



Article

An Improved Parameter Estimation Method for High-Efficiency Multi-GNSS-Integrated Orbit Determination

Xingyuan Yan ^{1,2} , Chenchen Liu ^{3,*}, Meng Yang ^{1,2}, Wei Feng ^{1,2} and Min Zhong ^{1,2}

¹ School of Geospatial Engineering and Science, Sun Yat-sen University, Zhuhai 519082, China; yanxy35@mail.sysu.edu.cn (X.Y.)

² Key Laboratory of Natural Resources Monitoring in Tropical and Subtropical Area of South China, Ministry of Natural Resources, Guangzhou 510300, China

³ School of Artificial Intelligence, Shenzhen Polytechnic, Shenzhen 518055, China

* Correspondence: liuchen@szpt.edu.cn

Abstract: The increased number of satellites and stations leads to the serious time consumption of the integrated precise orbit determination (POD), especially in the current global navigation satellite system (GNSS) with more than 120 satellites. To improve the computational efficiency of multi-GNSS-integrated POD, this paper proposed an improved parameter estimation method based on intel oneAPI high-performance computing, where the inactive parameters are eliminated in a batch mode. Compared with the classical estimation method based on the “one-by-one” elimination, the efficiencies were significantly improved with ratios of 2.53, 4.21, and 5.38 for 79, 126, and 171 stations’ GPS/BDS/Galileo/GLONASS-integrated POD, respectively. The elapsed time of the improved method by using 126 stations was the same as that of 79 stations’ POD by the classical estimation method. In terms of precision, the one-dimensional root mean square error (RMS) reductions were 0.1 cm (7%), 34.3 cm (11%), 1.9 cm (18%), 0.4 cm (8%), 0.2 cm (13%), and 0.4 cm (13%) for GPS, BDS GEO, BDS IGSO, BDS MEO, Galileo, and GLONASS satellites, respectively.

Keywords: precise orbit determination; global navigation satellite system; parameter elimination; time consumption; Intel oneAPI high-performance computing



Citation: Yan, X.; Liu, C.; Yang, M.; Feng, W.; Zhong, M. An Improved Parameter Estimation Method for High-Efficiency Multi-GNSS-Integrated Orbit Determination. *Remote Sens.* **2023**, *15*, 2635. <https://doi.org/10.3390/rs15102635>

Academic Editors: Yunbin Yuan and Baocheng Zhang

Received: 26 April 2023

Revised: 15 May 2023

Accepted: 15 May 2023

Published: 18 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multiple global navigation satellite systems (multi-GNSS) can improve reliability and reduce convergence time for precise point positioning (PPP) [1–3]. GNSS satellite orbit and clock offsets generated by precise orbit determination (POD) are prerequisite products of PPP, and their accuracy directly affects PPP performance [4–6]. To better provide multi-GNSS precise services, a pilot project named a multi-GNSS experiment (MGEX) of the International GNSS Service (IGS) [7] deploys monitoring stations around the world for navigation satellites’ POD, and the number of these stations is increasing year by year [5,8]. In such high-volume data processing, more accurate and reliable orbit and clock offset products can be obtained [9–11], but this further increases the burden of multi-GNSS POD data processing. In addition, POD has been widely used for precise modeling of satellite errors, such as antenna phase center correction [12–14], and empirical solar radiation pressure modeling [15–17], which usually requires POD data processing for one year or even several years. Thus, higher efficiency data processing can achieve higher precision by adding more stations’ observations and can reduce the period of the entire data processing tasks.

Parameters to be estimated include station coordinates, satellite orbits, station and satellite clock offsets, zenith total delay (ZTD), and carrier-phase ambiguities for POD. These large dimensional matrix operations greatly limit the speed of POD data processing while all parameters are kept to solve the normal equation system (NEQ) [18]. A “one-by-one” parameter-elimination method for processing huge GNSS networks was developed by

Ge et al. [18] to improve computational efficiency, where only active parameters were kept, and others were eliminated one by one as soon as they became inactive. With the addition of new constellations of the BDS-2 regional system and early Galileo, before 2018, there were about 80 satellites in multi-GNSS [5,19,20]. Although the one-by-one elimination is adopted, the computational efficiency is still seriously affected due to the expansion of NEQ after the addition of new satellite parameters. An alternative method was used to divide the joint POD into three steps [20]. In addition to GNSS POD, high-efficiency Leo-Earth Orbit (LEO) POD was also divided into two steps, where GNSS POD was obtained by ground stations and then GNSS PPP for LEO [21–23]. However, in theory, a more rigorous “one-step” integrated solution can provide better GNSS and LEO products [21,22].

On the other hand, parallel techniques are also used to improve the efficiency of GNSS solutions. Distributed memory computing techniques have been used for processing massive GPS network datasets [24,25]. Some parallel processing strategies have been developed to speed up epoch-wise PPP or baseline solutions [26,27]. Moreover, Open Multiprocessing (OpenMP) has emerged to support multi-platform shared memory multiprocessing programming [28,29], and OpenMP-based parallelism has been introduced into the extended Kalman filter for real-time GPS network solutions [30]. For multi-GNSS POD, a new parallel elimination of the inactive parameters was realized for improving the efficiency of multi-GNSS POD [31]. Parallelism is always used to improve real-time data processing efficiency; however, it sacrificed CPU and memory usage, and it was confirmed that the architecture of computers entirely limits the performance of the parallel algorithm [31]. Thus, these parallel algorithms are not fully applicable to improve the efficiency of the postprocessing tasks, especially for the completion of the later global navigation system construction of BDS-3 and Galileo [8,32], and about 40~50 newly launched satellites were added to the routine POD. In this case, the number of satellites has increased by about 45%, and this will bring a new challenge to multi-GNSS-integrated POD.

Intel released a unified programming model named oneAPI that simplifies the development process of heterogeneous computing in different architectures and maximizes performance to meet the needs of different workloads. As two important toolkits in oneAPI, the oneAPI HPC and oneAPI Base can provide performant science applications for developers [33]. Therefore, to retain the advantages of serial programming with small CPU and memory consumption, an improved parameter estimation method based on the oneAPI HPC was proposed for high-efficient multi-GNSS-integrated POD, while combining multi-day independent POD parallel to further improve the overall efficiency of the long-time period data processing tasks. First, the main processing of GNSS POD was introduced, and then the major time-consuming procedures were obtained. Next, the improved estimation method was designed based on serial programming of the parameter batch elimination. Afterward, the performance of the classical estimation methods based on the “one-by-one” elimination and multi-threaded parallel elimination and the improved method were compared. Finally, the efficiency of the improved method was verified by different stations and satellites, and the accuracy and time consumption of GPS/BDS/Galileo/GLONASS-integrated POD were analyzed.

2. GNSS Precise Orbit Determination

The reduced dynamic orbit determination method is widely used for GNSS satellites, where the initial condition state parameters of the position and velocity vectors and solar radiation pressure empirical model parameters were estimated by ground observation. In this section, the GNSS observation model, the main process of POD, and the elapsed time decomposition in the least squares adjustment (LSQ) will be introduced.

2.1. GNSS Observation Model

The pseudo-range and carrier-phase equations of the GNSS satellite are as follows [34–36]:

$$\begin{cases} p_i^{j,G} = \rho_i^{j,G} - c\delta t^{j,G} + c\delta t_{i,G} + M_i^{j,G} \cdot T_i + \delta I_i^{j,G} + \epsilon_i^{j,G} \\ p_i^{j,C} = \rho_i^{j,C} - c\delta t^{j,C} + c\delta t_{i,G} + ISB_{i,C_G} + M_i^{j,C} \cdot T_i + \delta I_i^{j,G} + \epsilon_i^{j,C} \\ p_i^{j,E} = \rho_i^{j,E} - c\delta t^{j,E} + c\delta t_{i,G} + ISB_{i,E_G} + M_i^{j,E} \cdot T_i + \delta I_i^{j,G} + \epsilon_i^{j,E} \\ p_i^{j,R} = \rho_i^{j,R} - c\delta t^{j,R} + c\delta t_{i,G} + IFB_{i,R_G}^{j,R} + M_i^{j,R} \cdot T_i + \delta I_i^{j,G} + \epsilon_i^{j,R} \end{cases} \quad (1)$$

$$\begin{cases} \lambda^{j,G} \cdot \varphi_i^{j,G} = \rho_i^{j,G} - c\delta t^{j,G} + c\delta t_{i,G} + M_i^{j,G} \cdot T_i - \delta I_i^{j,G} + \lambda^{j,G} \cdot N_i^{j,G} + \epsilon_i^{j,G} \\ \lambda^{j,C} \cdot \varphi_i^{j,C} = \rho_i^{j,C} - c\delta t^{j,C} + c\delta t_{i,G} + ISB_{i,C_G} + M_i^{j,C} \cdot T_i - \delta I_i^{j,C} + \lambda^{j,C} \cdot N_i^{j,C} + \epsilon_i^{j,C} \\ \lambda^{j,E} \cdot \varphi_i^{j,E} = \rho_i^{j,E} - c\delta t^{j,E} + c\delta t_{i,G} + ISB_{i,E_G} + M_i^{j,E} \cdot T_i - \delta I_i^{j,E} + \lambda^{j,E} \cdot N_i^{j,E} + \epsilon_i^{j,E} \\ \lambda^{j,R} \cdot \varphi_i^{j,R} = \rho_i^{j,R} - c\delta t^{j,R} + c\delta t_{i,G} + IFB_{i,R_G}^{j,R} + M_i^{j,R} \cdot T_i - \delta I_i^{j,R} + \lambda^{j,R} \cdot N_i^{j,R} + \epsilon_i^{j,R} \end{cases} \quad (2)$$

where i and j represent the station and satellite, respectively, and the right superscript of G , C , E , and R represents GPS, BDS, Galileo, and GLONASS; p_i^j and φ_i^j are the raw pseudo-range and carrier phase observations, and λ^j is the wavelength of carrier phase; ρ_i^j is the geometry distance between i and j ; $c\delta t^{j,G}$ is satellite clock offset; the receiver clock offsets of the other systems are referenced to the GPS ($c\delta t_{i,G}$), and aligned to their own systems by introducing ISB_{i,C_G} , ISB_{i,E_G} , and $IFB_{i,R_G}^{j,R}$ for BDS, Galileo and GLONASS, respectively; T_i is the zenith tropospheric delay of station i , and M_i^j is the mapping function of signal direction from satellite j ; δI_i^j is the equivalent distance caused by the ionospheric delay; $\lambda^j \cdot N_i^j$ is the equivalent distance of the ambiguity; ϵ_i^j and ϵ_i^j are the noise of pseudo-range and carrier phase, respectively. The first-order term of ionospheric delay can be eliminated by the ionospheric-free (IF) linear combination of the two different frequency observations in the GNSS.

