GPS Positioning: A Mathematical Perspective

Nathan Lunceford

Table of contents

0.1 Overview

The Global Positioning System (GPS) is a satellite-based navigation system consisting of 24 satellites equipped with atomic clocks, orbiting the Earth at an altitude of 20,200 km. By measuring the travel time of signals transmitted from satellites to a receiver, GPS calculates the receiver's position in three-dimensional space. Each signal defines a sphere with a radius equal to the distance traveled by the signal, and the receiver's position is at the intersection of these spheres.

To determine an accurate position (x, y, z) and synchronize the receiver's clock, at least four satellites are required. The system of nonlinear equations representing the sphere intersections can be formulated as:

$$r_i(x, y, z, d) = \sqrt{(x - A_i)^2 + (y - B_i)^2 + (z - C_i)^2} - c(t_i - d) = 0$$

where d is a clock correction factor for the receiver. Solving this system reveals both the receiver's coordinates and the corrected clock time.

However, the system faces challenges due to:

1. **Receiver Clock Inaccuracy**: Affordable GPS receivers lack the precision of satellite atomic clocks, necessitating the inclusion of d in the equations.
2. **Ill-Conditioning**: The system becomes sensitive to errors when satellites are clustered close together in the sky.
3. **Environmental Factors**: Signal transmission speed is affected by atmospheric interference and obstacles, introducing further inaccuracies.

To address these challenges, we will implement the following strategies:

- **Numerical Root-Finding Methods**: Techniques such as Newton-Krylov and Gauss-Newton will be employed to iteratively solve the system of equations. Special attention will be given to mitigating sensitivity to initial guesses and managing ill-conditioned scenarios.
- **Error Analysis**: The system's sensitivity to input timing errors will be evaluated using the **Error Magnification Factor (EMF)**, which quantifies how small variations in signal timing propagate and impact positional accuracy.

Using precise satellite positions and signal timing data, the system delivers accurate navigation but relies on advanced mathematical techniques to handle complex nonlinearities and maintain reliability. This report focuses on exploring and solving the GPS positioning problem through numerical and analytical approaches.

1 Numerical Root-Finding Approach to GPS Positioning

1.1 Objective

This study aims to determine the position of a GPS receiver (x, y, z) and correct its clock bias d by solving a system of nonlinear equations. These equations represent the distances between the receiver and multiple satellites using signal travel times. To achieve precise results, we use iterative root-finding methods to solve the system.

1.2 Background

In a GPS system, the position of a receiver is determined by measuring the time it takes for signals to travel from multiple satellites. By multiplying the signal travel time by the speed of light c, the distance between each satellite and the receiver can be calculated. This relationship, which involves the receiver's unknown position (x, y, z) and a clock correction d, is modeled as a system of nonlinear equations:

$$\sqrt{(x - A_i)^2 + (y - B_i)^2 + (z - C_i)^2} = c(t_i - d), \quad i = 1, 2, 3, 4$$

Where:

- (A_i, B_i, C_i) are the known coordinates of the satellites.
- t_i are the measured signal travel times.
- $c \approx 299,792.458\,\text{km/s}$ is the speed of light.
- (x, y, z) are the unknown coordinates of the receiver in the **Earth-Centered, Earth-Fixed (ECEF)** coordinate system.
- d is the receiver's clock bias, which accounts for inaccuracies in its internal clock.

1.2.1 Earth-Centered, Earth-Fixed (ECEF) Coordinates

The ECEF coordinate system is a global Cartesian coordinate system defined as follows:

- The origin is located at the Earth's center of mass.
- The x-axis points toward the intersection of the Equator and the Prime Meridian.
- The y-axis lies in the equatorial plane, pointing 90° east of the x-axis.
- The z-axis aligns with the Earth's rotational axis, pointing toward the North Pole.

In this system, both the satellite positions (A_i, B_i, C_i) and the receiver position (x, y, z) are expressed relative to the Earth's center, making it possible to calculate precise distances between the receiver and satellites.

