



**DATE: 22ND MARCH 2011**

**CSC 160**

**GAUSSIAN ELIMINATION PROJECT—SOLVING SYSTEMS OF EQUATIONS**

# **GAUSSIAN ELIMINATION PROJECT— SOLVING SYSTEMS OF EQUATIONS**

---

**ABSTRACT:**

THIS CODE FOR THIS PROJECT EXAMINES, IMPLEMENTS, AND ENHANCES A WAY OF SOLVING A SYSTEM OF LINEAR EQUATIONS USING GAUSSIAN ROW REDUCTION TECHNIQUES. THIS CODE IS THEN ANALYZED IN TERMS OF CORRECTNESS, TIMING AND DISTRIBUTION TESTS. FURTHERMORE, THE PERTURBATION EXPERIMENT AND OTHER CONDITIONED SYSTEMS ARE EXAMINED TO TEST THE GAUSSIAN REDUCTION TECHNIQUES WITH AND WITHOUT PIVOTING. PLEASE NOTE THAT ALL PROGRAMMING IS COMPLETED THROUGH THE PROGRAM MATLAB

---

# Theory

---

The goal of this project is to solve a system of equations using Gaussian Elimination. The following spread explains the basic theories that were used in order to program all parts of the code, including how to transpose and multiply matrixes (as these are later used in the research):

## Transpose Matrix (Appetizer 1)

When a matrix is transposed the rows of the matrix are exchanged with the columns of the matrix. The following example shows a theoretical example of what a transpose of a matrix looks like mathematically (courtesy of Wikipedia):

$$\begin{bmatrix} 1 & 2 \end{bmatrix}^T = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}.$$

## Matrix Multiplication (Appetizer 2)

The multiplication of a matrix is a cross product of two arrays of numbers. Another way to consider this is by considering this to be the dot product of the row of one matrix times the column of the second matrix. With this regard the inner dimensions of the matrix must match. The following portray the mathematical and theoretical representations of the matrix:

$$\text{if } A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ and } B = \begin{bmatrix} e & f \\ g & h \end{bmatrix} \text{ then } {}^{(1)}AB = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix} \quad (AB)_{i,j} = \sum_{k=1}^p A_{ik}B_{kj},$$

Note that the code for this matrix multiplication should therefore test the inner dimensions of the matrices before performing the multiplication.

## Gaussian Reduction (Main Course 1)

The process of Gaussian Reduction is used in order to solve a system of linear equations. A system of linear equations can be rewritten in terms of a matrix in the following form (courtesy of Wikipedia):

$$\begin{aligned} 2x + y - z &= 8 & (L_1) \\ -3x - y + 2z &= -11 & (L_2) \\ -2x + y + 2z &= -3 & (L_3) \end{aligned}$$

Therefore, the Gaussian Elimination algorithm applied to the [augmented matrix](#) begins with:

$$\left[ \begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array} \right]$$

Note that in this case the vector of constants is combined with the parameter matrix in order to form an augmented matrix. While this differs from the suggested code, the section on 'modifications' explains some key changes to the suggested code to enhance the performance of the system.

The next step of this process is to use row reduction techniques in order to create a matrix of upper triangular form. For instance subtracting  $-3/2$  of the first row from the second row will result in a zero in the (2,1) position.

$$\left[ \begin{array}{ccc|c} 1 & \frac{1}{3} & \frac{-2}{3} & \frac{11}{3} \\ 0 & 1 & \frac{2}{5} & \frac{13}{5} \\ 0 & 0 & 1 & -1 \end{array} \right]$$

That is, it is in [row echelon form](#).

At the end of the algorithm, if the [Gauss–Jordan elimination](#)(zeros under and above the leading 1) is applied:

$$\left[ \begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -1 \end{array} \right]$$

That is, it is in [reduced row echelon form](#), or row canonical form.

Thus repeating this process the upper triangular form can be obtained as portrayed to the right.

Repeating this system from the last row and moving back up we can find the reduced row echelon form.

Thus the solution to this system is given by the solution vector:  $\langle 2, 3, -1 \rangle$ , In other words  $x=2, 3=y, -1=z$ .

Note that during this process it is important to first check that this is a square matrix, because the given algorithm is invalid for matrixes that are not square—the number of unknown variables solved by the system of equations must be matched by an equal number of functions.

Furthermore it must be ensured that none of the diagonal entries are zeroes because if the diagonals are zero, then it might imply that the function has a changing variable and pivoting is needed to solve this problem. Note that the next section deals with pivoting, using it to solve the system.