2.2. Main Processes of POD

The main process of POD is shown in Figure 1. The initial orbit and partial derivatives can be obtained by a numerical integrator, where the initial condition state parameters and the perturbation accelerations during the satellite motion were considered, and some external tables are used to support integration, such as pole bulletin (finals2000A.data), earth gravity field model (EGM2008), JPL DE405 planetary ephemeris, and satellite metadata. The empirical parameter set (ECOM1 or ECOM2) is used to compensate for the solar radiation pressure perturbation [37,38].

Afterward, GNSS observations of satellite-to-ground distance from the station network are used to estimate corrections of the initial condition state parameters. To obtain clean observations, the turboedit algorithm is used to realize observation data preprocessing and cycle slip detection of carrier-phase observations [39]. Combined with the initial orbit, partial derivatives, and GNSS observations, the LSQ adjustment is established. Next, residuals editing is performed according to posterior analysis, and smaller outliers and cycle slips can be detected in this processing. When all cycle slips are detected and estimated as new ambiguities in LSQ, the fixed solution of POD can be obtained by applying the constraints of fixed double-differenced ambiguities to the NEQ system [40].

In a POD process, the LSQ usually requires four iterations for different purposes, i.e., three for residuals editing and the last for ambiguity fixing. The running time of other core processes in POD was counted: parallel data preprocessing took about 4 min for 168 stations; orbit integrating, residuals editing, and ambiguity fixing of 120 satellites were about 2.6, 2.6, and 5.4 min, respectively. However, the time consumption of the LSQ can reach several hours, and the internal processes should be analyzed.

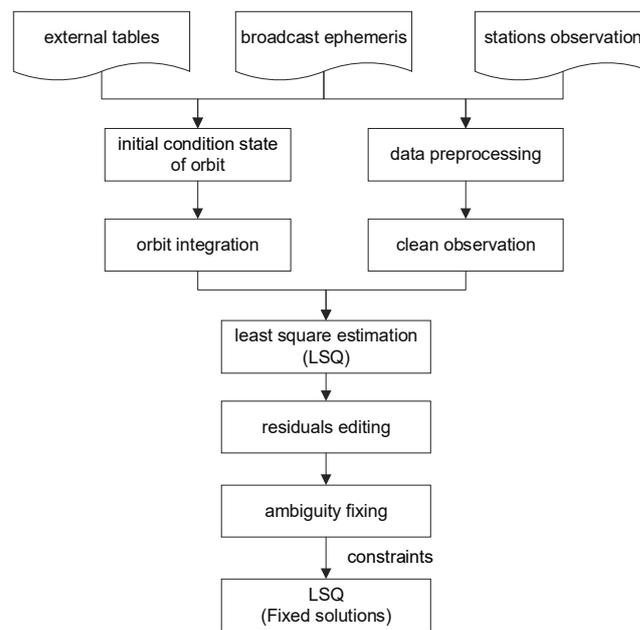


Figure 1. Process of GNSS precise orbit determination.

Processes in LSQ are decomposed as shown in Figure 2. First, observation equations are constructed, including calculating pre-fit residual vector l in Equation (3) and the partial derivative for design matrix A in Equation (3). Second, the NEQ system is cumulated by observation equations. Next, the inactive parameters in the NEQ system are eliminated, and the index of the NEQ system and local parameters of each station should be updated in time. An epoch-to-epoch cycle from parts one to three is executed, while all observations are processed. Finally, the NEQ system is solved to obtain the estimations of orbit, earth rotation parameter (ERP), inter-system bias (ISB) or inter-frequency bias (IFB) [41,42], station coordinate and other parameters, and the eliminated parameters in the third step are recovered, and the residuals from observations are calculated for subsequent residuals editing.

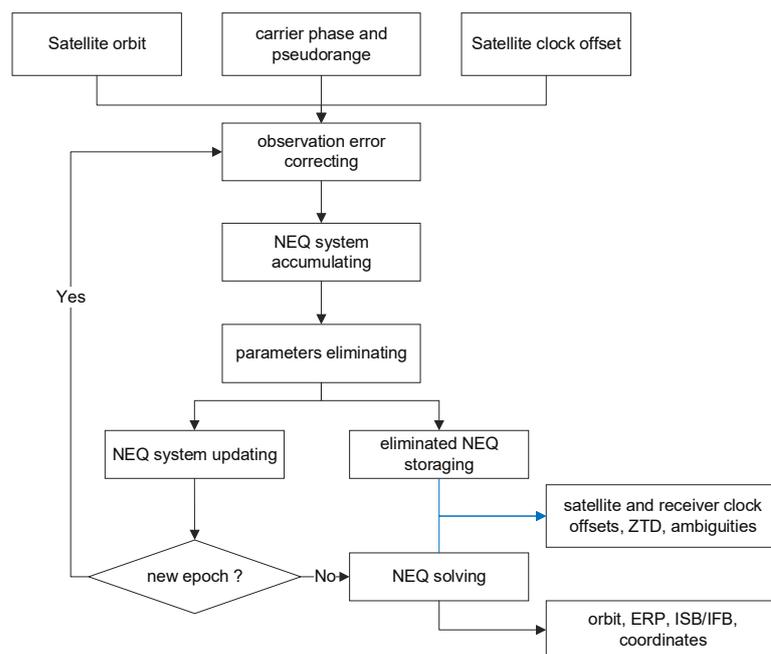


Figure 2. LSQ process chart. The blue arrow is performed after solving NEQ.

2.3. Elapsed Time Decomposition in LSQ

A POD with 96 satellites and 133 stations was carried out for time consumption analysis. The elapsed times of observation modeling, accumulating NEQ, and eliminating parameters were about 2.211, 0.2, and 22 s per epoch, shown in Figure 3a–c, respectively. The number of eliminated inactive parameters for each epoch is shown in Figure 3d, which increased significantly due to the ZTD parameters estimated with an interval of 2 h. In this case, the elapsed time for elimination can reach 40 s per epoch.

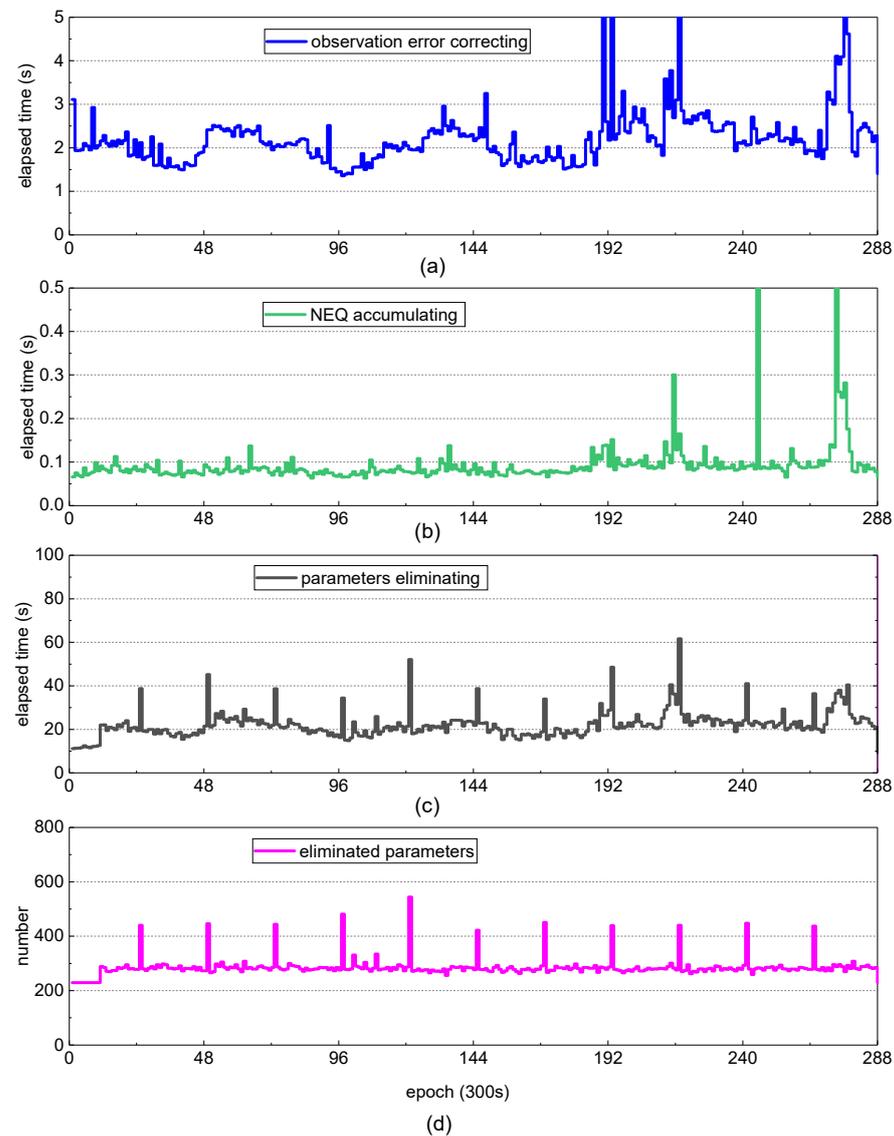


Figure 3. Elapsed time decomposition in LSQ: (a) the elapsed time for error correcting, (b) the elapsed time for error correcting for accumulating NEQ, (c) the elapsed time for inactive parameters eliminating, and (d) the number of inactive parameters.

