**FiPPP User Manual**

**Version 1.0**

**Authors: *Chao Hu, Ruiguang Wang, Changpeng Tao***

10, July, 2023

Contents

[Introduction 1](#_Toc140524835)

[Features of FiPPP 1](#_Toc140524836)

[Main Functions 1](#_Toc140524837)

[Build new project and run FiPPP 3](#_Toc140524838)

[Build new project 3](#_Toc140524839)

[FiPPP Configuration 10](#_Toc140524840)

[Output file format description 13](#_Toc140524841)

[Data Visualization 15](#_Toc140524842)

[Build project 15](#_Toc140524843)

[Configuration 16](#_Toc140524844)

[Structure of Software 21](#_Toc140524845)

[Appendix 29](#_Toc140524846)

[PPP Models 29](#_Toc140524847)

[General Observation model 29](#_Toc140524848)

[BDS-3 New Observation Combination Model 33](#_Toc140524849)

[Examples 35](#_Toc140524850)

[Acknowledgement 46](#_Toc140524851)

[Contact author 46](#_Toc140524852)

[References 46](#_Toc140524853)

Introduction

The multi-GNSS and multi-frequency location-based services (LBS) are the hot topic in GNSS community. It is also suggested that the abundant signal information can optimize the performances of positioning, navigation and timing (PNT) applications. Therefore, the open-source single- to five-frequency precise point positioning software (FiPPP) is proposed to fully use the whole received multi-frequency and multi-GNSS observations, including BDS-3, GPS and Galileo. The newly open-source FiPPP is implemented and organized by C++ computing language, which should be compiled, debugged and run on the Windows operating system. Apart from the positioning service, the multi-GNSS and multi-frequency DCB and code OSB can also be estimated by FiPPP. Meanwhile, regardless of positioning and DCB/OSB parameters, other related results including tropospheric delay, ambiguity, model residuals, clock offset and ionospheric delay can also be obtained based on the FiPPP. To show the performance of FiPPP, useful scripts and visualization tool are also provided for solution presentation, such as PDOP, convergence time and equation residuals.

Features of FiPPP

The received system- and frequency-wide GNSS observations can be fully introduced into PPP solution, where the sequential least square filtering algorithm is used to estimated the epoch-wise unknown parameters. The observations of BDS-3, GPS and Galileo from single to five frequencies can be flexibly integrated into FiPPP software. For example, the random selection of BDS-3 signals B1C/B2a/B1I/B3I/B2 frequencies can be set in PPP estimation.

Main Functions

The main functions of current version of FiPPP software include:

* Supports single- to five-frequency uncombined PPP solution
* Supports GPS, BDS-3 and Galileo, and its combination
* Multi-frequency and multi-GNSS standard single point positioning
* Multi-GNSS dual- to five-frequency IF combined PPP
* System-wide and frequency-wide integrated GNSS PPP
* Multi-GNSS and multi-frequency kinematic PPP (under testing)
* Supports the selection of frequencies, such as BDS-3 IF combination with B1C/B2a+B1C/B2+B1I/B3I or B1C/B2a/B2+B1I/B3I
* Corrects GNSS observations by the issued DCB, OSB and IFCB products
* Estimation of GPS third frequency IFCB parameters
* Multi-frequency DCB and code OSB estimation and its format output
* Supports the single-difference dual-frequency uncombined partial PPP-AR; and UPD/IRC single-difference dual-frequency IF combined PPP-AR (under testing)
* Supports visualization with debug log and interface

Apart from the functions mentioned above, some interesting tools can also be obtained from FiPPP, such as: setting the orders of BDS-3 signals used, and [iGMAS GPS PIFCB products](http://igmas.users.sgg.whu.edu.cn/products/download/directory/products/upd) to correct GPS third frequency IFCB. The newly open-source FiPPP is continuously updated and improved. The structure of FiPPP software is drawn in Figure 1.

