

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,700

Open access books available

139,000

International authors and editors

175M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# On Non-Linearity and Convergence in Non-Linear Least Squares

---

Orhan Kurt

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.76313>

---

## Abstract

To interpret and explain the mechanism of an engineering problem, the redundant observations are carried out by scientists and engineers. The functional relationships between the observations and parameters defining the model are generally nonlinear. Those relationships are constituted by a nonlinear equation system. The equations of the system are not solved without using linearization of them on the computer. If the linearized equations are consistent, the solution of the system is ensured for a probably global minimum quickly by any approximated values of the parameters in the least squares (LS). Otherwise, namely an inconsistent case, the convergence of the solution needs to be well-determined approximate values for the global minimum solution even if in LS. A numerical example for 3D space fixes coordinates of an artificial global navigation satellite system (GNSS) satellite modeled by a simple combination of first-degree polynomial and first-order trigonometric functions will be given. It will be shown by the real example that the convergence of the solution depends on the approximated values of the model parameters.

**Keywords:** nonlinear equation system, objective function, least squares, convergence, consistency

---

## 1. Introduction

There are two main computing classes, these are hard and soft computing. Scientists and engineers generally prefer the first-class computing because they can easily establish an explicit mathematical relationship between the model parameters and their data (observations), not in the second class. The relationships between the parameters and data can be linear or nonlinear.

If the relations are nonlinear, they should be linearized via Taylor expansion [1–7]. Therefore, the linear models can be solved by linear algebra [8–15].

To overcome complicated real-life problems whose mathematical models are not known, the soft computing techniques have been developed in the last decades. We can count well-known techniques, some as artificial neural network (ANN), artificial intelligence (AI), machine learning (ML), deep learning (DP), fuzzy logic (FL) and genetic algorithms (GA) [16–18]. The techniques inspired by the human intelligence and learning processes can be very time-consuming according to the data given in run due to their processing based on the trial-and-error method. If these techniques are roughly defined, data (experimental outcomes and observations) are separated into two parts in them, learning (or training) data and test data. Mathematical (functional and/or stochastic) relations between data and model parameters are learned from the learning data. The handled model is tested by means of the test data. After that, the trained and developed model, if meets expectations, is used to estimate for producing unobserved data for the scientific (or engineering) problems [16–18].

In the soft computing techniques, the linear algebra is also a very effective tool to solve the problem as in the hard computing ones. For this reason, we should take a short overview on linear algebra used in science and engineering [16–18].

## 2. Linear algebra and objective functions

Linear algebra has two basic problems. A solution of linear equations system is one of them; the other is the eigendecomposition. In this chapter, we will use both of them upon a linear equation system as a combined form (Eqs. (8)–(11)) in which we will solve the linear equations system by means of the singular value decomposition related with the eigendecomposition (or the matrix diagonalization) [8, 9, 13, 14].

Suppose an estimated unknown vector  $\hat{\mathbf{x}}_u = \mathbf{x} + \hat{\boldsymbol{\delta}}$  (in interested model) and an experimental data (or observations which are stochastic variables) vector  $\mathbf{y}_n = \hat{\mathbf{y}} - \hat{\boldsymbol{\varepsilon}}$  [in which an estimated data and error (residual) vectors are in order of  $\hat{\mathbf{y}}$  and  $\hat{\boldsymbol{\varepsilon}}$ ] by an objective function and their covariance matrices  $\boldsymbol{\Sigma}_{\hat{\mathbf{x}}} = \boldsymbol{\Sigma}_{\mathbf{x}} = \hat{\sigma}_0^2 \mathbf{Q}_{\mathbf{x}}$  (for the unknowns) and  $\boldsymbol{\Sigma}_{\mathbf{y}} = \sigma_0^2 \mathbf{P}^{-1}$  (for the data), respectively, with a priori variance  $\sigma_0^2$  and a posteriori variance  $\hat{\sigma}_0^2$ . Note that  $\hat{\mathbf{x}}$  is a non-stochastic vector before estimation, where an approximated values vector is  $\mathbf{x}$  for  $\hat{\mathbf{x}}$  (hat-sign “^” shows an estimated value for interested parameter according to an objective function). In addition,  $n$ ,  $m$  and  $u$  are the observation number, the equation number and the unknown number, respectively.

Start with a linear or nonlinear functions vector  $\mathbf{f}_m(\hat{\mathbf{y}}, \hat{\mathbf{x}}) = \mathbf{0}$ , we can have a linear mathematical model with a weight matrix ( $\mathbf{P} = \sigma_0^2 \boldsymbol{\Sigma}_{\mathbf{y}}^{-1}$ ) of the observations for  $m = n$ :

$$\boldsymbol{\varepsilon}_n = \mathbf{A}_{n,u} \boldsymbol{\delta}_u - \mathbf{l}_n, \quad \mathbf{P}_{n,n}, \quad (1)$$

$$\mathbf{A}_{n,u} = \left. \frac{\partial \mathbf{f}(\hat{\mathbf{y}}, \hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} \right|_{\hat{\mathbf{y}}, \hat{\mathbf{x}}=\mathbf{y}, \mathbf{x}} \quad \text{and} \quad \mathbf{l}_n = \mathbf{f}(\mathbf{y}, \mathbf{x}).$$

Mathematical model between data and unknowns can be established by Taylor expansion for any model. However, if  $m$  pieces function vector  $\mathbf{f}_m(\hat{\mathbf{y}}, \hat{\mathbf{x}}) = \mathbf{0}$  is not transformed into  $\hat{\mathbf{y}}_n - \mathbf{f}_n(\hat{\mathbf{x}}) = \mathbf{0}$  (for  $m = n$ ), the error in variable solution as in total least squares (TLS) method can be preferred. Therefore,  $\mathbf{f}_m(\hat{\mathbf{y}}, \hat{\mathbf{x}}) = \mathbf{0}$  (for  $m \neq n$ ) should be differenced as following:

$$\mathbf{B}_{m,n} \boldsymbol{\varepsilon}_n - \mathbf{A}_{m,u} \boldsymbol{\delta}_u + \mathbf{I}_m = \mathbf{0} \quad \mathbf{P}_{n,n} \quad (2)$$

where

$$\mathbf{B}_{m,n} = \left. \frac{\partial \mathbf{f}(\hat{\mathbf{y}}, \hat{\mathbf{x}})}{\partial \hat{\mathbf{y}}} \right|_{\hat{\mathbf{y}}, \hat{\mathbf{x}} = \mathbf{y}, \mathbf{x}_0}.$$

Most of science and engineering problems can be modeled as  $\hat{\mathbf{y}} - \mathbf{f}(\hat{\mathbf{x}}) = \mathbf{0}$  ( $m = n$ ). Therefore, the functional model named as indirect adjustment method in the adjustment literature [3–7] in geomatics engineering has been preferred in the chapter. The weight matrix ( $\mathbf{P}_{n,n}$ ) of observations (stochastic variables) would be accepted as a unit matrix  $\mathbf{P}_{n,n} = \mathbf{I}_{n,n}$  in here for simplicity.

## 2.1. Objective functions

A generalization for objective functions is  $L_p$  – Norm ( $p = 1, 2, 3, 4, \dots, \infty$ ) [9, 10]. The first-degree objective function is  $L_1$ -norm estimation which is accepted as a robust estimation method in just linear models [9–11].

$$\mathbf{i}^T |\boldsymbol{\varepsilon}| \mapsto \min \quad L_1 - \text{norm estimation (Least absolute residuals)}, \quad (3)$$

$$\boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} \mapsto \min \quad L_2 - \text{norm estimation (Least squares)}, \quad (4)$$

$$|\boldsymbol{\varepsilon}|_{\max} \mapsto \min \quad L_\infty - \text{norm estimation (Minmax absolute residuals)}, \quad (5)$$

$$\mathbf{i} = [1 \ 1 \ \dots \ 1]^T.$$

The second-degree objective function is  $L_2$ -norm estimation which is known as least squares (LS) method and widely used in hard and soft computations.

The last-degree objective function is  $L_\infty$ -norm estimation which is known as minmax method. In fact, the soft computing techniques use this objective while it applies the trial-and-error method in their learning stages. Eq. (1) under  $L_1$ -norm and  $L_\infty$ -norm is also solved by means of linear programming methods, for this reason; the methods may give several solutions (as being in trial-and-error method) to any interested problem [10, 11].

