# Design Specifications

## 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.

Software

### 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.

### Data flow diagram

### Generate dataset

### Algorithm 1

### Algorithm 2

### Flow charts

### Generate dataset

### Algorithm 1

### Algorithm 2

## 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.

## Experimental methodology/supporting theory/developed code

## Time line

* Design, experimental designs

## Major subsections of project

* Generate dataset:
  + I need to be able to generate a large, sparse, positive semi-definite matrix. I am using a method described in “paper name” which creates a diagonally dominant matrix.
* Algorithm 1
  + I need to implement the algorithm described in “paper name”. 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 “paper name” 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.
* Algorithm 2
  + James Saunderson has proposed an alternative approach to the algorithm described in “paper name”, which I will be developing an implementation for. To properly test it, I need to be able to generate the dataset.
* Compare A1 and A2
  + I will be performing some comparisons on the effectiveness of Algorithm 1 and 2. To do so, I need both algorithms working and need to be able to generate the dataset.
* Parallel Algorithm 2
  + Due to the nature of the approach proposed in Algorithm 2, there is a lot of room for parallelism. Once the original algorithm is working, I will modify it to be parallel.
* Compare A1 and Parallel A2
  + Once Algorithm 2 is parallel, I will be able to perform some comparisons on Algorithm 1 and the parallelised Algorithm 2.
* 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 algorithm 1 and parallel algorithm 2 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 algorithm 1 and parallel algorithm 2 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.