Each equation represents a sphere centered at a satellite's position, with the radius equal to the calculated distance. The receiver's position is determined as the intersection point of these spheres. The inclusion of d corrects for the receiver's clock bias, which is less accurate than the highly precise atomic clocks on board the satellites.

1.3 Methodology

A numerical root-finding approach is employed to solve this system of nonlinear equations. Specifically, Python's `fsolve` function from the `scipy.optimize` library is utilized. This method iteratively refines estimates for (x, y, z, d) until the residuals—the differences between the measured and calculated distances—are minimized.

1.3.1 Formulation of the Problem

The system of equations is reformulated as residual functions f_i, where each function evaluates to zero when the system is satisfied:

$$f_i(x, y, z, d) = \sqrt{(x - A_i)^2 + (y - B_i)^2 + (z - C_i)^2} - c(t_i - d), \quad i = 1, 2, 3, 4$$

The root-finding process involves solving for (x, y, z, d) such that all $f_i(x, y, z, d) = 0$.

1.3.2 Numerical Solution

The computational steps are as follows:

1. **Input Data:** Known satellite positions (A_i, B_i, C_i) and signal travel times t_i are provided as inputs, along with the speed of light c.
2. **Initial Guess:** A starting point of $(x_0, y_0, z_0) = (0, 0, 6370)$ km is assumed, placing the receiver near the Earth's surface. The initial clock correction is set to $d_0 = 0$.
3. **Iterative Solver:** The `fsolve` function iteratively adjusts (x, y, z, d) to minimize the residuals f_i. The algorithm terminates when all residuals approach zero, indicating convergence to a solution.

The implementation was tested using the following dataset:

- Satellite positions:
  - $(15, 600, 7, 540, 20, 140)$
  - $(18, 760, 2, 750, 18, 610)$
  - $(17, 610, 14, 630, 13, 480)$
  - $(19, 170, 610, 18, 390)$
- Signal travel times:

$$t = [0.07074, 0.07220, 0.07690, 0.07242] \text{ (in seconds)}$$

> Accompanying Code
>
> The **accompanying code implementation** defines the residual function, applies the `fsolve` solver, and outputs the computed results.

1.4 Results

The numerical root-finding method produced the following results:

- **Receiver Position (in km):**

$$(x, y, z) = (-41.77271, -16.78919, 6370.0596)$$

- **Clock Correction (in seconds):**

$$d = -3.201566 \times 10^{-3}$$

These values were verified against expected results, confirming the accuracy of the numerical solution. The computed receiver position aligns with the Earth's surface, and the clock correction accounts for the slight bias in the receiver's timing.

1.5 Limitations

While the numerical root-finding approach proved effective, several limitations should be noted:

1. **Sensitivity to Ill-Conditioning:** The system may become ill-conditioned when satellite positions are clustered, amplifying numerical errors and reducing accuracy.
2. **Dependence on Initial Guesses:** Poor initial guesses can lead to non-convergence or convergence to an incorrect solution.
3. **Computational Cost:** Iterative methods like `fsolve` require multiple evaluations of the residuals, making them computationally expensive for real-time applications.
4. **Lack of Analytical Insights:** The numerical solution provides no explicit relationships between the variables, limiting its use for sensitivity analysis or theoretical exploration.

1.6 Motivation for an Improved Approach

The limitations of the numerical root-finding method highlight the need for a more robust and efficient solution. Specifically:

- A method that avoids iterative guesswork.
- Improved handling of ill-conditioned systems.
- A more **analytical approach** that isolates variables and reduces the problem complexity.

To address these concerns, the next section introduces a **determinant-based analytical approach**. This method linearizes the system of equations, isolates variables explicitly, and reduces the problem to solving a single quadratic equation for the clock correction d.

1.7 Conclusion

The numerical root-finding approach provides an approximate solution to the GPS equations but often struggles with sensitivity, inefficiency and a lack of robustness. These limitations highlight the need for a more reliable analytical method which will be discussed in the next section.

