, GNDEC Ludhiana. Guru Nanak Dev Engineering College was established by the Nankana Sahib Education Trust Ludhiana. The Nankana Sahib Education Trust i.e NSET was founded in memory of the most sacred temple of Sri Nankana Sahib, birth place of Sri Guru Nanak Dev Ji. With the mission of Removal of Economic Backwardness through Technology Shiromani Gurudwara Parbandhak Committee i.e SGPC started a Poly technical was started in 1953 and Guru Nanak Dev Engineering College was established in 1956.

NSET resolved to uplift Rural areas by admitting 70% of students from these rural areas ever year. This commitment was made to nation on 8th April, 1956, the day foundation stone of the college building was laid by Dr. Rajendra Prasad Ji, the First President of India. The College is now ISO 9001:2000 certified.

Guru Nanak Dev Engineering College campus is spread over 88 acres of prime land about 5 Kms

from Bus Stand and 8 Kms from Ludhiana Railway Station on Ludhiana-Malerkotla Road. The college campus is well planned with beautifully laid out tree plantation, pathways, flowerbeds besides the well maintained sprawling lawns all around. It has beautiful building for College, Hostels, Swimming Pool, Sports and Gymnasium Hall Complex, Gurudwara Sahib, Bank, Dispensary, Post Office etc. There are two hostels for boys and one for girls with total accommodation of about 550 students. The main goal of this institute is:

- To build and promote teams of experts in the upcoming specialisations.
- To promote quality research and undertake research projects keeping in view their relevance to needs and requirements of technology in local industry.
- To achieve total financial independence.
- To start online transfer of knowledge in appropriate technology by means of establishing multipurpose resource centres.

1.1 Testing and Consultancy Cell

My Six Weeks Institutional Training was done by me at TCC i.e Testing And Consultancy Cell, GNDEC Ludhiana under the guidance of Dr. H.S.Rai Dean Testing and Consultancy Cell. Testing and Consultancy Cell was established in the year 1979 with a basic aim to produce quality service for technical problems at reasonable and affordable rates as a service to society in general and Engineering fraternity in particular.



Figure 1.2: Testing and Consultancy Cell

Consultancy Services are being rendered by various Departments of the College to the industry, State Government Departments and Entrepreneurs and are extended in the form of expert advice in design, testing of materials & equipment, technical surveys, technical audit, calibration of instruments, preparation of technical feasibility reports etc. This consultancy cell of the college has given a new dimension to the development programmers of the College. Consultancy projects of over Rs. one crore are completed by the Consultancy cell during financial year 2009-10.

Ours is a pioneer institute providing Consultancy Services in the States of Punjab, Haryana, Himachal, J&K and Rajasthan. Various Major Clients of the Consultancy Cell are as under:

- Northern Railway, Govt. of India
- Indian Oil Corporation Ltd.
- Larson & Turbo.
- Multi National Companies like AFCON & PAULINGS.
- Punjab Water Supply & Sewage Board
- Power Grid Corporation of India.
- National Building Construction Co.
- Punjab State Electricity Board.
- Punjab Mandi Board.
- Punjab Police Housing Corporation.
- National Fertilizers Ltd.
- GLADA, Ludhiana

CHAPTER 2

INTRODUCTION TO PROJECT

2.1 Overview

2.1.1 What is Octave?

Octave is an open-source interactive software system for numerical computations and graphics. It is particularly designed for matrix computations: solving simultaneous equations, computing eigenvectors and eigenvalues and so on. In many real-world engineering problems the data can be expressed as matrices and vectors, and boil down to these forms of solution.

In addition, Octave can display data in a variety of different ways, and it also has its own programming language which allows the system to be extended. It can be thought of as a very powerful, programmable, graphical calculator.

Octave makes it easy to solve a wide range of numerical problems, allowing you to spend more time experimenting and thinking about the wider problem. Octave was originally developed as a companion software to undergraduate course book on chemical reactor design.

It is currently being developed under the leadership of Dr. J.W. Eaton and released under the GNU General Public Licence. Octaves usefulness is enhanced in that it is mostly syntax compatible with MATLAB which is commonly used in industry and academics.

2.1.2 What Octave is not?

Octave is designed to solve mathematical problems numerically, that is by calculating values in the computer's memory. This means that it can't always give an exact solution to a problem, and it should not be confused with programs such as Mathematica or Maple, which give symbolic solutions by doing the algebraic manipulation.

This does not make it better or worse, its just used for different tasks. Most real mathematical problems (particularly engineering ones!) do not have neat symbolic solutions.

2.1.3 Who uses Octave?

Octave and MATLAB are widely used by engineers and scientists, in both industry and academia for performing numerical computations, and for developing and testing mathematical algorithms.

For example, NASA use it to develop spacecraft docking systems;

- Jaguar Racing use it to display and analyse data transmitted from their Formula 1 cars.
- Sheffield University use it to develop software to recognise cancerous cells.
- It makes it very easy to write mathematical programs quickly, and display data in a wide range of different ways.

2.1.4 Why not use a normal highlevel language, e.g. C++ ?

C++ and other industry-standard programming languages are normally designed for writing general-purpose software. However, solutions to mathematical problems take time to program using C++, and the language does not natively support many mathematical concepts, or displaying graphics.

Octave is specially designed to solve these kind of problems, perform calculations, and display the results.

2.2 The Existing System

There are few existing systems for solving this particular problem like Sage or other softwares but they dont have following features required by our mentor. These system were not open source and free web based software that were need. All exiting system suffers from at least one of the following system.

2.2.1 Limitations of previous system

- No batch mode.
- Don't give output as png images.
- They are costly.
- They don't save output.
- They need installation and a lot of system resources.
- They don't save user session.
- Earlier it was on CGI, now moved to PHP.