### Gaussian Reduction with Pivoting (Main Course 2)

It was mentioned earlier that pivoting may occur if one of the entries on the diagonal are 0. If this happens, then the rows can be switched to make the row with the zero on the diagonal below the actual diagonal. This is called partial pivoting. An example of partial pivoting is as follows (courtesy of Wikipedia):

$$\left[ \begin{array}{ccc|c} 1 & -1 & 2 & 8 \\ 0 & 0 & -1 & -11 \\ 0 & 2 & -1 & -3 \end{array} \right]$$

For example in the situation depicted to the right, while the Gaussian Reduction above does not account for the zero in the second row, partial pivoting allows us to switch the second and third rows thus eliminating the zero on the diagonal.

If there is a zero on the diagonal that cannot be eliminated, then the matrix is singular, in other words it cannot be solved. Note that when the rows are switched, the corresponding rows of the vector of constants also have to be switched.

### Random Test (Desert)

Note that for any real parameter matrix and coefficient vector, if the augmented matrix is not singular, then it can be said that there exists a solution. This applies for any random non-integer non-positive number set also.

# Method and Code Modifications (Extensive Research)

---

The algorithms as stated above and as given on the course website are used to implement the Gaussian reduction code, first without and then including the partial pivoting.

All of these original codes are included in the compressed zip folder named OriginalMethodandCode.

After the author created the code as was suggested in the texts, a further step of refining was conducted. The following considerations were taken in order to make the code more efficient in time and accuracy measurements, as well as storage of the file.

## Some Considerations

- It was noted in the theory above that the matrix of parameters and that of the constant vector can be combined in order to create an augmented matrix. Using this method allows a more efficient and faster row reduction calculation as where originally the same row has to be swapped in two different matrixes, in the modified code only one swap is necessary.
- When matrixes are initialized, the full size is first initialized then the spaces filled in. This speeds up the process because then the function does not need to continuously expand and grow, thus increasing the speed and functionality of the script.
- It must be noted that while it enables a clear understanding of the several different steps of Gaussian reduction, the numerous embedded functions in the `gauss_reduce` function impede the computer processing speed. It is much faster to execute a single operation rather than calling several embedded operations. Thus another major consideration taken into account was that many of the sub functions of the original function were eliminated and the code concatenated to the main `gauss_reduce` function's file. Note that while this may seem to make the code more complicated, for the user, this is more efficient—there is only one file that contains the entirety of the function. This new code is noted in the code name by the letter 'n'.
  - Note that the m file for the research extension of this project is named `main_research` in order to make it distinct from the m file that was created in order to test the Gaussian reduction formulas.

The remainder of this report analyzes the effectiveness of this method in terms of correctness, timing, and distribution as well as the perturbation experiment and ill conditioned systems.

# Correctness Test

---

The main aim of this project is to solve a system of linear equations. Thus it is only natural to need to test whether or not the solutions obtained through this project result in the expected answers.

## Method

In order to verify whether or not the process of Gaussian Elimination is consistent and that it works, the vector of constants can be found from the system of equations and the solutions to this system of equations. This can be mathematically represented in the following way (courtesy of Wikipedia):

### Matrix equation

The vector equation is equivalent to a [matrix](#) equation of the form

$$A\mathbf{x} = \mathbf{b}$$

where  $A$  is an  $m \times n$  matrix,  $\mathbf{x}$  is a [column vector](#) with  $n$  entries, and  $\mathbf{b}$  is a column vector with  $m$  entries.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

The number of vectors in a basis for the span is now expressed as the [rank](#) of the matrix.

Thus by multiplying the matrix of parameters ( $A$ ) with the solution vector ( $\mathbf{x}$ ), we expect to obtain the original vector of constants ( $\mathbf{b}$ ).

Note that it is important to remember  $A\mathbf{x}$  is not the same as  $\mathbf{x}A$ .

Because the examples of matrixes generated by the random test are not whole integers, when performing various operations and performing the Gaussian elimination, the values of the matrix may change slightly but noticeably enough to be detected by the `==` operator. For this reason the floating point numbers are compared and contrasted using matlab's built-in `eps()` function.