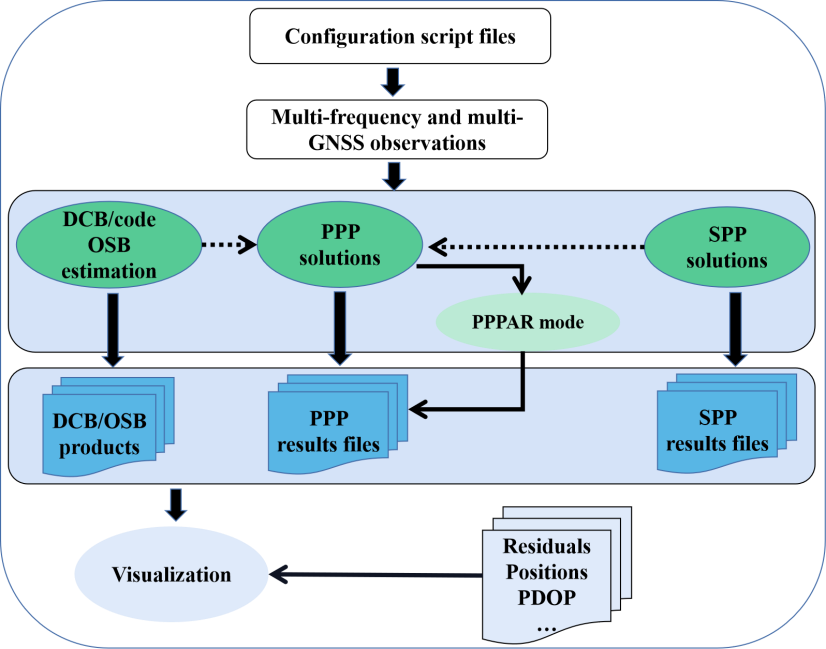


Figure 1 Main features of FiPPP software and its flowchart. The system- and frequency-wide PPP solution is supported by the flexible IF combination and UC.

To present more details of FiPPP software, the main functions and its processing methods are shown in Figure 2.

Figure 2 The main steps of FiPPP data processing procedure

Build new project and run FiPPP

Similar with other open-source C++ PPP software, the project management of FiPPP is organized by CMAKE platform tool. The Windows operating system is used to compile and run the FiPPP. It should be noted that the Eigen library is cited to conduct the matrix operations in the FiPPP. Before building a new FiPPP software, it is necessary to download and install the Windows version of [CMAKE](https://cmake.org/download/) and [Visual Studio](https://visualstudio.microsoft.com/zh-hans/downloads/) (VS) for VS users, and download and install [TDM-GCC](https://jmeubank.github.io/tdm-gcc/download/) for GNU users. Meanwhile, it is recommended to use Visual Studio IDE for VS users, and CLion IDE for run and debug the FiPPP. Note that the author uses JetBrains [CLion](https://www.jetbrains.com/clion/) 2022.3 software and Visual Studio 2022 Community software platform for developing and debug under Windows 11. The CMAKE version is 3.23.1 and the C/C++ compiler is gcc 10.3.0.

Build new project

* VS user

Firstly, download the FiPPP package from: GPS toolbox webpage.

Secondly, open the CMAKE GUI window. Select “*where is the source code:*” and “*Where to build the binaries:*”; the addresses of the defined folders should be created in the corresponding files. Then, click “*Configure*” and select the Visual Studio version in “*Specify the generator for this project*”. Click the button “*Finish*” to end the CMAKE configuration. As shown in Figure 3, click “*Generate*” and “*Open Project*”. You can also find the “FiPPP.sln” file in the “*build*” folder to open the project as shown in Figure 4.



(a) CMake GUI window



(b) Configure option

Figure 3 CMAKE GUI configure options for building new project



Figure 4 Configuring and Generating of new project

Builddone

Figure 5 Compiling the FiPPP software after building

After the completion of project building, user can open the “FiPPP.sln” project file under the “*build*” folder, or directly click “*Open Project*” as shown in Figure 5.

Step 1: set the “*FiPPP Solver*” as the first startup project as shown in Figure 6.



Figure 6 Setting the first startup project

Step 2: add the “*Command Arguments*” and “*Working Directory*” in the FiPPP Solver property page as shown in Figure 7. Then, users can find the “*FiPPP.conf*” under the “*FiPPP*” folder. It should be noted that user needs input “-*c FiPPP.conf*” in “*Command Arguments*”, and the path of “*FiPPP.conf*” should be added in “*Working Directory*”.



Figure 7 Inputting *Command Arguments* and *Working Directory*

Step 3: Because of the global stations used in DCB/OSB estimation, the “*Heap Reserve Size*” and “*Stack Reserve Size*” should be increased to “*1600000000*” as shown in Figure 8. *Note that the “Stack Reserve Size” value needs to be set according to your own computer configuration. If it is too large, the FiPPP cannot be started normally. If it is too small, the EsitimateBias function cannot be used.*



Figure 8 Modifying the reserve size

run

Figure 9 The Run window of FiPPP processing

* GNU user

Firstly, download the FiPPP package from: GPS toolbox webpage.

Secondly, load the FiPPP project, after cmake completed, you can add program arguments " -c FiPPP.conf" and add working directory "D:\FiPPP" (This path is the path of the "FiPPP.conf" file on your computer.) in Run/Debug Configurations, then the project is ready for running. The personal configuration as shown in Figure 3.

Note that you should install the MinGW on your Windows computer for gcc compiler.



