## **Computing Assignment 1**

## **Finite-Precision Errors in GE for Large Linear Systems**

The goal of this assignment is to estimate the matrix size N\* at which the error resulting from Gaussian Elimination (GE) is as large as the solution. N\* can be found by ploting log10(N) vs ℇres, where N is the array of matrix sizes and ℇres is the array of typical errors for coresponding N. To achieve an accurate estimate, the array of N values, and the number of experiments for each (Nexp) must be fine tuned.

**Determining Values of Nexp and N.**

As the number of experiments for each N value increases, the better our estimate will be for the typical GE error. Nexp needs to be its maximum value at which the computation time is still efficient. To find this value, execute an array of experiments each with a different value of Nexp ranging from 200 to 10000 incrementing by 200, and N = 128. Plotting the resulting ℇres vs Nexp yeilds Figure 1.

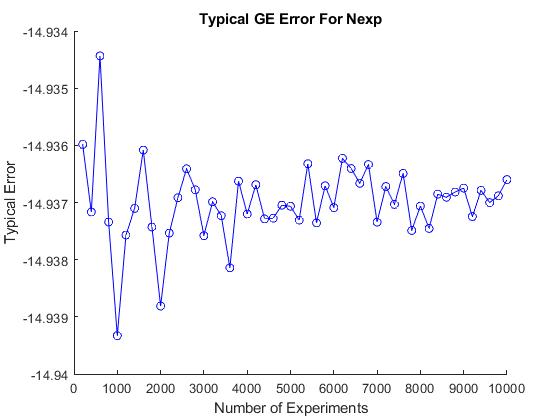
*Figure 1: Typical GE error for Nexp*

Figure 1 resembles a Damped Sine Wave, which confirms that more experiments means better accuracy. The waveform stablizes at Nexp ≈ 3000, at which the computation time is approximatly 2 seconds, an acceptable efficency.

Let Nexp = 3000.

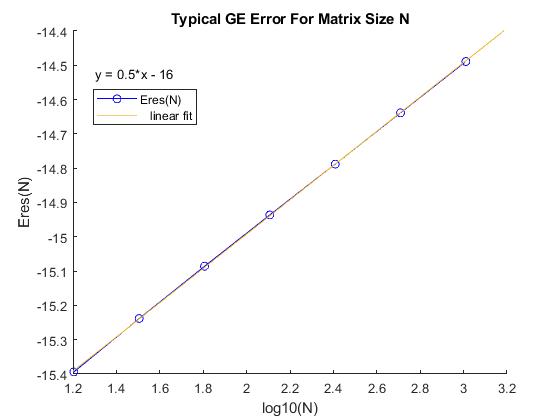
N\* can be estimated by taking the best fit line of the plot log10(N) vs ℇres. To have an acurate best-fit line we need our x values – log10(N) – to be evenly spaced. Therfore N must be an array of values following an exponetial sequence.

Let N = [16, 32, 64, 128, 256, 512, 1024].

The exponential function with base 2 was chosen for its gradual slope, which ensures maximum efficency.

**Solving For N\***

Ploting log10(N) vs ℇres generates the graph seen in figure 2.

*Figure 2: Typical GE Error For Matrix Size N.*

Taking the best fit line of ℇres and sloving for ℇres ≈ 0, we get log10(N\*) ≈ 32. Solving further we get:

N\* ≈ 1032.

At a matrix size of 1032 our typical GE error is approximatlly zero, meaning that each digit of the solution is corrupt; the resulting error is as large as the solution.

GE has the Big O notation of O(n3), using GE for a matrix of size N\* would result in an exceptionally large time complexity. None of today’s computers are robust enough to solve this problem efficently.