ECE174 Computer Assignment #2: Global Positioning System (GPS) Algorithm

Christopher Yin
A11892138

Problem Statement

The pseudorange - an approximation of the distance – between an object and several earth-orbiting satellites can be used to accurately locate the object in space. Pseudorange measurements are obtained by sending signals from satellites at known coordinates to a receiver on the surface of the earth. The true range could theoretically be calculated from the speed of light and the time required for the signal to pass from the satellite to the receiver. However, the exact signal speed deviates from the speed of light in a vacuum unpredictably due to atmospheric conditions, and the travel time cannot be accurately determined due to errors in the receiver clock. Therefore, the pseudorange, rather than the true range, is calculated for each satellite. By establishing a nonlinear least-squares problem to model this situation, the receiver location can still be accurately determined. In this report, the Steepest Descent and Gauss-Newton algorithms will be developed and deployed to determine the true receiver location from simulated pseudorange data.

Technical Procedure

All locations will be specified in a geostationary reference coordinate system originating at the earth's center. Distances will be given in units of Earth Radii (ER), with $1\ ER = 6370\ km$.

1. Linearization

To begin with, a linearized model of the pseudorange must be developed. Let the position of the lth satellite be denoted by coordinate vector $S_l = (x_l, y_l, z_l)^T$, with the position of the receiver indicated by $S = (x, y, z)^T$. The true range, $R_l(S)$ is defined as below with $\Delta S = S - S_l$:

$$R_l(S) = \|S - S_l\| = \|\Delta S_l\| = (\Delta S_l^T \Delta S_l)^{\frac{1}{2}} \qquad (1)$$

The gradient of $R_l(S)$ yields the unit vector pointing from the satellite location to the receiver, which is the direction of the steepest increase of $R_l(S)$. This can be derived as follows:

$$\nabla_S R_l(S) = \left(\frac{\partial R_l(S)}{\partial S}\right)^T$$

$$\frac{\partial R_l(S)}{\partial S} = \frac{\partial (\Delta S_l^T \Delta S_l)^{\frac{1}{2}}}{\partial S} = \frac{\partial (\Delta S_l^T \Delta S_l)^{\frac{1}{2}}}{\partial \Delta S} \cdot \frac{\partial \Delta S}{\partial S}$$

$$= \frac{1}{2}(\Delta S_l^T \Delta S_l)^{-\frac{1}{2}} \cdot \frac{\partial(\Delta S_l^T \Delta S_l)}{\partial \Delta S} \cdot \frac{\partial(S - S_l)}{\partial S}$$

$$= \frac{1}{2R_l(S)} \cdot 2\Delta S_l^T \cdot I$$

$$\frac{\Delta S_l^T}{R_l(S)} = r_l^T(S)$$

$$\nabla_S R_l(S) = \left(\frac{\partial R_l(S)}{\partial S}\right)^T = \frac{\Delta S_l}{R_l(S)} = r_l(S) \rightarrow \frac{\partial R_l(S)}{\partial S} = r_l^T(S) \qquad (2)$$

Where $r_l(S)$ is the aforementioned unit vector.

The pseudorange deviates from the true range because of two factors- inaccuracy in the receiver clock, and random noise introduced by atmospheric conditions modulating the speed of light signals. The former is encapsulated by adding a constant, systematic clock bias b to the true range. The latter is accounted for by adding the random noise term $\nu_l \sim N(0, \sigma^2)$, which is i.i.d for every satellite. The pseudorange $y[k]$ for measurement k is thus given by:

$$y[k] = h(X) + \nu[k], k = 1, \dots, m \qquad (3)$$

Where

$$X = \binom{S}{b} \in \mathbb{R}^4, h(X) = R(S) + eb \in \mathbb{R}^4$$

$$y[k] = \begin{pmatrix} y_1[k] \\ y_2[k] \\ y_3[k] \\ y_4[k] \end{pmatrix}, \quad R(S) = \begin{pmatrix} R_1(S) \\ R_2(S) \\ R_3(S) \\ R_4(S) \end{pmatrix}, \quad e = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \nu[k] = \begin{pmatrix} \nu_1[k] \\ \nu_2[k] \\ \nu_3[k] \\ \nu_4[k] \end{pmatrix}$$

Assuming the satellite locates S_l are known, this yields four equations for four unknowns- the three coordinates of the receiver location S, and the clock bias error b.