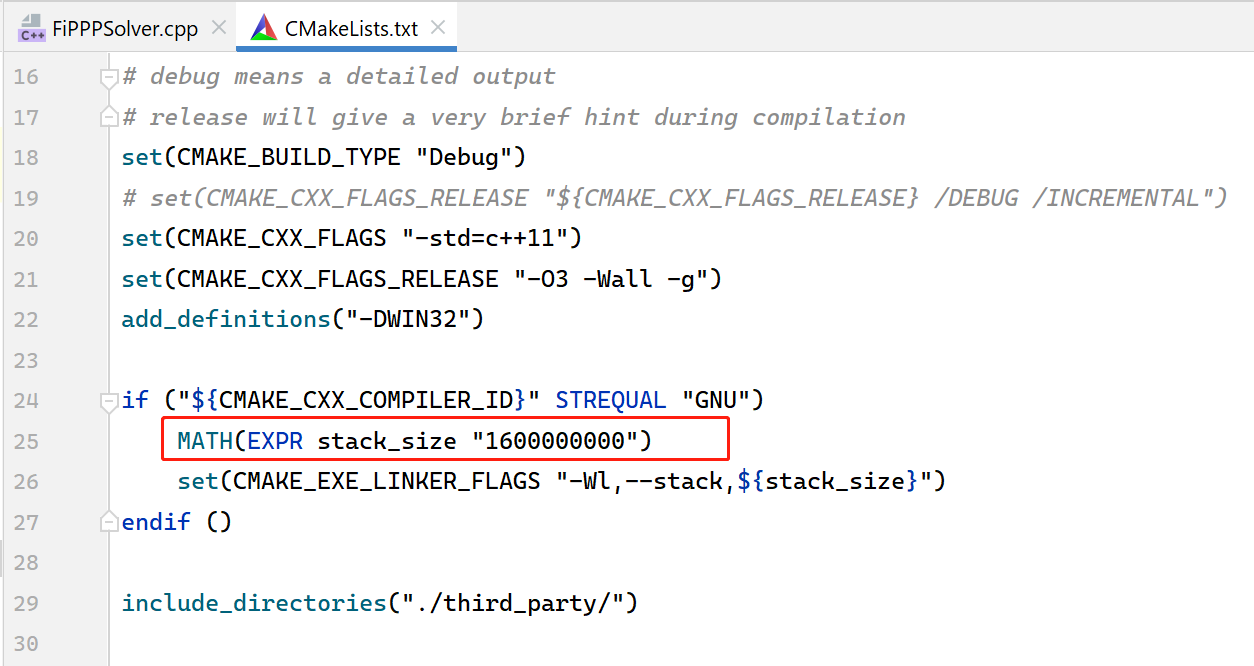
Figure 10 Personal configuration on CLion platform

1. Open project



Figure 11 Open project on CLion platform

Note that the "*stack\_size*" value needs to be set according to your own computer configuration. If it is too large, the FiPPP cannot be started normally. If it is too small, the *EsitimateBias* function cannot be used. For GNU users, it can be modified in '. / FiPPP / CMakeLists.txt ', as shown in the following.



2. Cmake



Figure 12 Cmake on CLion platform

3. Add DLL files

Since FiPPP links to dynamic libraries, users need to copy the DLL files in"./third\_party/CLAPACKlib/lapackbin/" to " ./ cmake-build-debug-tdm\_gcc103/src/FiPPPSolver " (This path is the path of "FiPPPSolver.exe" after "CMAKE" on your computer.).



Figure 13 Add DLL files

4. Configure



Figure 14 Setting Configures

5. Build



Figure 15 Build project

6. Run



Figure 16 Run project

FiPPP Configuration

Before running the FiPPP software, the configuration file should be set based on the users’ needs. The default processing options can be found in “*./ FiPPP /FiPPP.conf*”.

The configuration file mainly contains five sections:

I. *PPP processing strategies configuration*

II. *PPP-AR processing strategies configuration*

III. *Estimation and generation of DCB and code OSB product*

IV. *Input file configurations*

V. *Output file configurations*

An example of Configuration file is presented as Figure 17, where ‘#’ represents the annotation of function. Users can select different process options according to the positioning tasks and the received observations. It is very important to carefully check the correctness of parameters’ configurations before running FiPPP software. For example, if the users want to conduct the kinematics ionosphere-free (IF) combined PPP solution with GPS, Galileo and BeiDou (PPP\_KIN\_IF12\_GEC), the value of GNSS system, “PPPModel” and “ionoopt” should be set as GEC, 3 and 3, respectively.