## 2.2. Rank deficiencies in linear models

While a rank is a number that indicates a linear independent column, the number of the coefficient matrix of unknowns in a linear equation system, a rank deficiency represents a linear dependent column number (if it is smaller than the row number) of the coefficient matrix. Inconsistency in the solution stage of a linear equation system results from the (rank) deficiencies.

Defining the rank of  $\mathbf{A}_{n,u}$  by  $\text{rank}(\mathbf{A}) = r$ , a condition  $r \leq u \leq \min(m, n)$  is always satisfied. In general,  $n = m$  in well-known (or the indirect) LS used in many scientific problems.

Denoting the rank defect  $d$  letter, we can define two type defects [12].

$$d_s = n - r, \quad \text{Surjectivity ("onto" mapping)} \quad (6a)$$

$$d_i = u - r. \quad \text{Injectivity ("one-to-one" mapping)} \quad (6b)$$

Objective functions are used to remove the surjectivity defect  $d_s$  occurred by the redundant observations. The injectivity defect  $d_i$  can consist of three reasons in the estimation problem [12].

*Datum defects (d-defects)* are closely related to the origin of the spatial system. The defect arises if the data do not carry any information to cover the absolute spatial position of the problem given.

*Configuration (Design) defects (c-defects)* occur from weak geometric relation among data and unknowns. To avoid the defect, we can be careful and planned when picking data (whose interval or/and place) and choosing the consistent mathematical model (can use auxiliary variables instead of original ones).

*Ill-conditioning defects (i-defects)* arise from the large intervals among the elements of the coefficient matrix of unknowns. Norming the matrix can reduce ill-conditioning defects but cannot remove it fully. *I-defects* and *c-defects* cannot be separated from each other easily [12].

The defects lead to the failure of any given problem to be solved properly. Since the unknown coefficient matrix cannot be inverted by *regular (ordinary) inverse* methods, we should use *pseudo inverse* to overcome the effects of the defects [8, 9, 13–15]. Eigenvalue and singular value decompositions can be used effectively for the pseudoinverse. Denoting a positive definite symmetric matrix  $\mathbf{N}$  (that is always satisfied for  $\mathbf{N} = \mathbf{A}^T \mathbf{A}$  or  $\mathbf{N} = \mathbf{A} \mathbf{A}^T$ ), its pseudoinverse is:

$$\mathbf{Q}_{u,u} = \mathbf{N}^+ = \mathbf{S} \mathbf{\Lambda}^+ \mathbf{S}^T = \mathbf{V} \mathbf{\Sigma}^+ \mathbf{U}^T, \quad \text{Pseudoinverse of } \mathbf{N} \quad (7a)$$

$$\mathbf{N}_{u,u} = \mathbf{S} \mathbf{\Lambda} \mathbf{S}^T = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T, \quad \text{For a positive definite symmetric matrix} \quad (7b)$$

$$\mathbf{\Lambda}^+ = \mathbf{\Sigma}^+ = \begin{bmatrix} \mathbf{\Lambda}_r^{-1} & \mathbf{0}_{r,d} \\ \mathbf{0}_{r,d} & \mathbf{0}_{d,d} \end{bmatrix}.$$

Since  $\mathbf{N}_{u,u}$  is a positive definite symmetric matrix in the LS,  $\mathbf{S} = \mathbf{U} = \mathbf{V}$ . If there is no defect in a matrix  $\mathbf{N}$ ,  $\mathbf{N}^{-1} = \mathbf{N}^+$ . Therefore, we can use pseudoinverse safely in any given problem [8, 9, 13–15].

### 2.3. Hard computing

Linearizing from nonlinear functions to their linear form by means of Taylor expansion, a linear equation system is to be handled as Eq. (1). To avoid complicated proofs in the solution of an equation system, the simplified mathematical model can be written in the following (statically rotation invariant [1]) numerical computation form.

$$\mathbf{A}_{n,u} \boldsymbol{\delta}_u = \mathbf{l}_n, \quad \mathbf{P} = \mathbf{I}. \quad (8)$$

Meet two states to solve Eq. (8),  $n \leq u$  and  $n > u$ . The solution for the former state  $n \leq u$  is achieved by means of auxiliary variables vector  $\boldsymbol{\lambda}_n$  which can be defined as  $\boldsymbol{\delta}_u = \mathbf{A}_{u,n}^T \boldsymbol{\lambda}_n$ . In fact, the auxiliary vector  $\boldsymbol{\lambda}_n$  is named as a Lagrange multipliers vector or an eigenvalues vector in a homogenous equations system in which  $\mathbf{l} = \mathbf{0}$  for Eq. (8) [9]. Putting back  $\boldsymbol{\delta} = \mathbf{A}^T \boldsymbol{\lambda}$  into Eq. (8), we compute  $\boldsymbol{\lambda}$  first:

$$\hat{\boldsymbol{\lambda}}_n = \mathbf{Q}_{n,n} \mathbf{l}_n, \quad \mathbf{Q}_{n,n} = (\mathbf{A} \mathbf{A}^T)^+. \quad (9)$$

And then  $\boldsymbol{\delta}$  and its variance-covariance matrix if we know the statistical uncertainty of observations ( $\boldsymbol{\Sigma}_l = \boldsymbol{\Sigma}_y$ ) are calculated by Eq. (10) and the low error propagation, respectively. We can only calculate the variance-covariance matrix of estimations as in Eq. (10) due to  $\hat{\sigma}_0^2 = 0$  and taking  $\boldsymbol{\Sigma}_y = \mathbf{I}$  in the chapter.

$$\hat{\boldsymbol{\delta}} = \mathbf{A}^T \hat{\boldsymbol{\lambda}} = \mathbf{A}^T \mathbf{Q} \mathbf{l}, \quad \boldsymbol{\Sigma}_{\hat{\boldsymbol{\delta}}} = \sigma_0^2 \mathbf{A}^T \mathbf{Q} \mathbf{Q} \mathbf{A}, \quad (10a)$$

$$\hat{\mathbf{x}} = \mathbf{x} + \hat{\boldsymbol{\delta}}, \quad \boldsymbol{\Sigma}_{\hat{\mathbf{x}}} = \boldsymbol{\Sigma}_{\hat{\boldsymbol{\delta}}}, \quad (10b)$$

$$\hat{\mathbf{x}}^T \hat{\mathbf{x}} \mapsto \min. \quad (10c)$$

In the state ( $n \leq u$ ),  $\mathbf{A} \hat{\boldsymbol{\delta}} - \mathbf{l} = \hat{\boldsymbol{\varepsilon}} = \mathbf{0}$  should be provided. If not, continue solution until  $\max(|\hat{\boldsymbol{\delta}}|) \leq \text{thres} = 5e - 12$  (or  $\max(|\hat{\boldsymbol{\varepsilon}}|) \leq \text{thres} = 5e - 12$ ) by taking  $\mathbf{x} = \hat{\mathbf{x}}$  in every iteration step.  $\hat{\mathbf{x}}^T \hat{\mathbf{x}}$  will be the smallest at end of the solution.

Solution to the second state  $n > u$  is a situation encountered in many scientific and engineering problems. Multiplying both sides of Eq. (8) by  $\mathbf{A}_{u,n}^T$  the normal equation system is established and solved with Eq. (11):

$$\hat{\boldsymbol{\delta}}_u = \mathbf{Q} \mathbf{A}^T \mathbf{l}, \quad \mathbf{Q}_{u,u} = (\mathbf{A}^T \mathbf{A})^+, \quad (11a)$$

$$\hat{\mathbf{x}} = \mathbf{x} + \hat{\boldsymbol{\delta}}, \quad \boldsymbol{\Sigma}_{\hat{\mathbf{x}}} = \hat{\sigma}_0^2 \mathbf{Q}, \quad (11b)$$

$$\hat{\sigma}_0^2 = \frac{\hat{\boldsymbol{\varepsilon}}^T \hat{\boldsymbol{\varepsilon}}}{n - r}, \quad \text{A posteriori variance } (r = \text{rank}(\mathbf{A})) \quad (11c)$$

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{A} \hat{\boldsymbol{\delta}} - \mathbf{l}, \quad (11d)$$

$$\hat{\boldsymbol{\varepsilon}}^T \hat{\boldsymbol{\varepsilon}} \mapsto \min. \quad L_2 - \text{norm estimation (Least Square)} \quad (11e)$$

End the solution if the condition ensured is  $\max(|\hat{\boldsymbol{\delta}}|) \leq \text{thres} = 5e - 12$ ; otherwise, continue the iteration with  $\mathbf{x} = \hat{\mathbf{x}}$ .