3. Improved Parameter Estimation Method

Assuming the observation equations are as follows:

$$v = A\hat{X} + l, P \quad (3)$$

$$v_{X_0} = X - X_0 = \hat{X}, P_{X_0} \quad (4)$$

where A is the design matrix, \hat{X} is the vector of the estimated parameters, v is the residual vector, l is the observed minus computed distance (O–C) vector, P is the weight matrix,

and X_0 and P_{X_0} are a priori values of X and their weight matrix, respectively. Then, the corresponding normal equations (NEQ) are

$$(A^T P A + P_{X_0}) \hat{X} = -A^T P l \tag{5}$$

$$N \hat{X} = -W \tag{6}$$

It is worth mentioning that the number of parameters involved in each station–satellite pair is significantly less than the number of global parameters \hat{X} . Therefore, the design matrix A is a sparse matrix with many zero elements. To improve the efficiency, the NEQ of $A^T P A$ is usually calculated by multiplying the non-zero elements of matrix A , instead of direct matrix multiplication. In this process, it is necessary to establish the index information of the local station to the global parameter \hat{X} . The criterion of adjustment is $v^T P v + v_{X_0}^T P_{X_0} v_{X_0} \rightarrow \min$, and it can be expanded into the following Equation:

$$\begin{aligned} v^T P v + v_{X_0}^T P_{X_0} v_{X_0} &= (A \hat{X} + l)^T P (A \hat{X} + l) + \hat{X}^T P_{X_0} \hat{X} = (\hat{X}^T A^T P + l^T P) (A \hat{X} + l) + \hat{X}^T P_{X_0} \hat{X} \\ &= \hat{X}^T A^T P A \hat{X} + \hat{X}^T A^T P l + l^T P A \hat{X} + l^T P l + \hat{X}^T P_{X_0} \hat{X} = \hat{X}^T N \hat{X} + 2 \hat{X}^T W + l^T P l = \hat{X}^T W + l^T P l \end{aligned} \tag{7}$$

The parameters to be estimated in POD can be divided into three categories: (1) constant parameters, such as station coordinates and orbital parameters (initial orbital position and velocity parameters, solar radiation pressure parameters); (2) time-dependent process parameters, including troposphere zenith total delay (piecewise linear model), receiver clock offsets, satellite clock offsets (white noise model), and others; (3) ambiguity parameters of station–satellite pair. The corresponding NEQ system is shown as follows:

$$\begin{bmatrix} N_{\hat{X}_C \hat{X}_C} & N_{\hat{X}_C \hat{X}_P} & N_{\hat{X}_C \hat{X}_Y} \\ N_{\hat{X}_P \hat{X}_C} & N_{\hat{X}_P \hat{X}_P} & N_{\hat{X}_P \hat{X}_Y} \\ N_{\hat{X}_Y \hat{X}_C} & N_{\hat{X}_Y \hat{X}_P} & N_{\hat{X}_Y \hat{X}_Y} \end{bmatrix} \begin{bmatrix} \hat{X}_C \\ \hat{X}_P \\ \hat{X}_Y \end{bmatrix} = \begin{bmatrix} W_{\hat{X}_C} \\ W_{\hat{X}_P} \\ W_{\hat{X}_Y} \end{bmatrix} \tag{8}$$

where \hat{X}_C , \hat{X}_P , and \hat{X}_Y are the vectors of constant parameters, time-dependent parameters, and ambiguity parameters, respectively. As the tropospheric parameter of the random walk model, two adjacent time interval parameters can be considered in \hat{X}_P , and the constraints between them can be derived by the state transition matrix from the previous interval to the current ones. Then, the previous ones are set to inactive and eliminated together with other parameters.

Step 1: Set inactive parameters’ flag for \hat{X}_P and \hat{X}_Y vectors.

For an epoch solution of satellite and receiver clock offsets, they will be eliminated before the next epoch. The ZTD and ambiguity parameters can be estimated piecewise or by effective time intervals, and they will be eliminated if the end time is less than or equal to the current time. For illustration purposes, we randomly selected three ambiguities to show in Figure 4. The total number of the inactive parameters is denoted as n_{elim} , and the corresponding P_{X_0} (Equation (5)) of these parameters should be added into the NEQ system in advance.

Step 2: Obtain the block matrix of parameters to be eliminated.

The vectors of the entire parameters in NEQ system are only divided into two parts: “active” (\hat{X}') and “inactive” (\hat{R}), where \hat{R} is $n_{elim} \times 1$, \hat{X}' is $(n_{tot} - n_{elim}) \times 1$, $n_{tot} = n_C + n_P + n_Y$, n_C , n_P , and n_Y are the number of \hat{X}_C , \hat{X}_P , and \hat{X}_Y , respectively. The corresponding NEQ is divided into four blocks. The translation is essential to centralize the scattered parameters into a block \hat{R} . To gain efficiency, \hat{R} can be obtained by one-to-one “swap”, where the “last” of the eliminated parameter is swapped with the “first” active parameters. The two rows and columns of the “swap” need to be recorded, and the order of the parameters can be restored by repeated “swap”. Meanwhile, the W matrix also needs to be swapped correspondingly. The blue part in Figure 4 is for inactive parameters to

be eliminated later, among them, “1”, “2”, and “3” represent the randomly selected three ambiguity parameters.

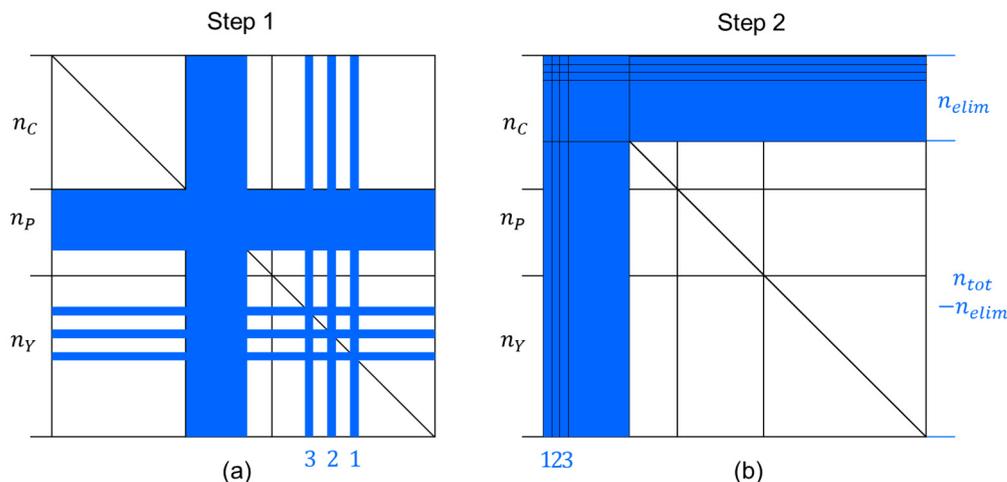


Figure 4. Diagrams of inactive parameters flag setting and centralization: (a) is for Step 1, and the blue part is for inactive parameters to be eliminated later, among them, “1”, “2”, and “3” represent the randomly selected three ambiguity parameters. (b) is for Step 2 after centralization, where the blue part is \hat{R} , and the white part is \hat{X}' .

Step 3: Batch elimination for inactive \hat{R} .

In the case of matrix operation, \hat{R} can be eliminated from the NEQ system by performing only once per epoch. The specific methods are as follows:

$$\begin{bmatrix} N_{\hat{R}\hat{R}} & N_{\hat{R}\hat{X}'} \\ N_{\hat{X}'\hat{R}} & N_{\hat{X}'\hat{X}'} \end{bmatrix} \begin{bmatrix} \hat{R} \\ \hat{X}' \end{bmatrix} = \begin{bmatrix} W_{\hat{R}} \\ W_{\hat{X}'} \end{bmatrix} \tag{9}$$

\hat{R} can be expressed by:

$$\hat{R} = N_{\hat{R}\hat{R}}^{-1} (W_{\hat{R}} - N_{\hat{R}\hat{X}'} \hat{X}') \tag{10}$$

Equation (9) can be rewritten without \hat{R} :

$$[N_{\hat{X}'\hat{R}} \quad N_{\hat{X}'\hat{X}'}] \begin{bmatrix} N_{\hat{R}\hat{R}}^{-1} (W_{\hat{R}} - N_{\hat{R}\hat{X}'} \hat{X}') \\ \hat{X}' \end{bmatrix} = W_{\hat{X}'} \tag{11}$$

$$(N_{\hat{X}'\hat{X}'} - N_{\hat{X}'\hat{R}} N_{\hat{R}\hat{R}}^{-1} N_{\hat{R}\hat{X}'}) \hat{X}' = W_{\hat{X}'} - N_{\hat{X}'\hat{R}} N_{\hat{R}\hat{R}}^{-1} W_{\hat{R}} \tag{12}$$

The NEQ system after elimination can be denoted by:

$$\bar{N}_{\hat{X}'\hat{X}'} \hat{X}' = \bar{W}_{\hat{X}'} \tag{13}$$

where $\bar{N}_{\hat{X}'\hat{X}'} = N_{\hat{X}'\hat{X}'} - N_{\hat{X}'\hat{R}} (N_{\hat{R}\hat{R}}^{-1}) N_{\hat{R}\hat{X}'}$, $\bar{W}_{\hat{X}'} = W_{\hat{X}'} - N_{\hat{X}'\hat{R}} (N_{\hat{R}\hat{R}}^{-1}) W_{\hat{R}}$. The index of \hat{R} and \hat{X}' to global parameter \hat{X} in Equation (12) should be stored in a binary file, which is convenient to recover the eliminated parameters \hat{R} by using Equation (10) after obtaining \hat{X}' . When the inactive \hat{R} is eliminated, the variables (Equation (7)) in the final solution also need to be updated accordingly. The specific expressions can be obtained as follows:

$$\begin{aligned} v^T P v + v_{X_0}^T P_{X_0} v_{X_0} &= \hat{X}^T W + I^T P I = \hat{R}^T W_{\hat{R}} + \hat{X}'^T W_{\hat{X}'} + I^T P I = (W_{\hat{R}}^T - \hat{X}'^T N_{\hat{X}'\hat{R}}) N_{\hat{R}\hat{R}}^{-1} W_{\hat{R}} + \hat{X}'^T W_{\hat{X}'} + I^T P I \\ &= W_{\hat{R}}^T N_{\hat{R}\hat{R}}^{-1} W_{\hat{R}} - \hat{X}'^T N_{\hat{X}'\hat{R}} N_{\hat{R}\hat{R}}^{-1} W_{\hat{R}} + \hat{X}'^T W_{\hat{X}'} + I^T P I \\ &= W_{\hat{R}}^T N_{\hat{R}\hat{R}}^{-1} W_{\hat{R}} + \hat{X}'^T (W_{\hat{X}'} - N_{\hat{X}'\hat{R}} N_{\hat{R}\hat{R}}^{-1} W_{\hat{R}}) + I^T P I = W_{\hat{R}}^T N_{\hat{R}\hat{R}}^{-1} W_{\hat{R}} + \hat{X}'^T W_{\hat{X}'} + I^T P I \end{aligned} \tag{14}$$

$l^T Pl$ should be updated after eliminating \hat{R} , with:

$$l^T Pl = l^T Pl - W_{\hat{R}}^T N_{\hat{R}\hat{R}}^{-1} W_{\hat{R}} \tag{15}$$

To set up the runtime environment of the high-efficiency matrix operation, only two toolkits need to be installed: Intel oneAPI Base Toolkit and Intel oneAPI HPC Toolkit. When the program is implemented, the matrix inverse $N_{\hat{R}\hat{R}}^{-1}$ can be efficiently implemented by calling DPOTRF and DPOTRI functions. DGEMM function can be called for Matrix multiplication, for instance, $N_{tmp} = N_{\hat{X}'\hat{R}}(N_{\hat{R}\hat{R}}^{-1})$. The DGEMV function is called when a matrix is multiplied by a vector, as in $W_{tmp} = N_{\hat{R}\hat{X}'}(N_{\hat{R}\hat{R}}^{-1})W_{\hat{R}}$. Using Equations (10), (14) and (15), \hat{R} can be eliminated and $\bar{N}_{\hat{X}'\hat{X}'}$ obtained as the white part in Figure 5a. In the global NEQ system, except for the $\bar{N}_{\hat{X}'\hat{X}'}$, the rest of the parts are set to zero.