Figure 17 An example of FiPPP software configuration file

Output file format description

The multi-GNSS and multi-frequency observations from IGS-MGEX stations can be input into FiPPP to conduct the PPP solution. User can download the precise satellite products of any GNSS Analyst Center (AC) to build the observation equations. The way to name the output files is based on the processing mode, combination strategy, GNSS system used, station name and the day of year (DOY) of observation files. For example, “*PPP\_KIN\_IF12\_GEC\_OFF\_ARHT2550.22O.pos*” means the dual-frequency IF kinematics PPP based on GEC observations from ARHT station during DOY 255, 2022, where *OFF* denotes the IF combination of BDS-3 newly modulated signals are not included in PPP solution. Furthermore, the final filename extension is different from the results, such as satellite elevation angle is “*ele*”, ambiguity is “*amb*”, troposphere “trop”, PDOP is “pdop”, pseudorange residual is “*resp*” and phase residual is “*resl*”, etc.

* **Position file: PPP\_KIN\_IF12\_OFF\_GEC\_BRUX2550.22O.pos**

Format: year month day hour minute second DOY SOD X Y Z E N U RMS Flag

Flag: 6: float solution, 7: fix solution.

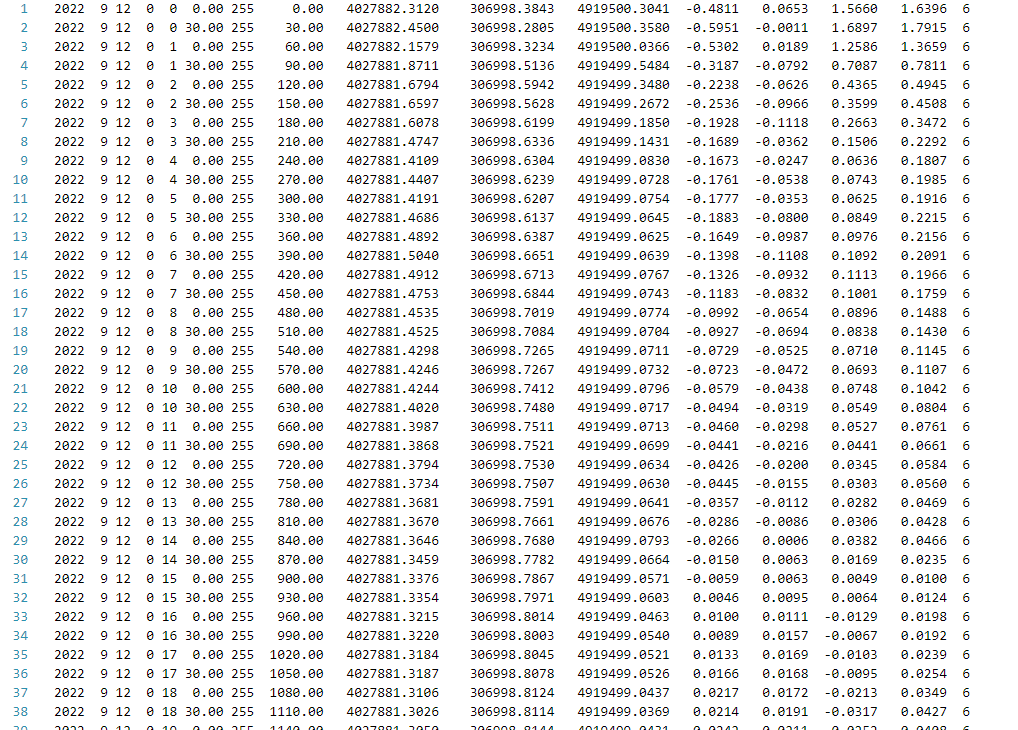


Figure 18 The format of positioning results file

* **Parameters file: PPP\_KIN\_IF1213\_E\_BRUX2550.22O.state**

"#" indicates the current epoch information; the first column is the satellite number, where an abbreviated description before each column is presented. For instant, “BLC15E” represents the phase ambiguity value of E1 and E5a dual-frequency IF combination; “CS” is the indication of cycle slip detection status, where “1” denotes a cycle slip of current epoch is occurred; “LC15E” represents the phase equation residual of E1 and E5a dual-frequency IF combination PC15E represents the residual value of the corresponding pseudorange equation; azimuth represents the satellite azimuth angle in degrees; dtSat represents the satellite clock in seconds; elevation represents the satellite elevation angle.

For other individual parameter files, the interpretations are consistent with this document, and will not be described one by one.