Relationships between nonlinearity and LS in a multidimensional surface have been shown by Teunissen et al. [1, 2]. The authors argued the relation on some simple examples and gave some analytical solutions for them. But, they highlighted that those types of analytical solutions have not been given for every problem and emphasized that suitable Taylor expansions have been useful to the solution not being transformed into the analytical ones.

### 3. Geometry of a combination of polynomial and trigonometric functions

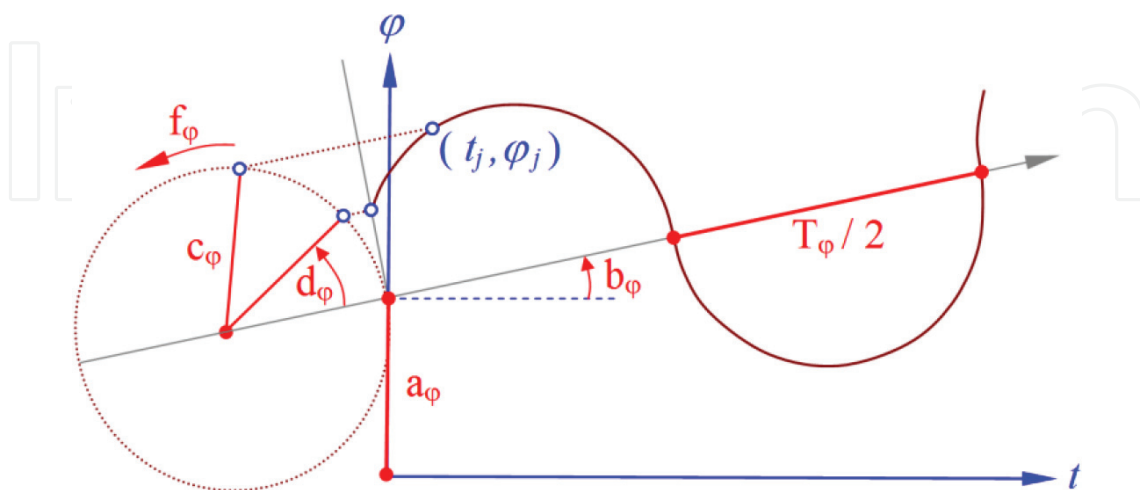
These type functions can be used in defining the orbits of artificial satellites (and celestial bodies). Also, the numerical example part of this chapter, to estimate those type functions, will be inspected and applied on a real example. To foresee a model for any problem we should interpret the model parameter and comprehend the geometry of the model (**Figure 1**).

With respect to independent variable time  $t$ , a combination function of  $p = 1$  degree polynomial and order  $q = 1$  trigonometric function(s) [a combination of polynomial degree and trigonometric order (CPT)] to be estimated in the chapter is:

$$\phi_j = a_\phi + b_\phi t_j + c_\phi \sin(d_\phi + e_\phi t_j), \quad (12)$$

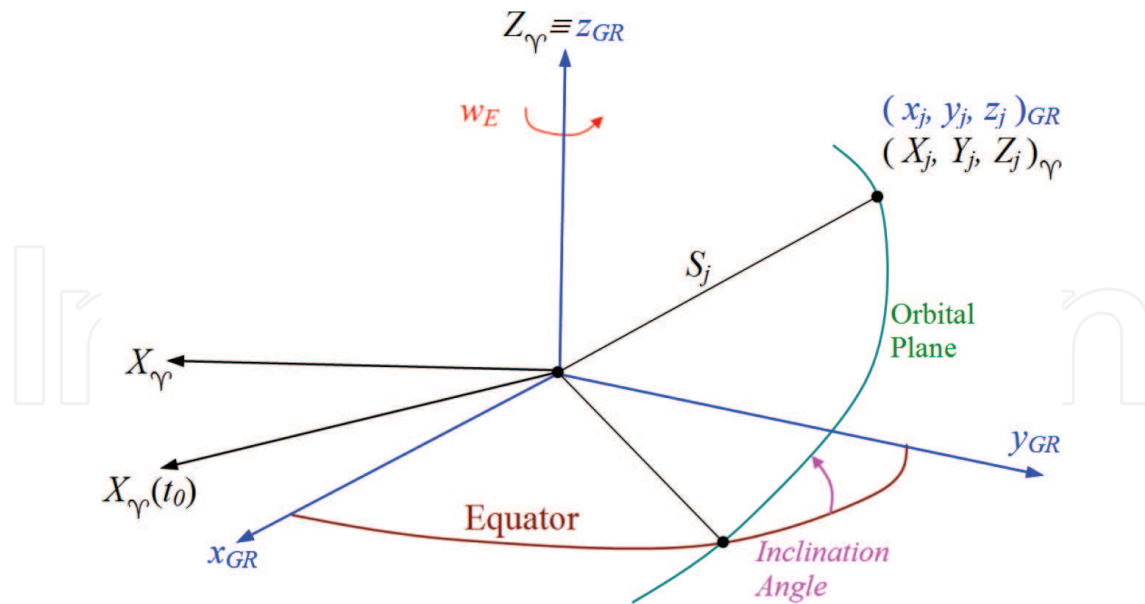
$$\phi_j \in \{X_j, Y_j, Z_j, S_j\}, \quad j \in \{1, 2, \dots, n\}.$$

where  $(t_j, \phi_j)$  are data given. In Eq. (12), translation  $a_\phi$  and slope  $b_\phi$  are elements of a line equation which is a first-order polynomial of CPT function. The other model parameters in the trigonometric part of Eq. (12) are defined as an amplitude  $c_\phi$ , and an initial phase  $d_\phi$  and a frequency (or angular velocity)  $e_\phi = 2\pi/T_\phi$  (a period  $T_\phi$ ) of a wave (**Figure 1**).



**Figure 1.** The geometry of a first-degree and first-order combination of polynomial and trigonometric (CPT) function.





**Figure 2.** Earth (GR) and space-fixed (Y) coordinates for an artificial satellite.

In this chapter, the functions  $\phi_j$  are the coordinate components  $\{X_j, Y_j, Z_j\}$  incoming from a precise orbit file and the geometric distances  $S_j = \sqrt{X_j^2 + Y_j^2 + Z_j^2}$  as a function of the components. However, nonperiodic earth-fixed coordinates (GR) in the SP3 file should be transformed to the periodic space-fixed coordinates (Y); why is Eq. (12) suitable for the space-fixed coordinates, not earth-fixed ones (as seen from **Figure 3** in the numerical example part) (**Figure 2**)?

For this propose, an easy transformation into any epoch (e.g., it can be taken as the first epoch  $t_0$  of the data) is carried out by:

$$\mathbf{X}_{Y,j} = \mathbf{R}_3(\theta_j) \mathbf{x}_{GR,j}, \quad \theta_j = -w_E t_j, \quad (13a)$$

$$\mathbf{x}_{GR,j} = \mathbf{R}_3(-\theta_j) \mathbf{X}_{Y,j}, \quad \mathbf{R}_3(-\theta_j) = \mathbf{R}_3^T(\theta_j) = \mathbf{R}_3^{-1}(\theta_j), \quad (13b)$$

$$\mathbf{R}_3(\theta_j) = \begin{bmatrix} \cos \theta_j & \sin \theta_j & 0 \\ -\sin \theta_j & \cos \theta_j & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{X}_{Y,j} = \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix}_Y, \quad \mathbf{x}_{GR,j} = \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix}_{GR}.$$

where  $w_E$  and  $\mathbf{R}_3$  are in order of the angular velocity of earth and well-known orthogonal rotation matrix around the third axis (**Figure 2**).

A solution of nonlinear Eq. (12) is realized in the following order. Linearizing Eq. (12) by Taylor expansion and omitting the terms greater than or equal to quadratic ones, the linear equation system as given by Eq. (8) is obtained. The explicit form of the Eq. (8) with respect to the approximate values of unknowns for a CPT is:



$$\mathbf{A}_j = \begin{bmatrix} 1 & t_j & \sin(d_0 + f_0 t_j) & c_0 \cos(d_0 + e_0 t_j) & t_j c_0 \cos(d_0 + e_0 t_j) \end{bmatrix}, \quad (14a)$$

$$\mathbf{l}_j = \left[ \phi_j - \{a_0 + b_0 t_j + c_0 \sin(d_0 + e_0 t_j)\} \right], \quad j \in \{1, 2, \dots, n\}. \quad (14b)$$

We can use a recursive solution for Eq. (14) instead of the batch solution as Eq. (11) because of its solution velocity.

$$\hat{\delta} = \mathbf{Q} \left( \sum_{j=1}^n \mathbf{A}_j^T \mathbf{l}_j \right), \quad \mathbf{Q} = \left( \sum_{j=1}^n \mathbf{A}_j^T \mathbf{A}_j \right)^{-1} = \left( \sum_{j=1}^n \mathbf{A}_j^T \mathbf{A}_j \right)^+. \quad (15)$$

Continuation of the solution of Eq. (15) can be performed according to Eq. (11). The model given by Eq. (12) is a simple model to determine the satellite orbit motions. For more complicated models, the readers can utilize [19–25] resources.

