Project No. 2

"Reality Check 4, Numerical Analysis by Timothy Sauer"
(3rd Edition, Pages 248-251)

GPS, Conditioning, and Non Linear Least Squares

November 23, 2022

Submitted by:
Liteshwar Rao

Numerical Methods, AI-5003
Topics in Scientific Computing, MAT-5402
Fall 2022
Yeshiva University

Contents

1 Abstract

This project explores the application of numerical methods in GPS positioning covering satellite coordinates, receiver position near earth, and transmission time. Specifically, an attempt has been made to use multivariate **Newton-Raphson** method to solve or find roots of non-linear systems of equations followed by **quadratic formula** to find exact roots of the equation, application of **reduced row echelon form (rref)** to solve systems of linear equations, conditioning of GPS problem, measure input and output errors with respect to changes in coordinates, and calculate error magnification factor when the satellites are bunched and unbunched. Lastly, the project uses **Gauss-Newton** method to solve non-linear least squares problem. The project concludes by stating whether GPS error and conditioning number can be reduced by adding more satellites by comparing the results when satellites are unbunched with that of bunched satellites.

2 Introduction[1]

The Global Positioning System (GPS) consists of 24 satellites carrying atomic clocks, orbiting the earth at an altitude of $20,200$ km. Each satellite has a simple mission: to transmit carefully synchronized signals from predetermined positions in space, to be picked up by GPS receivers on earth. The receivers use the information, with some mathematics (described shortly), to determine accurate (x, y, z) coordinates of the receiver. If three satellites are available, then three spheres are known, whose intersection consists of two points, as shown in Figure 1. One intersection point is the location of the receiver. The other is normally far from the earth's surface and can be safely disregarded.

However, there is a major problem with this analysis. First, although the transmissions from the satellites are timed nearly to the nanosecond by onboard atomic clocks, the clock in the typical low-cost receiver on earth has relatively poor accuracy. If we solve the three equations with slightly inaccurate timing, the calculated position could be wrong by several kilometers. Fortunately, there is a way to fix this problem. The price to pay is one extra satellite. Define d to be the difference between the synchronized time on the (now four) satellite clocks and the earth-bound receiver clock. Denote the location of satellite i by (Ai, Bi, Ci).

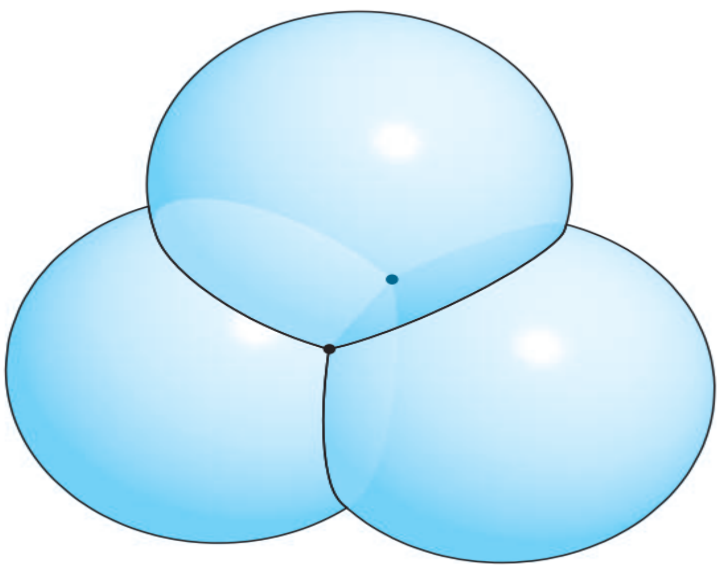

Figure 1 Three intersecting spheres

2.1 Equations to be solved

We need to solve following system (4.38) for (x, y, z, d) for given values of A_i, B_i, C_i, t_i, c , where $i = 1, 2, 3, 4$ respectively for four satellites.

$$(x - A_1)^2 + (y - B_1)^2 + (z - C_1)^2 = (c(t_1 - d))^2 \tag{1}$$
$$(x - A_2)^2 + (y - B_2)^2 + (z - C_2)^2 = (c(t_2 - d))^2 \tag{2}$$
$$(x - A_3)^2 + (y - B_3)^2 + (z - C_3)^2 = (c(t_3 - d))^2 \tag{3}$$
$$(x - A_4)^2 + (y - B_4)^2 + (z - C_4)^2 - (c(t_4 - d))^2 \tag{4}$$

[1]Reality Check, GPS, conditioning, and nonlinear least squares, page 248, Timothy Sauer, 3rd edition

Two further problems emerge when GPS is deployed. First is the conditioning of the system of equations (4.38). We will find that solving for (x, y, z, d) is ill-conditioned when the satellites are bunched closely in the sky. The second difficulty is that the transmission speed of the signals is not precisely c. Furthermore, the signals may encounter obstacles on earth before reaching the receiver, an effect called multipath interference. To the extent that these obstacles have an equal impact on each satellite path, introducing the time correction d on the right side of (4.38) helps. In general, however, this assumption is not viable and will lead us to add information from more satellites and consider applying **Gauss–Newton** to solve a least squares problem.