2 Determinant-Based Analytical Approach to GPS Positioning

2.1 Objective

The objective of this section is to solve the GPS equations **analytically** using a **determinant-based approach**, unlike numerical root-finding methods, this analytical technique avoids iterative guesswork by isolating the variables (x, y, z, d) explicitly. By systematically transforming the original nonlinear system into a more manageable linear form, this method ensures greater stability and computational efficiency which is critical for real-time or precision GPS applications.

The approach has been implemented in Python using the **SymPy** library to symbolically manipulate the equations, solve the system and isolate the unknowns step-by-step.

2.2 Problem Formulation

The GPS receiver's position (x, y, z) and clock offset d are determined from four satellite equations of the form:

$$\sqrt{(x - A_i)^2 + (y - B_i)^2 + (z - C_i)^2} = c(t_i - d), \quad i = 1, 2, 3, 4$$

Here:

- (A_i, B_i, C_i) are known satellite coordinates.
- t_i are measured signal travel times.
- $c \approx 299792.458\,\text{km/s}$ is the speed of light.
- (x, y, z) and d are the unknown receiver coordinates and clock correction.

Each equation represents a sphere with the satellite at its center. The receiver lies at the intersection of these four spheres. However, the presence of square roots and the unknown d makes the system inherently nonlinear and challenging to solve directly.

2.3 Methodology

2.3.1 Reducing the Nonlinear System to a Linear Form

To eliminate the square roots, we square each equation. After doing so, we have:

$$(x - A_i)^2 + (y - B_i)^2 + (z - C_i)^2 = c^2(t_i - d)^2$$

We then **subtract the equations for satellites** $i = 2, 3, 4$ **from the equation for** $i = 1$. This subtraction removes the $x^2 + y^2 + z^2$ terms and results in three linear equations in the four unknowns x, y, z, d:

$$xu_x^{(j)} + yu_y^{(j)} + zu_z^{(j)} + du_d^{(j)} + w^{(j)} = 0, \quad j = 1, 2, 3$$

where $u_x^{(j)}, u_y^{(j)}, u_z^{(j)}, u_d^{(j)}, w^{(j)}$ are constants determined by the satellite positions and travel times.

At this point, we have **three linear equations** with **four unknowns**:

$$xu_{x1} + yu_{y1} + zu_{z1} + du_{d1} + w_1 = 0$$

$$xu_{x2} + yu_{y2} + zu_{z2} + du_{d2} + w_2 = 0$$

$$xu_{x3} + yu_{y3} + zu_{z3} + du_{d3} + w_3 = 0$$

> Accompanying Code
>
> In the accompanying code, this system of equations is constructed symbolically using **SymPy**, with the `simplify()` function applied to ensure clarity and precision in the linearized expressions.

Why Solve for the Variables in Terms of d?

With three linear equations and four unknowns, the system is underdetermined. We cannot directly solve for all four variables at once. To proceed, we:

- Treat d as a parameter.
- Express x, y, z as linear functions of d:

$$x = f_x(d), \quad y = f_y(d), \quad z = f_z(d)$$

Once x, y, z are known in terms of d, we substitute these functions back into one of the original nonlinear equations. This will yield a **single quadratic equation in** d. Solving that quadratic equation gives us d. With d in hand, we can easily find x, y, z.

This approach reduces the complexity: instead of trying to solve four nonlinear equations simultaneously, we simplify the problem to solving one quadratic equation after isolating variables in terms of d.

Introducing the Determinant Equation

To isolate x, y, z in terms of d, we use a **determinant-based approach**. By arranging our linear equations in a matrix form and considering certain determinants, we can:

- Identify linear dependencies,
- Isolate one variable at a time,
- Avoid ambiguity and instability.

The idea is to construct a determinant from the system of equations and carefully manipulate it so that one variable (e.g., x) can be extracted in terms of y, z, d and constants. We then repeat or apply a similar reasoning for y and z.