2.2.2 Advantages of shifting the system from CGI to PHP:

- PHP was specially designed for a websites, the facilities that web designers typically want in a scripting language are built into it.
- Another convenience is its handling of form input.
- Accessing databases is just as easy. There are built-in facilities in PHP to access MySQL, Dbase, Oracle, InterBase, and so on.

- Need to MIME encode your message? There's a function to do it for you too.
- When PHP scripts runs, If in case you get error then messages will be like pinpointing the offending lines in your code to help you locate the error.

2.3 User Requirement Analysis

1. Perform most of difficult Calculation work.
2. Make it work like batch mode. so, that user can give inputs together and relax.
3. Help M.Tech and Civil Engineer to analysis structure.
4. Automatic calculation of modal force and modes.
5. Reduce the time for analysis.
6. Provide on-line way to analysis so that individual does not have to install anything.

2.4 Feasibility Study

This study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness.

A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Carrying out a feasibility study involves information assessment, information collection and report writing. The information assessment phase identifies the information that is required to answer the three questions set out above.

Once the information has been identified, you should question information sources to discover the answers to these questions Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

A feasibility study is designed to provide an overview of the primary issues related to a business idea. The purpose is to identify any make or break issues that would prevent your business from being successful in the marketplace. In other words, a feasibility study determines whether the business idea makes sense. A thorough feasibility analysis provides a lot of information necessary for the business plan. For example, a good market analysis is necessary in order to determine the project's feasibility. This information provides the basis for the market section of the business plan.

The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards.

Objectives of feasibility study are listed below:

- To analyze whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

2.5 Types of Feasibility

2.5.1 Technical Feasibility

Technical feasibility is one of the first studies that must be conducted after the project has been identified. In large engineering projects consulting agencies that have large staffs of engineers and technicians conduct technical studies dealing with the projects. In individual agricultural projects financed by local agricultural credit corporations, the technical staff composed of specialized agricultural engineers, irrigation and construction engineers, and other technicians are responsible for conducting such feasibility studies. The Technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system. This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. When writing a feasibility report, the following should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study.
- The part of the business being examined.
- The human and economic factor.
- The possible solutions to the problem.

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. A Technical feasibility also performs the following tasks.

- Analyzes the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP the project is technically feasible for development.

2.5.2 Economic Feasibility

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

Economic feasibility is the cost and logistical outlook for a business project or endeavor. Prior to embarking on a new venture, most businesses conduct an economic feasibility study, which is a study that analyzes data to determine whether the cost of the prospective new venture will ultimately be profitable to the company. Economic feasibility is sometimes determined within an organization, while other times companies hire an external company that specializes in conducting economic feasibility studies for them.

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. Economic feasibility determines whether the required software is capable of generating financial gains for an organization. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Economic analysis is the most frequently used method to determine the cost/benefit factor for evaluating the effectiveness of a new system. In this analysis we determine whether the benefit is gained according to the cost invested to develop the project or not. If benefits outweigh costs, only then the decision is made to design and implement the system. It is important to identify cost and benefit factors, which can be categorized as follows:

- Development Cost
- Operation Cost

This System is Economically feasible with 0 Development and Operating Charges as it is developed in Qt Framework and Octave which is open source technology and is available free of cost on the internet.

2.5.3 Operational Feasibility

Operational feasibility is a measure of how well a project solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. All the operations performed in the software are very quick and satisfy all the requirements.

2.5.4 Technological Feasibility

Technological feasibility is carried out to determine whether the project has the capability, in terms of software, hardware, personnel to handle and fulfill the user requirements. The assessment is based on an outline design of system requirements in terms of Input, Processes, Output and Procedures. Automated Building Drawings is technically feasible as it is built up using various open source technologies and it can run on any platform.

2.5.5 Behavioral Feasibility

Behavioral feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. It is a measure of how well the solution of problems or a specific alternative solution will work in the organization. It is also measure of how people feel about the system. If the system is not easy to operate, than operational process would be difficult. The operator of the system should be given proper training. The system should be made such that the user can interface the system without any problem.

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture, and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters such as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyzes whether users will adapt to a new software.

- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?
- The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.6 Objective of Project

Web-Octave is a web based software and the main objectives of this project is to -:

1. To inspire M.Tech students to automate their work and do programming.
2. Perform most of difficult Calculation work.
3. Make it work like batch mode. so, that user can give inputs together and relax.
4. Accept inputs from the user.
5. Help M.Tech and Civil Engineer related to OCTAVE.
6. Automatic calculation of modal force and modes.
7. Reduce the time for analysis.
8. Generates the final output in the form of png image.
9. Provide on-line way to analysis so that individual does not have to install anything.
10. Save the session of a user.

This project use classes instead of simple dispatch functions.

3.1 Software Requirement Analysis

Software requirement analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

3.1.1 Functional Requirements

- **Specific Requirements:** This phase covers the whole requirements for the system. After understanding the system we need the input data to the system then we watch the output and determine whether the output from the system is according to our requirements or not. So what we have to input and then what we'll get as output is given in this phase. This phase also describe the software and non-function requirements of the system.
- **Input Requirements of the System**
 1. Guess points
 2. Precision
 3. Step-size in case of iterative methods.
 4. Required point at which value is to be found
 5. User can define his/her problem in function file
- **Output Requirements of the System**
 1. Final output after iterations.
 2. Graphs wherever possible in form of images.
- **Special User Requirements**
 1. It can save sessions of concerned person.
 2. Taking bulk input values in form of functions too.
- **Software Requirements**
 1. Programming language: Octave 4.0
 2. software: \LaTeX
 3. Web Languages: php
 4. Database: Mysql
 5. Documentation: Doxygen 1.8.3
 6. Text Editor: Vim
 7. Operating System: Ubuntu 14.04 or up
 8. Revision System: Git

3.1.2 Non functional requirements

1. Scalability: System should be able to handle a number of users. For e.g., handling around thousand users at the same time.
2. Usability: Simple user interfaces that a layman can understand.
3. Speed: Processing input should be done in reasonable time i.e. we can say maximum 24 hrs.