3 Activity 1

We employ the **Multivariate Newton-Raphson** method to solve for the position and time correction using four satellites by finding (x, y, z, d). We use maximum number of iteration $= 100$ denoting $\alpha_1, \alpha_2, \ldots, \alpha_{100}$ and tolerance level $= 10^{-6}$ denoting $\frac{\sqrt{||\boldsymbol{X_{i+1}} - \boldsymbol{X_i}||_2}}{\sqrt{4}} < 10^{-6}$ the **RMSE** of vector of unknowns as stopping conditions. We test the Multivariate Newton's Method for 4 satellites (A_i, B_i, C_i, t_i) with a known answer to see if we can obtain the same solution. To solve for $\alpha = (x, y, z, d)$ the Multivariate Newton Method takes the Jacobian matrix of System $F(\alpha) = (f_1, f_2, f_3, f_4)$ seen below.

$$DF(\alpha) = \begin{bmatrix} \partial f_1/\partial x & \partial f_1/\partial y & \partial f_1/\partial z & \partial f_1/\partial t \\ \partial f_2/\partial x & \partial f_2/\partial y & \partial f_2/\partial z & \partial f_2/\partial t \\ \partial f_3/\partial x & \partial f_3/\partial y & \partial f_3/\partial z & \partial f_3/\partial t \\ \partial f_4/\partial x & \partial f_4/\partial y & \partial f_4/\partial z & \partial f_4/\partial t \end{bmatrix} = \begin{bmatrix} 2(x - A_1) & 2(y - B_1) & 2(z - C_1) & 2c^2(t_1 - d) \\ 2(x - A_2) & 2(y - B_2) & 2(z - C_2) & 2c^2(t_2 - d) \\ 2(x - A_3) & 2(y - B_3) & 2(z - C_3) & 2c^2(t_3 - d) \\ 2(x - A_4) & 2(y - B_4) & 2(z - C_4) & 2c^2(t_4 - d) \end{bmatrix}$$

Beginning with an initial vector α_0 we iterate through $n \in N$ steps of the process of solving:

$$\begin{cases} DF(\alpha_k)s = F(\alpha_k) \\ \alpha_{k+1} = \alpha_k - s \end{cases}$$

$$for\, k = 0, 1, 2, \ldots, n$$

We are given $\alpha_0 = (0, 0, 6370, 0)$ as initial vector. We get the desired level of accuracy in four iterations and our solution vector and given solution is given below in the **table 1** for comparison. We see that our solution is exactly matching with the given solution. We can also see in **table 2** that at this solution the equations 1,2,3,4 given above are approximately zero.

Table 1: Values of x, y, z, d

Variable	**Our solution**	**Given solution**
x	-4.17727096e+01	-41.77271
y	-1.67891941e+01	-16.78919
z	6.37005956e+03	6370.0596
d	-3.20156583e-03	$-3.201566X10^{-3}$

Table 2: Values of equations 1,2,3,4

Equation	**Value**
1	-1.78813934e-07
2	-1.19209290e-07
3	-1.19209290e-07
4	-1.19209290e-07

We see that **Multivariate Newton Raphson** method has quadratic rate of convergence if initial choosen roots are closer to the actual roots. We demonstrate this rate of convergence in the **table 3** below. We see that at each successive iteration the square of the value of $||\boldsymbol{F(X_k)}||_\infty$ for $k = 0, 1, 2, \ldots$ is lesser than its value in the previous iteration.

Table 3: Demonstration of quadratic convergence

Iteration no. (k)	$\|\|\boldsymbol{F(X_k)}\|\|_\infty$
1	954699.9125281572
2	309.60563373565674
3	3.272294998168945e-05
4	1.7881393432617188e-07

The execution time involves defining functions, their Jacobians, and iterating Multivariate Newton-Raphson method is approximately 0.01 seconds.