## 4. Numerical example

For a nonlinear estimation of CPT functions, some numerical examples are chosen from GNSS {Global navigation satellite systems = GPS (USA) + GLONASS (RU), GALILEO (EU), COMPASS (CHN)} artificial satellite orbits whose coordinates are downloaded from the internet address <ftp://ftp.glonass-iac.ru/MCC/PRODUCTS/17091/final/Sta19426.sp3> [26].

For this purpose, two estimation software have been developed in 64Bit Python (in accordance with the 2.7 and 3.6 version) and 32Bit C++ (Code::Blocks) environments to see the convergence rate of the mathematical model given in Eq. (12) [27, 28]. The computed elements of CPT functions for the selected four satellites R01 (GLONASS), G03 (GPS), E01 (GALILEO) and C06 (COMPASS) are summarized in **Table 1** in which they are ordered from the nearest satellite to the farthest one.

The motions of the CPT functions estimated satellites (in **Table 1**) with respect to earth- (left column of **Figure 3**) and space-fixed (right column of **Figure 3**) coordinate systems are demonstrated in **Figure 3**. Moreover, coherence between the estimated CPT function (black solid line) of the C06 satellite and its data points given (colorful circles) is represented in detail in **Figure 4**.

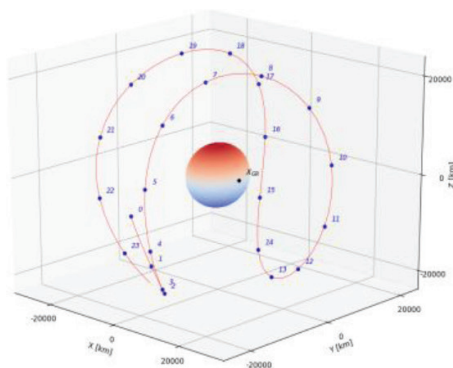
We know that accuracy of precise SP3 file coordinates is about  $\sigma_0 = \pm 5$  cm. If we compare the value with its estimations given in **Table 1**, we can say that our predicted model is not meet our demands. We should expand the model by raising the degree of polynomial part or/and order of trigonometric part of CPT functions. In fact, we can readily see that the projected model with Eq. (12) will never cover the data. The model is only chosen for this chapter. The more suitable model established on Keplerian orbital elements can be found in the orbit determination literature and in [18–20].

Comparing the solution velocities (from the iteration numbers with respect to  $5e - 12$  threshold in **Table 1**) in different platforms, we can say that the solution velocities in 64Bit Python are generally better than 32Bit ones.

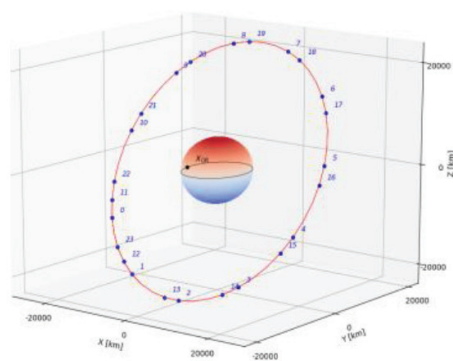
Sat.	$\varphi$	X	Y	Z	Unit	S
R01 (RU)	<i>iter</i>	5	4	4	—	10
	<i>iter++</i>	5	4	5	—	9
	$a_\varphi$	-11.860	2.420	2.771	km	<u>25,508.091</u>
	$b_\varphi$	0.028	0.003	0.052	km/h	0.001
	$c_\varphi$	25,474.503	11,178.453	22,965.361	km	8.127
	$d_\varphi$	-60°54'04.37"	-23°03'56.93"	-30°31'00.32"	deg	42°41'35.87"
	$e_\varphi$	31°57'53.92"	-31°57'44.92"	-31°57'56.31"	deg/h	-31°58'13.55"
	$T_\varphi$	11 <sup>h</sup> 05'44.37"	11 <sup>h</sup> 15'47.54"	11 <sup>h</sup> 15'43.53"	h	11 <sup>h</sup> 15'37.46"
	$\pm \hat{\sigma}_0$	2.902	1.266	2.653	km	0.211
	<i>iter</i>	6	4	4	—	30
	<i>iter++</i>	6	4	4	—	34
	$a_\varphi$	-17.545	7.332	-1.824	km	<u>26,561.324</u>
	$b_\varphi$	-0.132	-0.045	-0.085	km/h	-0.003
	$c_\varphi$	24,800.602	17,959.166	21,757.547	km	14.211
G03 (USA)	$d_\varphi$	66°34'39.73"	-48°06'43.68"	-7°51'47.16"	deg.	-72°24'19.52"
	$e_\varphi$	30°04'56.03"	30°04'52.50"	30°05'00.02"	deg./h	-30°08'11.17"
	$T_\varphi$	11 <sup>h</sup> 58'01.91"	11 <sup>h</sup> 58'3.31"	11 <sup>h</sup> 58'00.32"	h	11 <sup>h</sup> 56'44.42"
	$\pm \hat{\sigma}_0$	4.898	3.648	3.961	km	0.183
	<i>iter</i>	4	5	4	—	32
	<i>iter++</i>	5	5	4	—	93
	$a_\varphi$	1.866	1.358	-4.673	km	<u>29,600.332</u>
	$b_\varphi$	0.052	-0.048	0.004	km/h	-0.000
	$c_\varphi$	21,349.011	26,031.918	24,878.432	km	3.720
	$d_\varphi$	-35°22'15.46"	85°57'22.67"	-16°23'29.47"	deg.	8°29'58.23"
	$e_\varphi$	-25°34'13.06"	-25°34'14.87"	25°34'18.57"	deg./h	-25°42'10.68"
	$T_\varphi$	14 <sup>h</sup> 04'43.81"	14 <sup>h</sup> 04'42.81"	14 <sup>h</sup> 04'40.78"	h	14 <sup>h</sup> 00'22.19"
	$\pm \hat{\sigma}_0$	1.796	1.428	1.058	km	0.236
	<i>iter</i>	7	6	6	—	28
E01 (EU)	<i>iter++</i>	10	7	7	—	29
	$a_\varphi$	-407.495	228.037	266.178	km	<u>42,175.353</u>
	$b_\varphi$	61.945	-6.376	-7.892	km/h	-0.358
	$c_\varphi$	41,716.806	24,832.649	34,235.695	km	227.599
	$d_\varphi$	-24°21'36.29"	59°24'50.73"	67°36'9.42"	deg.	18°32'27.85"
	$e_\varphi$	-15°07'02.75"	-15°02'16.41"	-15°02'36.74"	deg./h	-14°57'17.11"
	$T_\varphi$	23 <sup>h</sup> 48'48.86"	23 <sup>h</sup> 56'22.30"	23 <sup>h</sup> 55'49.94"	h	24 <sup>h</sup> 04'21.41"
	$\pm \hat{\sigma}_0$	56.552	38.018	51.901	km	0.491
	<i>iter</i>	7	6	6	—	28
	<i>iter++</i>	10	7	7	—	29
	$a_\varphi$	-407.495	228.037	266.178	km	<u>42,175.353</u>
	$b_\varphi$	61.945	-6.376	-7.892	km/h	-0.358
	$c_\varphi$	41,716.806	24,832.649	34,235.695	km	227.599
	$d_\varphi$	-24°21'36.29"	59°24'50.73"	67°36'9.42"	deg.	18°32'27.85"
C06 (CHN)	$e_\varphi$	-15°07'02.75"	-15°02'16.41"	-15°02'36.74"	deg./h	-14°57'17.11"
	$T_\varphi$	23 <sup>h</sup> 48'48.86"	23 <sup>h</sup> 56'22.30"	23 <sup>h</sup> 55'49.94"	h	24 <sup>h</sup> 04'21.41"
	$\pm \hat{\sigma}_0$	56.552	38.018	51.901	km	0.491

**Table 1.** Computed elements of the CPT functions for G03, R01, E01, C06 satellites by  $Iter_{MAX} = 1000$  and  $thres = 5e-12$  in loops {iteration numbers of 64Bit Python and 32Bit C++ software in windows are denoted as *iter* and as *iter++* respectively}.