The value of `eps()` as stored in the matlab library is `2.2204e-016`. This is the minimal distance between two floating point numbers that matlab can recognize. Thus if the absolute value of the difference between any two given numbers is smaller than the value of `eps()`, we can deduce that according to matlab the two numbers are equal. The code for this algorithm is completed in the `correctness_test.m` file, and is executed in the main file. (Note that for the solution of the matrix to be correct all of the solutions in the solution vector must match)

## Data and Analysis

The main file tests the correctness of the Gaussian elimination code 1000 times; the results can be seen by executing the m-file.

## Summary

It is found that the Gaussian Elimination project leads to a decreasing accuracy as the number of terms in the matrix increase. For a 2x2 matrix accuracy was 0.328, 3x3 it was 0.045, for 4x4 0.009. This is because there are a greater number of solutions in the solution vector, thus it is less likely that they are all correct. This low accuracy is also partially due to the usage of the `eps()` function.

# Timing Test

## Matrix equation

The vector equation is equivalent to a matrix equation of the form

$$A\mathbf{x} = \mathbf{b}$$

where  $A$  is an  $m \times n$  matrix,  $\mathbf{x}$  is a column vector with  $n$  entries, and  $\mathbf{b}$  is a column vector with  $m$  entries.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

The number of vectors in a basis for the span is now expressed as the rank of the matrix.

Theoretically the time it takes to compute the solution of a  $N \times N$  system is proportional to  $N^3$ . The following tests the speed for a range of different ranges of  $N$  and compares the relationship to a cubic one.

## Method

Matlab's inbuilt functions tic and toc allow the user to measure the time that passes between two parts of a running simulation. In order to keep the timing consistent, a separate timing function called timing\_test.m is created. The time is measured only during the gaussian\_elimination\_n, and not for the generation of a random system. The same process is repeated for the partial pivoting function also. The spread below portrays the results and graphs obtained.

## Data and Analysis

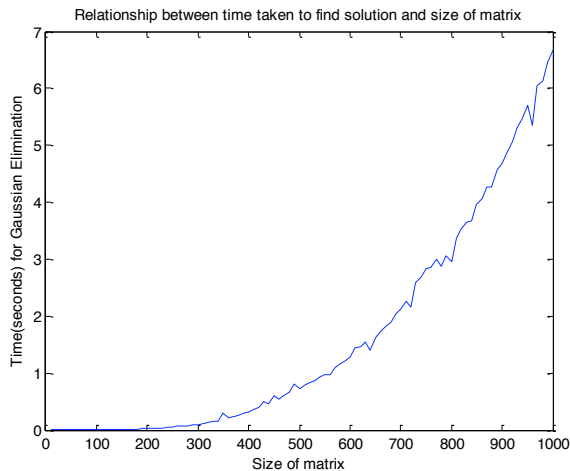
### Gaussian Reduction

For  $N=10$ , time was 0.007216 seconds

$N=100$ , time was 0.007818 seconds

$N=1000$ , time was 6.957945

The following graphs show this progression from  $N=10:1000$  in 10  $N$  intervals:



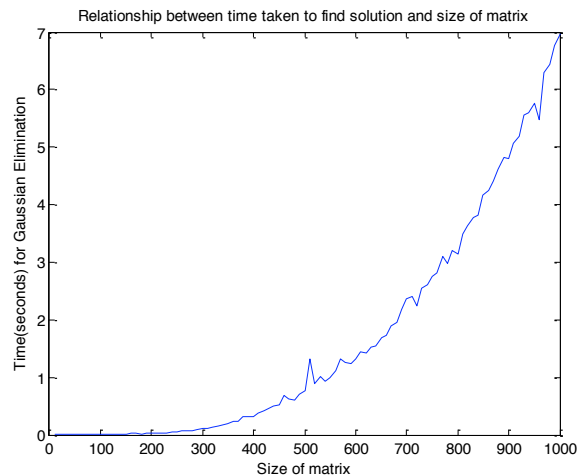
### Gaussian Reduction with Partial Pivoting

For  $N=10$ , time was 0.0011663 seconds

$N=100$ , time was 0.008028 seconds

$N=1000$ , time was 7.082447

The following graphs show this progression from  $N=10:1000$  in 10  $N$  intervals:



## Summary

The graphs above both definitely portray exponential relationships. Note that any spikes can be explained through spikes in the computer's processing systems (however there was an attempt to

minimize this by closing all other programs) It can be seen that the partial pivoting results in a larger time taken. This is evident when comparing the N=1000 time results, the time taken for partial pivoting is slightly larger. Specifically it is approximately 2% slower.