4 Activity 2

We solve the same problem here, finding (x, y, z, d) as we did in activity 1 but we use quadratic formula instead of multivariate **Newton-Raphson** method. To setup the problem, following steps are required:

- Subtract equations 2,3,4 from equation 1 and get three linear equations in (x, y, z, d). The thee equations are below:

$$2(A_4 - A_1)x + 2(B_4 - B_1)y + 2(C_4 - C_1)z - 2c^2(t_4 - t_1)d - c^2(t_1^2 - t_4^2) + A_1^2 - A_4^2 + B_1^2 - B_4^2 + C_1^2 - C_4^2 = 0 \tag{5}$$

$$2(A_3 - A_1)x + 2(B_3 - B_1)y + 2(C_3 - C_1)z - 2c^2(t_3 - t_1)d - c^2(t_1^2 - t_3^2) + A_1^2 - A_3^2 + B_1^2 - B_3^2 + C_1^2 - C_3^2 = 0 \tag{6}$$

$$2(A_2 - A_1)x + 2(B_2 - B_1)y + 2(C_2 - C_1)z - 2c^2(t_2 - t_1)d - c^2(t_1^2 - t_2^2) + A_1^2 - A_2^2 + B_1^2 - B_2^2 + C_1^2 - C_2^2 = 0 \tag{7}$$

- The above system of equation is written into matrix-vector notation by taking the coefficient of x, y, z, d and denoting them by matrix A, unknown vector x, y, z, d as a column vector u and all constant terms by a column vector b . This can be rewritten as $Au = b$.

$$\begin{bmatrix} 2(A_4 - A_1) & 2(B_4 - B_1) & 2(C_4 - C_1) & -2c^2(t_4 - t_1) \\ 2(A_3 - A_1) & 2(B_3 - B_1) & 2(C_3 - C_1) & -2c^2(t_3 - t_1) \\ 2(A_2 - A_1) & 2(B_2 - B_1) & 2(C_2 - C_1) & -2c^2(t_2 - t_1) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ d \end{bmatrix} = \begin{bmatrix} c^2(t_1^2 - t_4^2) - A_1^2 + A_4^2 - B_1^2 + B_4^2 - C_1^2 + C_4^2 \\ c^2(t_1^2 - t_3^2) - A_1^2 + A_3^2 - B_1^2 + B_3^2 - C_1^2 + C_3^2 \\ c^2(t_1^2 - t_2^2) - A_1^2 + A_2^2 - B_1^2 + B_2^2 - C_1^2 + C_2^2 \end{bmatrix}$$

- Setup the augmented matrix to solve for x, y, z, d execute reduced ow echelon form (rref). However, we have 4 unknowns and 3 equations, the system is inconsistent. We get following form after using rref and then we express x, y, z in terms of d, where r_{ij} denotes element of the below matrix in row i and column j.

$$\begin{bmatrix} 1 & 0 & 0 & r_{14} & | & r_{15} \\ 0 & 1 & 0 & r_{24} & | & r_{25} \\ 0 & 0 & 1 & r_{34} & | & r_{35} \end{bmatrix}$$

$$x = -r_{14}d + r_{15} \tag{8}$$

$$y = -r_{24}d + r_{25} \tag{9}$$

$$z = -r_{34}d + r_{35} \tag{10}$$

- Substitute equations 8,9,10 into equation 1 and get one quadratic equation in d.

$$(A_1 + dr_{14} - r_{15})^2 + (B_1 + dr_{24} - r_{25})^2 + (C_1 + dr_{34} + -r_{35})^2 - c^2(d - t_1)^2 \tag{11}$$

- Now, we have quadratic equation in d which can be solved for d using the quadratic formula:

$$d = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{12}$$

- After getting the solutions for d we use equations 8,9,10 to find the coordinates x, y, z.

Since we have a quadratic equation, this system has two solutions. However, we disregard one of the solutions for which the elevation data, the z coordinate, does not reflect the point on the surface of the earth. It is better to use quadratic formula to find exact solutions for this problem because a slight approximation even to the level of *thousandths* can create an error of meters on the surface of the earth. The reason is also hinted in the activity itself on page 248 in the last paragraph that slight inaccurate timing can cause the calculated position go wrong by several kilometers. Thus, to have correct position and reception of signals in terms of transmission time we must measure them exactly.

