ECE 4094

Project A

Design Specifications

	First Name	Last name	ID	Email
Supervisor 1	James	Saunderson		James.Saunderson@monash.edu
Supervisor 2	Jonathan	Li		Jonathan.Li@monash.edu
Student 1	Simon	Teshuva	24195537	Srtes1@student.monash.edu

Contents

1. Code Sharing and Backing up

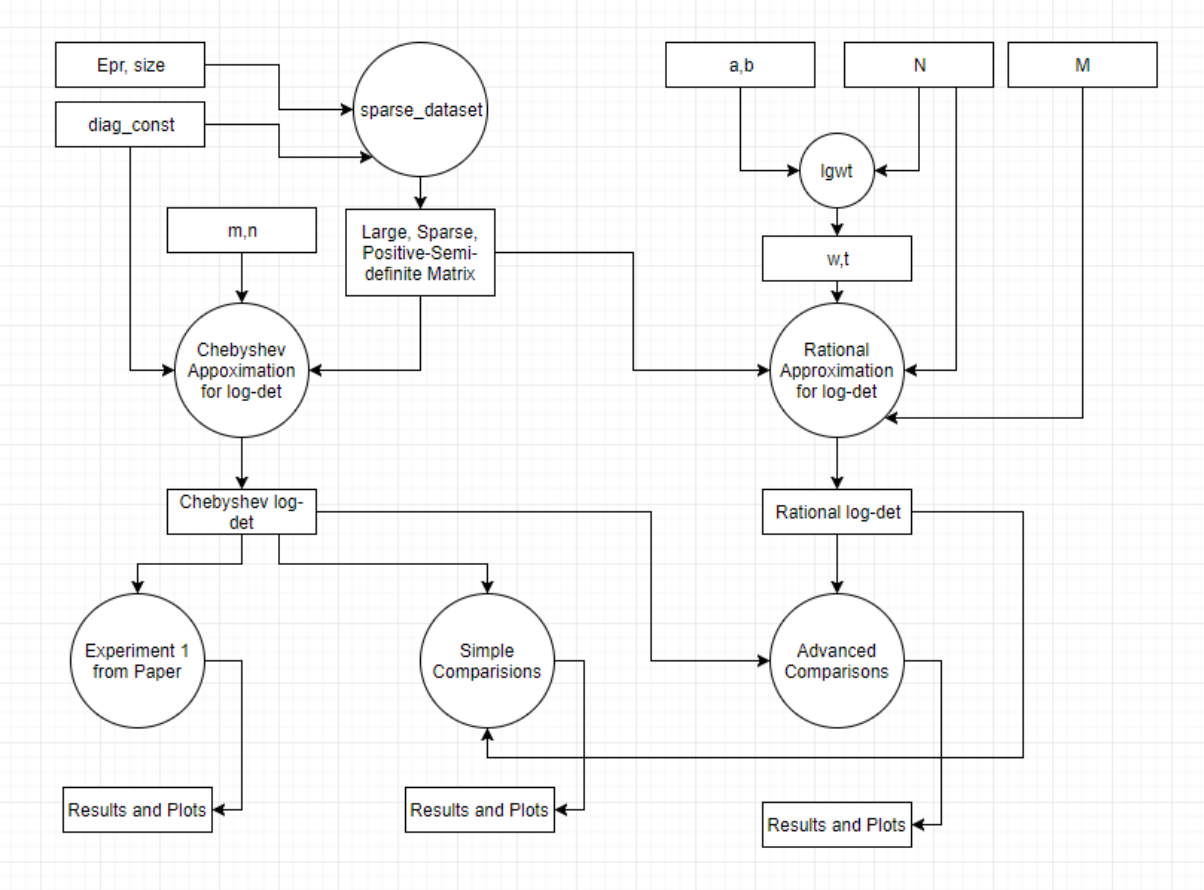
As this project will primarily involve coding and report writing I will need a repository for version control. I have chosen to use GitHub as it is common industry practice.