To linearize the pseudorange, the nonlinear function $h(X)$ can be linearized using a multivariate Taylor series expansion with $\Delta X = X - X_0$:

$$h(X) = h(X_0 + \Delta X) = h(X_0) + \frac{\partial h(X_0)}{\partial X} \Delta X \qquad (4)$$

The Jacobian matrix is shown to be:

$$H(x) = \frac{\partial h(X_0)}{\partial X} = \left(\frac{\partial h(X_0)}{\partial S} \; \frac{\partial h(X_0)}{\partial b}\right) = \begin{pmatrix} \frac{\partial R_1(S_0)}{\partial S} & 1 \\ \frac{\partial R_2(S_0)}{\partial S} & 1 \\ \frac{\partial R_3(S_0)}{\partial S} & 1 \\ \frac{\partial R_4(S_0)}{\partial S} & 1 \end{pmatrix} = \begin{pmatrix} r_1^T(S_0) & 1 \\ r_2^T(S_0) & 1 \\ r_3^T(S_0) & 1 \\ r_4^T(S_0) & 1 \end{pmatrix} \tag{5}$$

Therefore, the linearization of the pseudorange is given by:

$$y \approx h(X_0) + H(X_0)\Delta X_0 + \nu \tag{6}$$

As the true range increases, the linearized approximation becomes more accurate. This is because linearization with a Taylor Series approximation relies on the ability to ignore higher order terms. The second order term in the Taylor Series approximation for the pseudorange is reported below:

$$\frac{1}{2}\Delta X \frac{\partial^2}{\partial X^2} h(X_0)\Delta X = \frac{1}{2}\Delta X \frac{\partial}{\partial X} \nabla_{\mathrm{X}} h(X_0)\Delta X \tag{7}$$

For simplicity's sake, this can be treated as a linearization of the range function $R_l(S)$, as this is what induces nonlinearity in the $h(X)$ function. Therefore, consider this term analogous to the following:

$$\frac{1}{2}\Delta S \frac{\partial^2}{\partial S^2} R_l(S_0)\Delta S = \frac{1}{2}\Delta S \frac{\partial}{\partial S} \nabla_{\mathrm{S}} R_l(S_0)\Delta S \tag{8}$$

The term $\frac{\partial}{\partial S} \nabla_{\mathrm{S}} R_l(S_0)$ can be simplified to:

$$\frac{\partial}{\partial S} r_l(S_0) = \frac{1}{R_l(S_0)} \left(I - r_l(S_0) r_l(S_0)^T\right)$$

The second order term in the Taylor Series expansion for $h(x)$ thus becomes of the order of:

$$\frac{1}{2}\Delta S \frac{1}{R_l(S_0)} \left(I - r_l(S_0) r_l(S_0)^T\right)\Delta S \tag{9}$$

Here, the term ΔS represents a small magnitude displacement of S from the original position of S_0. $R_l(S_0)$, on the other hand, is a comparatively enormous magnitude measurement of the distance from the satellite to the receiver, which is expected to be within the range of $20{,}000\ km$. Therefore, with ΔS in the numerator and $R_l(S_0)$ in the denominator, this term has very small magnitude, especially when compared with the first order term in this Taylor expansion. By increasing $R_l(S_0)$, the magnitude of this second order term decreases further, and it becomes more justified to ignore these and any higher order terms to construct the linearization of $h(X)$.

2. Algorithm Development

Gauss-Newton

Using the linearization of the pseudorange obtained above, define an error function:

$$e(X) = y - h(X) \approx y - h(\hat{X}_k) - H(\hat{X}_k)\Delta X_k = \Delta y_k - H(\hat{X}_k)\Delta X_k \qquad (10)$$

$$\text{With } \Delta y_k = y - h(\hat{X}_k)$$

For the kth measurement of the pseudorange, assuming zero noise measurements (ν term can be ignored) and an identity norm weighting matrix in both the domain and codomain, define a loss function as follows:

$$l_k(X) = \frac{1}{2}\|e(X_k)\|^2 = \frac{1}{2}\|y[k] - h[k]\|^2 = \frac{1}{2}\left\|\Delta y_k - H(\hat{X}_k)\Delta X_k\right\|^2 \qquad (11)$$

Minimizing this loss function provides a least-squares solution to the linearized inverse problem:

$$\Delta y_k = H(\hat{X}_k)\Delta X_k \qquad (12)$$

Because $h(X)$ is clearly one-to-one, $H(X)$ must also be one-to-one, allowing a pseudoinverse to be formed:

$$\Delta X_k = \left(H^*(\hat{X}_k)H(\hat{X}_k)\right)^{-1} H^*(\hat{X}_k)\,\Delta y_k$$

$$H^*(\hat{X}_k) = H^T(\hat{X}_k)\mathrm{W} = H^T(\hat{X}_k)$$

$$\Delta X_k = \left(H^T(\hat{X}_k)H(\hat{X}_k)\right)^{-1} H^T(\hat{X}_k)\,\left(y - h(\hat{X}_k)\right) \qquad (13)$$

For the loss function given above, the gradient is derived as:

$$\nabla_X l(X) = -\left(\frac{\partial h(X)}{\partial(X)}\right)^T \left(y - h(X)\right) = H^T(\hat{X}_k)\left(y - h(X)\right) \qquad (14)$$