We get following results after executing rref:

$$\begin{bmatrix} 1 & 0 & 0 & -10.7491778814199 & | & -41.7382953702257 \\ 0 & 1 & 0 & 623.677195294780 & | & -18.7859377036771 \\ 0 & 0 & 1 & 83788.8071424892 & | & 6101.80417737350 \end{bmatrix}$$

As per the steps mentioned above, we get following coordinates in terms of d:

$$x = 10.7491778814199 * d - 41.7382953702257 \tag{13}$$

$$y = -623.67719529478 * d - 18.7859377036771 \tag{14}$$

$$z = 6101.8041773735 - 83788.8071424892 * d \tag{15}$$

And the quadratic equation in d is given below:

$$-82854564582.5318 * d^2 + 15077167846.7619 * d + 49119806.6409449 \tag{16}$$

Solving for d, the two solutions are -0.00320157 and 0.18517305. Solving for x, y, z coordinates with both d the results are presented in **table 4** below:

Table 4: Values of x, y, z, d

Coordinates	$\boldsymbol{d_1 = -0.00320157}$	$\boldsymbol{d_2 = 0.18517305}$
x	-41.7727095708270	-39.7478373481468
y	-16.7891941065242	-134.274144360665
z	6370.05955922334	-9413.62455373576

We can see that for $d_2 = 0.18517305$ we get the $z = -9413.62455373576$ which is much larger than the radius of the earth (6370) km. Thus, we disregard this second solution. We can check that the solutions x, y, z we get for $d_1 = -0.00320157$ match with the solution mentioned in activity 1.

The execution time, involves solving for d and x, y, z for both values of d, is approximately 0.005 seconds.

5 Activity 4

In this activity we set up a test of the conditioning of the GPS problem. The satellite positions (A_i, B_i, C_i) is defined by for $i = 1, 2, 3, 4$

$$A_i = \rho \cos\phi_i \cos\theta_i \tag{17}$$
$$B_i = \rho \cos\phi_i \sin\theta_i \tag{18}$$
$$C_i = \rho \sin\phi_i \tag{19}$$

and ϕ_1, θ_i at equispaced values,

$$(\phi_i, \theta_i) = \left(\frac{i\pi}{8}, \frac{(i-1)\pi}{2}\right) \tag{20}$$
$$\rho = 26570\,\text{km} \tag{21}$$
$$d = 0.0001 \tag{22}$$
$$c \approx 299792.458\,\text{km/sec} \tag{23}$$
$$R_i = \sqrt{A_i^2 + B_i^2 + (C_i - 6370)^2} \tag{24}$$
$$t_i = d + R_i/c \tag{25}$$

First, we solve for x, y, z using quadratic formula as we did in activity 2 with $A_i, B_i, C_i, R_i, t_i, c, d$ calculated from above. We get following results after executing rref:

$$\begin{bmatrix} 1 & 0 & 0 & 1270.90445499704 & | & 0.127090445500859 \\ 0 & 1 & 0 & -1118.66226324829 & | & -0.111866226324805 \\ 0 & 0 & 1 & 86702.1069978110 & | & 6378.67021069978 \end{bmatrix}$$

As per the steps mentioned above, we get following coordinates in terms of d:

$$x = 0.127090445500859 - 1270.90445499704 * d \tag{26}$$
$$y = 1118.66226324829 * d - 0.111866226324805 \tag{27}$$
$$z = 6378.67021069978 - 86702.106997811 * d \tag{28}$$

And the quadratic equation in d is given below:

$$-82355395912.4289 * d^2 + 15630851505.094 * d - 1562261.59655011 \tag{29}$$

Solving for d, the two solutions are $1.00000000e-04$ and $1.89697539e-01$. Solving for x, y, z coordinates with both d the results are presented in **table 5** below:

Table 5: Values of x, y, z, d

Coordinates	$\boldsymbol{d_1 = 1.00000000e-04}$	$\boldsymbol{d_2 = 1.89697539e-01}$
x	1.17114651310146e-12	-240.960357550957
y	9.22872889219661e-15	212.095612594024
z	6370.00000000000	-10068.5061524256

We can see that for $d_2 = 1.89697539e - 01$ we get the $z = -10068.5061524256$ which is much larger than the radius of the earth (6370) km. Thus, we disregard this second solution. The plot of four satellites is given in **figure 2** below.