2. Software

2.1 Language

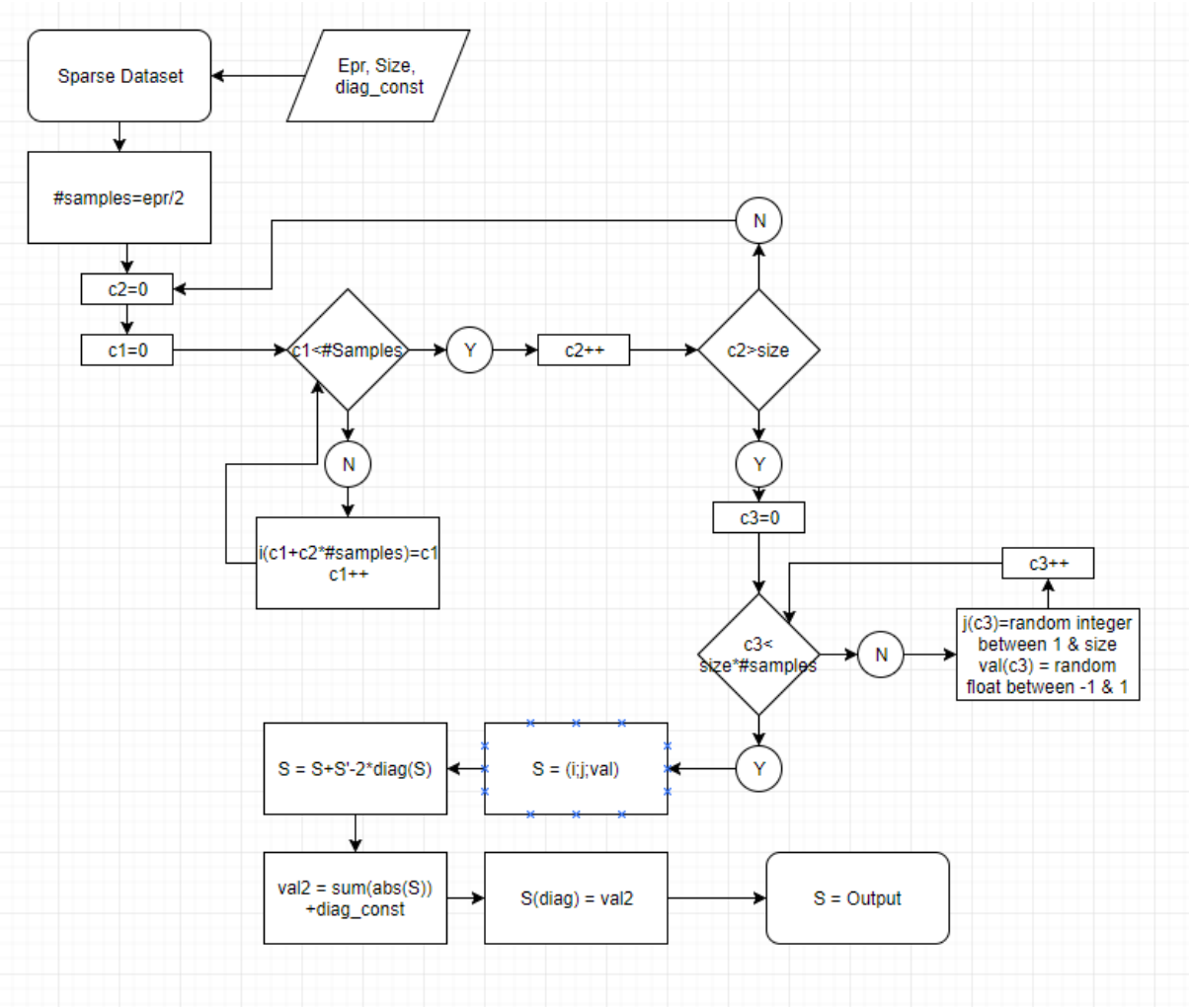
When choosing a programming language to code in for this project I had three main choices: MATLAB, Python, and C. I have chosen to use MATLAB as it has the largest set of libraries and pre-made functions, and is the easiest to code in. Furthermore, I expect that the thing that will bottleneck performance in this project is the power of the computer used. However, should I find that it is the choice of language that is bottlenecking performance, I will port my code over to Python. If the language choice is still bottlenecking performance, then I will port the code to C.