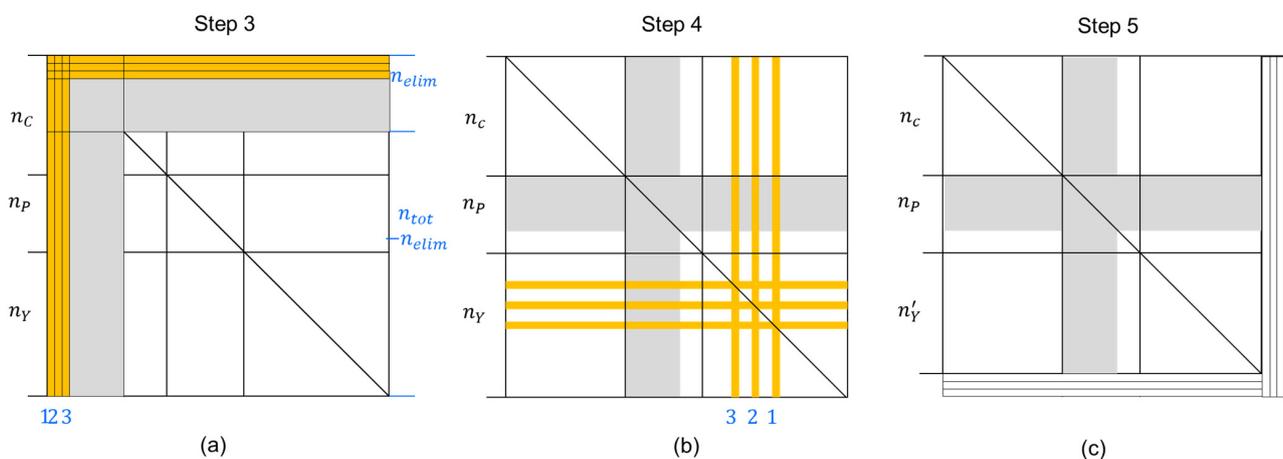


Figure 5. Diagrams of inactive parameters batch elimination and restoration: (a) NEQ after elimination, “1”, “2”, and “3” represent the randomly selected three ambiguity parameters, (b) NEQ after swap, (c) NEQ after compression. The yellow part is for eliminated ambiguities; gray part is for satellite and receiver clock offsets and inactive ZTDs.

Step 4: Restore the NEQ structure and update the parameter index.

After elimination, the parameter structure of the NEQ system needs to be restored as the order of \hat{X} rather than $[\hat{R} \ \hat{X}']^T$, like the structure of Equation (5), and the NEQ is shown as Equation (16):

$$\begin{bmatrix} \bar{N}_{\hat{X}_C\hat{X}_C} & \bar{N}_{\hat{X}_C\hat{X}_P} & \bar{N}_{\hat{X}_C\hat{X}_Y} \\ \bar{N}_{\hat{X}_P\hat{X}_C} & \bar{N}_{\hat{X}_P\hat{X}_P} & \bar{N}_{\hat{X}_P\hat{X}_Y} \\ \bar{N}_{\hat{X}_Y\hat{X}_C} & \bar{N}_{\hat{X}_Y\hat{X}_P} & \bar{N}_{\hat{X}_Y\hat{X}_Y} \end{bmatrix} \begin{bmatrix} \hat{X}_C \\ \hat{X}_P \\ \hat{X}_Y \end{bmatrix} = \begin{bmatrix} \bar{W}_{\hat{X}_C} \\ \bar{W}_{\hat{X}_P} \\ \bar{W}_{\hat{X}_Y} \end{bmatrix} \tag{16}$$

The specific operation is to “swap” again according to Step 2. Corresponding graphical expression, the NEQ system structure after the “swap” of all parameters is shown in Figure 5b. Since the process parameter vector \hat{X}_P always exists in the NEQ system, its position should be kept and not compressed (gray part). However, the ambiguity parameter (yellow part) should be compressed, and the dimension n_Y of the ambiguity parameter \hat{X}_Y might vary with different epochs. After the compression, the final structure of NEQ is shown in Figure 5c, its dimension becomes $n_{tot_new} = n_c + n_p + n_{Y'}$, and the indexes from local parameters (station related) to global parameters (NEQ) should be updated. If a new ambiguity parameter appears in the next epoch, it can be added to the end of the parameter vector \hat{X}_Y , and n_Y updated accordingly.

When the epoch-loop from Step 1 to Step 4 ends, the inactive \hat{X}_P and \hat{X}_Y are eliminated once again, and only the \hat{X}_C are kept in the final NEQ system to participate in the final inversion and complete the LSQ. The estimations of \hat{X}_C can be obtained by:

$$\hat{X}_C = \bar{N}_{\hat{X}_C \hat{X}_C}^{-1} \cdot \bar{W}_{\hat{X}_C} \quad (17)$$

A posteriori error of unit weight σ_0 can be calculated as:

$$\sigma_0 = \sqrt{\frac{l^T P l - \hat{X}_C^T \cdot W_{\hat{X}_C}}{n_{obs_total} - n_{par_total}}} \quad (18)$$

where n_{obs_total} is the total number of observations, and n_{par_total} is the total number of parameters to be estimated in POD, which includes \hat{X}_C and all eliminated \hat{X}_P and \hat{X}_Y .

4. Experiment with the Multi-GNSS-Integrated POD

To effectively illustrate the effect of the improved method, three schemes of comparative experiments were designed. The specific experimental schemes were as follows:

- Considering the classical estimation method based on the “one-by-one” elimination of “inactive” parameters had great success in huge GNSS networks POD, it was chosen as a reference group, named **Scheme I**;
- The “one-by-one” elimination method could be assisted with OpenMP to realize parallel computing, hence further improving the efficiency of multi-GNSS-integrated POD. Four threads were used to perform parameter elimination in parallel, named **Scheme II**;
- The improved estimation method based on the oneAPI HPC library was named **Scheme III**.

A Dell computer was selected as the platform for multi-GNSS POD data processing. An Intel(R)-Core (TM) i9-10900k @3.70 GHz CPU with 10 logical cores and a solid-state drive with 64 GB memory were equipped on this computer. To analyze the POD efficiency of the above parameter-elimination methods in cases of different stations and satellites, we randomly selected 79, 126, and 171 stations from the IGS/MGEX network to conduct GPS (G), GPS/BDS (GC), GPS/BDS/Galileo (GCE), and GPS/BDS/Galileo/GLONASS (GCER) satellites' POD, respectively. Afterward, the LSQ time-consuming was analyzed. The general error correction and parameter setting of the data processing are shown in Table 1.

Table 1. The data and error corrections for multi-GNSS-integrated POD.

Type	Descriptions
stations	IGS/MGEX station network [43];
Period	Days of year (DOYs) from 013 to 019, 2022;
Observations	zero-difference carrier phase and pseudo-range elevation weight; cut-off is 7°;
orbital arc	24 h;
Solar radiation	GPS/GLONASS: ECOM2 [38];
pressure model	Galileo: ECOM1+a priori model [16,37]; BDS2: ECOM1; BDS3: ECOM1 + a priori model [17];
Inter-system biases (ISB) and Inter-frequency biases (IFB)	ISB between BDS2/3 and GPS, Galileo, and GPS; constant parameter per station; IFB per satellite and station pair between GLONASS and GPS; Constraint that sum of all ISBs and IFBs are zero was added;

Table 1. Cont.

Type	Descriptions
Ionospheric delay	Ionosphere-free (IF) combination; GPS: L1/L2; BDS: B1I/B3I; Galileo: E1/E5a; GLONASS: R01/R02;
Tropospheric delay	Zenith total delay (ZTD): 2-h interval; Saastamoinen [44] + Global Map Function (GMF) [45]; Horizontal gradient: 24-h interval;
Antenna phase center correction (PCC)	For both satellites and receivers, phase center correction model is from igs14_2196.atx [46,47]; BDS and Galileo receivers PCC was using GPS L1 and L2 instead.

5. Validation of the Multi-GNSS POD

5.1. Time Consumption Analyzed for Multi-GNSS-Integrated POD

The LSQ of 96 satellites (GCE) and 133 stations were selected to analyze the variation of time consumption with different epochs. The detailed results of Schemes I–III for LSQ are shown in Figure 6, and the number of eliminated parameters was also plotted.

