Nick Thompson

Dr. Morago

CSC-340-001

26 September 2019

Programming Assignment 1 – Gaus-Jordan Elimination and Variations

- Summary of the assignment goals:

  I wrote four programs that implement each of the algorithms. I followed the algorithms step-by-step exactly as they are covered in the algorithms you provided. I included comments throughout my code highlighting where I have implemented each step of each algorithm. I only used the approved list methods and numpy.dot() to verify that my inverse matrix was correct (you said this was okay).

- A written, non-mathematical description of what each algorithm does and how
  - Gaus-Jordan Elimination
    - Create a matrix consisting of the given system of equations. Then look at each column of the matrix and find the highest number in it (ignoring negative signs). When you look for this number, only look in the lower left diagonal of the matrix (including the diagonal). If at any point that diagonal has a zero in it, exit the program. If the row that you found the number in appears later in the matrix, swap it with the current row to bring the largest number into the current row. Divide the current row by the number you found earlier. This will eventually make it so all the diagonal numbers are 1. For every other row, subtract it by the value you get from multiplying the value of the index at that row and that column by the value of the index at the current row and the current column. This will eventually make it so every number not on a diagonal is zero. After this process is complete, you will be left with the identity matrix and the corresponding values in the right most column of the matrix.
  - Matrix Inverse using Gaus-Jordan Elimination
    - Create a matrix consisting of the given matrix and the identity matrix of the same dimensions. Then look at each column of the matrix and find the highest number in it (ignoring negative signs). When you look for this number, only look in the lower left diagonal of the matrix (including the diagonal). If at any point that diagonal has a zero in it, exit the program. If the row that you found the number in appears later in the matrix, swap it with the current row to bring the largest number into the current row. Divide the current row by the number you found earlier. This will eventually make it so all the diagonal numbers are 1. For every other row, subtract it by the value you get from multiplying the value of the index at

that row and that column by the value of the index at the current row and the current column. This will eventually make it so every number not on a diagonal is zero. After this process is complete you will be left with the identity matrix on the left half and the inverse of the original matrix on the right half. I recommend you multiply the original matrix by the inverse matrix to be sure the returned value is the identity matrix.

- Gaussian Elimination
  - Create a matrix consisting of the given system of equations. Then look at each column of the given matrix and find the highest number in it (ignoring negative signs). When you look for this number, only look in the lower left diagonal of the matrix (including the diagonal). If at any point that diagonal has a zero in it, exit the program. If the row that you found the number in appears later in the matrix, swap it with the current row to bring the largest number into the current row. For all rows that appear later in the matrix, subtract it by the value you get from dividing the value at the index of the current row and current column by the value that appears on the diagonal of the current row times the value that appears on the current row and current column, this process will eventually give you a matrix where the lower left part of it (excluding the diagonal) is all zeros. After this process is complete, you have to work back though the matrix starting at the bottom to solve the system of equations by substituting the values into the system of equations.

- Matrix Determinant using Gaussian Elimination
  - Look at each column of the given matrix and find the highest number in it (ignoring negative signs). When you look for this number, only look in the lower left diagonal of the matrix (including the diagonal). If at any point that diagonal has a zero in it, exit the program. If the row that you found the number in appears later in the matrix, swap it with the current row to bring the largest number into the current row. Throughout this process, keep track of how many times a row is swapped. For all rows that appear later in the matrix, subtract it by the value you get from dividing the value at the index of the current row and current column by the value that appears on the diagonal of the current row times the value that appears on the current row and current column, this process will give you a matrix where the lower left part of it (excluding the diagonal) is all zeros. Take (-1) raised to the number of times a row was swapped and multiply it by all the diagonal values in the matrix. If the determinate is zero, then the matrix is singular.

- Discussion of any issues encountered:

I encountered many issues. The main issue I encountered was figuring out how to precisely translate the algorithm's pseudo code into Python. I especially had much trouble correctly indexing my matrices. Aside from that, I struggled with the subtraction part of Gaus-Jordan Elimination because I did not set a variable before the loop which meant that what I was subtracting by was changing each iteration. I also struggled because I naïvely made a variable to keep track of if the magnitude was negative; however, I did not properly change it. Furthermore, I made many mistakes when trying to figure out how to implement the substitution part of Gaussian Elimination. Finally, I encountered many issues regarding machine precision and floats.