## Distribution Test

---

The following spread examines the distribution of solutions for the Gaussian Elimination for a 3x3 matrix. The mean and standard deviation of each of the variables is calculated. Thus there are three means and standard deviations.

### Method

The mean of a set of data can be defined as the sum of all of the values of the data divided by the number of pieces of data that exist for the set of values. The standard deviation is the measurement of diversity in data. The following formulas (courtesy of Wikipedia) are the mathematical representations of the mean and standard deviations for a given set of data:

Let  $X$  be a random variable with mean

$$E[X] = \mu.$$

Here the operator  $E$  denotes the average

$$\sigma = \sqrt{E[(X - \mu)^2]}.$$

In this case there are three sets of data for each of the variables, thus the mean and standard deviation are calculated for each set of data separately, and then compared to each other.

Because in this case the matrices are generated are composed of floating point numbers that encompass a range from -100 to 100. Thus theoretically it is expected that the mean of each of the solutions will not relate to each other, but the mean is very close to the value of 0. Furthermore, the more accurate the data is (the closer the mean to the median, 0), the smaller the standard deviation.

The code that calculates the mean and standard deviation are in the m file names deviation\_test.m, the results due to a single run of the function is given below.

### Data and Analysis

The data received below for the mean and standard deviations are for one run of the function

```
Distribution Test
Here the distribution in terms of standard deviation and mean for 3x3 NxN matrices are compared
This is the three means, and std of a 3x3 matrix when calculated 100 times
mean =
    -2.6654    -0.69525    -1.0587
std =
    12.775     10.171     10.986
```

As predicted, the mean is close to 0 because it is approximately the middle point of the data range. The std is comparable to each other while is good suggesting that the data for each of the three variables are reasonable. Furthermore, in this instance the std is narrow suggesting that the data spans a more compact range. Note that the greatest displacement of mean from 0 corresponds to the greatest std.

## Summary

The calculation of mean and std for large arrays of data are especially useful and applicable in math and statistic courses such as biology etc. The function used to calculate this as shown above has purpose.

# Perturbation Experiment and Effect of Pivoting

---

The Perturbation Experiment seeks to find how changing the values of a parameter matrix and its corresponding coefficients can affect the solution of the system. This is done by adding a 1% noise to the original system as is described below:

## Method

The perturbation experiment is conducted in several steps. First a random matrix is generated and solved, the solution is saved. 1% noise is now added to the given matrix. Thus if the original range of values was from -100 to 100, each coefficient of the matrix has a random number between -1 and 1 added to it. The perturbed system is then solved and the difference of the solution of the perturbed system and the original system is found and analyzed.

According to the course webpage, this error can be calculated in the following form:

$$E = \sqrt{(\Delta x_1)^2 + (\Delta x_2)^2 + (\Delta x_3)^2 + \dots + (\Delta x_{10})^2}.$$

where delta is the difference between the two solution vectors

In order to minimize errors, this process is iterated 100 times, and the mean and standard deviation for the set of errors is calculated. This same process is then repeated for a range of values from [-1.0,1.0] to [-0.00001,0.00001] and the resultant mean and standard deviation are given in the graphs as follows in the data and analysis section.

## Data and Analysis

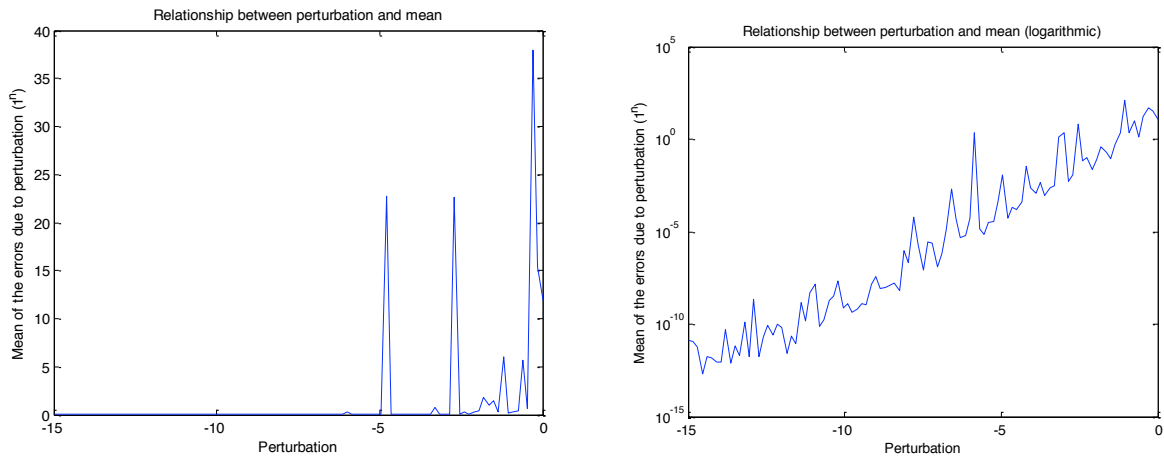
For one run of the 10x10 matrix, error was 2.0638, this changed with each run. Thus it is only natural to plot and analyze this relationship, as shown in the data collected below