Substituting this into the equation above yields:

$$\Delta X_k = \left(H^T(\hat{X}_k)H(\hat{X}_k)\right)^{-1} \nabla_X l(\hat{X}_k) \qquad (15)$$

This correction term, multiplied by a positive, real-valued step-size parameter α_k, can be iteratively calculated and added to the current estimate to yield a new estimate:

$$\hat{X}_{j+1} = \hat{X}_j + \Delta X_k = \hat{X}_j + \alpha_k\left(H^T(\hat{X}_j)H(\hat{X}_j)\right)^{-1} H^T(\hat{X}_j)\left(y - h(\hat{X}_j)\right) \qquad (16)$$

Where at the optimal solution the gradient of the loss function will be 0 and correction term will also be 0. The functions $h(X)$ and $H(X)$ are defined in (3) and (5), respectively. Because this method is known to tolerate larger step sizes, a constant $\alpha = 1$ will be used. Equation (16) will be iterated until the change in estimate is less than $1E11$ or $1E6$ iterations have occurred.

Steepest Descent

More generally, for the nonlinear loss function defined above,

$$dl = \frac{\partial l(X)}{\partial X} dx \qquad (17)$$

Linearizing about current estimate $\hat{X}_k$ allows the following to be stated:

$$\Delta l(\Delta X_k) = l(X) - l(\hat{X}_k) = l(\hat{X}_k + \Delta X_k) - l(\hat{X}_k) \approx \frac{\partial l(\hat{X}_k)}{\partial X} \Delta X_k = \left(\nabla_X l(\hat{X}_k)\right)^T \Delta X_k \qquad (18)$$

The correction term can be calculated as:

$$\Delta X_k = -\alpha_k \nabla_X l(\hat{X}_k) = -\alpha_k H^T(\hat{X}_k)(y - h(X)) \qquad (19)$$

Because the gradient gives the direction of steepest ascent, the negative of this correction term is added to the current estimate for gradient descent. This iteration is encapsulated by the following equation:

$$\hat{X}_{j+1} = \hat{X}_j + \alpha_j H^T(\hat{X}_j)\left(y - h(\hat{X}_j)\right) \qquad (20)$$

This method is more sensitive to step size, so a range of step sizes will be used: $\alpha = [0.001\ 0.01\ 0.1]$. The same termination criteria as for the Gauss-Newton method will be employed here.

3. Simulation

The true receiver position S and satellite positions S_l for $l = 1,2,3,4$ will be designated (in ER):

$$S = (1.000000000, 0.000000000, 0.000000000)^T$$
$$S_1 = (3.585200000, 2.070000000, 0.000000000)^T$$
$$S_2 = (2.927400000, 2.927400000, 0.000000000)^T$$
$$S_3 = (2.661200000, 0.000000000, 3.171200000)^T$$
$$S_4 = (1.415900000, 0.000000000, 3.890400000)^T$$

The true clock bias error b will be taken as:

$$b = 2.354788068 \times 10^{-3}\ ER$$

The following initial estimates will be used:

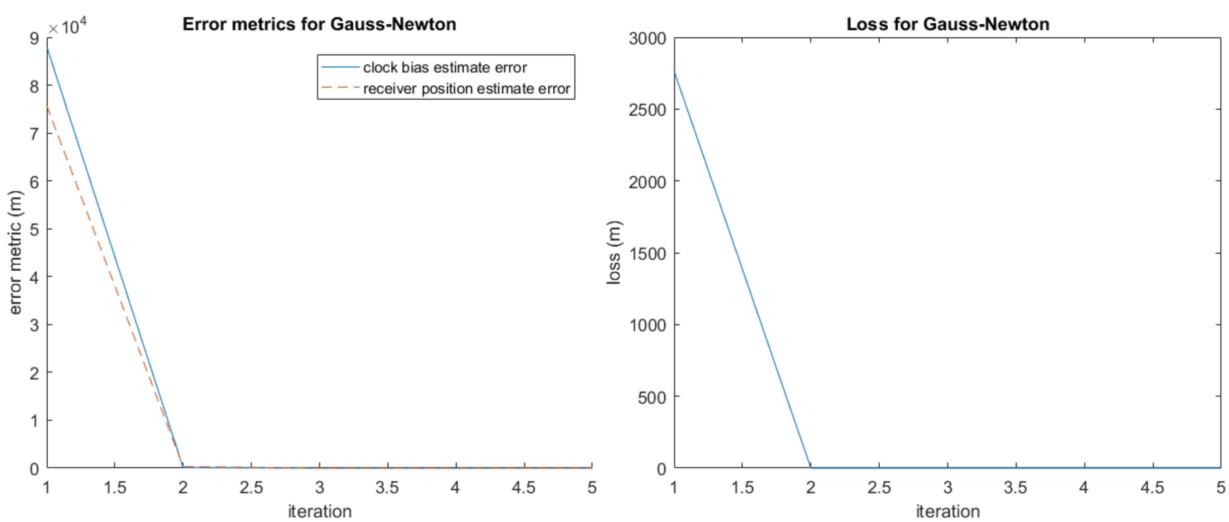
$$\hat{S}(0) = (0.93310, 0.25000, 0.258819)^T\ ER$$
$$\hat{b}(0) = 0\ ER$$