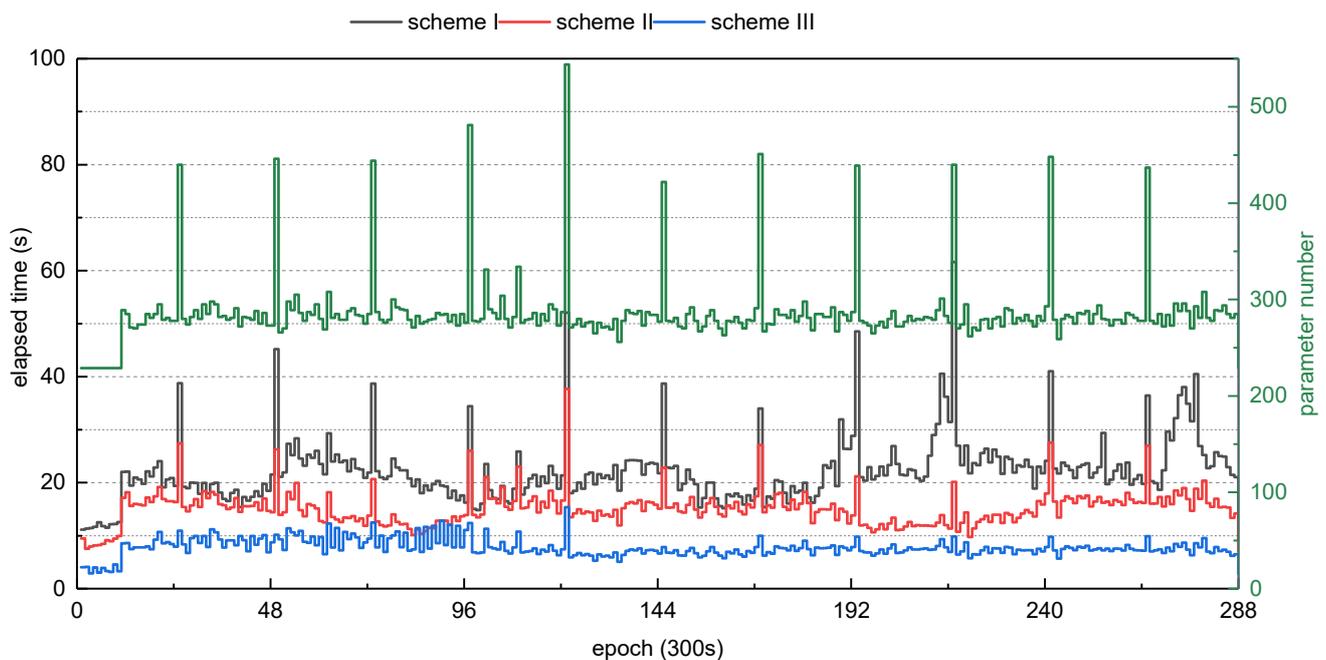


Figure 6. Elapsed time of different schemes and the number of eliminated parameters. The green line is for the number of eliminated parameters; the black, red and blue lines are elapsed time for Schemes I–III, separately.

When “one-by-one” elimination was adopted in Scheme I, the whole LSQ took 2 h: 6 min: 49.2 s; among them, the time spent on the eliminated-parameter recovery was 145.458 s. When parallel elimination was used in Scheme II with four threads, the LSQ was shortened to 1 h: 20 min: 25.6 s. Compared to 22 s of Scheme I, the average time of parameter elimination per epoch was 13.113 s, which was reduced by about 41%. In the case of the improved method in Scheme III, it took 41 min; 48.6 s for one LSQ, which was reduced by about 67%; and took 79.037 s for parameter recovery by matrix calculation in Equation (10).

Considering that multiple POD tasks are executed at the same time to further improve the efficiency of data processing, four groups of POD data processing were set up. For

Schemes I and III, the CPU utilization rates were about 45% and 40%. Since there were 10 cores in this CPU and four groups of POD used, the CPU utilization was consistent with the theoretical rate of 40%. For Scheme II, parallel computing can improve the LSQ efficiency; however, it also increases the CPU and memory overheads. To further quantitatively analyze the CPU resource occupation caused by multi-thread parallelism, five days of POD results were analyzed. The LSQ elapsed time with four and two simultaneous POD groups are shown in Figure 7, and the corresponding CPU utilization rates were 100% and 66%, respectively. In this case, the LSQ times of Scheme II with four groups and two groups POD were 4 h: 18 min: 46.171 s and 3 h: 08 min: 32.266 s. Compared with the elapsed time of 4 h: 52 min: 56.651 s in Scheme I, the improvements in LSQ efficiency in Scheme II were 0 h: 34 min: 10.475 s (12%) and 1 h: 44 min: 24.385 s (36%) for four and two simultaneous POD groups, respectively. The main reason was that when the CPU utilization rate was 100%, part of the parameter elimination needed to wait until the previous work finished and released some CPU cores, this process reduced the parallel processing efficiency. Thus, in the following work, only two rather than four groups of POD were solved simultaneously to fully exploit the capabilities of parallel elimination in Scheme II.

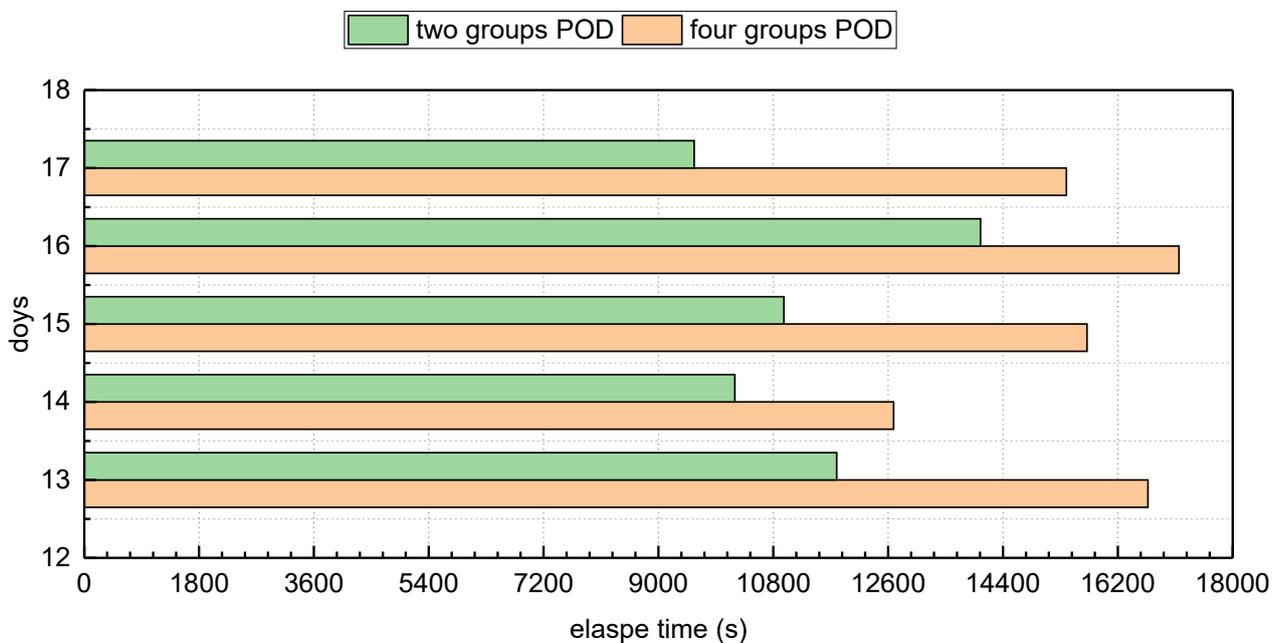


Figure 7. Elapsed time of LSQ using two or four threads in Scheme II.

It should be noted that the 171 stations were selected randomly, without considering the data quality to optimize the stations. Based on the precise orbit determination of 171 stations, the 126 stations were obtained by removing the stations whose residual RMS value of carrier-phase observation was greater than 16 mm. Therefore, the selected 126 stations can be regarded as optimization by the data quality control with a posterior residual analysis. Similarly, 79 stations were optimized based on 126 stations' POD, which removed the stations whose carrier residual RMS was greater than 10 mm. Considering the influence of different distributions on the accuracy of orbit determination when the number of stations was fixed, the distributions of 171, 126, and 79 stations are shown in Figures 8–10, respectively. According to the strategy mentioned in Table 1, one-week data were used for GNSS POD analysis. Table 2 lists the GNSS satellites involved in POD.

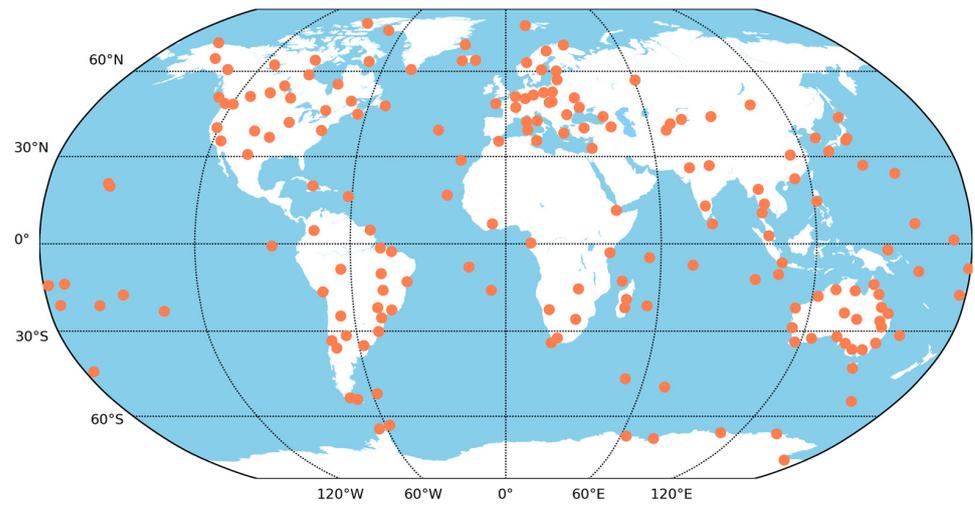


Figure 8. Distribution of the 171 stations for POD.

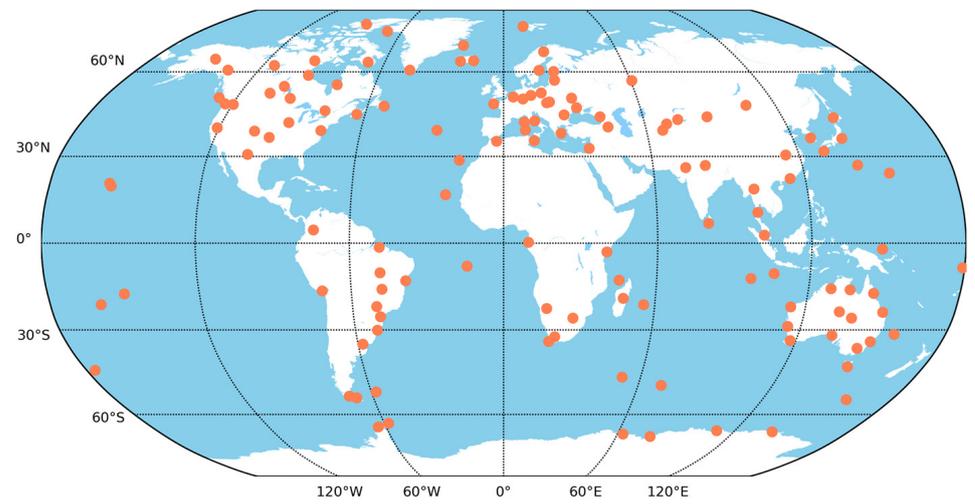


Figure 9. Distribution of the 126 stations for POD.