3.2 Other Specifications

A Software Requirements Analysis for a software system is a complete description of the behavior of a system to be developed. It include functional Requirements and Software Requirements. In addition to these, the SRS contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation.

- **Purpose:** Web Octave Tool is a web based software and the main purpose of this project is to:
 1. Perform most of difficult Calculation work.
 2. Make it work like batch mode. so, that user can give inputs together and relax.
 3. Help M.Tech and Civil Engineer to analysis structure.
 4. Automatic calculation of modal force and modes.
 5. Reduce the time for analysis.
 6. Provide on-line way to analysis so that individual does not have to install anything.
- **Users of the System**
 1. Client : Clients are the end users that benefit from this software. They just provide input and gets output.Client of this system:
 - (a) Researcher or student-: They have knowledge of working of procedures and what input is being provided.

3.3 DFDs

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs of Web-Octave is as following:-

1. Data flow LEVEL 0 fig 3.1
2. Data flow LEVEL 1 fig 3.2

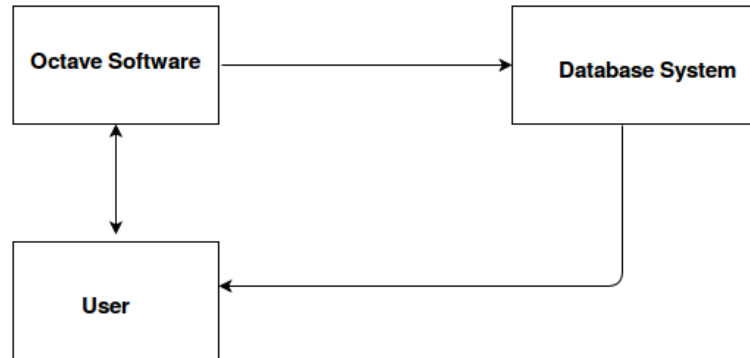


Figure 3.1: Data flow LEVEL 0

Now the Basic Data Flow:-

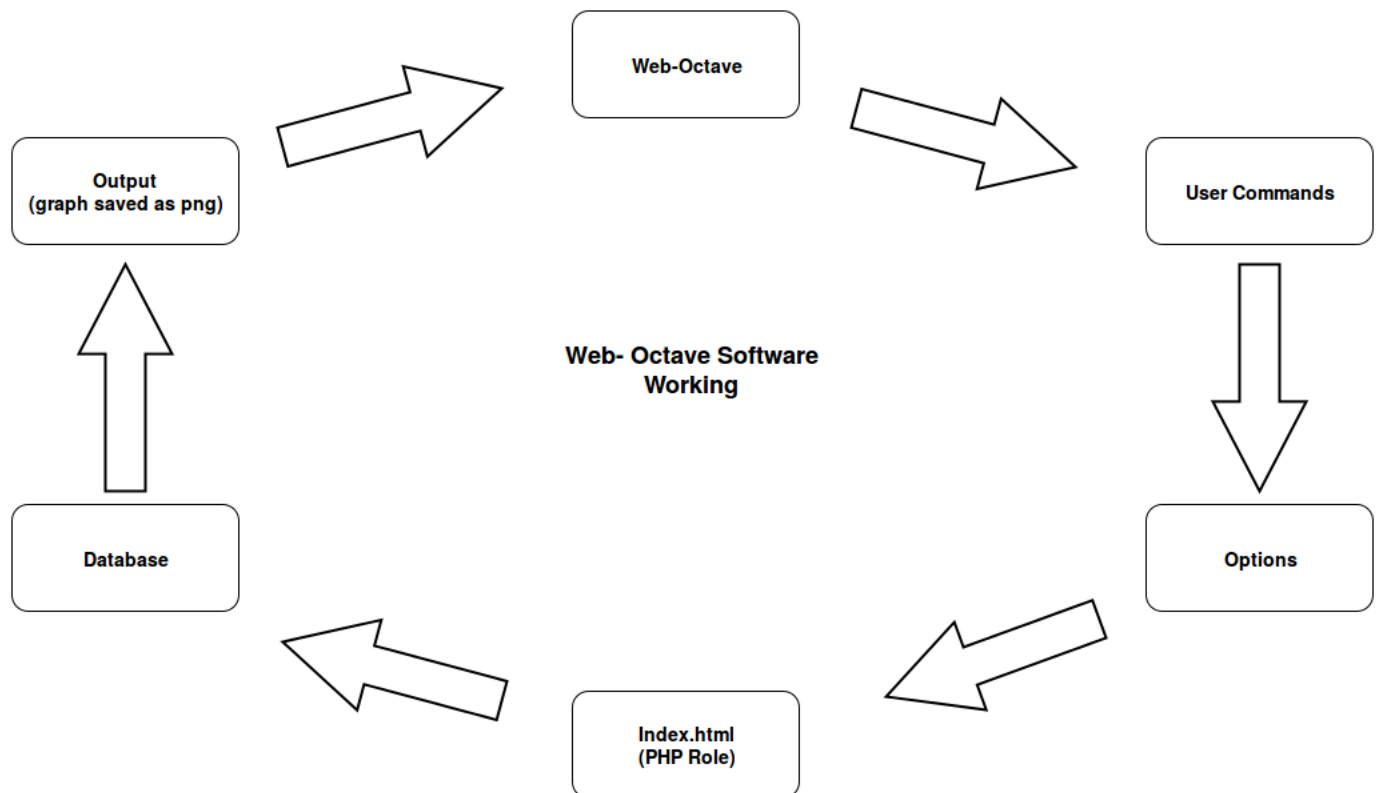


Figure 3.2: Data Flow LEVEL 1

3.4 Flowchart

A flowchart is a type of diagram that represents an algorithm, work flow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. Flowcharts are used in designing and documenting simple processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart are:

1. A processing step, usually called activity, and denoted as a rectangular box.
2. A decision, usually denoted as a diamond.