The mean of the system should approach 0 as the value of perturbation decreases because it implies that there is a lower perturbation in the solution, so the errors are increasingly smaller. This is portrayed in both of the graphs:

Note that this process is repeated for partial pivoting also. Note that while theoretically this also approaches a similar result, however the mean should seem to have a smaller error in the partial pivoting version, this data shows the magnitude of the error is still in 10<sup>-12</sup> for both cases. So partial pivoting does not result in a significantly lower error.

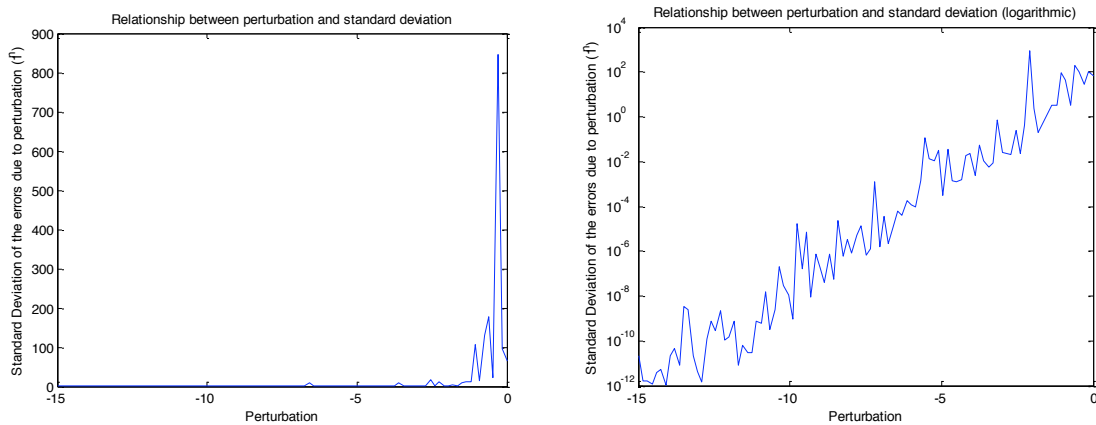


## Mean



As can be seen in this graph, as the perturbation decreases so does the mean of the errors, the mean of the errors approaches 0 suggesting that the error due to perturbation decreases. Note that the logarithmic scale shows a somewhat linear decrease (because the scale is logarithmic). Note that the spikes occur due to anomalies in the data as the matrices are generated at random, despite this the data still follows a sure path.

## Standard Deviation

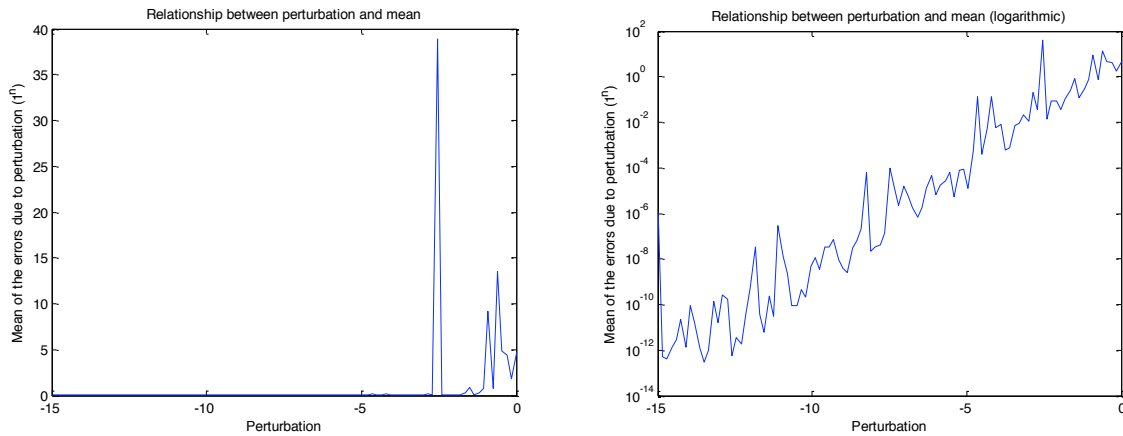


Similarly, this graph shows that the standard deviation of the errors also falls as the perturbation falls. The standard deviation approaches 0 because all the values of error are more condensed and approach zero. Note that the logarithmic scale shows a somewhat linear decrease (because the scale is logarithmic).