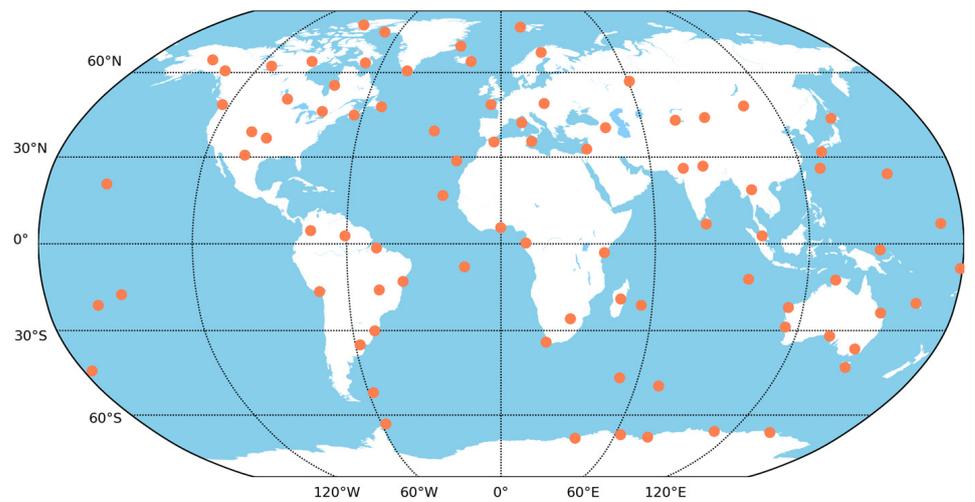


Figure 10. Distribution of the 79 stations for POD.

Table 2. Satellite PRN involved in multi-GNSS POD.

Systems	PRN List
GPS (G)	G01, G02, G03, G04, G05, G06, G07, G08, G09, G10, G11, G12, G13, G14, G15, G16, G17, G18, G19, G20, G21, G22, G24, G25, G26, G27, G28, G29, G30, G31, G32
GLONASS (R)	R01, R02, R03, R04, R05, R06, R07, R08, R09, R10, R11, R12, R13, R14, R15, R17, R18, R19, R20, R21, R22, R23, R24
BDS (C)	C01, C02, C03, C04, C05, C06, C07, C08, C09, C10, C11, C12, C13, C14, C16, C19, C20, C21, C22, C23, C24, C25, C26, C27, C28, C29, C30, C32, C33, C34, C35, C36, C37, C38, C39, C40, C41, C42, C43, C44, C45, C46
Galileo (E)	E01, E02, E03, E04, E05, E07, E08, E09, E11, E12, E13, E15, E18, E19, E21, E24, E25, E26, E27, E30, E31, E33, E36

Figure 11 shows the elapsed time of three schemes with different satellite systems and stations. It can be seen from Figure 11a–c that as the number of systems and satellites increases, a significant increment of LSQ time consumption can be observed, especially for the GLONASS. For ISBs of the BDS and Galileo, they were only related to the station with a number of $n_{site} \times 1$; however, GLONASS adopted the IFB related to the station–satellite pair, and its parameter number was $n_{site} \times 24$. The IFBs significantly increased the number of parameters, thus greatly increasing LSQ time consumption.

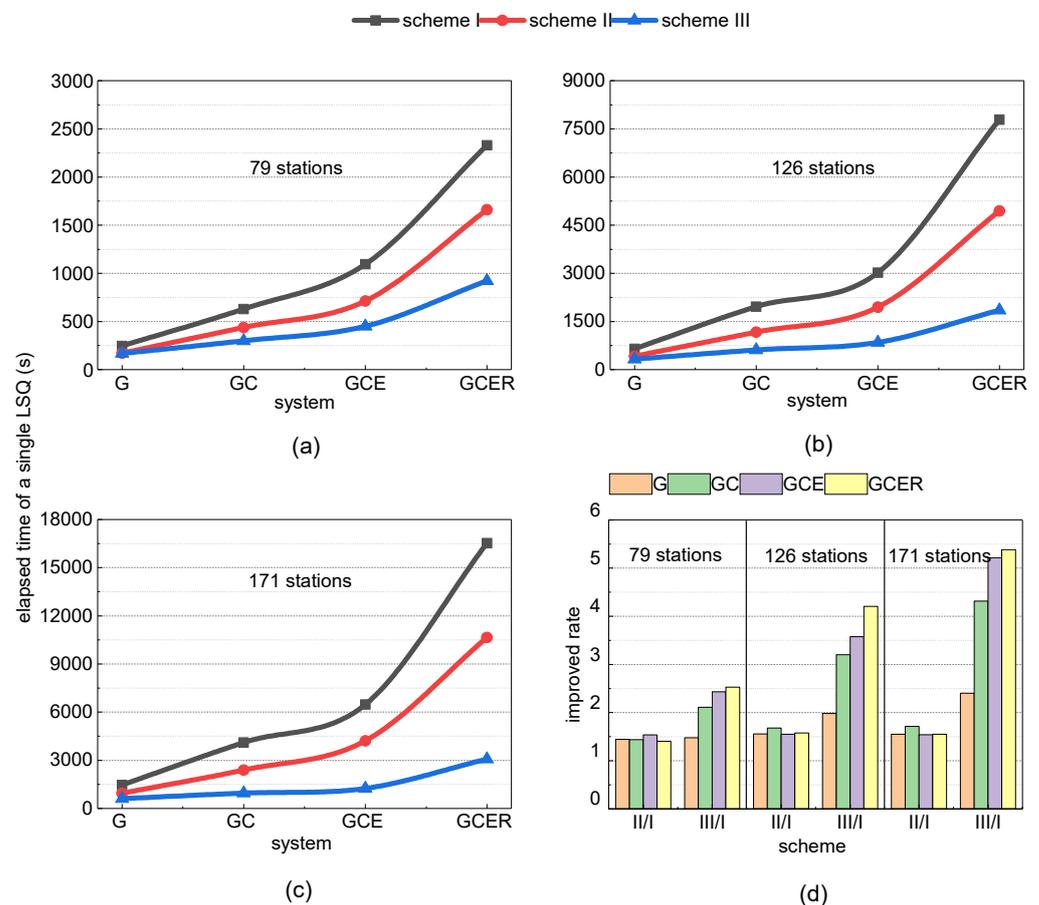


Figure 11. Elapsed time of a single LSQ for three schemes with different systems: (a–c) are the results for 79, 126, and 171 stations, and (d) is the improved rate of Schemes II and III concerning those of Scheme I.

The specific time statistics are shown in Table 3. As seen from Figure 11 and Table 3, when the number of satellites was small, for instance, single-system GPS, the difference in the elapsed time in the three methods was relatively small, which was about 2–4 min, 5–10 min, and 10–24 min for 79, 117, and 171 stations, respectively. With the increasing number of satellites or stations, the difference between the three methods became more

and more significant. For GCER POD, the maximum difference in elapsed time between Schemes I and III can reach 225 min when 171 stations were used.

Table 3. Elapsed time of LSQ for three schemes with different systems and stations, the unit is hh: min: s.

Systems	79 Stations			126 Stations			171 Stations		
	I	II	III	I	II	III	I	II	III
G	0:04:06	0:02:50	0:02:47	0:10:46	0:06:55	0:05:26	0:24:22	0:15:43	0:10:08
GC	0:10:29	0:07:18	0:04:58	0:32:42	0:19:30	0:10:13	1:08:19	0:39:53	0:15:50
GCE	0:18:14	0:11:52	0:07:30	0:50:21	0:32:27	0:14:05	1:47:55	1:10:10	0:20:41
GCER	0:38:51	0:27:40	0:15:22	2:09:45	1:22:23	0:30:50	4:35:25	2:57:30	0:51:11

To describe the improvement of time consumption visually and conveniently, the improved ratio of elapsed time was adopted for analysis. Figure 11d and Table 4 show the detailed ratios of Schemes II and III. Compared with Scheme I, the elapsed time of Scheme II was shortened by about 1.45, 1.59, and 1.59 for 79, 126, and 171 stations, respectively. Compared with Scheme I, the time of Scheme III was shortened by about 2.14, 4.32, and 4.33 for 79, 126, and 171 stations. It was worth noting that Scheme II had a stable ratio of around 1.5, and did not change with the increased number of stations or satellites, because the parallelism saved running time through multi-threads when the algorithm remained unchanged. For Scheme III, the improvement ratio kept higher with the increased number of stations and satellites, which reflected that the improved method had more advantages in multi-GNSS-integrated POD with the larger number of stations.

Table 4. Improved ratios of LSQ elapsed time with different stations and systems.

Systems	79 Stations		126 Stations		171 Stations	
	II/I	III/I	II/I	III/I	II/I	III/I
G	1.45	1.48	1.56	1.98	1.55	2.40
GC	1.44	2.11	1.68	3.20	1.71	4.32
GCE	1.54	2.43	1.55	3.58	1.54	5.22
GCER	1.40	2.53	1.57	4.21	1.55	5.38

Afterward, the characteristics of LSQ elapsed time in G-, GC-, GCE-, and GCER-integrated POD of the three methods were analyzed with the number of increased stations, and the specific relationship is shown in Figure 12. For Schemes I and II, the slope of the curve became steeper with the increased stations, especially for GCER POD (green line). However, for Scheme III, even for the GCER with the largest number of satellites, the function of the elapsed time with the number of stations was still close to a linear trend with a smaller slope than the other two methods.

To study the realization of GCER-integrated POD within a fixed time range, the appropriate number of stations can be determined by using the function expression between LSQ time consumption and the number of stations in Figure 12, and when the number of stations was determined, the distribution of stations could be optimized to ensure a higher POD accuracy. The quadratic polynomial model was selected to solve the model coefficients (green line in Scheme III), and the specific function was $dt = 0.0805x^2 + 3.2471x + 163.21$, $x \in [79, 171]$. When 100 stations were selected, the LSQ took about 21.5 min for an arc of 24 h. If the time span of generation ultra-rapid orbit is limited to 2 h, 100 stations need 86 min for four LSQ cycles, and the remaining 34 min can be used for orbit integrating, ambiguity fixing, residual editing, data preprocessing, and others.

Considering that POD is an overall process, we also analyzed the computation time of overall orbit determination in Table 5. For Schemes I–III, only the LSQ methods were different. When the POD was carried out, four iterations of the LSQ were used. Thus, the elapsed time differences of the POD were larger than those of the LSQ, especially since the largest difference between Schemes I and III was nearly about 15 h in the GCER's POD by using 171 stations.