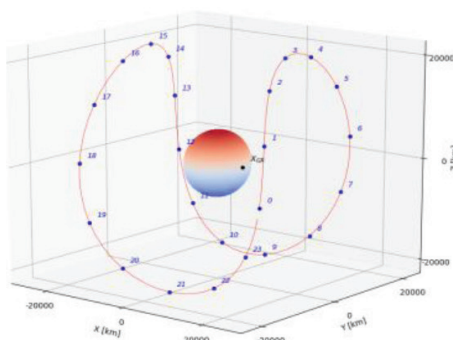
If we chose the threshold as  $51e-13$ , we can see the distinctions of solution convergences between 64Bit and 32Bit running on the estimations of C06 satellite from 1000 (X), 8 (Y), 7 (Z), 1000 (S) in 32Bit C++ and 10 (X), 8 (Y), 6 (Z), 28 (S) in 64Bit Python in Windows. In here, 1000 is the maximum iteration number. If the mathematical model would be more complicated and its data number would be bigger than the number used in the example part, we would see the state more prominently.



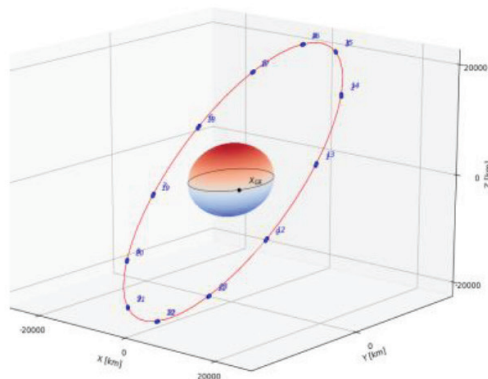
R01  
(a Mid Earth  
Orbit in  
GLONASS)



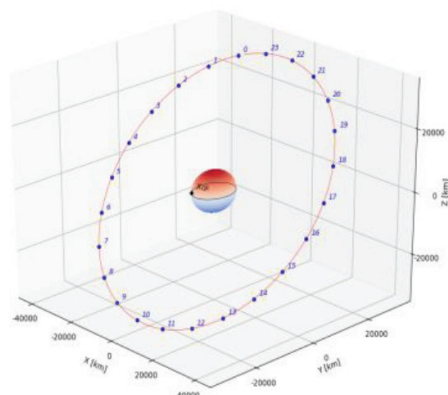
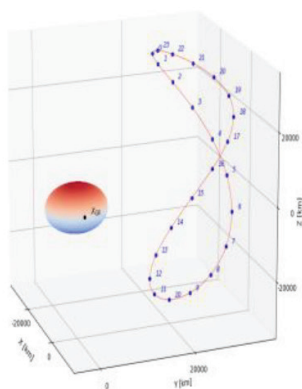
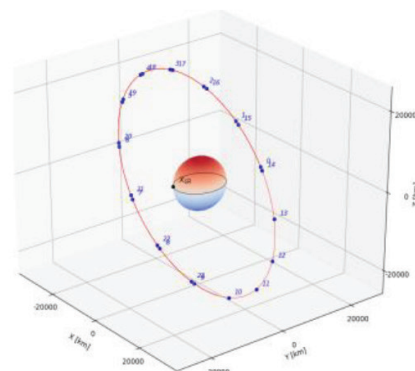
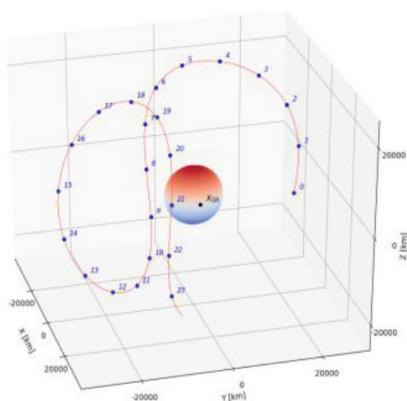
G03  
(a Mid Earth  
Orbit in GPS)



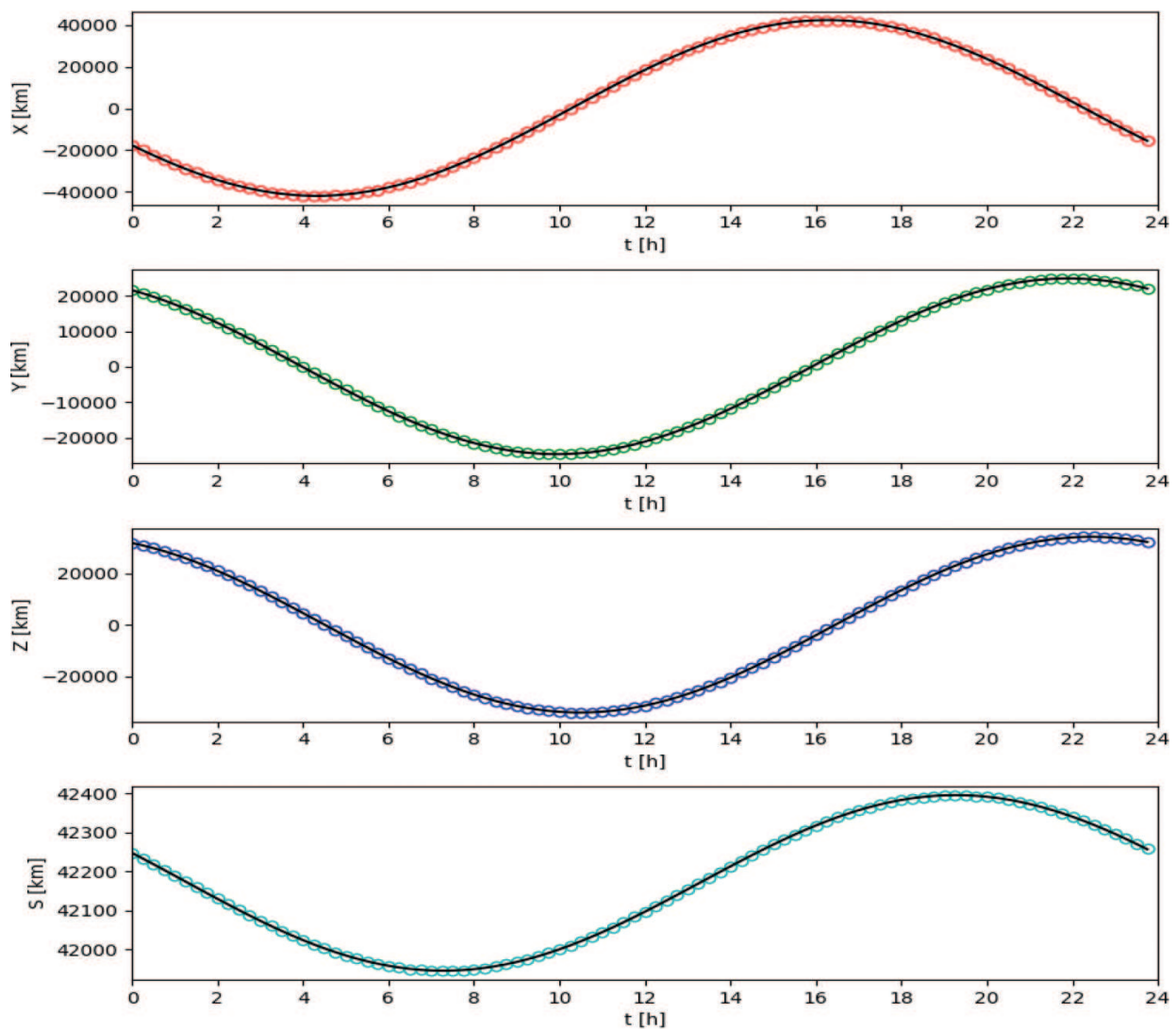
E01  
(a Mid Earth  
Orbit in  
GALILEO)



C06  
(Earth-fixed  
inclined  
satellite in  
COMPASS, a  
Geostationary  
Earth Orbit)



**Figure 3.** Earth (left) and space (right) fixed orbital traces (see the appendix) with time tags of R01, G03, E02, C06 satellites and the motion of X-coordinate axis shown as GR ( $X_{GR}(t_0)$ ) position on the intersection Greenwich meridian and equator) symbol at  $t_0$  (=2017 April 01 00:00:00).