Note again that any spikes in data are partially caused due to the random generation of matrices.

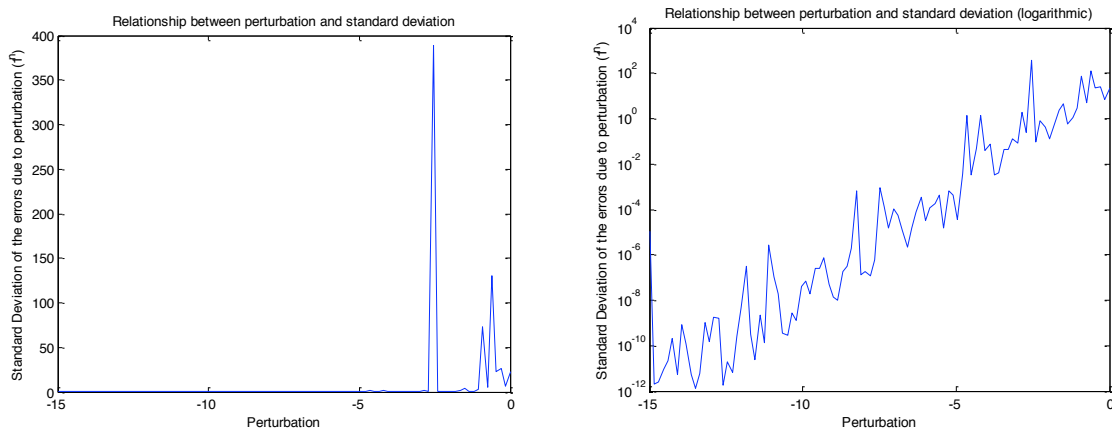
This process is repeated for the partial pivoting systems

## Mean (partial pivoting)



As can be seen in this graph, as the perturbation decreases so does the mean of the errors, the mean of the errors approaches 0 suggesting that the error due to perturbation decreases. Note that the logarithmic scale shows a somewhat linear decrease (because the scale is logarithmic). Note that the spikes occur due to anomalies in the data as the matrices are generated at random, despite this the data still follows a sure path.

## Standard Deviation (partial pivoting)



Similarly, this graph shows that the standard deviation of the errors also falls as the perturbation falls. The standard deviation approaches 0 because all the values of error are more condensed and approach zero. Note that the logarithmic scale shows a somewhat linear decrease (because the scale is logarithmic). Note again that any spikes in data are partially caused due to the random generation of matrices.

## **Summary**

As can be seen in the graphs above, both the standard deviation and mean decreases exponentially as the perturbation decreases. This is the expected result as explained above.

# Ill conditioned systems

---

A random ill conditioned system is created (as described in the method) and then tested using the perturbation experiment. The mean and standard deviation are compared to the perturbation results as found above.

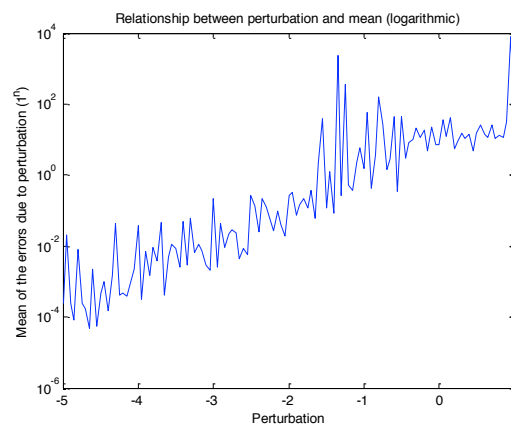
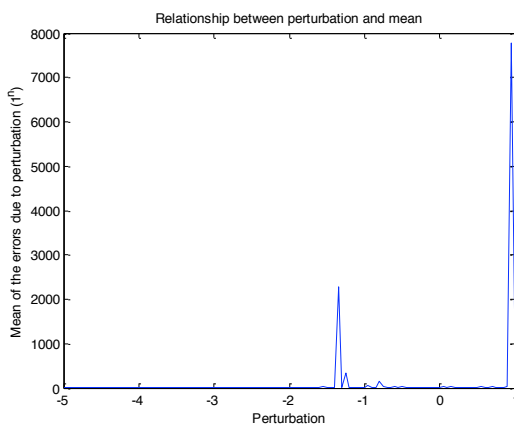
## Method