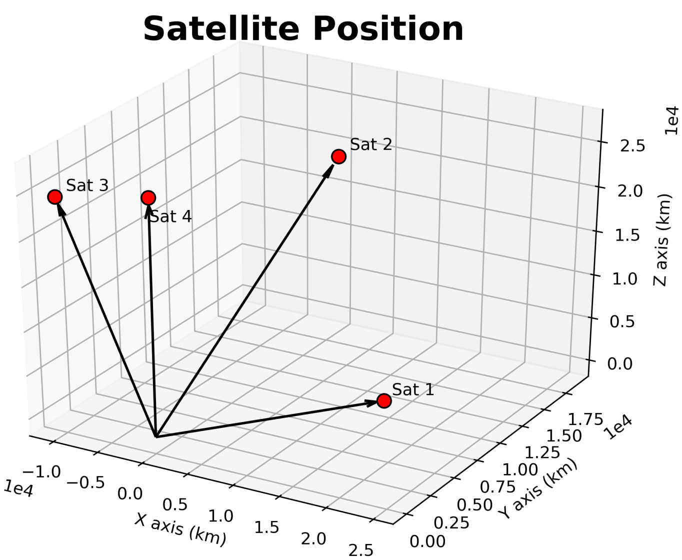

Figure 2 Position of Satellites

Now, we set up conditioning of the GPS problem by changing the t_i by $\Delta t_i = 10^{-8}$ defined as input error and calculate the change in coordinates x, y, z defined as output error $||(\Delta x, \Delta y, \Delta z)||_\infty$ corresponding to the change in t_i. Then, we define **error magnification factor** (emf) as below and the **condition number** of the problem to be the maximum error magnification factor for all small Δt_i.

$$\boldsymbol{emf} = \frac{||(\Delta x, \Delta y, \Delta z)||_\infty}{c||(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4)||_\infty} \tag{30}$$

$$(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4) = (+\epsilon, +\epsilon, +\epsilon, -\epsilon) \tag{31}$$

$$(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4) = (+\epsilon, +\epsilon, -\epsilon, -\epsilon) \tag{32}$$

$$(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4) = (+\epsilon, -\epsilon, +\epsilon, -\epsilon) \tag{33}$$

$$(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4) = (+\epsilon, -\epsilon, -\epsilon, -\epsilon) \tag{34}$$

$$\text{with}\, \epsilon = 10^{-8}$$

For each equation 31 to 34, we calculate new solutions $\bar{x}, \bar{y}, \bar{z}, \bar{d}$ by changing the t_i, change in position defined by $||(x, y, z) - (\bar{x}, \bar{y}, \bar{z})||_\infty$ and emf as defined in equation 30 by using quadratic formula as we did earlier. For each equation 31 to 34 we get two solutions for d and as stated earlier the second solution is disregarded because it gives the z coordinate which does not reflect the point on the surface of the earth. The results are presented below in **table 6**:

Table 6: Values of $x, y, z, d,$ emf, input error, output error

Variable	**equation 31**	**equation 32**	**equation 33**	**equation 34**
x	0.00800303974453537	-0.00141483300624948	0.00800303975065556	-0.00141483300703993
y	0.00163956589402865	-0.00426858230105415	0.00882672124720273	0.00291857656454204
z	6370.01641452326	6370.00542750150	6370.01641452368	6370.00542750144
d	0.00010004475295049491	0.00010000810419460163	0.00010004475294567976	0.00010000810419522358
input error	0.0029979245784220246	0.0029979245784220246	0.0029979245784220246	0.0029979245784220246
output error	0.0164145232674855	0.00542750149907079	0.0164145236849436	0.00542750144541060
emf	5.47529560470977	1.81041962767709	5.47529574395879	1.81041960977797

From the table 6 above we can see that the maximum position error (output error) in meters = 16.4145236849436 meters for equation 33 and condition number of the problem (maximum emf) = 5.47529574395879 for equation 33.

6 Activity 5

In this activity we do what we did in activity 4 above. We calculate initial x, y, z, d and then by changing t_i we calculate new $\bar{x}, \bar{y}, \bar{z}, \bar{d}$, input error, output error, emf. Everything remains same here, but satellites are tightly grouped based on ϕ_i, θ_i as given below for $i = 1, 2, 3, 4$.

$$(\phi_i, \theta_i) = \left(\frac{\pi}{2} + (i-1)\frac{5\pi}{100.2}, \frac{(i-1).5.2.\pi}{100}\right) \tag{35}$$

We get following results after executing rref:

$$\begin{bmatrix} 1 & 0 & 0 & -0.491122051149574 & | & -4.91122680705649e-5 \\ 0 & 1 & 0 & 470.740626800584 & | & 0.0470740635173058 \\ 0 & 0 & 1 & 90787.6580889053 & | & 6379.07876580349 \end{bmatrix}$$

As per the steps mentioned above, we get following coordinates in terms of d:

$$x = 0.491122051149574 * d - 4.91122680705649e - 5 \tag{36}$$

$$y = 0.0470740635173058 - 470.740626800584 * d \tag{37}$$

$$z = 6379.07876580349 - 90787.6580889053 * d \tag{38}$$

And the quadratic equation in d is given below:

$$-81632897415.4348 * d^2 + 15795763269.4758 * d - 1578759.99775505 \tag{39}$$

Solving for d, the two solutions are $1.00000000e-04$ and $1.89697539e-01$. Solving for x, y, z coordinates with both d the results are presented in **table 7** below:

Table 7: Values of x, y, z, d

Coordinates	$\boldsymbol{d_1 = 1.00000000e-04}$	$\boldsymbol{d_2 = 1.89697539e-01}$
x	-6.29556137107215e-11	0.0931155324075295
y	8.37253454066822e-10	-89.2512645624731
z	6369.99999999460	-10843.1165866237

We can see that for $d_2 = 1.89697539e-01$ we get the $z = -10843.1165866237$ which is much larger than the radius of the earth (6370) km. Thus, we disregard this second solution. The plot of four satellites is given in **figure 3** below.