2.2 Data flow diagram

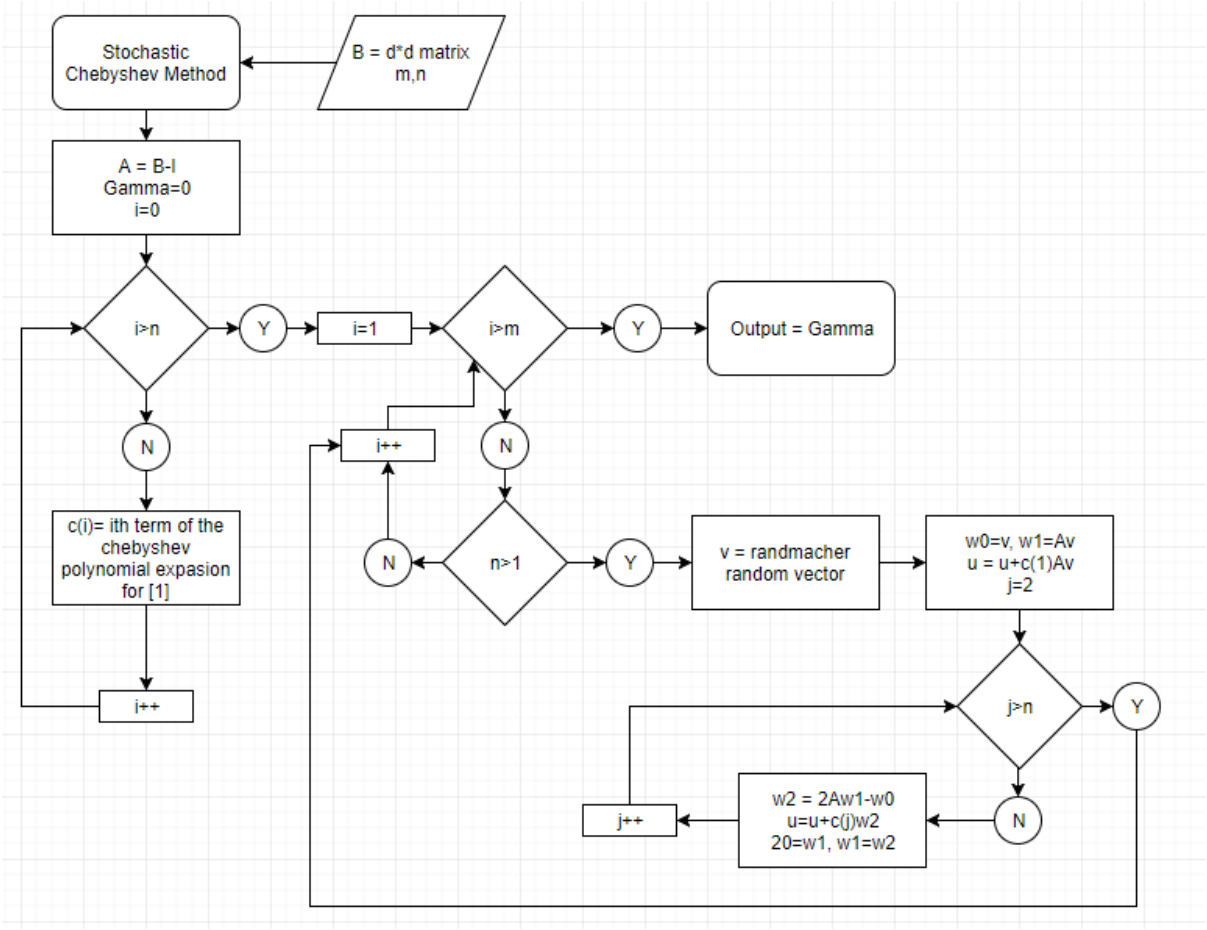


2.3 Flow charts

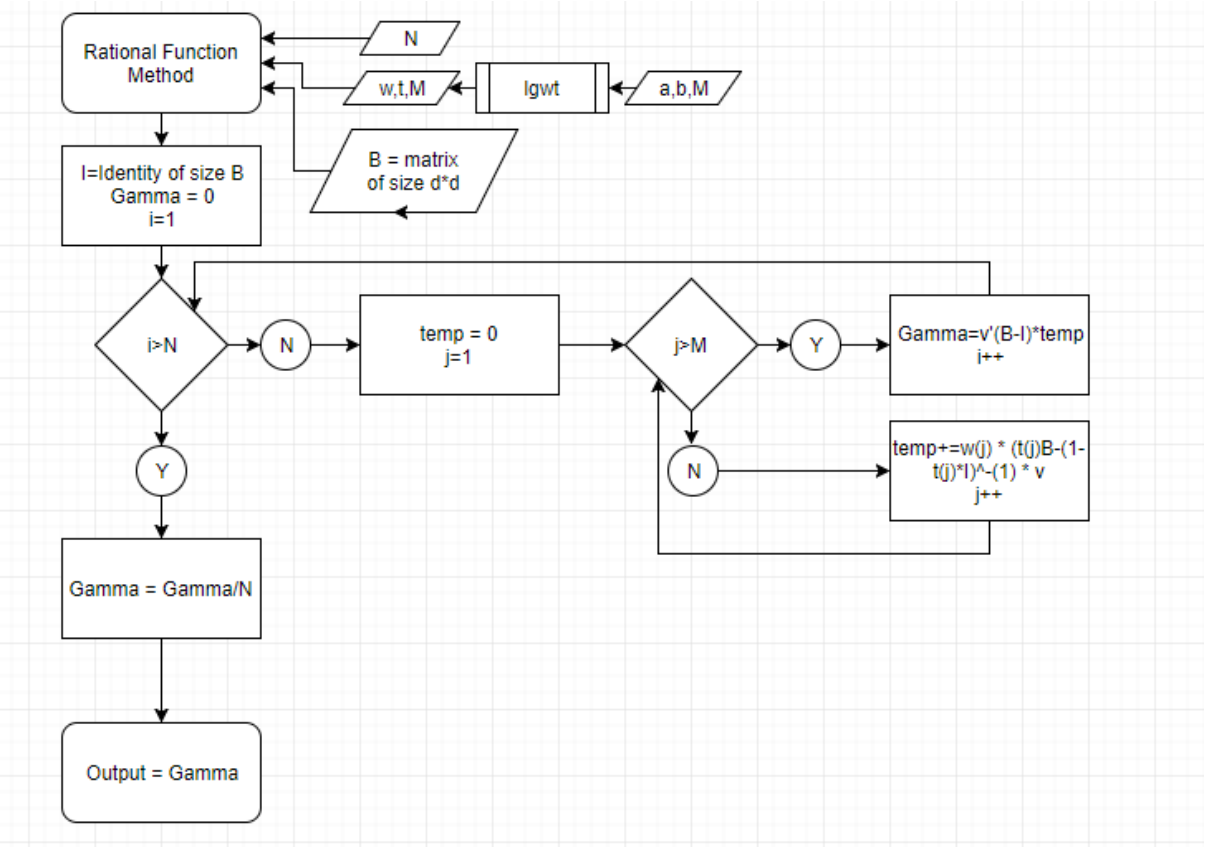
2.3.1 Generate dataset



2.3.2 Chebyshev Algorithm



2.3.3 Rational Algorithm



3. Computing Power Needed

I need a computer which can run MATLAB and some third-party libraries such as Chebfun and lgwt. I also need a computer with enough RAM to store the matrices I will be working on and perform computations on them. The largest matrix I expect to use will be of size 6e7 by 6e7 and using MATLAB's "whos" function I found that a Sparse matrix of this size needs around 10.8GB. As a result, I feel that my home desktop will be sufficient for this project. It has a GTX1060 graphics card, i5-7600 processor, a 250GB SSD, a 2TB HDD and 16GB of RAM. I expect that when I implement parallelism for the Rational Function method I may need access to a computer with more cores than mine. If that is the case I will try and use on of the more powerful computers in the Department of Electrical and Computer Systems Engineering or in the Faculty of Information Technology.