The pseudorange values y_l are generated from Equation (3) with $v = 0$. This assumes only one measurement $(m = 1)$ per satellite. The Steepest Descent and Gauss-Newton Algorithms will be applied to this data using the step sizes and termination conditions previously specified to yield estimates for S and b.

Results

Gauss-Newton

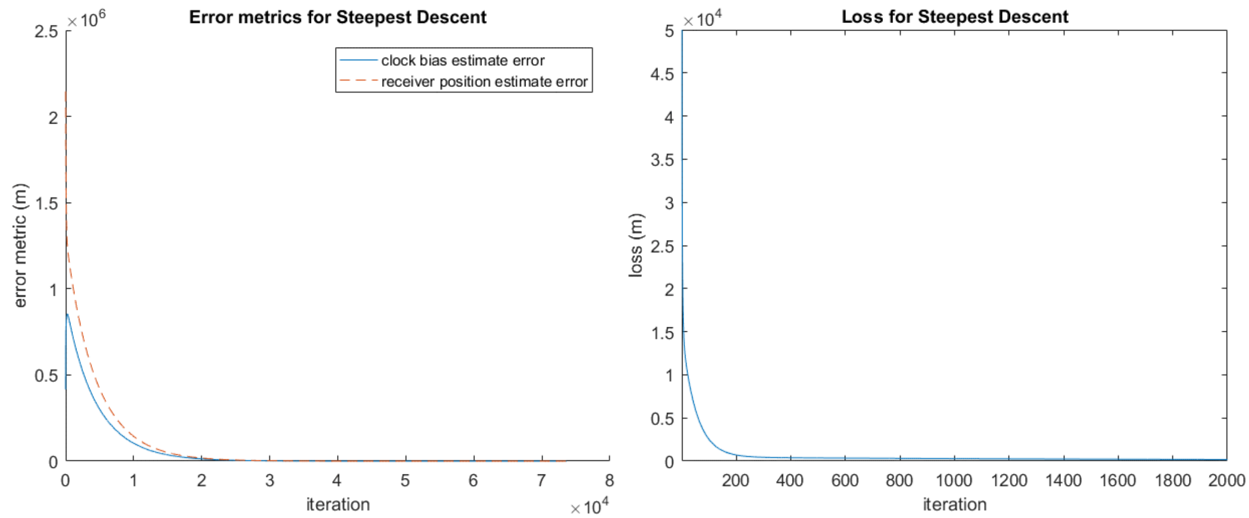
The Gauss-Newton method successfully located the receiver within __ iterations. The following graphs display the loss, receiver position estimate error ($\|\hat{S}(k) - S\|$), and clock bias estimate error ($|\hat{b}(k) - b|$) in meters vs the iteration number k.



Thus, the termination condition is met by the 5th iteration, and the error for both clock bias and receiver position are easily reduced to effectively zero with this rapid convergence rate.

Steepest Descent

A step size of 0.1 was found to be sufficient to locate the receiver. However, the Steepest Descent method achieves convergence at a far slower rate than the Gauss-Newton method, and the receiver location was obtained only after __ iterations. The following graphs display the loss, receiver position error, and clock bias estimate error in meters vs the iteration number.



Thus, the termination condition is met after 73,602 iterations. The loss graph is truncated along the x-axis to better show the initial drop-off of the loss function. This convergence rate is far slower than that of Gauss-Newton, although the error for both clock bias and receiver position are still reduced to effectively zero.

Conclusions

In this report two methods for locating an object using GPS pseudorange data from four satellites are derived and demonstrated on simulated data. While both the Steepest Descent and Gauss-Newton methods were able to successfully locate the receiver once appropriate step sizes were chosen, the Gauss-Newton accomplished this in roughly 10^4 fewer iterations than the Steepest Descent. However, the Gauss-Newton method does require the calculation of a matrix inverse, a computationally intensive task. By comparison, the Steepest Descent method entails a much lighter computational burden. Therefore, the application must be considered when choosing which algorithm to implement. For example, in the case of convolutional neural nets (CNNs), the sheer number of parameters that are trained may indicate that the computational cost of matrix inversion is prohibitively expensive, and a simple Steepest Descent Algorithm should be used.

Additionally, the simulated data used in this report assumed zero noise. In reality the noise introduced by the effect of atmospheric conditions on the speed of light will need to be taken into account.