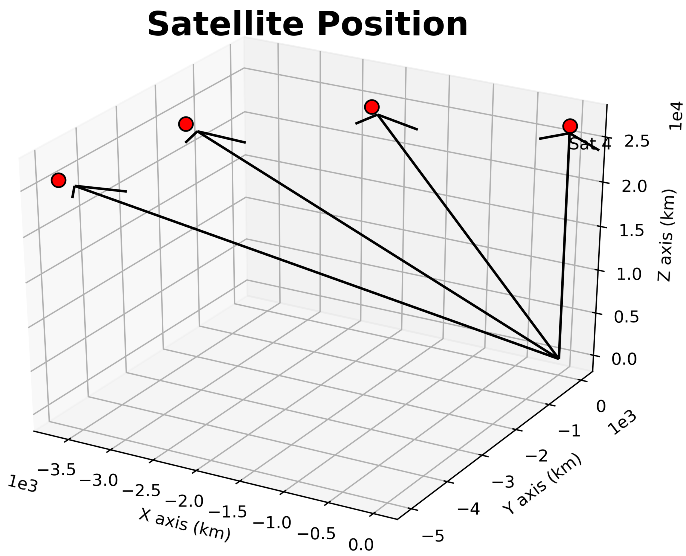

Figure 3 Position of Satellites

The results are presented below in **table 8** after setting up the conditioning of the GPS problem:

Table 8: Values of x, y, z, d, emf, input error, output error

Variable	**equation 31**	**equation 32**	**equation 33**	**equation 34**
x	-0.0098619 2514254343	0.0472346 296088492	-0.149615 013851172	-0.0924654 835840050
y	0.349281 035045689	-0.725930 335267656	1.530865 84586420	0.4550425 52592169
z	6368.092 69781956	6373.471 92857916	6362.472 44684141	6367.854 64648235
d	9.36479146 5929276e-05	0.00011159 106344940835	7.490059014 447573e-05	9.2853853 14970767e-05
input error	0.00299792 45784220246	0.00299792 45784220246	0.00299792 45784220246	0.00299792 45784220246
output error	1.907302 17503278	3.47192 858456219	7.52755 315318700	2.14535 351225095
emf	636.207 524619149	1158.11 071751167	2510.921 45792046	715.6129 03570833

From the table 8 above we can see that the maximum position error (output error) in meters = 7527.55315318700 meters for equation 33 and condition number of the problem (maximum emf) = 2510.92145792046 for equation 33.

The tightly grouped satellites give the larger change in position (output error) as compared to what we have in activity 4 (loosely grouped satellites). We can also see that the error magnification factor and condition number are extremely large for tightly grouped satellites.

7 Activity 6

In this activity we attempt to answer whether the GPS error and condition number can be reduced by adding satellites. We use the same procedure as we did in activity 4 but with the following changes:

- Total satellites are now eight.
- Application of Gauss-Newton least squares since we have eight equations with four unknowns.
- Change in ϕ_i, θ_i as below for $i = 1, 2, \ldots, 8$

$$(\phi_i, \theta_i) = \left(\frac{i.\pi}{16}, \frac{(i-1).\pi}{4}\right) \tag{40}$$

- Change in Δt as below:

$$(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4, \Delta t_5, \Delta t_6, \Delta t_7, \Delta t_8) = (+\epsilon, +\epsilon, +\epsilon, -\epsilon, +\epsilon, +\epsilon, +\epsilon, -\epsilon) \tag{41}$$
$$(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4, \Delta t_5, \Delta t_6, \Delta t_7, \Delta t_8) = (+\epsilon, +\epsilon, -\epsilon, -\epsilon, +\epsilon, +\epsilon, -\epsilon, -\epsilon) \tag{42}$$
$$(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4, \Delta t_5, \Delta t_6, \Delta t_7, \Delta t_8) = (+\epsilon, -\epsilon, +\epsilon, -\epsilon, +\epsilon, -\epsilon, +\epsilon, -\epsilon) \tag{43}$$
$$(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4, \Delta t_5, \Delta t_6, \Delta t_7, \Delta t_8) = (+\epsilon, -\epsilon, -\epsilon, -\epsilon, +\epsilon, -\epsilon, -\epsilon, -\epsilon) \tag{44}$$
$$\text{with } \epsilon = 10^{-8}$$