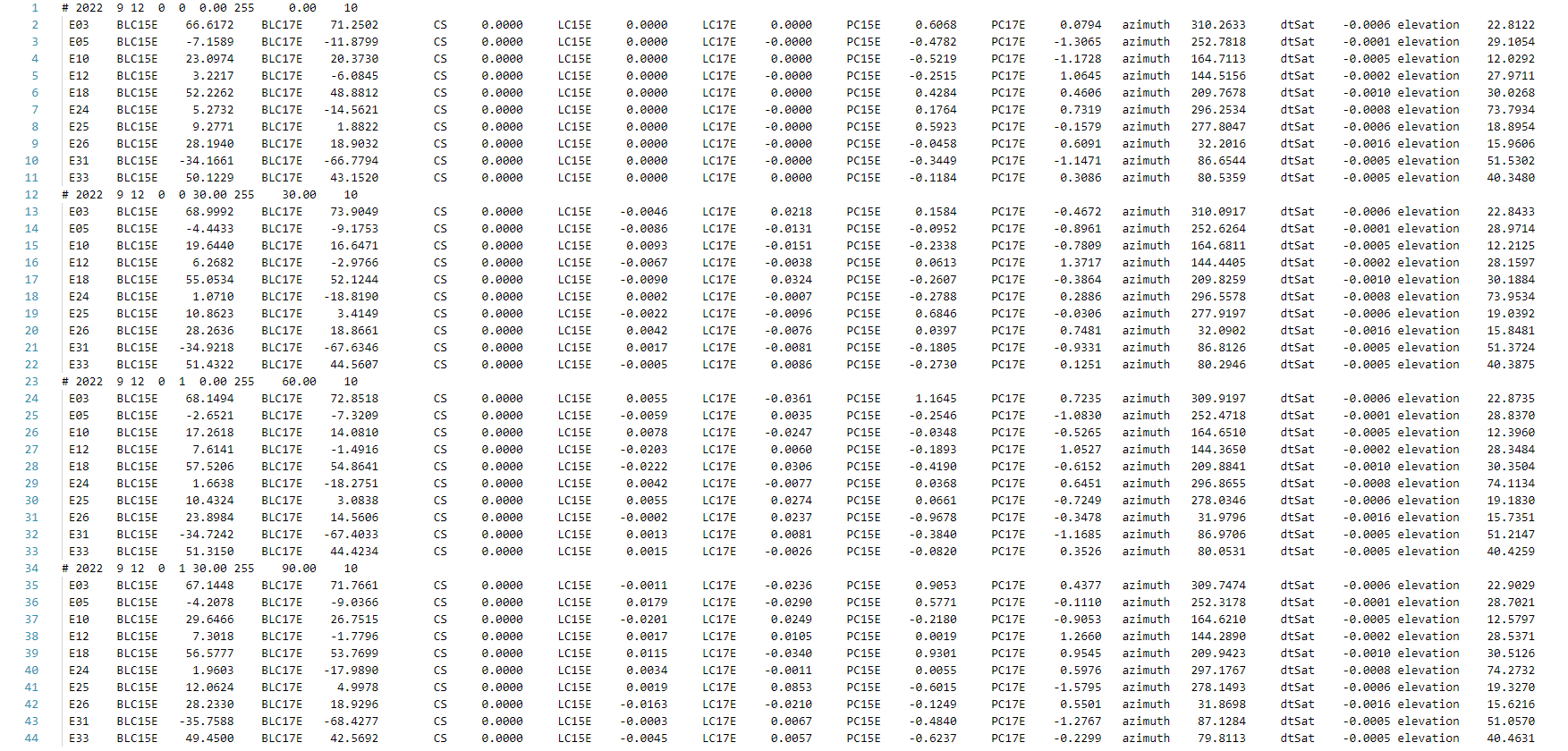


Figure 19 The format of parameter results file

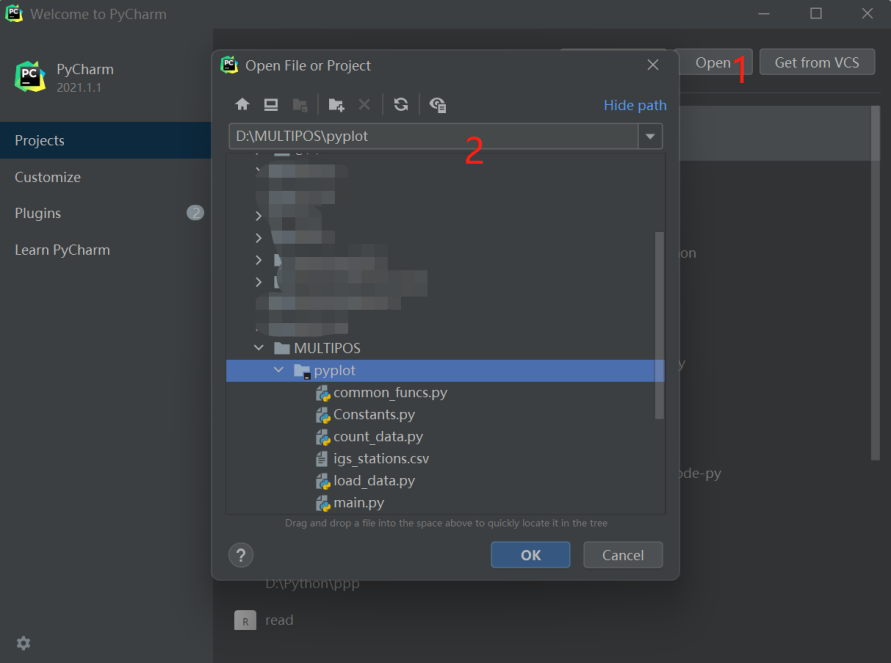
Data Visualization

Apart from the result files, FiPPP software also provides some useful python scripts to show and compare the estimated parameters, which can be found at “*folder/tools*”. These tools can be compiled by pycharm or visual studio code editor.

Build project

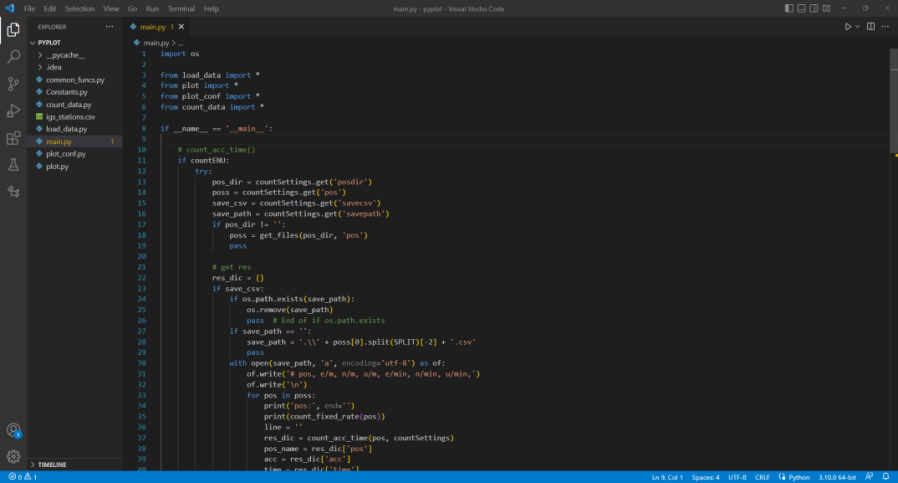
* **pycharm user**

Open “pycharm” and open “pyplot” or open the “pyplot” folder with pycharm.



* **visual studio code user**

Open the pyplot folder with visual studio code



Configuration

The default processing options can be found in “*./pyplot/plot\_conf.py*”.

The configuration file includes the functions of counting convergence accuracy and time, drawing station location, showing positioning