4.4 Numerical Approaches

With (x, y, z, t) denoting the unknowns (receiver position and clock error), the nonlinear system to be solved can be written as

$$(x-3)^2+(y-2)^2+(z-3)^2-[(10010.00692286-t)\cdot c]^2=0 \qquad \text{(1 a)}$$
$$(x-1)^2+(y-3)^2+(z-1)^2-[(10013.34256381-t)\cdot c]^2=0 \qquad \text{(1 b)}$$
$$(x-5)^2+(y-7)^2+(z-4)^2-[(10016.67820476-t)\cdot c]^2=0 \qquad \text{(1 c)}$$
$$(x-1)^2+(y-7)^2+(z-3)^2-[(10020.01384571-t)\cdot c]^2=0 \qquad \text{(1 d)}$$
$$(x-7)^2+(y-6)^2+(z-7)^2-[(10023.34948666-t)\cdot c]^2=0 \qquad \text{(1 e)}$$
$$(x-1)^2+(y-4)^2+(z-9)^2-[(10030.02076857-t)\cdot c]^2=0 \qquad \text{(1 f)}$$

where $c = 0.299792458$ (in the unit of 1,000 km/ms).

The following subsections will describe different ideas of solving this system:

i. Linearize the system - then solve as a square- or overdetermined system dependent on the number of equations, and

ii. Newton's method directly

4.4.1. Linearization.

The equations (1 a) - (1 f) are nonlinear, but if we expand all the squares, each equation will take the form $x^2 + y^2 + z^2 + t^2 c^2 + \{\textit{linear terms}\} = 0$, i.e. if we subtract one of the equations (say the last one) from the rest, all nonlinearities will vanish, and we are left with a linear system

$$\begin{bmatrix} 4 & -4 & -12 & 3.59751 \\ 0 & -2 & -16 & 2.99792 \\ 8 & 6 & -10 & 2.39834 \\ 0 & 6 & -12 & 1.79875 \\ 12 & 4 & -4 & 1.19917 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = \begin{bmatrix} 35971.1 \\ 29957.2 \\ 24031.4 \\ 17993.5 \\ 12059.7 \end{bmatrix}$$

This overdetermined system can be solved in least square sense with the methods in Chapter III.1.4, giving

$$x = 5.0000\ ,\ y = 3.0000\ ,\ z = 1.0000\ ,\ t = 10{,}000\ .$$

If all the data was exact and our calculations had no rounding error, it would have made no difference which equation we selected initially to subtract from the others. With errors present, we ought to repeat this calculation, starting with each of the equations in turn. Then, at the end, we can take the average of the obtained positions - this would likely be more accurate than the individual results.

If we only have four satellites visible, the first elimination would give us 3 linear equations in 4 unknowns. The *echelon form* (Section III.1.1) allows us to express 3 of the unknowns in terms of the 4th one. Substituting these expressions into the equation we started with, we get a quadratic equation in the remaining (the 4th) unknown. This quadratic will typically have two solutions, only one of which will correspond to a 'reasonable' position. We can then immediately find the remaining unknowns. Again, to minimize errors, it is recommended to repeat the calculation the four ways, and average the results.

4.4.2. Newton's Method.

It was a very unusual circumstance that allowed the nonlinear terms in all but one of the equations (1 a) - (1 f) to be eliminated. Linearization - as it occurrs in Newton's method - is much more general. It does then not rely on any particular coincidences between the equations; nor do we end up with one equation less to work with (to use direct linearization, we would need signals from 5 satelites when 4 should have sufficed). With numerical linearization, the process becomes iterative, and relies on the knowledge of a starting guess. How close such guess has to be varies from problem to problem. As we will see, this is not a difficulty at all in the present case.

If we have an approximation x_n, y_n, z_n, t_n , we get the updates to a new approximation

$$x_{n+1} = x_n + \Delta x_n\ ,\ \ y_{n+1} = y_n + \Delta y_n\ ,\ z_{n+1} = z_n + \Delta z_n\ ,\ t_{n+1} = t_n + \Delta t_n$$

by solving the (overdetermined) linear system

$$\begin{bmatrix} 2(x_n-3) & 2(y_n-2) & 2(z_n-3) & 2c^2(10010.00692286-t_n) \\ 2(x_n-1) & 2(y_n-3) & 2(z_n-1) & 2c^2(10013.34256381-t_n) \\ 2(x_n-5) & 2(y_n-7) & 2(z_n-4) & 2c^2(10016.67820476-t_n) \\ 2(x_n-1) & 2(y_n-7) & 2(z_n-3) & 2c^2(10020.01384571-t_n) \\ 2(x_n-7) & 2(y_n-6) & 2(z_n-7) & 2c^2(10023.34948666-t_n) \\ 2(x_n-1) & 2(y_n-4) & 2(z_n-9) & 2c^2(10030.02076857-t_n) \end{bmatrix} \begin{bmatrix} \Delta x_n \\ \Delta y_n \\ \Delta z_n \\ \Delta t_n \end{bmatrix} =$$

$$= \begin{bmatrix} (x_n-3)^2+(y_n-2)^2+(z_n-3)^2-(c(10010.00692286-t_n))^2 \\ (x_n-1)^2+(y_n-3)^2+(z_n-1)^2-(c(10013.34256381-t_n))^2 \\ (x_n-5)^2+(y_n-7)^2+(z_n-4)^2-(c(10016.67820476-t_n))^2 \\ (x_n-1)^2+(y_n-7)^2+(z_n-3)^2-(c(10020.01384571-t_n))^2 \\ (x_n-7)^2+(y_n-6)^2+(z_n-7)^2-(c(10023.34948666-t_n))^2 \\ (x_n-1)^2+(y_n-4)^2+(z_n-9)^2-(c(10030.02076857-t_n))^2 \end{bmatrix}$$

(cf. ....). It now remains to find a start guess. Knowing that signal travel times cannot be negative, one can for example choose t_0 as the shortest time recorded, i.e. $t_0 = 10010.00692286$. Let us also guess that we are at location $x = y = z = 0$. This is an extremely coarse guess; the errors of 5000 and 3000 kms in the *x*- and *y*-directions resp. are the size of a continent. The one thousand km initial error in height places us deep in the earth's interior. The iterations proceed as follows:

n	*x*	*y*	*z*	*t*
0	0.0	0.0	0.0	10010.007
1	6.368727	0.374601	-2.403971	9985.218
2	4.984063	3.018241	1.046303	10000.266
3	5.000160	2.999808	0.999585	9999.998
4	5.000000	3.000000	1.000000	10000.000

We see the typical signs of quadratic convergence - a doubling of correct digits once the iterations have 'settled in'. Here, full convergence is obtained after just 4 iterations - a typical situation when a 'reasonable' guess is available. The numerical errors are at this point reduced to better than 1 m in distance, and 1 μs in time.