- Calculate A_i, B_i, C_i, R_i, t_i as we did in activity 4 and 5 but for eight satellites with the given angles ϕ_i, θ_i as above.
- The good initial vector is $(0, 0, 6370, 0)$ because we have seen in activity 1 in table 2 that at this vector, equation 1,2,3, and 4 are approximately equal to zero with the tolerance level of 10^{-6}. In other words, this vector is very close to the roots of these equations, and we know Gauss-Newton converge locally only, if initial guess is sufficiently close to the actual roots.
- To check if it actually converged to a minimum, we must check following inequality for equation 1,2,3, and 4 by taking their derivative with respect to $(x, y), (y, z), (x, z)$ and evaluate them at the identified solution vector.

$$\left| r_i \frac{\partial^2 r_i}{\partial x_j \partial x_k} \right| << \left| \frac{\partial r_i}{\partial x_j} \frac{\partial r_i}{\partial x_k} \right| \tag{45}$$

<u>Setting up Gauss-Newton Least Squares method:</u>[2]
Consider the nonlinear system:

$$\begin{aligned} r_1(x1, \ldots, x_n) &= 0 \\ &\vdots \\ r_m(x1, \ldots, x_n) &= 0 \end{aligned} \tag{46}$$

[2]Gauss-Newton method, page 240, Timothy Sauer, 3rd edition

of m equations in n unknowns, with $m \geq n$

The square residual is the function of $E(x_1, \ldots, x_n)$ as shown below.

$$E(x_1, \ldots, x_n) = \frac{1}{2}(r_1^2 + \ldots + r_m^2) = \frac{1}{2}r^T r \tag{47}$$

To minimize the residual function, take the derivative of equation 46 and equate to zero:

$$\nabla E(X) = \nabla \left(\frac{1}{2} r^T r = 0 \right) \tag{48}$$

$$\nabla E(X) = \frac{1}{2}(r^t Dr + r^T Dr) = r^T Dr = 0 \tag{49}$$

Let $F(X) := r^T Dr$, and solve $F(X) = 0$ using Multivariate Newton method. Since, Multivariate Newton method acts on a column vector function, let $F(X) := (r^T Dr)^T = (Dr)^T r$ and solve $F(X) = 0$. Using Multivariate Newton method we know it gives us following for iterations to solve for s:

$$DF(x_i)s = F(x_i) \tag{50}$$

$$X_{i+1} = x_i - s \tag{51}$$

We need $DF(x_i)$, which is equal to:

$$DF(x) = D[(Dr)^T r] \tag{52}$$

$$= (Dr)^T.Dr + r.D[(Dr)^T] \tag{53}$$

We ignore the terms $r.D[(Dr)^T]$. So, $DF(x) \approx (Dr)^T.Dr$. Putting this into equation 49 we get the following:

$$(Dr)^T.Dr.s = (Dr)^T.r \tag{54}$$

$$\text{Let's define } A := Dr \text{ we get} \tag{55}$$

$$A^T.A.s = A^T.r \tag{56}$$

$$x_{i+1} = x_i - s \tag{57}$$

which we solve for s using QR factorization since it is same as normal equations

We get the initial solutions x, y, z, d without change in Δt_i using the algorithm described above. We get $(x, y, z, d) = (2.85359262e - 12, 7.11034112e - 13, 6.37000000e + 03, 1.00000000e - 04)$ in the fourth iteration as our initial solution. The plot of eight satellites are shown below in **figure 4**.

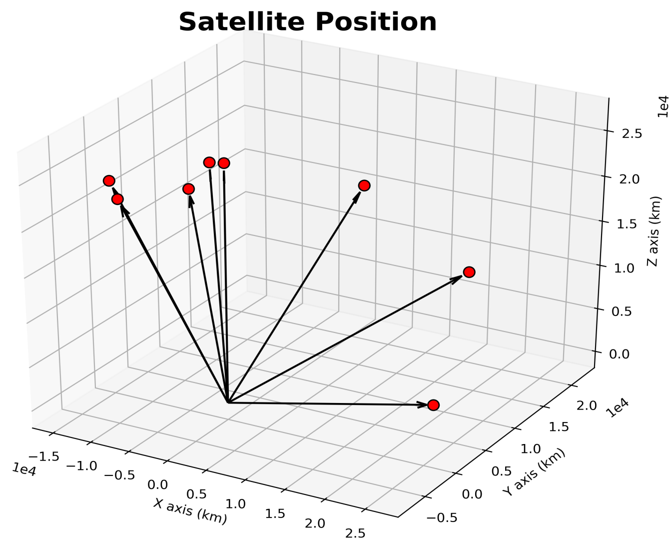

Figure 4 Position of Satellites

Now, we set up the conditioning of the GPS problem using Δt_i and get the new solutions for each equation no. 41,42,43,44 using the same Gauss-Newton method. The results are presented below in the **table 9** below:

