APPENDIX I

SIMULATION TOOL

A.1 INTRODUCTION

Matrix Laboratory (MATLAB) is an exclusive language for programming; also it is a fourth generation and numerical computing of multi-paradigm. It lets the process of user interface creation, interfacing with various languages (Python, Java, C, C++ and Fortran), matrix manipulations, algorithm execution and data and functions plotting. It is mainly intended for the computing of symbolic access and MUPAD engine. Some Packages additionally such as simulation of multi-domain graphical, dynamic design based system model and embedded and Simulink.

Basically, it is used in various environment and application. It is begin with the programming of algebra and it is simple. It is proceed with the batch job and interactive sessions. It precedes the program with the concept of expression, constants, variables, statements, etc.

Figure A.1: Basic View of MATLAB
It has frequent in-built commands and functions of math which I used for the calculation of mathematical, numerical methods and plots generating. It is constructed round the scripting language and involved in executed files and mathematical shell.

A. Variables

Variables are used for the declaration purposes using the operator “=”.

It is a conditional language typed due to the alignment variables. The values are constant initially and then it will change as per the computation process and the function. For example: \( x = 10; \)

B. Vectors and matrices

It define the values of variables as array and consisting the value with initial start at 1 and increment by 2 and it will terminate when it exceeding. The syntax of array is defined as

\( \text{init} : \text{increment} : \text{terminator}. \)

For instance: array = 1:2:9; array = 1 3 5 7 9

C. Structures

It defines the structure of the array and it supports the name of the field for the look ups and manipulation. Inappropriately, MATLAB JIT will not sustenance the structure of MATLAB. Therefore the cost of the structure will come in to account.

D. Functions

In creation of function, the file names have to relate with the first function and the function valid is begin with the character alphabetic. Also, it include with the underscores, letters and numbers. It is frequently case sensitive.

E. Function handles

It introduced by the lambda calculus and processes the functions which are executed either in nested or .m files.

F. Classes and object-oriented programming

It supports the programming of object-oriented which contains packages, semantics of pass by value and references, classes, virtual dispatch and inheritance. However, the
significant process of convention calling and syntax is varying from other languages. It depends on the classes of value or reference to handle it.

G. Graphics and graphical user interface programming

It helps to handle the graphics interface features and it guide to design for user easy understanding. Also, it used to plot the features in graphical wise.

![Figure A.2: Plotted vector of X and Y](image)

![Figure A.3: Graphical representation of plotted vector](image)

For example, the vector of x and y function is plotted as shown in Figure A.2. The graphical interfaces can be planned with the environment of GUI design tool. The example graph is given in Figure A.3.
H. Interfacing with other languages

The function calls and the subroutines are programmed using the Fortran or C. The function wrapper is passed by involving the data types. The object file is the dynamic loadable and processed by compiling and executing. It includes the Interface with python language. The libraries are included according to the mechanism. As replacements to the MuPAD based available Toolbox of Symbolic Math and connected to Maple or Mathematical. It exported and imported the MathML.

A.2 Computational Mathematics of MATLAB's Power

It is used for mathematics computational and the common calculations are as following.
Dealing with Matrices and Arrays

✓ 2-D and 3-D Plotting and graphics
✓ Linear Algebra
✓ Algebraic Equations
✓ Integration
✓ Transforms
✓ Curve Fitting
✓ Non-linear Functions
✓ Statistics
✓ Data Analysis
✓ Calculus and Differential Equations
✓ Numerical Calculations
✓ Various other special functions

A.3 Features of MATLAB

Following are the basic features of MATLAB –

✓ It is a high-level language for numerical computation, visualization and application development.
✓ It also provides an interactive environment for iterative exploration, design and problem solving.

✓ It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.

✓ It provides built-in graphics for visualizing data and tools for creating custom plots.

✓ MATLAB’s programming interface gives development tools for improving code quality maintainability and maximizing performance.

✓ It provides tools for building applications with custom graphical interfaces. It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

A.4 Big Data with MATLAB

How to work with huge and fast data sets: Big data mentions to the studied improvement in the data rate for creation and analysis. A primary driver of this trend is the ever increasing digitization of information. The number and types of acquisition devices and other data generation mechanisms are growing all the time.

Big data sources include streaming data from instrumentation sensors, satellite and medical imagery, video from security cameras, as well as data derived from financial markets and retail operations. Big data sets from these sources can contain gigabytes or terabytes of data, and may grow on the order of megabytes or gigabytes per day.

Big data represents an opportunity for analysts and data scientists to gain greater insight and to make more informed decisions, but it also presents a number of challenges. Big data sets may not fit into available memory, may take too long to process, or may stream too quickly to store. Standard algorithms are usually not designed to process big data sets in reasonable amounts of time or memory. There is no single approach to big data. Therefore, MATLAB provides a number of tools to tackle these challenges.