2.3.2 Isolating x in Terms of d Using the Determinant

Step-by-Step for x:

1. **Set Up the Determinant:**

   Consider a matrix formed from the coefficient vectors $\mathbf{u_x}, \mathbf{u_y}, \mathbf{u_z}, \mathbf{u_d}, \mathbf{w}$. A key construction is:

$$\det[\mathbf{u_y} \mid \mathbf{u_z} \mid x\mathbf{u_x} + y\mathbf{u_y} + z\mathbf{u_z} + d\mathbf{u_d} + \mathbf{w}] = 0$$

This determinant equals zero because if $-\mathbf{w}$ lies in the span of the other vectors, the system is consistent. Expanding this determinant along the third column will separate terms involving x, y, z, d.

$$\det \begin{bmatrix} u_{y1} & u_{z1} & xu_{x1} + yu_{y1} + zu_{z1} + du_{d1} + w_1 \\ u_{y2} & u_{z2} & xu_{x2} + yu_{y2} + zu_{z2} + du_{d2} + w_2 \\ u_{y3} & u_{z3} & xu_{x3} + yu_{y3} + zu_{z3} + du_{d3} + w_3 \end{bmatrix} = 0$$

2. **Expand the Determinant**:
   Expanding along the third column gives:

$$\det[\dots] = (xu_{x1} + yu_{y1} + zu_{z1} + du_{d1} + w_1) \cdot \det \begin{bmatrix} u_{y2} & u_{z2} \\ u_{y3} & u_{z3} \end{bmatrix}$$

$$-(xu_{x2} + yu_{y2} + zu_{z2} + du_{d2} + w_2) \cdot \det \begin{bmatrix} u_{y1} & u_{z1} \\ u_{y3} & u_{z3} \end{bmatrix}$$

$$+(xu_{x3} + yu_{y3} + zu_{z3} + du_{d3} + w_3) \cdot \det \begin{bmatrix} u_{y1} & u_{z1} \\ u_{y2} & u_{z2} \end{bmatrix}$$

3. **Group Terms Involving x, y, z, d**:
   Collect coefficients for each variable. For example for x, the coefficient C_x is:

$$C_x = u_{x1} \cdot \det \begin{bmatrix} u_{y2} & u_{z2} \\ u_{y3} & u_{z3} \end{bmatrix} - u_{x2} \cdot \det \begin{bmatrix} u_{y1} & u_{z1} \\ u_{y3} & u_{z3} \end{bmatrix} + u_{x3} \cdot \det \begin{bmatrix} u_{y1} & u_{z1} \\ u_{y2} & u_{z2} \end{bmatrix}$$

   On expansion, you get an equation of the form:

$$C_x x + C_y y + C_z z + C_d d + T = 0$$

   where C_x, C_y, C_z, C_d, T are combinations of determinants and known constants derived from the satellite data.

4. **Isolate x:**

   To solve for x, rearrange the equation:

$$x = -\frac{C_y y + C_z z + C_d d + T}{C_x}$$

   However, at this stage, x is still expressed in terms of y, z, d. To get x purely in terms of d, you must similarly isolate y and z in terms of d.

5. **Repeat for y and z:**

   By constructing similar determinant equations and performing analogous expansions, you isolate y and z as linear functions of d:

$$y = f_y(d), \quad z = f_z(d)$$

   Once $y(d)$ and $z(d)$ are known, substitute them back into the expression for x:

$$x = f_x(d)$$

Now all three spatial variables are functions of d:

$$x = f_x(d), \quad y = f_y(d), \quad z = f_z(d)$$

Accompanying Code

The accompanying code uses SymPy's `linear_eq_to_matrix` to extract the coefficient matrix and split it into parts:

- A_{xyz}: Coefficients for x, y, z,
- A_d: Coefficients for d.

The resulting system is:

$$A\_xyz \begin{bmatrix} x \\ y \\ z \end{bmatrix} = -A_d d - \mathbf{w}$$

Solving this system for x, y, and z in terms of d is achieved using the `LUsolve` function. The symbolic solutions $x(d)$, $y(d)$, and $z(d)$ are simplified and stored.

