Jeremy Thomas

Assignment 2 Report

For this assignment I had to write four programs. I implemented Gaus-Jordan elimination and used it to find the inverse of a matrix. I also implemented Gaussian elimination and used it to find the determinate of a matrix.

In the Gaus-Jordan elimination problem I made a function that took an augmented matrix and used it to find the answer to x, y, and z. The function iterated over a column value and went through different steps of Gaus-Jordan elimination while iterating over the rows. When starting in the first column it would iterates over the rows for each value in that column to find the magnitude, and if a new magnitude were found it would swap the pivot row with the magnitude row. Once swapping was complete it would enter a similar for loop that would carry out division on the pivot row. After division does subtraction on all rows but the pivot row. This was done by multiplying the current value in the column by a set variable in the pivot row. This value was used to subtract and replace the value in the matrix. After subtraction is carried out, it increments the col by one and continues the process. When the algorithm is finished with the matrix it pulls the last index in each row and assigns them to values x, y, and z.

Inversion with Gaus-Jordan was the same process but done with an identity matrix augmented with a identity matrix. When the algorithm is finished with the matrix it pulls the last three values in reach row to display the inverted matrix.

Gaussian elimination was very similar to the Gaus-Jordan algorithm up to a certain point. The algorithm started by iterating over a col value. The swapping was done much like Gaus-Jordan, the pivot row was set to the same value as the column to make sure it was swapping on or below the diagonal. After swapping there was no division. The subtraction process differed from Gaus-Jordan in which it did not iterate over all rows and look to see if the row was the pivot row. The subtraction loop always iterated over rows based on a value of column value + 1. After subtraction, the col value would increase. After the matrix was done, values were pulled from the resulting matrix to do backwards substitution. The formula used were: "z = u / r", "y = (t - (q * z)) / p", and "x = (s - ((n * y) + (v * z))) / m".

For finding the determinate using Gaussian elimination a similar process was done, but I added a count to keep track of how many times the algorithm swapped rows. I pulled the necessary values from the diagonals of the matrix and used this formula to solve: "((-1) ** swap_count) * r * p * m".

I had issues with figuring out a way to iterate by column when my data is always group by rows. It was trial by error until I figured it out. Writing out the process on paper helped me figure out a way to do it.