residuals, presenting satellite sky map and tropospheric delay values, PDOP values, ambiguity values, pseudorange and phase residual values, etc.

* **Count convergence accuracy and time**

According to the corresponding options set by the user, the convergence accuracy and time of each station in the E, N, and U directions can be obtained statistically as shown in Figure 20. It can be generated and stored as a “CSV” file according to user requirements as Figure 21.

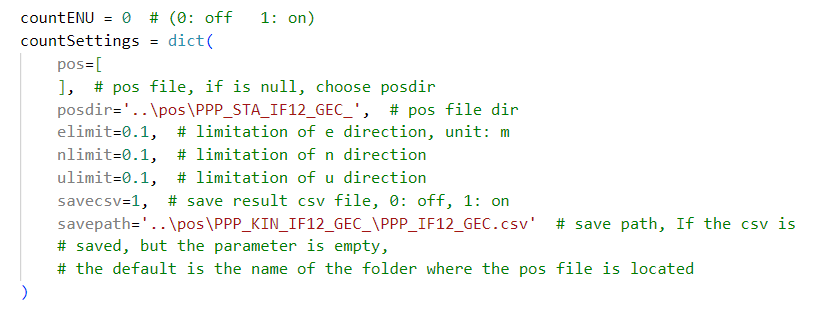


Figure 20 Part of configuration file to count E, N and U information



Figure 21 Results of E, N and U information in CSV file

* **Plot stations distribution**

The user can input the folder where the observation file is located or the name of the observation station to draw the location of the observation station as Figures 22 and 23.

