MATH 447 Project 1 (Summary)

(1) For problem number one, because our group was composed of three members we were asked to solve for the receiver position (x,y,z) by solving for x, y and z and then inputting each solution into the equation for the first satellite on p. 238. By assigning new variables and constants towards the goal of finding each solution, a quadratic formula was found. The task was much more difficult than we had envisioned it to be but after some laborious work done outside of MATLAB, Kevin Talbott was able to find the GPS receiver position after inputting each of the satellite positions into the newly formed quadratic equation.

A .pdf file is attached that displays the step-by-step process towards finding the quadratic equation.

(2) Not Required

(3) For problem number three, we were asked to solve the system of equations for the four satellites using the Multivariable Newton’s Method in MATLAB. We began this problem by first taking the partial derivatives with respect to x, y and z of each of the four equations. Our resulting Jacobian Matrix was then a 4x4 matrix. By using the symbolic function in MATLAB we were able to make an m-file that would carry out the Multivariable Newton’s Method after having input the given initial values. To make sure the method would converge to the values we found in problem number one a for loop was added. After executing our m-file we saw that the system converged to the x, y, z, and d values after a few iterations.

(4) For problem number four, we used a variation of the m-file we created for problem number three except we’ve redefined the satellite positions in relation to spherical coordinates rho, phi and theta for the four satellites. For rho we chose to use the rho that was the distance from the Earth to the GPS satellite, 26570 km. For each of the satellite positions (for satellites 1 through 4) we used “unbunched” coordinates and used the equations on p. 239 with phi being between 0 and 90 degrees, restricted to the upper hemisphere, and theta being between 0 and 360 degrees.

After inputting our redefined satellite positions, we then defined equations in MATLAB for the satellite ranges of the four satellites. Using the four satellite ranges we were able to redefine the travel times for satellites 1 through 4.

Once we found the new travel times for the satellites we randomly added and subtracted the travel times by 10^-8 seconds to compensate for the time delay that each satellites’ atomic clock would have. Using the change in the transmission time and multiplying the infinity norm by **c** we were able to deduce our Backward Error. By calculating the change in coordinates found after each iteration we were able to deduce our Forward Error.

Attached is the MATLAB file

(5) For problem number five, the spherical coordinates, phi and theta were changed so that they were within 5% of one another. By changing the spherical coordinates in this manner we brought the satellites closer together or in other words making them “bunched,” and by doing this we will have an even greater position error.

After running our MATLAB program, we noticed that the EMF increased greatly compared to our results from problem number four.

Attached is the MATLAB file

(6) For problem number six, we disregarded our “bunched” satellite configuration in favor of the “unbunched” spherical coordinates from problem number four. Now we’ve added four more satellites to our configuration in hope of seeing if these new additions will give us a smaller condition number, smaller EMF and a smaller GPS error.

We began by defining four more satellites and inputting them into our m-file thereby redefining matrices that were added earlier. Our initial matrix labeled x0, had entries defined as x=0, y=0, z=0, and d=.0001. Using the Gauss-Newton Method and going through many iterations we arrived at the maximum GPS error and the condition number.

Attached is the MATLAB file