**Figure 4.** Temporal changing of space fixed coordinates of C06. The circles and solid lines represent the data points and estimated functions under LS respectively for X (red), Y (green), Z (blue), S (cyan).

Approximated values and the loop element for the unknowns are computed following the order in all solutions of the satellites when running in 32Bit C++ and 64Bit Python platforms for the estimations in **Table 1**.

$$a_0 = 0.0, b_0 = 0.0, c_0 = \max\{\phi_j\}, \quad j \in \{1, 2, \dots, n = 96\}$$

$$d_0 = \arcsin(\phi_1/c_0), e_0 = \{\arcsin(\phi_2/c_0) - \arcsin(\phi_1/c_0)\}/\{t_2 - t_1\}$$

$$\mathbf{x} = [a_0 \quad b_0 \quad c_0 \quad d_0 \quad e_0]^T$$

Maximum iteration number and threshold loop elements are  $iter_{MAX} = 1000$  and  $thres = 5e - 12$  to break the iteration loop.

Taking the approximated values as  $\mathbf{x} = [0 \ 0 \ thres \ 0 \ 0]^T \approx \mathbf{0}^T$  and the same loop elements given above, the iteration numbers are handled as 138 (X), 178 (Y), 78 (Z), 16 (S) in 64Bit Python. For  $\mathbf{x} = [0 \ 0 \ c_0 \ 0 \ 0]^T$  they are 33 (X), 235 (Y), 96 (Z), 11 (S) in 64Bit Python. Different approximated value selections cause different iteration numbers (namely convergence rate).

#### 4.1. An expanded model example by an auxiliary cosine wave

Since the estimated standard deviations  $\hat{\sigma}_0 = \{\pm 56.552, \pm 38.018, \pm 51.901, \pm 0.491\}$  (for X, Y, Z, S in **Table 1**) of the CPT functions for the coordinates of the C06 satellite are not statically equal to their expected values ( $\sigma_0 = \pm 5$  cm), the CPT model should be expanded. As an example, three more unknowns are added to the model given in Eq. (12)

$$\phi_j = a_\phi + b_\phi t_j + c_\phi \sin(d_\phi + e_\phi t_j) + f_\phi \cos(g_\phi + h_\phi t_j)$$

The added terms represent an amplitude  $f_\phi$ , an initial phase  $g_\phi$  and a frequency  $h_\phi$  of a new wave carried by first (sine) wave. After the first estimation with respect to Eq. (12),

we can choose the approximate values of the new parameters as

$$f_0 = \max(abs(\hat{\epsilon}))$$

$$g_0 = \arcsin(\hat{\epsilon}_1/f_0)$$

$$h_0 = \{\arcsin(\hat{\epsilon}_2/f_0) - \arcsin(\hat{\epsilon}_1/f_0)\}/\{t_2 - t_1\}$$

from  $\hat{\epsilon}_j = [\phi_j - \{\hat{a} + \hat{b} t_j + \hat{c} \sin(\hat{d} + \hat{e} t_j)\}]$ ,  $j \in \{1, 2, \dots, n = 96\}$ . The approximate value vector of the expanded model by a new wave is:

$$\mathbf{x} = [a_0 \ b_0 \ c_0 \ d_0 \ e_0 \ f_0 \ g_0 \ h_0]^T = [\hat{a} \ \hat{b} \ \hat{c} \ \hat{d} \ \hat{e} \ f_0 \ g_0 \ h_0]^T$$

The approximate values are substituted in the following linearized model as initial values for the loop in LS estimation.

$$\mathbf{l}_j = [\phi_j - \{a_0 + b_0 t_j + c_0 \sin(d_0 + e_0 t_j) + e_0 \cos(h_0 + g_0 t_j)\}]$$

$$\mathbf{A}_j^T = \begin{bmatrix} 1 \\ t_j \\ \sin(d_0 + e_0 t_j) \\ c_0 \cos(d_0 + e_0 t_j) \\ t_j c_0 \cos(d_0 + e_0 t_j) \\ \cos(g_0 + h_0 t_j) \\ -f_0 \sin(g_0 + h_0 t_j) \\ -t_j f_0 \sin(g_0 + h_0 t_j) \end{bmatrix}$$

After the evaluation, the improved solution for the C06 satellite is represented in **Table 2**. We can readily see the improvements upon the downs of the standard deviations from  $\hat{\sigma}_0 = \{\pm 56.552, \pm 38.018, \pm 51.901, \pm 0.491\}$  (**Table 1**) into  $\hat{\sigma}_0 = \{\pm 0.178, \pm 0.137, \pm 0.191, \pm 0.003\}$



Sat.	$\varphi$	X	Y	Z	Unit	S
	<i>iter</i>	27	158	53	—	19
C06 (CHN)	$a_\varphi$	223.514	149.296	182.012	km	42,169.956
	$b_\varphi$	1.515	0.224	0.060	km/h	0.055
	$c_\varphi$	42,064.675	66.183	91.527	km	225.161
	$d_\varphi$	$-25^\circ 07' 54.78''$	$11^\circ 15' 24.23''$	$3^\circ 15' 58.76''$	deg	$19^\circ 47' 37.54''$
	$e_\varphi$	$-15^\circ 02' 24.75''$	$30^\circ 04' 30.44''$	$30^\circ 04' 02.09''$	deg/h	$-15^\circ 02' 43.26''$
	$f_\varphi$	110.710	24,808.034	34,217.994	km	0.838
	$g_\varphi$	$5^\circ 08' 57.82''$	$30^\circ 28' 55.61''$	$-22^\circ 21' 14.60''$	deg.	$8^\circ 56' 45.50''$
	$h_\varphi$	$30^\circ 10' 58.43''$	$15^\circ 02' 17.24''$	$-15^\circ 02' 19.27''$	deg./h	$-29^\circ 40' 58.52''$
	$\pm \hat{\sigma}_0$	0.178	0.137	0.191	km	0.003

**Table 2.** The results by the expanded model for the C06 satellite in Python 3.6 in windows.

(Table 2) in kilometers for the C06 satellite. We can develop the last model more by means of the same manner if we want.

As another example, the expanded model has been trained on the coordinate components of R01 GLONASS satellite by means of Python 3.6 software on Windows. We can also see the improvements upon the downs of the standard deviations (*its iteration numbers*) from  $\hat{\sigma}_0 = \{\pm 2.902$  (5),  $\pm 1.266$  (4),  $\pm 2.653$  (4),  $\pm 0.211$  (10)} (Table 1) to  $\hat{\sigma}_0 = \{\pm 0.226$  (76),  $\pm 0.448$  (32),  $\pm 0.201$  (43),  $\pm 0.037$  (57)} in kilometers for the R01 satellite.

Condition numbers computed from a rate of maximum and minimum eigenvalues or a rate of singular values under LS are very effective tools for determining the consistency as well. Therefore, a larger condition number can cause larger iteration number (related to convergence rate), We can see those states from the CPT estimations of the C06 satellite with the iteration (*iter*) and condition numbers (*cond*). These are given for X, Y, Z, S as *iter* = {7, 6, 6, 28} and *cond* = { $3.4e + 13$ ,  $2.0e + 12$ ,  $2.8e + 12$ ,  $1.3e + 09$ } (Table 1), and as *iter* = {27, 158, 53, 19} and *cond* = { $9.5e + 14$ ,  $2.3e + 13$ ,  $3.0e + 13$ ,  $2.6e + 10$ } (Table 2).

## 5. Conclusions

In this chapter, the least squares (LS) estimations of the artificial satellite orbital movements by a combination of polynomial and trigonometric (CPT) functions have been given after a general overview has been made on the hard and soft computations. In practice, the orbital motions are modeled on Keplerian orbital elements. In contrary to this, the coordinate components have been selected for this chapter due to the nonlinear relations of the components and the unknowns which are the elements of CPT functions. The relations cause inconsistencies in the LS solutions. The inconsistencies result from the two injectivity defects, *c-defects* and *i-defects*. We can readily see the defects from the differences of the convergence rates (in other words the iteration numbers) in different computer platforms and architectures as shown in the chapter. The defects are not fully removed as long as not change the mathematic models. However, we can surpass the effects of those defects in part by means of the pseudoinverse based on the eigendecomposition or the singular value decomposition (SVD) as in here. The

surjectivity defect ( $d_s$ ) of the CPT functions not including the datum defects ( $d$ -defects) was eliminated by the LS objective function.