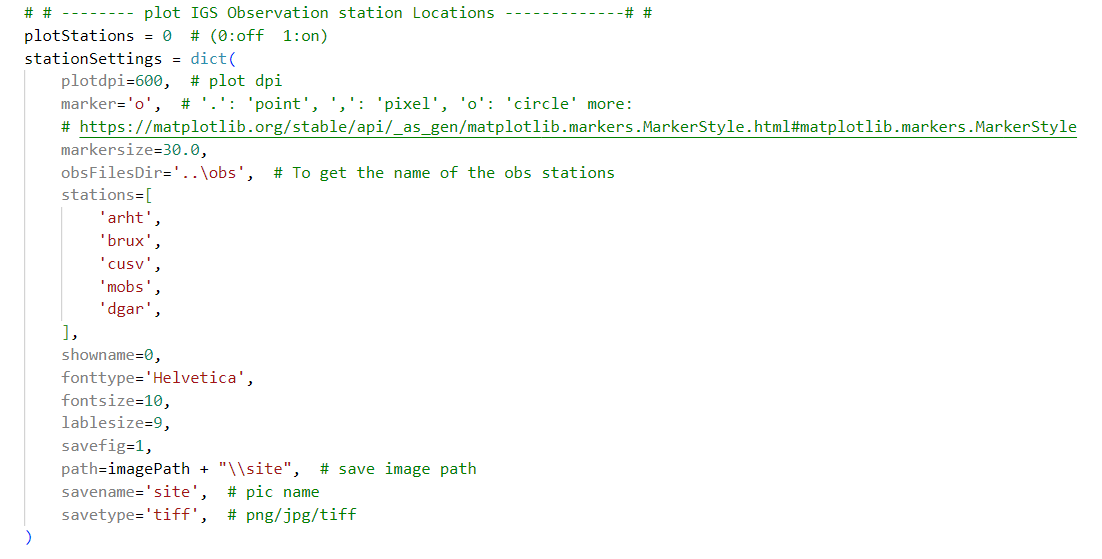


Figure 22 Part of configuration file for plotting station distribution



Figure 23 Stations distribution used in PPP solution

* **Plot positioning residuals**

Take the observation of ARHT station in DOY 255, 2022 as an example. The positioning residuals series can be presented as Figures 24 and 25. It can not only plot single positioning result file of single station, but also draw multiple positioning result files of single station. The legend is automatically generated according to the generated positioning result file name.

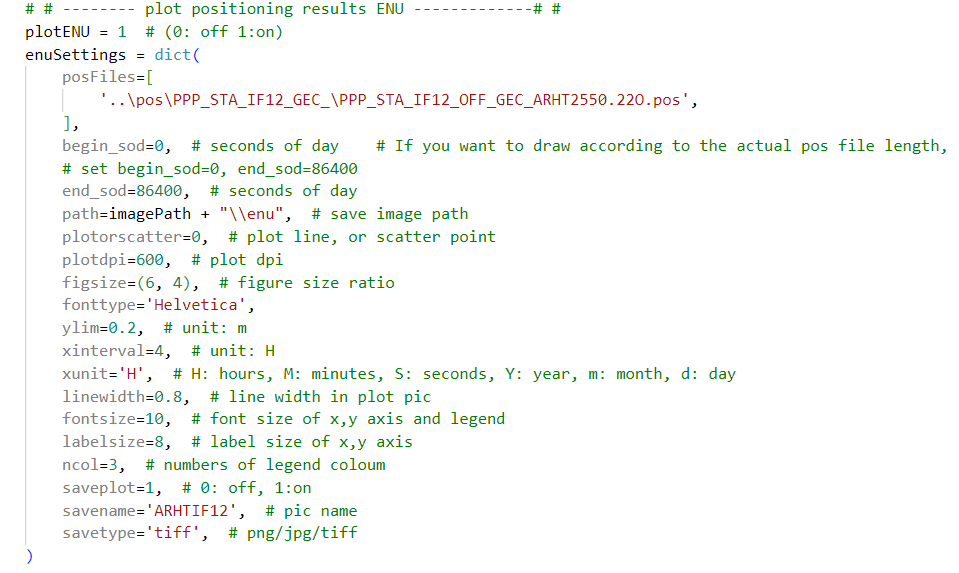


Figure 24 Part of configuration file for presenting positioning residuals



(a) Single positioning residual of ARHT



(b) Multi-positioning residual of BDS-3 five-frequency IF combination

Figure 25 The positioning residual of FiPPP software in DOY 255, 2022

* **Plot satellite sky map**

User can choose the “ele” file to draw satellite sky maps as follows.