Table 9: Values of x, y, z, d, emf, input error, output error

Variable	**equation 41**	**equation 42**	**equation 43**	**equation 44**
x	-2.83893582e-03	-1.15217935e-04	1.38699705e-03	4.11071132e-03
y	5.15876275e-04	5.18664395e-03	3.03089556e-03	7.70165995e-03
z	6.36999720e+03	6.37000428e+03	6.37000331e+03	6.37001039e+03
d	1.00002403e-04	1.00014211e-04	1.00010906e-04	1.00022713e-04
input error	0.00299792 45784220246	0.00299792 45784220246	0.00299792 45784220246	0.00299792 45784220246
output error	0.00283894	0.00518664	0.00331	0.01039
emf	0.94696706	1.7300782	1.10409716	3.46573095
iteration	5	3	3	3

From the table 9 above we can see that the maximum position error (output error) in meters = 10.39 meters for equation 44 and condition number of the problem (maximum emf) = 3.46573095 for equation 44. We have seen that the condition number and position error have reduced after increase in the number of satellites.

8 Conclusion

The aim of this project is to find whether the condition number and position error can be minimized by adding more number of satellites. The summary of the findings is shown below in **table 10**.

Table 10: Summary of findings

No. of satellites	4	4	8
Type	Loosely bunched	Tightly bunched	Loosely bunched
position error (in meters)	16.414	7527.553	10.39
condition number (approx)	5.475	2510.921	3.465

We can conclude following:

- The condition number and position number can be minimized further by adding more number of satellites.
- The satellites must be loosely bunched in order to achieve reduction in condition number and position error.
- The maximum GPS error, in meters, can go up to 1000's of meters if satellites are tightly bunched. Conversely, the GPS error is less than a few 100's meters if satellites are loosely bunched. Further, the GPS error can be reduced to under 5 meters if we have eight or more than eight loosely bunched satelittes.
- After changing the signs of $\Delta t_i's$ we see that the results are similar in terms of position error and emf. However, the signs of coordinates get flipped. The results are presented below in **table 11**. We obtain new results by flipping the sign of each Δt_i from positive to negative and vice-versa. For instance, in equation no. 41, we obtain new result by using the following. We followed same procedure for equation no 42, 43, and 44.

$$(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4, \Delta t_5, \Delta t_6, \Delta t_7, \Delta t_8) = (-\epsilon, -\epsilon, -\epsilon, +\epsilon, -\epsilon, -\epsilon, -\epsilon, +\epsilon) \tag{58}$$

Table 11: Values of x, y, z, d, emf, input error, output error after change in signs of $\Delta t_i's$

Variable	**equation 41**	**equation 42**	**equation 43**	**equation 44**
x	2.83893772e-03	1.15220664e-04	-1.38699230e-03	-4.11071298e-03
y	-5.15874756e-04	-5.18664339e-03	-3.03089170e-03	-7.70166363e-03
z	6.37000280e+03	6.36999572e+03	6.36999669e+03	6.36998961e+03
d	9.99975966e-05	9.99857894e-05	9.99890944e-05	9.99772872e-05
input error	0.00299792 45784220246	0.00299792 45784220246	0.00299792 45784220246	0.00299792 45784220246
output error	0.00283894	0.00518664	0.00331	0.01039
emf	0.94696769	1.73007801	1.10409716	3.46573095
iteration	3	3	3	3

References

[1] Gauss-Newton method script
https://www.youtube.com/watch?v=Kln0ZQ7sX8k&t=1080s

[2] Multivariate Newton-Raphson method script
https://www.youtube.com/watch?v=_05BHEuN0ME&t=443s

[3] Execution time of a cell
https://www.geeksforgeeks.org/how-to-check-the-execution-time-of-python-script/

[4] 3-D scatter plot
https://matplotlib.org/stable/gallery/mplot3d/scatter3d.html#sphx-glr-gallery-mplot3d-scatter3d-py

[5] SymPy library
https://www.youtube.com/watch?v=1yBPEPhq54M&list=PLkdGijFCNuVm4IfZlsZPEt4fPJHfl-0g5&index=4

[6] Calculating the norm by using NumPy
https://numpy.org/doc/stable/reference/generated/numpy.linalg.norm.html

[7] Execute rref
https://www.geeksforgeeks.org/python-sympy-matrix-rref-method/