For the sake of simplicity for readers, a simple CPT function has been chosen at first. After the initial estimation of the function, the estimated errors vector has been found. We have seen that the errors have had a periodic characteristic in time. So, a new wave defining the error characteristic and been able to carry by the first wave has been planned for expanding the CPTs. It is shown that we can expand a CPT function until ensuring statically equivalency between a priori and a posteriori variances. For instance, one may secure the equivalency of the variances if one would expand more by a new wave in the last estimated model in the same manner.

The convergence rates (upon the iteration numbers) of the LS estimation have been inspected according to the threshold ( $thres = 5e-12$ ) which is a good value for the estimation of the nonlinear CPT function. An algorithm compiled by different compilers and run in different architectures (with 32 Bit or 64 Bit) changes the convergence rate of the estimations in such as the inconsistent scientific problems. It is also observed that the iteration numbers change when the 64-bit Python software is run on Linux platform which has a different framework than Windows. But, the numbers have not been given in the example part of the chapter. Contrary to inconsistency model, namely in a consistent one, the iteration numbers can take equivalence values in all circumstances. Another way to determine the inconsistency of a model is to obtain its condition number which is computed from a rate of maximum and minimum eigenvalues or of singular values under LS. If the condition number is close to one, the projected model is accepted as a consistent model.

We can use the Soft Computing Methods (SCM) if not an exact mathematical relationship between the data and unknowns. The mathematical model is established by the trial-and-error method in training part of SCM by means of arbitrary weights and activation functions depending on SCM expert forecasts. For the solution of the SCM model during the training, we can use least absolute residuals (LAR) and minmax absolute residuals (MAR) objective functions by the linear programming or the LS estimation as in hard computing method (HCM). In the state, the inconsistency problem can erase whatever the solution method (LAR, MAR or LS) is. The inconstancy can be removed by means of experiences gained from HCMs.

Prior information is very important to select a suitable mathematical model for a scientific problem. For example, comparing a priori variance with a posteriori variance at the end of the estimation is a useful warning to the user to determine the correct mathematical model as seen from the expanded model in the example section of the chapter.

In numerical computation, there are two main phenomena which are the mathematical model (as a combination of functional and stochastic models) and objective function. The solution strategy is of no importance if the same mathematical model and objective function are preferred in the same problem of hard computing. All solution strategies always give same results, only their solution time spans can be distinct from each other (**Table 3**).

## Acknowledgements

I wish to thank TÜRKSAT A.Ş (<https://www.turksat.com.tr/en>) supporting this study.



## Appendix

<i>Epoch(j)</i>	$t_j - t_0$ [h]	$X_j$ [km]	$Y_j$ [km]	$Z_j$ [km]	$t_j$ [μsec]
0	0.00	-17531.307506	21541.792054	31834.209680	691.175390
1	0.25	-19992.147900	20678.913485	30924.464057	691.487182
2	0.50	-22367.333395	19727.437365	29882.214843	691.798804
3	0.75	-24646.589754	18691.353300	28711.797065	692.110527
4	1.00	-26820.032098	17575.021170	27418.102912	692.422088
5	1.25	-28878.208475	16383.154007	26006.563346	692.733923
6	1.50	-30812.142166	15120.799311	24483.127129	693.045689
7	1.75	-32613.372313	13793.318783	22854.237453	693.357416
8	2.00	-34273.992670	12406.366506	21126.806228	693.668853
9	2.25	-35786.688260	10965.865676	19308.186097	693.980453
10	2.50	-37144.769753	9477.983947	17406.140275	694.291853
11	2.75	-38342.205344	7949.107471	15428.810302	694.603337
12	3.00	-39373.649964	6385.813755	13384.681845	694.914789
13	3.25	-40234.471624	4794.843425	11282.548651	695.226157
14	3.50	-40920.774721	3183.071022	9131.474825	695.537828
15	3.75	-41429.420160	1557.474971	6940.755568	695.849062
16	4.00	-41758.042128	-74.893170	4719.876568	696.160359
17	4.25	-41905.061383	-1706.940006	2478.472204	696.471627
18	4.50	-41869.694990	-3331.562017	226.282782	696.782836
19	4.75	-41651.962330	-4941.677516	-2026.889008	697.094239
20	5.00	-41252.687387	-6530.258698	-4271.222183	697.405373
21	5.25	-40673.497209	-8090.363611	-6496.921900	697.716659
22	5.50	-39916.816531	-9615.167884	-8694.263981	698.028206
23	5.75	-38985.858544	-11097.996030	-10853.639319	698.339619
24	6.00	-37884.611845	-12532.352178	-12965.597898	698.651055
25	6.25	-36617.823589	-13911.950035	-15020.892223	698.962566
26	6.50	-35190.978917	-15230.741942	-17010.519896	699.273893
27	6.75	-33610.276761	-16482.946839	-18925.765115	699.585363
28	7.00	-31882.602129	-17663.077000	-20758.238873	699.896674
29	7.25	-30015.495009	-18765.963379	-22499.917620	700.208082
30	7.50	-28017.116067	-19786.779434	-24143.180195	700.519378
31	7.75	-25896.209299	-20721.063292	-25680.842806	700.830889
32	8.00	-23662.061860	-21564.738140	-27106.191889	701.142117
33	8.25	-21324.461276	-22314.130731	-28413.014646	701.453707

<i>Epoch(j)</i>	$t_j - t_0$ [h]	$X_j$ [km]	$Y_j$ [km]	$Z_j$ [km]	$t_j$ [μsec]
34	8.50	-18893.650283	-22965.987908	-29595.627135	701.764972
35	8.75	-16380.279535	-23517.491071	-30648.899728	702.076478
36	9.00	-13795.358449	-23966.268499	-31568.279851	702.387737
37	9.25	-11150.204459	-24310.405495	-32349.811865	702.699185
38	9.50	-8456.390949	-24548.452300	-32990.154024	703.010540
39	9.75	-5725.694161	-24679.429736	-33486.592421	703.321863
40	10.00	-2970.039352	-24702.832610	-33837.051887	703.633117
41	10.25	-201.446497	-24618.630817	-34040.103800	703.944417
42	10.50	2568.024186	-24427.268222	-34094.970801	704.255873
43	10.75	5326.326597	-24129.659289	-34001.528424	704.567076
44	11.00	8061.482740	-23727.183555	-33760.303648	704.878433
45	11.25	10761.636054	-23221.677952	-33372.470443	705.189684
46	11.50	13415.103835	-22615.427078	-32839.842340	705.501070
47	11.75	16010.428511	-21911.151470	-32164.862120	705.812547
48	12.00	18536.427516	-21111.993965	-31350.588697	706.124190
49	12.25	20982.241560	-20221.504257	-30400.681303	706.435990
50	12.50	23337.381077	-19243.621724	-29319.381091	706.747258
51	12.75	25591.770667	-18182.656653	-28111.490271	707.058650
52	13.00	27735.791345	-17043.269964	-26782.348927	707.370038
53	13.25	29760.320445	-15830.451549	-25337.809640	707.681516
54	13.50	31656.769034	-14549.497344	-23784.210083	707.992915
55	13.75	33417.116707	-13205.985271	-22128.343727	708.304435
56	14.00	35033.943650	-11805.750146	-20377.428827	708.615955
57	14.25	36500.459878	-10354.857705	-18539.075847	708.927258
58	14.50	37810.531573	-8859.577863	-16621.253481	709.238773
59	14.75	38958.704447	-7326.357314	-14632.253449	709.549980
60	15.00	39940.224086	-5761.791632	-12580.654219	709.861492
61	15.25	40751.053248	-4172.596961	-10475.283826	710.172943
62	15.50	41387.886081	-2565.581420	-8325.181952	710.484439
63	15.75	41848.159196	-947.616367	-6139.561422	710.796071
64	16.00	42130.059795	674.392412	-3927.769279	711.107495
65	16.25	42232.530676	2293.533436	-1699.247596	711.419275
66	16.50	42155.272132	3902.918094	536.505839	711.730566
67	16.75	41898.740894	5495.708947	2769.976807	712.042086
68	17.00	41464.146141	7065.147654	4991.673762	712.353663
69	17.25	40853.442432	8604.582424	7192.166646	712.665021