Working with Big Data in MATLAB:
64-bit Computing: The 64-bit version of MATLAB drastically increases the amount of data you can hold in memory – typically up to 2000 times more than any 32-bit program. While 32-bit programs limit you to addressing only 2 GB of memory, 64-bit MATLAB lets you address up to the physical memory limits of the OS. For Windows 8, that’s 500 GB for desktop versions and 4 TB for Windows Server.

Memory Mapped Variables: The `memmapfile` function in MATLAB lets you map a file, or a portion of a file, to a MATLAB variable in memory. This allows you to efficiently access big data sets on disk that are too large to hold in memory or that take too long to load.

Disk Variables: The `matfile` function lets you access MATLAB variables directly from MAT-files on disk, using MATLAB indexing commands, without loading the full variables into memory. This allows you to do block processing on big data sets that are otherwise too large to fit in memory.

Data store: Use the `datastore` function to access data that doesn’t fit into memory. This includes data from files, collections of files or, in conjunction with Database Toolbox, database tables. The `datastore` function allows you to define the data you want to import from your files or database tables, define the format to apply to your imported data, and manage the incremental import of your data, providing a means to iterate over big data sets using only a while loop.

Intrinsic Multicore Math: Many of the built-in mathematical functions in MATLAB, such as `fft`, `inv`, and `eig`, are multithreaded. By running in parallel, these functions take full advantage of the multiple cores of your computer, providing high-performance computation of big data sets.

GPU Computing: If you’re working with GPUs, GPU-optimized mathematical functions in Parallel Computing Toolbox provide even higher performance for big data sets.

Parallel Computing: Parallel Computing Toolbox provides a parallel for-loop that runs your MATLAB code and algorithms in parallel on multicore computers. If you use
MATLAB Distributed Computing Server, you can execute in parallel on clusters of machines that can scale up to thousands of computers.

Cloud Computing: You can run MATLAB computations in parallel using MATLAB Distributed Computing Server on Amazon’s Elastic Computing Cloud (EC2) for on-demand parallel processing on hundreds or thousands of computers. Cloud computing lets you process big data without having to buy or maintain your own cluster or data center.

Distributed Arrays: works with multidimensional matrices and arrays. Also, it distributed across the memory of a cluster of computers. Using this approach, you can store and perform computations on big data sets that are too large to fit in a single computer’s memory.

Map reduce : Use the Map reduce functionality built into MATLAB to analyze data that does not fit into memory. This is a powerful, and established programming technique that can be used to analyze data on your desktop, as well as run MATLAB analytics on the big data platform Hadoop.

Streaming Algorithms: Using System objects, you can perform stream processing on incoming streams of data that are too large or too fast to hold in memory. In addition, you can generate embedded C/C++ code from your MATLAB algorithms using MATLAB Coder, and run the resulting code on high-performance real-time systems.

Image Block Processing: The blockproc function in Image Processing Toolbox lets you work with really big images by processing them efficiently a block at a time. Computations run in parallel on multiple cores and GPUs when used with Parallel Computing Toolbox.

Machine Learning: Machine learning is helpful for extracting insights and developing predictive models with big data sets. A wide variety of machine learning algorithms including boosted and bagged decision trees, K-means and hierarchical clustering, K-nearest neighbor search, Gaussian mixtures, the expectation maximization algorithm, hidden Markov models, and neural networks are available in Statistics and Machine Learning Toolbox and Neural Network Toolbox.
Hadoop: With the Map reduce and Datastore functionality built into MATLAB, can develop algorithms on your desktop and directly execute them on Hadoop. To integrate MATLAB analytics with production Hadoop systems, use MATLAB Compiler to create applications or libraries from MATLAB Map reduce based algorithms.

A.5 Getting Started with Map Reduce

As the number and type of data acquisition devices grows annually, the sheer size and rate of data being collected is rapidly expanding. These big data sets can contain gigabytes or terabytes of data, and can grow on the order of megabytes or gigabytes per day. While the collection of this information presents opportunities for insight, it also presents many challenges. Most algorithms are not designed to process big data sets in a reasonable amount of time or with a reasonable amount of memory. Map reduce allows you to meet many of these challenges to gain important insights from large data sets.

Map reduce is a programming technique for analyzing data sets that do not fit in memory. You may be familiar with Hadoop® Map reduce, which is a popular implementation that works with the Hadoop Distributed File System (HDFS™). MATLAB® provides a slightly different implementation of the Map reduce technique with the map reduce function. The map reduce uses a data store to process data in small chunks that individually fit into memory. Each chunk goes through a Map phase, which formats the data to be processed. Then the intermediate data chunks go through a Reduce phase, which aggregates the intermediate results to produce a final result. The Map and Reduce phases are encoded by map and reduce functions, which are primary inputs to map reduce. There are endless combinations of map and reduce functions to process data, so this technique is both flexible and extremely powerful for tackling large data processing tasks.

The utility of the map reduce function lies in its ability to perform calculations on large collections of data. Thus, map reduce is not well-suited for performing calculations on normal sized data sets which can be loaded directly into computer memory and analysed with traditional techniques. Instead, use map reduce to perform a statistical or analytical calculation on a data set that does not fit in memory.
Each call to the map or reduce function by map reduce is independent of all others. For example, a call to the map function cannot depend on inputs or results from a previous call to the map function. It is best to break up such calculations into multiple calls to map reduce.