Following is flowchart of system showing flow of control and Data in the software:-

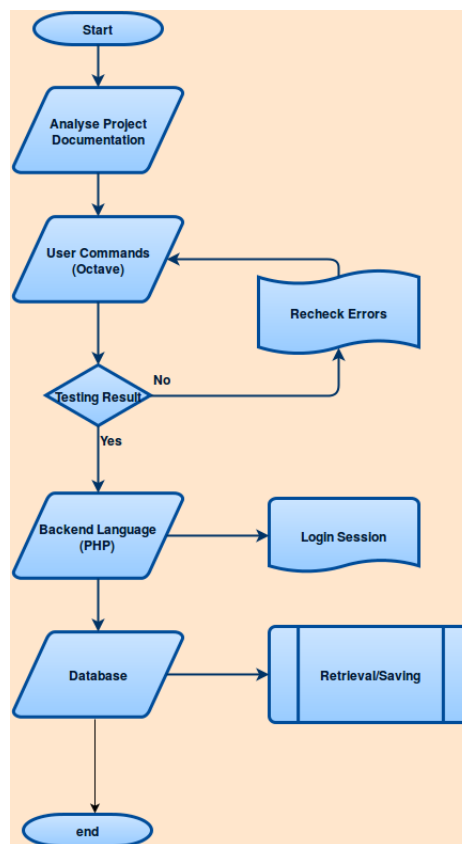


Figure 3.3: Flow diagram

3.5 UI Flow Diagram

UI Flow diagram tells how user will perceive different interface on click of different buttons or trigger. The rectangular blocks represent entities and Arrow represents change of view from one to another on the bases of button clicked mentioned near arrow.

3.6 Problem Formulation

When analytical solution of the mathematically defined problem is possible but it is time-consuming and the error of approximation we can use Octave. In this case the calculations are mostly made with use of computer because otherwise its highly doubtful if any time is saved. It is individually to decide what do we mean by "time-consuming analytical solution". In my discipline even very simple mechanical problems are solved numerically simply because of laziness.

When analytical solution is impossible, which was discussed by eg. Alexander Sadovsky. This means that we have to apply numerical methods in order to find the solution. This does not define that we must do calculations with computer although it usually happens so because of the number of required operations.

3.7 Facilities required for proposed work

3.7.1 Hardware Requirements

- Operating System: Linux/Windows
- Processor Speed: 512KHz or more
- RAM: Minimum 256MB

3.7.2 Software Requirements

- Software: octave gui
- Programming Language: octave

3.8 Methodology

- Studying various methods available to solve different problems of numerical analysis.
- Deciding various input and output parameters of methods.
- Making the approach modular.
- Graphical representation of solutions wherever possible.
- Generating documentation.

3.9 Project Work

Studied Previous System:

Before starting the project.

Learn octave:

Before starting with project, we have to go through the basics of Octave, such that there should not be any problem proceeding with project.

Get Familiar with Different methods and their algorithms:

We have gone through algorithms of these algorithms. Then implementation becomes easy.

Functions:

The user has been provided some test functions which he can use to test various.

Plots:

Octave provides fltk as the default toolkit. But we can use gnuplot for more accurate plotting by setting them as default toolkit.

Input:

Input values are taken from user or default values defined in the file are used.

Output:

The iterations are performed and it returns the output with the expected precision.

CHAPTER 4

DEVELOPMENT AND IMPLEMENTATION

4.1 Introduction to Languages

Front End languages are language that are used to give better user experience and user interface. These mainly include HTML, CSS, PHP. Some Frameworks like Bootstrap are also used with these basic languages.

4.1.1 HTML



Figure 4.1: HTML5 logo

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and HTML is a cornerstone technology, used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

4.1.2 CSS



Figure 4.2: CSS logo

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media.

Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts.

This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers.

CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions

```
p {
  color: red;
  text-align: center;
}
```

4.1.3 PHP



Figure 4.3: PHP logo

What is PHP?

- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)

- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

4.1.4 BootStrap



Figure 4.4: BootStrap logo

Bootstrap is a free and open-source collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It aims to ease the development of dynamic websites and web applications. Bootstrap is a front end framework, that is, an interface for the user, unlike the server-side code which resides on the "back end" or server.

4.2 Introduction to Octave



Figure 4.5: GNU Octave

GNU Octave is a high-level interpreted language, primarily intended for numerical computations. It provides capabilities for the numerical solution of linear and nonlinear problems, Image Processing and for performing other numerical experiments. It also provides extensive graphics capabilities for data visualization and manipulation. Octave is normally used through its interactive command line interface, but it can also be used to write non-interactive programs. The Octave language is quite similar to Matlab so that most programs are easily portable.

Octave is distributed under the terms of the GNU General Public License.

GNU/Linux systems

Executable versions of Octave for GNU/Linux systems are provided by the individual distributions. Distributions known to package Octave include: Debian, Fedora, Gentoo, and SuSE. These packages are created by volunteers. The delay between an Octave source release and the availability of a package for a particular GNU/Linux distribution varies. The Octave project has no control over that process. BSD systems

Executable versions of Octave for BSD systems are provided by the individual distributions. Both FreeBSD and OpenBSD have Octave packages. These packages are created by volunteers. The delay between an Octave source release and the availability of a package for a particular GNU/Linux distribution varies. The Octave project has no control over that process.