4. Experimental methodology

4.1 Implementing Parallelism

I will be using MATLAB's Parallel Computing Toolbox to implement parallelism in the Rational Function method. The method includes two nested loops where many approximations for the logdet are calculated. Each of these approximations are completely independent and can therefore all be calculated in parallel. The inner loop is used to sum the various parts of the rational function used to approximate the logdet. The outer loop is used to calculate a set of approximations for the logdet, after which, the average of these approximations is returned as the final result. I will start by implementing parallelism at the outer loop as it seems simpler to parallelise each of the approximations from each other, than to parallelise the steps within an approximation. Once the outer loop is parallelised however, I will attempt to also parallelise the inner loop and combine the two.

In terms of maximum speed up, the inner loop runs M times, while the out loop runs N times, so the maximum theoretical speedup possible is M*N, less the cost of running the parallel toolbox. In reality, the theoretical maximum speedup is never attained, and it is very hard to estimate without running the code. As a result, if I find that the parallelism increases performance by any meaningful amount, I will consider it an effective implementation of the parallelism.

4.2 Comparisons Between Methods

The three main criterion for determining the effectiveness of these methods are: relative accuracy (compared to exact methods), runtime, and space used. Runtime and space used can be compared in the same experiment. I will run the two algorithms over a wide set of data, ranging in size from $1e3 \times 1e3$ to $1e7 \times 1e7$. In addition, for each matrix size I will vary the density of non-zero elements, from 5 per row, to 50 per row. I will MATLAB's "tic" and "toc" functions to measure the time to execute, and the "whos" function to measure the space used. In addition, when computing the smaller matrices ($1e3 \times 1e3$ to $3e4 \times 3e4$), I can also compute the logdet using Cholesky Decomposition to get an exact solution. I can then compare the results from the Chebyshev algorithm and the Rational Function algorithm to the results from the Cholesky Decomposition to get the relative accuracy. This can be done both for the standard Rational Function algorithm and its parallelised version. After these comparisons have been performed I will need to devise more sophisticated comparisons.

5. Interrelatedness of Major Sections of the Project

- Generate dataset:
    - I need to be able to generate a large, sparse, positive semi-definite matrix. I am using a method described in [2] which creates a diagonally dominant matrix.
- Chebyshev Algorithm
    - I need to implement the algorithm described in [2]. being able to properly analyse the effectiveness of this algorithm relies on being able to properly generate the dataset described above.
- Experiment 1
    - There are a number of experiments performed in [2] that I will be replicating. To do so, I need to have the algorithm proposed in the paper working, as well as be able to generate the dataset described above.
- Rational Algorithm
    - James Saunderson has proposed an alternative approach to the algorithm described in [2], which I will be developing an implementation for. To properly test it, I need to be able to generate the dataset.
- Compare Chebyshev and Rational Algorithm
    - I will be performing some comparisons on the effectiveness of Chebyshev Algorithm and the Rational Function Algorithm. To do so, I need both algorithms working and need to be able to generate the dataset.
- Parallel Rational Algorithm
    - Due to the nature of the approach proposed in Rational Algorithm, there is a lot of room for parallelism. Once the original algorithm is working, I will modify it to be parallel.
- Compare Chebyshev and Parallel Rational
    - Once Rational Algorithm is parallel, I will be able to perform some comparisons on Chebyshev Algorithm and the parallelised Rational Algorithm.
- Advanced Dataset
    - Once the main parts of the project are completed I will attempt to generate a dataset which is not automatically diagonally dominant.
- Real world Data
    - I will be performing some analysis on a real-world dataset, in order to give the project more grounding in practical applications. In order to do this, I need Chebyshev Algorithm and Parallel Rational Algorithm working.
- GUI
    - Once all the coding for the project is done I will be generating a GUI to help viewers get a more intuitive understanding of what the code means and what it does. To begin work on this section, I need Chebyshev Algorithm and Parallel Rational Algorithm working. In addition, having the real-world dataset ready would give this section addition weight as the example used in the GUI can be a practical, real world one.

6. References

[1] $\log\left(1-\frac{(1-2\delta)x+1}{2}\right)$

[2] “Large-scale Log-determinant Computation through Stochastic Chebyshev Expansions”, I. Han, D. Maliotov, J. Shin