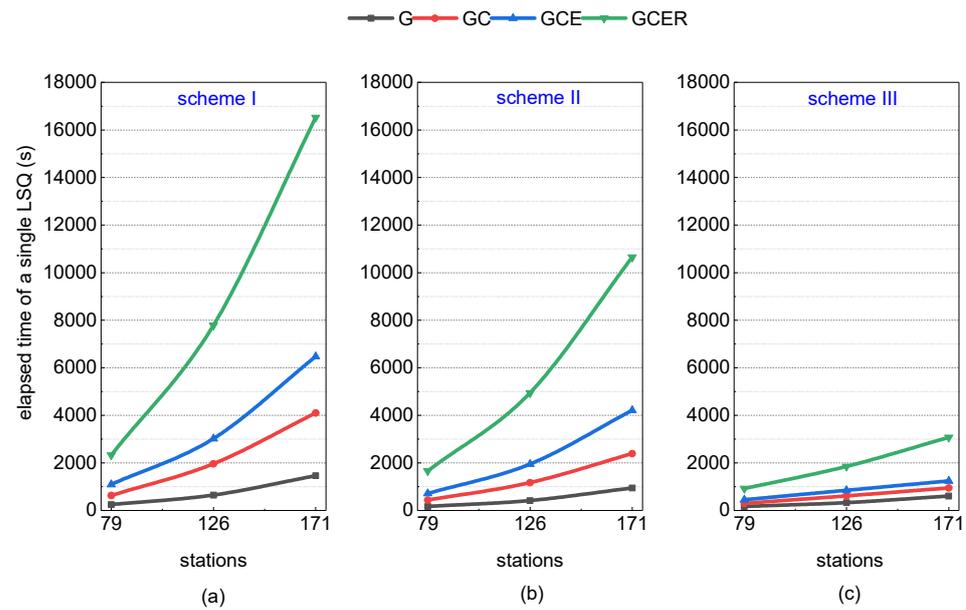


Figure 12. Elapsed time of a single LSQ for three schemes with different stations: (a–c) are the results for the Schemes I–III, respectively.

Table 5. Elapsed time of POD for three schemes with different systems and stations; the unit is hh: min: s.

Systems	79 Stations			126 Stations			171 Stations		
	I	II	III	I	II	III	I	II	III
G	0:37:14	0:32:10	0:31:58	1:08:54	0:53:30	0:47:34	2:07:50	1:33:14	1:10:54
GC	1:07:14	0:54:30	0:45:10	2:42:09	1:49:21	1:12:13	5:09:54	3:16:09	1:39:57
GCE	1:43:15	1:17:47	1:00:19	3:58:16	2:46:40	1:33:12	7:54:11	5:23:11	2:05:15
GCER	3:08:22	2:23:38	1:34:26	9:18:58	6:09:30	2:43:18	19:09:28	12:37:48	4:12:32

5.2. Precision for Multi-GNSS-integrated POD

In addition to analyzing the LSQ time consumption, it is also necessary to pay attention to the accuracy of POD with different stations. 1D RMS of the precise orbit was calculated based on GFZ MGEX Products as a reference, and the mean precision of the one-week POD result was obtained. From the comparison in Table 6, the orbit 1D RMS indicators of GPS, Galileo, and GLONASS satellites were 1.3–1.4 cm, 1.3–1.5 cm, and 2.7–3.1 cm, respectively. For the BDS, the 1D RMS indicators were 4.9–5.3 cm and 7.9–10.5 cm for MEO and IGSO satellites. However, the 1D RMS of the BDS GEO satellite was ranging from 234 to 315 cm. The possible reason is the poor geometry and solar radiation pressure model compared to that of MEO and IGSO satellites.

Table 6. One-dimensional (1D) RMS of POD with different stations; the unit is cm.

Satellites	79 Stations	126 Stations	171 Stations
GPS	1.4	1.3	1.3
BDS GEO	315.0	280.7	234.4
BDS IGSO	10.5	8.6	7.9
BDS MEO	5.3	4.9	4.9
Galileo	1.5	1.3	1.3
GLONASS	3.1	2.7	2.7

Improvement of the efficiency of different methods of parameter elimination should not affect the accuracy of the multi-GNSS POD; however, the improvement of the orbit accuracy will be achieved due to more observations provided by more stations, especially for the ultra-rapid orbit. It can be seen from Table 2 that the LSQ of the GCER POD based

on 126 stations can be achieved within 31 min by using the improved method (Scheme III). Compared with the 38 min of the “one-by-one” method (Scheme I) that used 79 stations to implement the LSQ of the GCER POD, the addition of more stations can achieve higher accuracy with a shorter elapsed time of the LSQ (Table 3) and whole POD (Table 5). When the number of stations was increased from 79 to 126, the accuracy improvements were 0.1 cm (7%), 34.3 cm (11%), 1.9 cm (18%), 0.4 cm (8%), 0.2 cm (13%), and 0.4 cm (13%) for GPS, BDS GEO, BDS IGSO, BDS MEO, Galileo and GLONASS satellites, respectively. If the number of stations was increased to 171, the accuracy of GEO and IGSO satellites improved by about 46.3 cm (16%) and 0.7 cm (8%), and the tracking stations distributed in the Asia-Pacific region could provide valid observations for GEO and IGSO satellites. However, there was no significant improvement in GPS, BDS MEO, Galileo, and GLONASS. The main reason is that the 126 stations were selected according to the global uniform distribution, which was nearly saturated for MEO satellites.

6. Conclusions

In this paper, an improved parameter estimation method based on Intel oneAPI HPC was proposed to improve computational efficiency. Through the analysis of LSQ time consumption with the increased number of stations, the improved method presents a nearly linear trend, and it is verified that the efficiency of the LSQ in GPS, BDS2, BDS3, Galileo, and GLONASS satellites’ integrated POD was improved by 4.33 times for 171 stations. In addition, the CPU consumption of the improved method was close to that of the serial “one-by-one” method, and then as many POD tasks as possible were performed simultaneously.

Based on this method, the GNSS POD time consumption and precision of 79, 126, and 171 stations were analyzed. The improved method can achieve higher accuracy of POD by using more stations within the same time requirement. However, when the number of stations was increased from 126 to 171, there was no significant improvement in the accuracy of the MEO satellites’ POD. Therefore, under the condition of ensuring the accuracy of POD, the data processing efficiency can be further improved by appropriately reducing the number of ground stations.

On the other hand, by analyzing the functional relationship between the number of stations and the LSQ time consumption in the GCER-integrated POD, it can be inferred that when the number of stations was about 100, the single LSQ time in POD was about 24 min. In this case, the ultra-rapid orbit products might be solved within 2 h, including data preprocessing, orbit integration, LSQ, residuals editing, ambiguity fixing, and other steps. In the future, multi-threaded parallel will be added to further improve the efficiency of the LSQ, and fast residuals editing will be implemented to reduce two LSQ iterations and then improve the efficiency of multi-GNSS-integrated POD.

Although the Intel oneAPI simplifies the development process of heterogeneous computing in different architectures and maximizes performance to meet the needs of different workloads, we did not test the proposed method in other CPU/GPU/FPGA architectures, and the performance of this method in other CPU architectures and whether it is limited to Intel CPU will require further investigation and analysis.

Author Contributions: Methodology, X.Y. and C.L.; software, X.Y.; validation, C.L. and M.Y.; formal analysis, investigation, X.Y. and C.L.; writing—original draft preparation, X.Y. and C.L.; review and editing, C.L., M.Y., W.F. and M.Z.; visualization, X.Y.; funding acquisition, W.F. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities, Sun Yat-sen University, grant numbers: 22lgqb09 and 22qntd0301.

Data Availability Statement: GNSS observation data used in this study are publicly available on the Crustal Dynamics Data Information System (CDDIS) at <https://cddis.nasa.gov/archive/gnss/data/>, accessed on 5 April 2022.