The first step of this process is to create an ill conditioned system. An ill conditioned system is created by first creating a 10x10 matrix as before. Then 3 rows are randomly selected of the 10 rows and one of them is exchanged with the average of the other two rows.

Using these ill conditioned systems, the perturbation experiment is repeated. And the results are compared. See the zipped folder for the code based on this algorithm, and the section below on data and analysis for the given system.

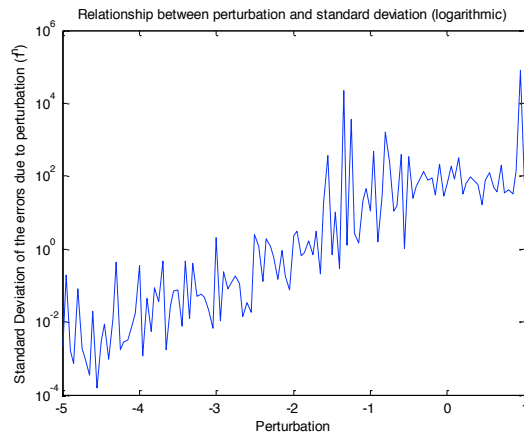
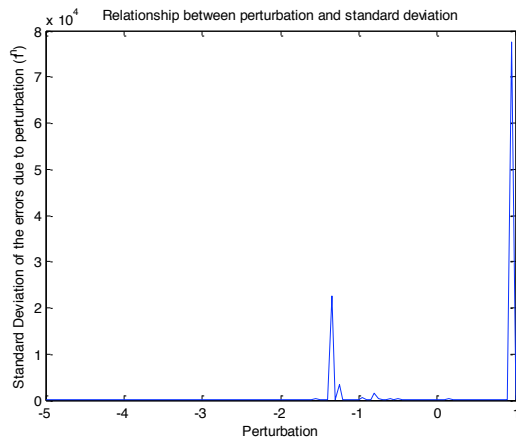
## Data and Analysis

### Mean



Note from the data above that the mean of the system also portrays an exponential relationship with the perturbation, and the error is close the that as was found for the regular system.

## Standard Deviation



Similarly, while the standard deviation of the system also portrays an exponential relationship, the error due to perturbation is much larger in the ill conditioned system. This is different from the regular systems because here the error is in the magnitude of  $10^{-2}$  whereas before it was  $10^{-4}$ . This shows that the ill system has caused the system to be harder to solve.

### **Summary**

Although not seemingly significant, the standard deviation of the ill conditioned system has a larger error, thus suggesting that the ill conditioned system is harder to solve because of the initial mix up.

The next spread explores the accuracy of the system as the size of the matrix increases.

# Accuracy of Results (Extra Credit)

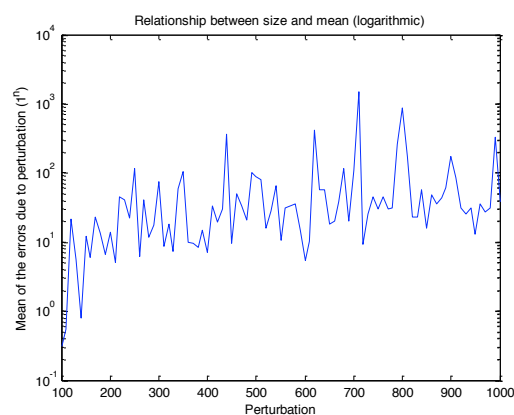
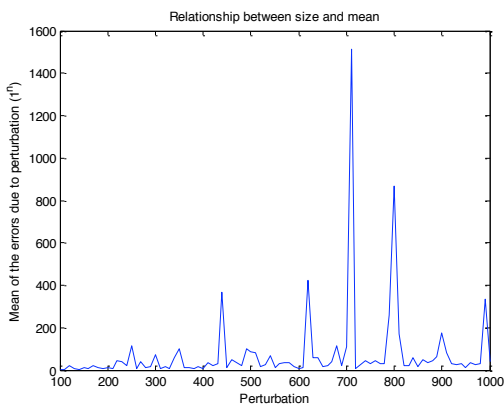
The main aim of this project is to solve a system of linear equations. But other than the speed of the calculations, another important factor is the accuracy of the data results. The following spread examines the accuracy of the mean and standard deviation of the errors due to results with the size of the matrix.