OS X

The Wiki has some instructions for installing Octave on OS X systems.

Windows

Windows binaries with corresponding source code can also be downloaded.

Sources

The latest released version of Octave is 4.0.3 released in July, 2016.

4.2.1 Installation of Octave

Installation of Octave is a very easy process. The current Octave version is: 4.0.3 Type the commands in the terminal:

```
$ sudo apt-add-repository ppa:octave/stable
```

```
$ sudo apt-get build-dep octave
```

\$ sudo apt-get install octave

This will install Octave on your pc or laptop.

The simplest way to compile this package is:

- 'cd' to the directory containing the package's source code and type './configure' to configure the package for your system.

Running 'configure' might take a while. While running, it prints some messages telling which features it is checking for.

- Type 'make' to compile the package.
- Optionally, type 'make check' to run any self-tests that come with the package, generally using the just-built uninstalled binaries.
- Type 'make install' to install the programs and any data files and documentation. When installing into a prefix owned by root, it is recommended that the package be configured and built as a regular user, and only the 'make install' phase executed with root privileges.
- Optionally, type 'make installcheck' to repeat any self-tests, but this time using the binaries in their final installed location. This target does not install anything. Running this target as a regular user, particularly if the prior 'make install' required root privileges, verifies that the installation completed correctly.
- You can remove the program binaries and object files from the source code directory by typing 'make clean'. To also remove the files that 'configure' created (so you can compile the package for a different kind of computer), type 'make distclean'. There is also a 'make maintainer-clean' target, but that is intended mainly for the package's developers. If you use it, you may have to get all sorts of other programs in order to regenerate files that came with the distribution.
- Often, you can also type 'make uninstall' to remove the installed files again. In practice, not all packages have tested that uninstallation works correctly, even though it is required by the GNU Coding Standards.
- Some packages, particularly those that use Automake, provide 'make distcheck', which can be used by developers to test that all other targets like 'make install' and 'make uninstall' work correctly. This target is generally not run by end users.

4.3 Ubuntu: An open source OS



Figure 4.6: Ubuntu

During my training, I also got familiar with a great and open source Operating System, Ubuntu. Firstly, it was quite difficult for a regular MS Windows user to port to Ubuntu. I did all of my project work using this vast operating system.

Ubuntu is a Debian-based Linux operating system, with Unity as its default desktop environment. It is based on free software and named after the Southern African philosophy of ubuntu (literally, "human-ness"), which often is translated as "humanity towards others" or "the belief in a universal bond of sharing that connects all humanity".

It came under Linux. A kernel normally used by many of the computer persons. You will rarely see a person who is unaware of the term Linux. From perspective of a computer simpleton the one who uses linux mostly shall be having a good knowledge regarding the working of shell, kernel etc.

Linux was created by Linus Torvalds. One of a gem of computer scientist who is popular for his OS.

Linus was one of the student in Finland and had read a book Operating Systems: Design and Implementation by Andrew S. Tanenbaum.

In this book the professor explained about the working of Kernel. To my strange is that he had given the whole source code of his kernel named MINIX in that book. Its really weird but acts as a lucky draw for Linus who took interest in this and with the help of MINIX he created a new OS named Linux. He had told about Andrew in his acknowledgement.

Ubuntu's goal is to be secure "out-of-the box". By default user's programs run with low privileges and cannot corrupt the operating system or other user's files. For increased security, the sudo tool is used to assign temporary privileges for performing administrative tasks, which allows the root account to remain locked and helps prevent inexperienced users from inadvertently making catastrophic system changes or opening security holes.

4.4 Working with Experimental Server



Figure 4.7: Server Communication

I had also done the whole project on ubuntu experimental server and had also learnt about making your system a server.

What is a Remote Server?

In simple words its nothing much but a Computer that is not attached to a users keyboard but over which he or she has some degree of control (like can see data of that computer, can retrieve or send data etc.)

For going deep you need to know about ssh (Secure Shell). I had written about it in my old blogs. You can Google it too.

I had done it using SSH. There are few terms related to this :

- SSH: It is a Secure Socket Shell, is a network protocol that provides administrators with a secure way to access a remote computer.
- MOSH: It is a software tool used to connect from a client computer to a server over the Internet, to run a remote terminal.
- Tmux: tmux is basically a terminal multiplexer. It is used so that within one terminal window we can open multiple windows and split-views.
- OpenSSH: It is a freely available version of the Secure Shell (SSH) protocol family of tools for remotely controlling or transferring files between computers. Traditional tools used to accomplish this is telnet which is not much secure.

In Unix, you can use SCP (the scp command) to securely copy files and directories between remote hosts without starting an FTP session or logging into the remote systems explicitly.

The scp command uses SSH to transfer data, so it requires a password.

Some of the useful commands in this for checking errors or for other purposes are:

- ll: This command is used to list the detail information of files and folder of a current directory.

- `tail -f error.log`: This is used for checking errors.
- `sudo apt-get install openssh-server`
- `sudo vim /etc/ssh/sshdconfig`
(To edit this as per your preferences. But first take a backup of this file for later default configurations if needed.)
- `sudo restart ssh`
(To check your ssh daemon is running or not.)
- `ps -A — grep sshd`
(This command should produce a line like this:
some-number ? 00:00:00 sshd)
- `ssh user@hostip`
(To enter into a remote server from some other system.)

4.5 Introduction to Doxygen



Figure 4.8: Doxygen logo

Doxygen is a documentation generator, a tool for writing software reference documentation. The documentation is written within code, and is thus relatively easy to keep up to date. Doxygen can cross reference documentation and code, so that the reader of a document can easily refer to the actual code.