Acknowledgments: Thanks for the data support of IGS/MGEX and iGMAS and the Intel oneAPI toolkit for scientific computing.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Zumberge, J.F.; Heflin, M.B.; Jefferson, D.C.; Watkins, M.M.; Webb, F.H. Precise Point Positioning for the Efficient and Robust Analysis of GPS Data from Large Networks. *J. Geophys. Res. Solid Earth* **1997**, *102*, 5005–5017. [CrossRef]
- Bahadur, B. A Study on the Real-Time Code-Based GNSS Positioning with Android Smartphones. *Measurement* **2022**, *194*, 111078. [CrossRef]
- Yan, X.; Liu, C.; Jiang, M.; Yang, M.; Feng, W.; Zhong, M.; Peng, L. Performance Analysis of Oceanographic Research Vessel Precise Point Positioning Based on BDS/GNSS RTK Receivers. *Measurement* **2023**, *211*, 112637. [CrossRef]
- Li, X.; Ge, M.; Dai, X.; Ren, X.; Fritsche, M.; Wickert, J.; Schuh, H. Accuracy and Reliability of Multi-GNSS Real-Time Precise Positioning: GPS, GLONASS, BeiDou, and Galileo. *J. Geod.* **2015**, *89*, 607–635. [CrossRef]
- Montenbruck, O.; Steigenberger, P.; Prange, L.; Deng, Z.; Zhao, Q.; Perosanz, F.; Romero, I.; Noll, C.; Stürze, A.; Weber, G.; et al. The Multi-GNSS Experiment (MGEX) of the International GNSS Service (IGS)—Achievements, Prospects and Challenges. *Adv. Space Res.* **2017**, *59*, 1671–1697. [CrossRef]
- Wu, Z.; Wang, Q.; Yu, Z.; Hu, C.; Liu, H.; Han, S. Modeling and Performance Assessment of Precise Point Positioning with Multi-Frequency GNSS Signals. *Measurement* **2022**, *201*, 111687. [CrossRef]
- Dow, J.M.; Neilan, R.E.; Rizos, C. The International GNSS Service in a Changing Landscape of Global Navigation Satellite Systems. *J. Geod.* **2009**, *83*, 191–198. [CrossRef]
- Prange, L.; Villiger, A.; Sidorov, D.; Schaer, S.; Beutler, G.; Dach, R.; Jäggi, A. Overview of CODE’s MGEX Solution with the Focus on Galileo. *Adv. Space Res.* **2020**, *66*, 2786–2798. [CrossRef]
- Steigenberger, P.; Rothacher, M.; Dietrich, R.; Fritsche, M.; Rülke, A.; Vey, S. Reprocessing of a Global GPS Network. *J. Geophys. Res. Solid Earth* **2006**, *111*, 3747. [CrossRef]
- Chen, H.; Jiang, W.; Ge, M.; Wickert, J.; Schuh, H. An Enhanced Strategy for GNSS Data Processing of Massive Networks. *J. Geod.* **2014**, *88*, 857–867. [CrossRef]
- Xie, W.; Huang, G.; Fu, W.; Shu, B.; Cui, B.; Li, M.; Yue, F. A Quality Control Method Based on Improved IQR for Estimating Multi-GNSS Real-Time Satellite Clock Offset. *Measurement* **2022**, *201*, 111695. [CrossRef]
- Steigenberger, P.; Fritsche, M.; Dach, R.; Schmid, R.; Montenbruck, O.; Uhlemann, M.; Prange, L. Estimation of Satellite Antenna Phase Center Offsets for Galileo. *J. Geod.* **2016**, *90*, 773–785. [CrossRef]
- Huang, G.; Yan, X.; Zhang, Q.; Liu, C.; Wang, L.; Qin, Z. Estimation of Antenna Phase Center Offset for BDS IGSO and MEO Satellites. *GPS Solut.* **2018**, *22*, 49. [CrossRef]
- Yan, X.; Huang, G.; Zhang, Q.; Wang, L.; Qin, Z.; Xie, S. Estimation of the Antenna Phase Center Correction Model for the BeiDou-3 MEO Satellites. *Remote Sens.* **2019**, *11*, 2850. [CrossRef]
- Rodriguez-Solano, C.J.; Hugentobler, U.; Steigenberger, P.; Bloßfeld, M.; Fritsche, M. Reducing the Draconitic Errors in GNSS Geodetic Products. *J. Geod.* **2014**, *88*, 559–574. [CrossRef]
- Montenbruck, O.; Steigenberger, P.; Hugentobler, U. Enhanced Solar Radiation Pressure Modeling for Galileo Satellites. *J. Geod.* **2015**, *89*, 283–297. [CrossRef]
- Yan, X.; Liu, C.; Huang, G.; Zhang, Q.; Wang, L.; Qin, Z.; Xie, S. A Priori Solar Radiation Pressure Model for BeiDou-3 MEO Satellites. *Remote Sens.* **2019**, *11*, 1605. [CrossRef]
- Ge, M.; Gendt, G.; Dick, G.; Zhang, F.P.; Rothacher, M. A New Data Processing Strategy for Huge GNSS Global Networks. *J. Geod.* **2006**, *80*, 199–203. [CrossRef]
- Prange, L.; Orliac, E.; Dach, R.; Arnold, D.; Beutler, G.; Schaer, S.; Jäggi, A. CODE’s Five-System Orbit and Clock Solution—The Challenges of Multi-GNSS Data Analysis. *J. Geod.* **2017**, *91*, 345–360. [CrossRef]
- Guo, J.; Xu, X.; Zhao, Q.; Liu, J. Precise Orbit Determination for Quad-Constellation Satellites at Wuhan University: Strategy, Result Validation, and Comparison. *J. Geod.* **2016**, *90*, 143–159. [CrossRef]
- Zhu, S.; Reigber, C.; König, R. Integrated Adjustment of CHAMP, GRACE, and GPS Data. *J. Geod.* **2004**, *78*, 103–108. [CrossRef]
- Li, B.; Ge, H.; Ge, M.; Nie, L.; Shen, Y.; Schuh, H. LEO Enhanced Global Navigation Satellite System (LeGNSS) for Real-Time Precise Positioning Services. *Adv. Space Res.* **2019**, *63*, 73–93. [CrossRef]
- Li, X.; Zhang, K.; Ma, F.; Zhang, W.; Zhang, Q.; Qin, Y.; Zhang, H.; Meng, Y.; Bian, L. Integrated Precise Orbit Determination of Multi-GNSS and Large LEO Constellations. *Remote Sens.* **2019**, *11*, 2514. [CrossRef]
- Serpelloni, E.; Casula, G.; Galvani, A.; Anzidei, M.; Baldi, P. Data Analysis of Permanent GPS Networks in Italy and Surrounding Region: Application of a Distributed Processing Approach. *Ann. Geophys.* **2006**, *49*, 897–928. [CrossRef]
- Boomkamp, H.; König, R. Bigger, Better, Faster POD Position Paper for Session on Precise Orbit Determination. In Proceedings of the IGS Workshop and Symposium, Berne, Switzerland, 1–5 March 2004; Volume 10. Available online: ftp://192.134.134.6/pub/igs/igs_cb/resource/pubs/04_rtberne/cdrom/Session9/9_0_Boomkamp.pdf (accessed on 31 February 2022).
- Li, L.; Lu, Z.; Chen, Z.; Cui, Y.; Kuang, Y.; Wang, F. Parallel Computation of Regional CORS Network Corrections Based on Ionospheric-Free PPP. *GPS Solut.* **2019**, *23*, 70. [CrossRef]
- Cui, Y.; Chen, Z.; Li, L.; Zhang, Q.; Luo, S.; Lu, Z. An Efficient Parallel Computing Strategy for the Processing of Large GNSS Network Datasets. *GPS Solut.* **2021**, *25*, 36. [CrossRef]

28. Costa, J.J.; Cortes, T.; Martorell, X.; Ayguade, E.; Labarta, J. Running OpenMP Applications Efficiently on an Everything-Shared SDSM. *J. Parallel Distrib. Comput.* **2006**, *66*, 647–658. [[CrossRef](#)]
29. Mironov, V.; Alexeev, Y.; Keipert, K.; D’ello, M.; Moskovsky, A.; Gordon, M.S. An Efficient MPI/OpenMP Parallelization of the Hartree-Fock Method for the Second Generation of Intel[®] Xeon Phi[™] Processor. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2017, Denver, CO, USA, 12–17 November 2017; Association for Computing Machinery, Inc.: New York, NY, USA, 2017.
30. Kuang, K.; Zhang, S.; Li, J. Real-Time GPS Satellite Orbit and Clock Estimation Based on OpenMP. *Adv. Space Res.* **2019**, *63*, 2378–2386. [[CrossRef](#)]
31. Chen, X.; Ge, M.; Hugentobler, U.; Schuh, H. A New Parallel Algorithm for Improving the Computational Efficiency of Multi-GNSS Precise Orbit Determination. *GPS Solut.* **2022**, *26*, 83. [[CrossRef](#)]
32. Yang, Y.; Mao, Y.; Sun, B. Basic Performance and Future Developments of BeiDou Global Navigation Satellite System. *Satell. Navig.* **2020**, *1*, 1. [[CrossRef](#)]
33. Intel oneAPI Intel[®]-Optimized Math Library for Numerical Computing. Available online: <https://www.intel.com/content/www/us/en/developer/articles.html> (accessed on 31 February 2022).
34. Leick, A.; Rapoport, L.; Tatarnikov, D. *GPS Satellite Surveying*; Wiley: Hoboken, NJ, USA, 2015.
35. Li, X.; Ge, M.; Lu, C.; Zhang, Y.; Wang, R.; Wickert, J.; Schuh, H. High-Rate GPS Seismology Using Real-Time Precise Point Positioning with Ambiguity Resolution. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 6165–6180. [[CrossRef](#)]
36. Ren, X.; Chen, J.; Li, X.; Zhang, X. Ionospheric Total Electron Content Estimation Using GNSS Carrier Phase Observations Based on Zero-Difference Integer Ambiguity: Methodology and Assessment. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 817–830. [[CrossRef](#)]
37. Beutler, G.; Brockmann, E.; Gurtner, W.; Hugentobler, U.; Mervart, L.; Rothacher, M.; Verdun, A. Extended Orbit Modeling Techniques at the CODE Processing Center of the International GPS Service for Geodynamics (IGS): Theory and Initial Results. *Manuscr. Geod.* **1994**, *19*, 367–386.
38. Arnold, D.; Meindl, M.; Beutler, G.; Dach, R.; Schaer, S.; Lutz, S.; Prange, L.; Sośnica, K.; Mervart, L.; Jäggi, A. CODE’s New Solar Radiation Pressure Model for GNSS Orbit Determination. *J. Geod.* **2015**, *89*, 775–791. [[CrossRef](#)]
39. Blewitt, G. An Automatic Editing Algorithm for GPS Data. *Geophys. Res. Lett.* **1990**, *17*, 199–202. [[CrossRef](#)]
40. Ge, M.; Gendt, G.; Rothacher, M.; Shi, C.; Liu, J. Resolution of GPS Carrier-Phase Ambiguities in Precise Point Positioning (PPP) with Daily Observations. *J. Geod.* **2008**, *82*, 389–399. [[CrossRef](#)]
41. Schaer, S.; Villiger, A.; Arnold, D.; Dach, R.; Prange, L.; Jäggi, A. The CODE Ambiguity-Fixed Clock and Phase Bias Analysis Products: Generation, Properties, and Performance. *J. Geod.* **2021**, *95*, 81. [[CrossRef](#)]
42. Liu, X.; Jiang, W.; Li, P.; Deng, Z.; Ge, M.; Schuh, H. An Extended Inter-System Biases Model for Multi-GNSS Precise Point Positioning. *Measurement* **2023**, *206*, 112306. [[CrossRef](#)]
43. IGS MGEX International GNSS Service, GNSS Constellations. Available online: <http://mgex.igs.org/index.php#Constellations> (accessed on 31 October 2019).
44. Saastamoinen, J. Atmospheric correction for the troposphere and stratosphere in radio ranging satellites. *Use Artificial Satell. Geod.* **1972**, *15*, 247–251.
45. Boehm, J.; Niell, A.; Tregoning, P.; Schuh, H. Global Mapping Function (GMF): A New Empirical Mapping Function Based on Numerical Weather Model Data. *Geophys. Res. Lett.* **2006**, *33*, L07304. [[CrossRef](#)]
46. Altamimi, Z.; Rebischung, P.; Métivier, L.; Collilieux, X. ITRF2014: A New Release of the International Terrestrial Reference Frame Modeling Nonlinear Station Motions. *J. Geophys. Res. Solid Earth* **2016**, *121*, 6109–6131. [[CrossRef](#)]
47. Villiger, A.; Dach, R.; Schaer, S.; Prange, L.; Zimmermann, F.; Kuhlmann, H.; Wübbena, G.; Schmitz, M.; Beutler, G.; Jäggi, A. GNSS Scale Determination Using Calibrated Receiver and Galileo Satellite Antenna Patterns. *J. Geod.* **2020**, *94*, 93. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.