2.3.3 Forming the Quadratic Equation in d

With $x(d), y(d), z(d)$ established, we return to an original nonlinear equation. For example:

$$\sqrt{(x - A_1)^2 + (y - B_1)^2 + (z - C_1)^2} = c(t_1 - d)$$

- Substitute $x(d), y(d), z(d)$ into the left-hand side.
- Square both sides to remove the square root.

After simplification, you obtain a **quadratic equation in d**:

$$ad^2 + bd + c = 0$$

Solving this quadratic equation using the quadratic formula:

$$d = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Choose the physically meaningful solution for d (the one placing the receiver near Earth's surface), then substitute d back into $x(d), y(d), z(d)$ to find the final coordinates (x, y, z).

Accompanying Code

1. Once $x(d)$, $y(d)$, and $z(d)$ are known, they are substituted back into one of the original nonlinear equations (e.g., the first equation). This substitution is performed symbolically in the code using `subs()`.
2. Simplifying the resulting equation produces a **quadratic equation in** d:

$$ad^2 + bd + c = 0$$

3. The coefficients of the quadratic equation are extracted using SymPy's `Poly` and `all_coeffs` functions. The quadratic formula is then applied to solve for d.
4. Among the solutions for d, the physically meaningful (real and close to zero) solution is selected. This step is automated in the code by evaluating the solutions and filtering for real roots.
5. The final values of x, y, and z are computed by substituting the selected d back into $x(d)$, $y(d)$, and $z(d)$.

The determinant-based analytical approach produces the following results:

- **Receiver Position (in km):**

$$(x, y, z) = (-41.77271, -16.78919, 6370.0596)$$

- **Clock Correction (in seconds):**

$$d = -3.201566 \times 10^{-3}$$

These results are consistent with the numerical solutions obtained in **numerical root-finding approach**, confirming the correctness of the analytical method.

2.4 Conclusion

This determinant-based analytical approach simplifies the original nonlinear GPS equations by transforming them into a linear form. It uses determinants to express x, y, z in terms of d and reduces the problem to solving a single quadratic equation for d.

Key Advantages:

- No iterative guesswork required, avoiding convergence issues.
- More stable, especially when satellites are poorly distributed.
- Provides a closed-form, analytical solution which offers deeper insight and efficiency.

By reducing the system to a linear form, isolating the variables in terms of d using determinants and solving a single quadratic equation for d, we achieve a more straightforward and reliable solution to the GPS positioning problem.

3 Conditioning Analysis of the GPS System

3.1 Objective

The objective of this section is to evaluate how the **conditioning** of the GPS system changes when the satellite positions are either **loosely distributed** or **tightly grouped**. Sensitivity to small errors in signal travel times t_i is analyzed using the **Error Magnification Factor (EMF)**. By comparing EMF values for both configurations we can identify how satellite geometry impacts the stability of the GPS positioning system.

3.2 Background

3.2.1 GPS Sensitivity and Conditioning

In GPS systems, small timing errors Δt_i in the satellite signals can result in significant positional errors. These errors arise from the **ill-conditioning** of the system of equations, which occurs when satellite positions are geometrically clustered.

The system is analyzed by:

1. **Introducing Perturbations**: A small timing error $\Delta t_i \approx 10^{-8}\,\text{s}$ (equivalent to 3 meters) is applied to each signal.
2. **Forward Error**: The resulting change in the computed position is measured as:

$$\|\Delta x, \Delta y, \Delta z\|_2 = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$$

3. **Error Magnification Factor (EMF)**: The EMF quantifies the sensitivity of the system to perturbations:

$$\text{EMF} = \frac{\|\Delta x, \Delta y, \Delta z\|_2}{c\,\|\Delta t_i\|_2}$$

   A **higher EMF** indicates a poorly conditioned system.

3.2.2 Satellite Geometry

- **Loosely Distributed Satellites**: Satellites are widely spaced across the sky, providing better geometric diversity. This reduces ill-conditioning and improves robustness.
- **Tightly Grouped Satellites**: Satellites are close together (within 5% of one another in spherical coordinates), leading to **geometric correlation** and increased sensitivity to timing errors.