## Method

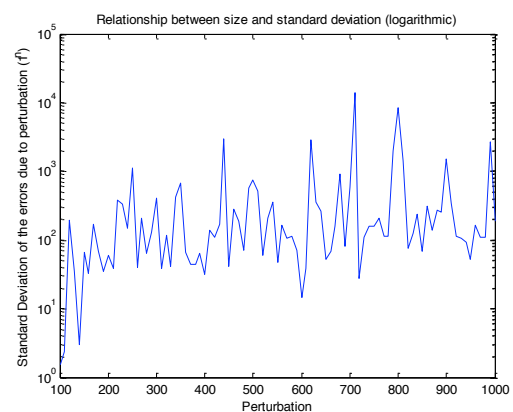
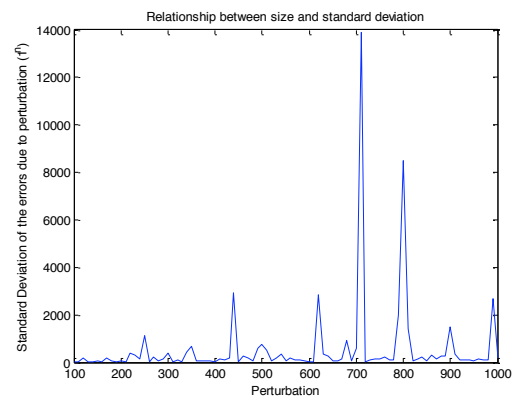
In the correctness test above, theoretically it was shown that as the matrix size increases, the error decreases, thus it can be expected that the mean error also increases as the size of the matrix increases. This is because there are more data points and more places that the round off errors can create errors in the function., In order to test the validity of this hypothesis, the following data charts are conjured comparing accuracy with size of matrix.

## Data and Analysis

### Mean



### Standard Deviation



## Summary

There is definitely some small amount of increase in error for both mean and standard deviation with an increase in the size of the matrix. This suggests that while not seemingly significant, they may cause the results of the data to be effected as the matrix size increases.

# Brief Conclusion

---

As the project progressed, several Gaussian elimination techniques were analyzed, both including and not including partial pivoting. The correctness, timing and distribution tests analyzed this form of solving problems. The perturbation test helped apply this solving of problems to a changing data and compare the results.

## **Application**

One application of this program is in statistics or math when simply solving a system of linear equations. The Gaussian Elimination techniques are used to solve systems of linear equations and find results.

Another application might be the perturbation experiment's calculation of error in mean and standard deviation. This can be used in experiments to compare a set of known literature values to experimentally determined data with uncertainty. The amount of perturbation correlates to the amount of uncertainty in the experimental data, and the mean and std are thus calculated.

## **Further Research**

While this research encompasses several different tests and even some ways to make the function faster, some further research might explore how to make the results more accurate and fast, by condensing the algorithm to fewer steps. The lower the amount of steps the higher the speed and accuracy of the system.

## Horror Matrices for Direct Methods

by Tim Davis

`Twas fill-in, and the slow transpose  
Did dir'ect factor in archive  
All tipsy were the banded 'ones  
And memory paths outgave.

"Beware the 'OrrorMat, my son!  
The cols that peak, the try's that catch!  
Beware the Netlib blurb, and shun  
the fractal banded mats!"

He took his direct slash in hand:  
Long time the max cond foe he sought ---  
So rested he by the elim' tree  
And stood while(1) in thought.

And as in out-core work he stood,  
The 'OrrorMat, with speyes of NaN,  
Came loading through the world-wide web  
And core dumped as it ran!

For one! to n! and 'til the end  
The direct slash went snicker-snack!  
He left it tril, and with diag  
He went condesting back.

"And hast thou slain the 'OrrorMat?  
Gzip to tars, my direct code!  
O fact'rize day! tril(U)! tril(A)!"  
He posted in his ode.

`Twas fill-in, and the slow transpose  
Did dir'ect factor in archive  
All tipsy were the banded 'ones  
And memory paths outgave.

Courtesy of <http://www.cise.ufl.edu/~davis/>