Robert Matthew Keezer

Calculus 3

Programming Project

3/31/2015

**Intro**

For this part of the project, I implemented an encoder method that creates two data streams, y1 and y2, from the input data stream x. It then multiplexes the two streams into one called y and outputs it. I then implemented both Jacobi and Gauss Seidel iteration methods in order to decode a given input y and output the original x using the original patterns.

**Part Two**

For the comparison of the two methods, Jacobi and Gauss Seidel, I tested them by using two constant 4x4 and 4x1 matrices with varying tolerances. Gauss Seidel is much faster than Jacobi because in Gauss Seidel after we calculate the value for x1, where x1 is in the first element of the guess vector, we immediately put the value back into the equation for the next element. Essentially we are updating the guess vector values immediately as we are calculating the other elements in the guess vector. This means we converge to the answer much quicker. This also agrees with the results for the test case given in the project outline. For x = (1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0) which encodes to a y, when putting the y into our Jacobi and Gauss Seidel methods, the Jacobi finds the answer in 3 iterations which the Gauss finds the answer in 2 when tolerance is 1e-8. As for this binary stream case, the length of n is unimportant because it converges so quickly, however if this were not a binary stream, the length of n would allow the Gauss Seidel method to find the answer quicker because it has more elements to update its guess per iteration.