<i>Epoch(j)</i>	$t_j - t_0$ [h]	$X_j$ [km]	$Y_j$ [km]	$Z_j$ [km]	$t_j$ [μsec]
70	17.50	40069.319938	10107.494927	9362.125250	712.976412
71	17.75	39115.191840	11567.526548	11492.356997	713.287987
72	18.00	37995.179006	12978.503915	13573.844046	713.599417
73	18.25	36714.092016	14334.463621	15597.779572	713.910828
74	18.50	35277.410608	15629.676058	17555.603122	714.221952
75	18.75	33691.260665	16858.668299	19439.034930	714.533584
76	19.00	31962.388796	18016.245942	21240.109087	714.845281
77	19.25	30098.134631	19097.513875	22951.205467	715.156764
78	19.50	28106.400907	20097.895874	24565.080300	715.468390
79	19.75	25995.621474	21013.152993	26074.895308	715.779824
80	20.00	23774.727299	21839.400671	27474.245306	716.091334
81	20.25	21453.110591	22573.124521	28757.184187	716.402951
82	20.50	19040.587166	23211.194732	29918.249189	716.714457
83	20.75	16547.357150	23750.879046	30952.483389	717.026029
84	21.00	13983.964160	24189.854253	31855.456321	717.337474
85	21.25	11361.253075	24526.216175	32623.282652	717.649172
86	21.50	8690.326544	24758.488074	33252.638847	717.960633
87	21.75	5982.500334	24885.627475	33740.777748	718.272113
88	22.00	3249.257698	24907.031342	34085.541002	718.583548
89	22.25	502.202871	24822.539587	34285.369282	718.895203
90	22.50	-2246.986126	24632.436884	34339.310236	719.206528
91	22.75	-4986.605239	24337.452758	34247.024111	719.518326
92	23.00	-7704.972522	23938.759921	34008.787012	719.830016
93	23.25	-10390.476491	23437.970866	33625.491749	720.141620
94	23.50	-13031.624539	22837.132665	33098.646226	720.452998
95	23.75	-15617.091194	22138.719998	32430.369356	720.764247

**Table 3.** Space Fixed Coordinates of C06 inclined geostationary earth orbit in COMPASS (which is Chinese Global Positioning Satellite System) are transformed with respect to  $t_0$  from earth fixed coordinates downloaded from <ftp://ftp.glonass-iac.ru/MCC/PRODUCTS/17091/final/Sta19426.sp3> [26]  $\{t_0 = 2017.04.01-00:00:00$  (Civil Calendar) = 1942–518,400 (GPS week—week seconds)).

## Author details

Orhan Kurt

Address all correspondence to: [orhnkrt@gmail.com](mailto:orhnkrt@gmail.com)

Geomatics Engineering Department, Kocaeli University, Kocaeli, Turkey

## References

- [1] Teunissen PJG. The Geometry of Geodetic Inverse Linear Mapping and Non-linear Adjustment, Publications on Geodesy, New Series. Vol. 8, No. 1. Delft: Netherlands Geodetic Commission; 1985. p. 177
- [2] Teunissen PJG, Knickmeyer EH. Nonlinearity and least squares. CISM Journal ACSGC. 1988;42(4):321-330
- [3] Krakiwsky EJ. A Synthesis of Recent Advances in the Method of Least Squares, Lecture Notes 42, Department of Geo\_desy and Geomatics Engineering. Canada: University of New Brunswick; 1975
- [4] Mikhail EM. Observations and Least Squares. IEP Series Civil Engineering. New York: Thomas Y. Crowell Company; 1976. ISBN: 0-7002-2481-5
- [5] Cross PA. Advanced least squares applied to positioning-fixing, working paper No. 6, University of East London; 1994. ISSN: 0260-9142
- [6] Leick A. GPS Satellite Surveying. 2nd ed. A Wiley-Inter-science Publication, New York; 1995. ISBN-10: 0-471-30626-6
- [7] Gibbs B. Advanced Kalman Filtering, Least-Squares and Modeling: A Practical Handbook. John Wiley & Sons Inc.; 2011. ISBN 978-0-470-52970-6 (cloth)
- [8] Strang G, Borre K. Linear Algebra, Geodesy and GPS. Wellesley College; 1997. ISBN: 9780961408862, 0961408863
- [9] Koch KR. Parameter Estimation and Hypothesis Testing in Linear Models: Second, Updated and Enlarged Edition. Springer-Verlag Berlin/Heidelberg; 1999. ISBN 978-3-642-08461-4
- [10] Farebrother RW. L1-Norm and  $L_\infty$ -Norm Estimation, An Introduction to the Least Absolute Residuals, the Minimax Absolute Residual and Related Fitting Procedures. Heidelberg/New York/Dordrecht/London: Springer; 2013. DOI: 10.1007/978-3-642-36300-9
- [11] Barrodale I, Ve Roberts FDK. An improved algorithm for discrete L1-Norm. SIAM Journal of Numerical Analysis. 1973;10(5):839-848
- [12] Delikaraoglou D. Estimability analyses of the free networks of differential range observations to GPS satellites. In: Grafarend EW, Sanso F, editors. Optimization and Design Geodetic Network. Springer Verlag; 1985. pp. 196-220, 606. ISBN 3-540-15739-5
- [13] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical Recipes in C: The Art of Scientific Computing, 2nd ed. Cambridge University Press: Cambridge; 2002. ISBN 0-521-43108-5
- [14] Kiusalaas J. Numerical Methods in Engineering with Python 3. Cambridge/New York/Melbourne/Madrid/Cape Town/Singapore/Sao Paulo/Delhi/Mexico City: Cambridge University Press; 2013. ISBN 978-1-107-03385-6 Hardback

- [15] Kurt O. An integrated solution for reducing ill-conditioning and testing the results in non-linear 3D similarity transformations; 2017. <http://www.tandfonline.com/doi/full/10.1080/17415977.2017.1337762>
- [16] Furuta H, Nakatsu K, Hattori H. Applications of soft computing in engineering problems. In: Gregorio Romero Rey, Luisa Martinez Muneta, editors. *Modelling Simulation and Optimization*. InTechRijeka, Croatia; 2010. ISBN: 978-953-307-048-3. Available online at <http://www.intechopen.com/books/modelling-simulationand-optimization/applications-of-soft-computing-in-engineering-problems>
- [17] Kurhe AB, Satonkar SS, Khanale PB, Ashok S. *Soft Computing and Its Applications*. BIOINFO Soft Computing. 2011;1(1):05-07 Available online at <http://www.bioinfo.in/contents.php?id=304>
- [18] Jain AK, Mao J, Mohiuddin KM. Artificial Neural Networks: A Tutorial, Computer. IEEE; 1996. pp. 31-44 0018-9162/96/1996. [http://www.cogsci.ucsd.edu/~ajyu/Teaching/Cogs202\\_sp12/Readings/jain\\_ann96.pdf](http://www.cogsci.ucsd.edu/~ajyu/Teaching/Cogs202_sp12/Readings/jain_ann96.pdf)
- [19] Montenbruck O, Gill E. *Satellite Orbits: Models, Methods, and Applications*. Berlin Heidelberg: Springer-Verlag; 2001. ISBN 3-540-67280-X
- [20] Kaula WM. *Theory of Satellite Geodesy, Applications of Satellites to Geodesy*. Blaisdell Publishing Company, A Division of GINN and Company. Library of Congress Catalog Card Number: 65-14571; 1966
- [21] Vallado DA. *Fundamentals of Astrodynamics and Applications*. McGraw-Hill; 1997. ISBN: 0-07-066829-9
- [22] Seeber G. *Satellite Geodesy*, 2nd ed. Berlin/New York: Walter de Gruyter; 2003. ISBN: 9783110175493, 3-11-017549-5
- [23] Xiaofeng FU, Meiping WU. Optimal design of broadcast ephemeris parameters for a navigation satellite system. *GPS Solution*. 2012;16(4):439-448
- [24] Xu G, Xu J. *Orbits, 2nd Order Singularity-Free Solutions*. 2nd ed. Berlin/Heidelberg: Springer-Verlag; 2013. DOI: 10.1007/978-3-642-32793-3
- [25] Capderou M. *Handbook of Satellite Orbits*. Cham/Heidelberg/New York/Dordrecht/London: Springer; 2014. DOI: 10.1007/978-3-319-03416-4
- [26] IAC Precise Ephemeris File. <http://ftp.glonass-iac.ru/MCC/PRODUCTS/17091/final/Sta19426.sp3> [Accessed: January 01, 2018]
- [27] Code::Blocks Homepage. <http://www.codeblocks.org/>; 2016 [Accessed: January 28, 2016]
- [28] Python. <https://www.python.org/> [Accessed: January 04, 2017]