Doxygen supports multiple programming languages, especially C++, C, C#, Objective-C, Java, Python, IDL, VHDL, Fortran and PHP.[2] Doxygen is free software, released under the terms of the GNU General Public License.

4.5.1 Features of Doxygen

Doxygen is a tool to create a documentation for your program/project written in the languages like C, C++, Java, python and so on. It reads the well formatted and special doxygen comments to create the required documentation. This documentation is very important for the new developers who want to help in the development of the project. Documentation is one of the main pillar of an open-source project.

- Requires very little overhead from the writer of the documentation. Plain text will do, Markdown is support, and for more fancy or structured output HTML tags and/or some of doxygen's special commands can be used.
- Cross platform: Works on Windows and many Unix flavors (including Linux and Mac OS X).
- Comes with a GUI frontend (Doxywizard) to ease editing the options and run doxygen. The GUI is available on Windows, Linux, and Mac OS X.
- Automatically generates class and collaboration diagrams in HTML (as clickable image maps) and \LaTeX (as Encapsulated PostScript images).
- Allows grouping of entities in modules and creating a hierarchy of modules.
- Doxygen can generate a layout which you can use and edit to change the layout of each page.
- Can cope with large projects easily.

4.5.2 Installation of Doxygen

Doxygen can be installed using following commands:

```
$ git clone https://github.com/doxygen/doxygen.git
```

```
$ cd doxygen
```

```
$ ./configure
```

```
$ make
```

This will install Doxygen on your pc or laptop.

You can create the documentation using the graphical user interface (GUI) or console mode.

While writing the comments we have to follow a pattern with the tags i.e. before every tag we should have something special so that Doxygen can understand what are we creating. Actually Doxygen read these tags and place them at special location in the generated output. So, we have to specify them explicitly.

4.6 Introduction to Reveal-js & Reveal-md



Figure 4.9: MD & JS

Reveal-js is one of the framework of Javascript. This can be used for presentations purpose. Now before going to reveal-md lets talk about some fundamental things.

What is a Markup language?

Markup languages are designed for the processing, definition and presentation of text. The language specifies code for formatting, both the layout and style, within a text file. HTML and Markdown is an example of a widely known and used markup language.

Markdown is a lightweight Markup Language with simple plain text formatting syntax designed so that it can be converted to HTML and many other formats. It is created by John Gruber. It had '.md' or '.markdown' extention.

”Markdown is a text-to-HTML conversion software tool written in Perl for web writers.”

Moreover, to enable markdown feature of reveal.js, we need reveal-md. The Markdown feature of reveal.js is awesome, and has an easy (and configurable) syntax to separate slides. Use three dashes surrounded by two blank lines.

4.6.1 Installation of reveal-md

Installation of reveal-md is a very easy process. Type the commands in the terminal:

```
$ sudo apt-get install npm
```

```
$ sudo apt-get install nodejs-legacy
```

```
$ sudo npm install -g reveal-md
```

This will install reveal-md on your pc or laptop.

4.7 Introduction to L^AT_EX

L^AT_EX, I had never heard about this term before doing this project, but when I came to know about it's features, found it excellent. L^AT_EX is a document markup language and document preparation system for the T_EX typesetting program. Within the typesetting system, its name is styled as L^AT_EX.



Figure 4.10: Donald Knuth, Inventor Of T_EX typesetting system

Within the typesetting system, its name is styled as L^AT_EX. The term L^AT_EX refers only to the language in which documents are written, not to the editor used to write those documents. In order to create a document in L^AT_EX, a .tex file must be created using some form of text editor. While most text editors can be used to create a L^AT_EX document, a number of editors have been created specifically for working with L^AT_EX.

L^AT_EX is most widely used by mathematicians, scientists, engineers, philosophers, linguists, economists and other scholars in academia. As a primary or intermediate format, e.g., translating DocBook and other XML-based formats to PDF, L^AT_EX is used because of the high quality of typesetting achievable by T_EX. The typesetting system offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout and bibliographies.

L^AT_EX is intended to provide a high-level language that accesses the power of T_EX. L^AT_EX essentially comprises a collection of T_EX macros and a program to process L^AT_EX documents. Because the T_EX formatting commands are very low-level, it is usually much simpler for end-users to use L^AT_EX.

To run L^AT_EX on your own computer, you need to use a latex distribution. A distribution includes a latex program and (typically) several thousand packages.