3.3 Methodology

3.3.1 Steps to Solve

The analysis was conducted as follows:

1. **Satellite Position Generation**:
   - Loosely distributed satellites were generated using diverse spherical coordinates (ϕ and θ).
   - Tightly grouped satellites were generated by restricting both ϕ_i and θ_i within 5% of one another.
2. **Nominal and Perturbed Signal Times**:
   - For both configurations, nominal signal travel times t_i were calculated based on the range:

$$R_i = \sqrt{A_i^2 + B_i^2 + (C_i - 6370)^2}, \quad t_i = d + \frac{R_i}{c}$$

   - Each t_i was perturbed by $10^{-8}\,\text{s}$, while other times were held constant.

3. **Solve for Position**:
   - The GPS equations were solved numerically using `fsolve` for both nominal and perturbed travel times.
   - The positional error $\|\Delta x, \Delta y, \Delta z\|_2$ was computed for each perturbation.
4. **Calculate EMF**:
   The EMF was calculated using the formula:

$$\text{EMF} = \frac{\|\Delta x, \Delta y, \Delta z\|_2}{c\,\|\Delta t_i\|_2}$$

5. **Compare Results**:
   EMF values were compared for both the loosely distributed and tightly grouped satellite configurations.

Accompanying Code

The accompanying code performs the following steps:

1. **Generate Satellite Positions**:
   - Loosely spaced satellites use diverse ϕ and θ values.
   - Tightly grouped satellites have ϕ and θ within 5% of each other.
2. **Calculate Ranges and Times**:
   Nominal ranges R_i and travel times t_i are computed.
3. **Introduce Perturbations**:
   Each signal time t_i is perturbed by 10^{-8} s, and the GPS equations are solved numerically using `fsolve`.
4. **Compute Forward Error and EMF**:
   - The positional error is computed as the Euclidean distance between the nominal and perturbed positions.
   - EMF values are calculated for each perturbation.
5. **Comparison of Configurations**:
   EMF values for loosely and tightly grouped satellites are compared, and the maximum EMF is identified.

3.4 Results

The results of the EMF analysis for both satellite configurations are as follows:

3.4.1 Loose Satellites

Perturbation in t_i	EMF Value
t_1	2.207538
t_2	2.828432
t_3	2.856126
t_4	2.203932

Maximum EMF:

$$\text{EMF}_{\text{max}} = 2.856126$$

3.4.2 Tightly Grouped Satellites

Perturbation in t_i	EMF Value
t_1	1553.480087
t_2	2317.787515
t_3	3402.498373
t_4	466.068303

Maximum EMF:

$$\text{EMF}_{\text{max}} = 3402.498373$$

3.4.3 Comparison

Configuration	EMF Range	Maximum EMF	Sensitivity
Loose Satellites	2.2 to 2.8	2.856126	Low
Tightly Grouped	466 to 3402	3402.498373	Very High

3.5 Interpretation

The results demonstrate a clear relationship between satellite geometry and the conditioning of the GPS system:

1. **Loose Satellites**:
   - EMF values remain low (around 2–3), indicating that the system is **well-conditioned**.
   - Perturbations in signal times result in small position errors due to the geometric diversity of the satellites.
2. **Tightly Grouped Satellites**:
   - EMF values increase dramatically (up to 3402), showing that the system becomes **ill-conditioned**.
   - Small input errors are significantly amplified, leading to large position errors.
   - This sensitivity arises from the satellites' correlated geometry, where their signals cannot provide sufficient independent information.

3.6 Conclusion

This analysis highlights the critical role of satellite geometry in the conditioning of the GPS system:

- Loosely distributed satellites provide better geometric diversity and robustness, resulting in lower EMF values and greater accuracy.
- Tightly grouped satellites lead to poor conditioning, amplifying errors and making the system highly sensitive to small perturbations in signal travel times.

To achieve accurate and stable GPS positioning, it is crucial to use satellites that are well-distributed across the sky. This insight is especially important for improving satellite selection algorithms in GPS receivers.