Map reduce Algorithm Phases:

The map reduce moves each chunk of data in the input data store through several phases before reaching the final output. The following Figure A.4 outlines the phases of the algorithm for map reduce.

![Figure A.4: Phases for Map reduce](image)

The algorithm has the following steps:

The map reduce reads a chunk of data from the input data store using \([\text{data}, \text{info}] = \text{read}(\text{ds})\), and then calls the map function to work on that chunk. The map function receives the chunk of data, organizes it or performs a precursory calculation, and then uses the add function and add multi functions to add key-value pairs to an intermediate data storage object called a Key Value Store. The number of calls to the map functions by map reduce is equal to the number of chunks in the input data store.

After the map function works on all of the chunks of data in the data store, map reduces groups all of the values in the intermediate Key Value Store object by unique key. Next, map reduce calls the reduce function once for each unique key added by the map function. Each unique key can have many associated values. The map reduce passes
the values to the reduce function as a Value Iterator object, which is an object used to iterate over the values. The Value Iterator object for each unique key contains all the associated values for that key.

The reduce function uses the has next and get next functions to iterate through the values in the Value Iterator object one at a time. Then, after aggregating the intermediate results from the map function, the reduce function adds final key-value pairs to the output using the add and add multi functions. The order of the keys in the output is the same as the order in which the reduce function adds them to the final Key Value Store object. That is, map reduce does not explicitly sort the output.

**A.6 MATLAB Map reduce and Hadoop**

While MATLAB Map reduce is optimized for array-based analysis, it is fully compatible with Hadoop Map reduce, so you can run your Map reduce based algorithms within the Hadoop Map reduce framework:

Execute Map reduce based algorithms on Hadoop directly from the MATLAB desktop, using MATLAB Distributed Computing Server™. The Package Map reduce based algorithms for deploying to production Hadoop systems, using MATLAB Compiler.

a. Map reduce on Hadoop:

Execute MATLAB Map reduce based algorithms within Hadoop Map reduce to explore and analyse data that is stored and managed on Hadoop, using MATLAB Distributed Computing Server.

b. Run map reduce on a Hadoop Cluster:

Create applications and libraries based upon MATLAB Map reduce for deployment within production instances of Hadoop, using MATLAB Compiler.

c. Cluster Preparation:

Before running map reduces on a Hadoop® cluster, make sure that the cluster and client machine are properly configured. Consult your system administrator, or see Configure a Hadoop Cluster.
d. Output Format and Order:

When running map reduce on a Hadoop cluster with binary output (the default), the resulting Key Value Data store points to Hadoop Sequence files, instead of binary MAT-files as generated by map reduce in other environments. For more information, see the 'Output Type' argument description on the map reduce reference page. When running map reduce on a Hadoop cluster, the order of the key-value pairs in the output is different compared to running map reduce in other environments. If your application depends on the arrangement of data in the output, you must sort the data according to your own requirements. First, you must set environment variables and cluster properties as appropriate for your specific Hadoop configuration. See your system administrator for the values for these and other properties necessary for submitting jobs to your cluster.

e. Speed Up and Deploy Map reduce Using Other Products:

Execution Environment: To use map reduce with Parallel Computing Toolbox™, MATLAB® Distributed Computing Server™, or MATLAB Compiler™, use the map reduce r configuration function to change the execution environment for map reduce . This enables you to start small to verify your map and reduce functions, then quickly scale up to run larger calculations.

Running in Parallel: Parallel Computing Toolbox can immediately speeds up the map reduce algorithms by using the full processing power of multicore computers to execute applications with a parallel pool of workers. If you already have Parallel Computing Toolbox installed, then you probably do not need to do anything special to take advantage of these capabilities. For more information about using map reduce with Parallel Computing Toolbox, see Run map reduce on a Parallel Pool.

MATLAB Distributed Computing Server enables you to run the same applications on a remote computer cluster. For more information, including how to configure MATLAB Distributed Computing Server to support Hadoop® clusters, see Big Data.
Application Deployment: MATLAB Compiler enables you to create standalone map reduce applications or deployable archives, which you can share with colleagues or deploy to production Hadoop systems.

A.7 Data Storage

According to the proposed research work, the storage is consists in the free storage tool like Google drive, Drop Box, etc. The device is connected with the MATLAB for research work storage process. If off line means, it considers the hardware resource like server, storage capacity, etc.

A.7 Hadoop Integration

There are two ways to deploy Map reduce applications. You can create a deployable archive or you can create a standalone application to run against Hadoop. Create a deployable archive if you want to use a Hadoop scheduling framework or if you want to integrate MATLAB code with an existing Hadoop job flow. A deployable archive includes a datastore, a map function, and a reduce function. The execution of map reduce function generates a Hadoop command that submits a job to Hadoop.