- On Windows: MikT_EX or T_EXLive
- On Linux: T_EXLive
- On Mac: MacT_EX

4.7.1 Typesetting

L^AT_EX was first developed in 1985 by Leslie Lamport. In preparing a L^AT_EX document, the author specifies the logical structure using familiar concepts such as chapter, section, table, figure, etc., and lets the L^AT_EX system worry about the presentation of these structures. It therefore encourages the separation of layout from content while still allowing manual typesetting adjustments where needed.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
  \maketitle
  \LaTeX{} is a document preparation system
  for the \TeX{} typesetting program.
\end{document}
```

Apart from this lat.pdf; lat.aux, lat.log, lat.pdf files are created by default.

- AUX is a data file format used by Latex AUX is a data file format used by LaTeX. LaTeX is a macro package which uses TeX typesetting language in its documents. AUX files contain information used for cross-referencing, and is also used to transport information from one compiler run to the next.
- Some of the compilers are pdftex, Xelatex, Lualatex etc.
- A log file is usually a flat text file that contains a list of events that happen when a program was running, with one event on each line. Often times errors are recorded in log files.
- .pdf: The common output format for your document. Created by pdflatex/ xelatex

Happy Texing :)

4.8 Introduction to Github



Figure 4.11: Github Logo

GitHub is a Git repository web-based hosting service which offers all of the functionality of Git as well as adding many of its own features. Unlike Git which is strictly a command-line tool, Github provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

GitHub offers both paid plans for private repo handle everything from small to very large projects with speed and efficiency. ositories, and free accounts, which are usually used to host open source software projects. As of 2014, Github reports having over 3.4 million users, making it the largest code host in the world.

GitHub has become such a staple amongst the open-source development community that many developers have begun considering it a replacement for a conventional resume and some employers require applications to provide a link to and have an active contributing GitHub account in order to qualify for a job.

The Git feature that really makes it stand apart from nearly every other Source Code Management (SCM) out there is its branching model.

Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

This means that you can do things like:

- **Frictionless Context Switching.**
Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.
- **Role-Based Code lines.**
Have a branch that always contains only what goes to production, another that you merge work into for testing, and several smaller ones for day to day work.

- Feature Based Work flow.
Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.
- Disposable Experimentation.
Create a branch to experiment in, realize it's not going to work, and just delete it - abandoning the work with nobody else ever seeing it (even if you've pushed other branches in the meantime).

Notably, when you push to a remote repository, you do not have to push all of your branches. You can choose to share just one of your branches, a few of them, or all of them. This tends to free people to try new ideas without worrying about having to plan how and when they are going to merge it in or share it with others.

There are ways to accomplish some of this with other systems, but the work involved is much more difficult and error-prone. Git makes this process incredibly easy and it changes the way most developers work when they learn it.

4.8.1 What is Git?



Figure 4.12: Git Logo

Git is a distributed revision control and source code management (SCM) system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become the most widely adopted version control system for software development.

As with most other distributed revision control systems, and unlike most clientserver systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server. Like the Linux kernel, Git is free and open source software distributed under the terms of the GNU General Public License version 2 to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

4.8.2 Installation of Git

Installation of git is a very easy process. The current git version is: 2.0.4. Type the commands in the terminal:

```
$ sudo apt-get update
```

```
$ sudo apt-get install git
```

This will install the git on your pc or laptop.

4.8.3 Various Git Commands

Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. The commonly used Git command line instructions are:-

4.8.3.1 Create Repositories

Start a new repository or obtain from an exiting URL

```
$ git init [ project-name ]
```

Creates a new local repository with the specified name

```
$ git clone [url ]
```

Downloads a project and its entire version history

4.8.3.2 Make Changes

Review edits and craft a commit transaction

```
$ git status
```

Lists all new or modified files to be committed.

```
$ git add [file ]
```

Snapshots the file in preparation for versioning.

```
$ git commit -m "[descriptive message ]"
```

Records file snapshots permanently in version history.

4.8.3.3 Group Changes

Name a series of commits and combine completed efforts

\$ git branch

Lists all local branches in the current repository.

\$ git branch [branch-name]

Creates a new branch.

\$ git checkout [branch-name]

Switches to the specified branch and updates the working directory.

\$ git branch -d [branch-name]

Deletes the specified branch.

4.8.3.4 Synchronize Changes

Register a repository bookmark and exchange version history.

\$ git push [alias [branch]]

Uploads all local branch commits to GitHub.

\$ git pull

Downloads bookmark history and incorporates changes.

CHAPTER 5

EXPERIMENTAL RESULTS AND COMPARISON

5.1 Experimental Results

I had tried my project on different server also i.e Experimental Server here. I had tried it on both ubuntu 14.04 and 15.10. It works fine on both versions.

Below is one of the experimental result.

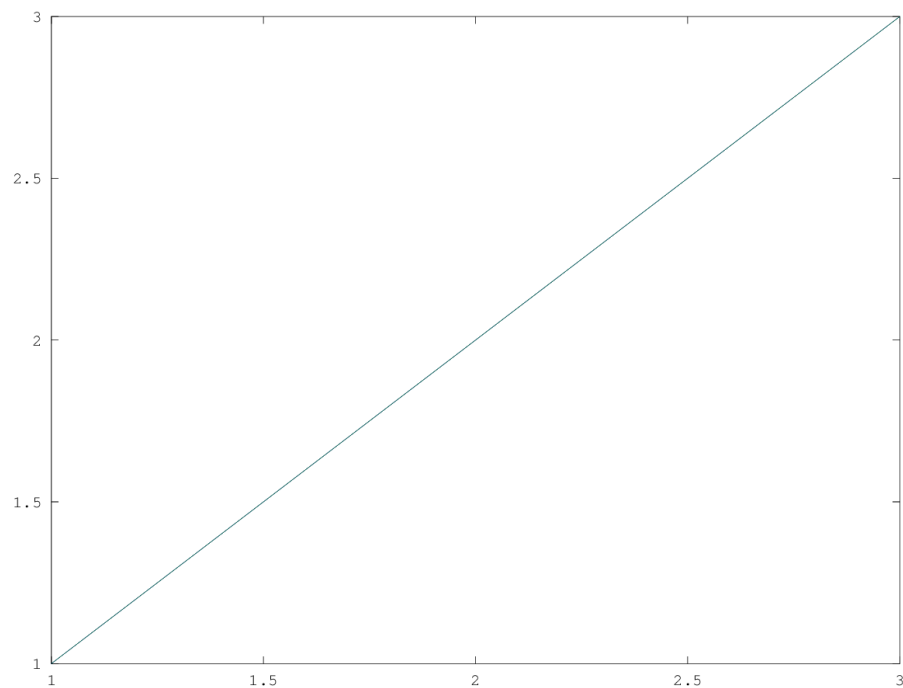


Figure 5.1: Graph Plotted

You may refer to my blogs also for detailed information.

Here is the url:

<https://deepti96.wordpress.com/>

The interface of Web-Octave looks like this but can be changed using CSS file.

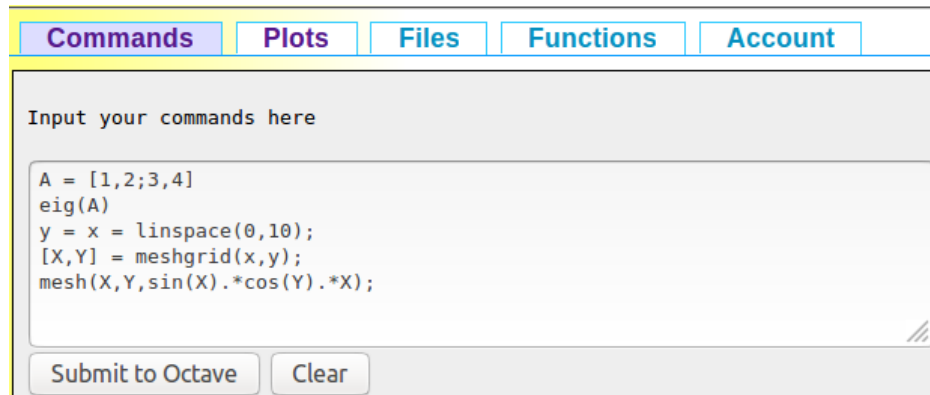


Figure 5.2: Code

Output of above code is:

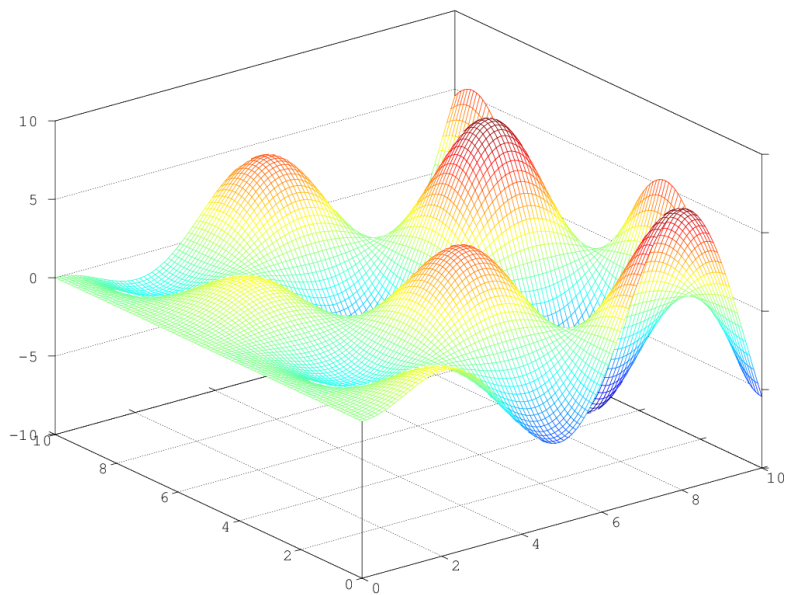


Figure 5.3: Output

5.1.1 Testing

Testing a program consists of providing the program with a set of test inputs (or test cases) and observing if the program behaves as expected. If the program fails to behave as expected, then the conditions under which failure occurs are noted for later debugging and correction.

This software had been taken through rigorous test to fully found potential causes of error and system failure and full focus have been given to cover all possible exceptions that can occur and cause failure of the software.

As this software is based on intensive background process it have been taken care that if correct input and email address are given then processing of user job can even continue or a least automatically restart even after server shuts down or even crash.

Overview of Octave versions	
Date	Publication Title
September 1999	Octave framework 1.0
September 2001	Octave framework 2.0
December 2001	Octave criteria 2.0
September 2003	Octave-S v0.9
March 2005	Octave-S v1.0
June 2007	Octave 3.x
March 2016	Octave 4.0.1
July 2016	Octave 4.0.3

Table 5.1: Octave Release

Test cases for main page(index.html)			
Input	Desired Output	Actual Output	Status
Inputs range exceeds	Alert user,Don't proceed	Alert user,Don't proceed.	Pass
Incomplete Command	Alert user about range. Don't proceed	Alert user about range. Don't proceed	Pass
PNG selected: No	Error	Don't proceed	Pass
Session: Yes	Show email field after Submit	Show email field after Submit	Pass
Help pressed	Show Detailed user help	Show Detailed user help	Pass

Table 5.2: Computational Analysis

Test cases for possible source of problems			
Input	Desired Output	Actual Output	Status
URL refreshed	Send to homepage	Send to homepage	Pass
server stops or rebooted	Start processing interrupted requests	Start processing interrupted requests	Pass

Table 5.3: Test case (general)

CHAPTER 6

CONCLUSION, SUMMARY AND FUTURE SCOPE

6.1 Future Scope

Octave is open source alternative to MATLAB. It provides for various mathematical and scientific functionalities and is being continuously improved by a wide community which is contributing for its development. Weboctave is web interface to octave. A server having Web Octave can be beneficial as it can be used for practicing purpose in colleges and also for Laboratory work in Research Centre. We can add more features like improving the login functionality also.

6.2 Technical and Managerial Lesson Learnt

I learned a lot by doing this project. During this period I got to learn a vast number of technologies. These are listed below :

- **Operating system:** Ubuntu
- **Languages used:** PHP, HTML, CSS, MySQL
- **Framework:** Reveal.js, Reveal-md
- **Typesetting:** LaTeX
- **Other Learnings:** Wordpress, Markdown

So during this project I learned all the above things. Above all I got to know how Softwares are developed from the scratch. Planning, designing, developing code, working in a team, testing etc. These are all very precious things I got to learn during this period.

- [1] Web-Octave, <https://github.com/dsdeptisharma/Web-Octave>
- [2] L^AT_EX Beginner's Guide By Stefan Kottwitz
- [3] My Blog, <http://deepti96.wordpress.com>
- [4] My Github Profile, <https://github.com/dsdeptisharma/>
- [5] Online Sources
- [6] Doxygen, www.doxygen.org