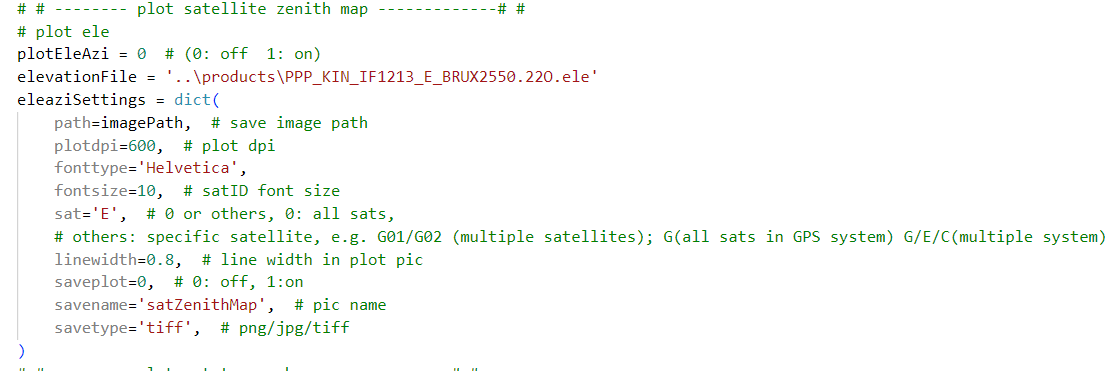


Figure 26 Part of satellite sky map in configuration file



Figure 27 Galileo satellites sky map of BRUX station

* **Plot PDOP value**

The user can choose the “pdop” file to draw the PDOP series.

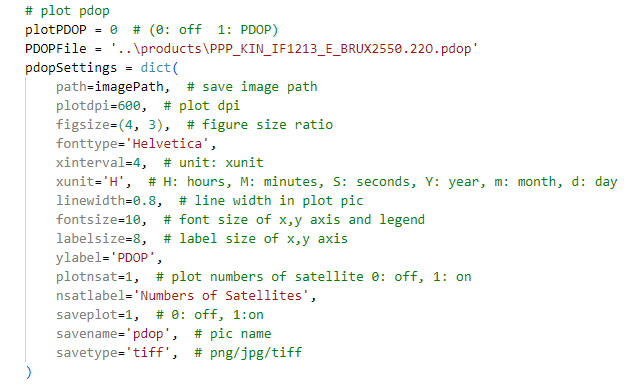


Figure 28 Part of plotting PDOP in configuration file



Figure 29 PDOP and numbers of satellites

* **Plot PPP model residual**

The user can choose the “resp/resl” file to draw the residual series of PPP model. Because of too many participating satellites, the legend is not displayed, where one color represents a satellite in figure.

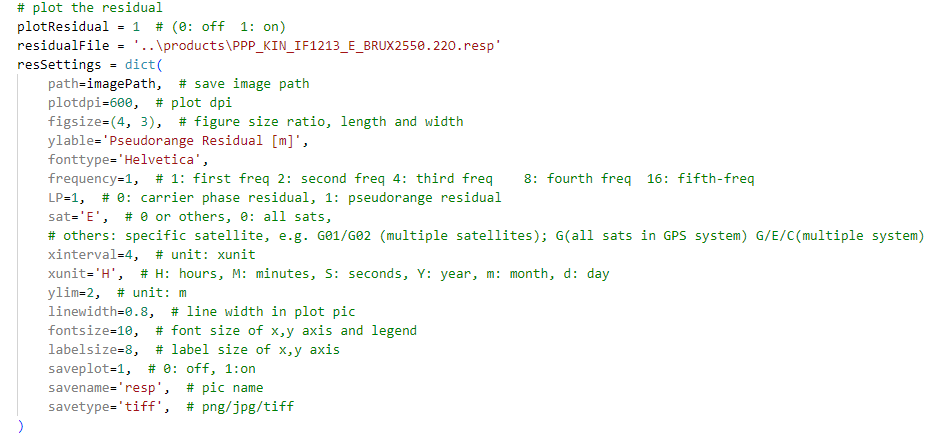


Figure 30 Part of plotting PPP model residual in configuration file



Figure 31 The pseudorange residual of PPP model

Structure of Software

**Main Function**

The main functions of FiPPP software include 15 important parts, which is listed as following:

1. Read Configuration File

2 Scan Files folder

3. Estimate DCB/Code OSB

3. Read Observation File

4. Match Product Files

5. Read Product Files

6. Scan and Modify Observation Data

7. Gross detection

8. Single Point Positioning

9. Cycle Slip Detection

10. Correct Bias

11. Form Observation Equations

12. Initialize Matrixes and Ambiguities

13. Parameters Estimation with Sequential Least Squares Filtering

14. Generate Result Files

15. PPP-AR with Integer Least Squares

**Important Subroutines**

**AboutOptions.hpp/cpp**

*Input parameters:*

The address of configuration file

Example: ./FiPPP.conf

*Output parameters:*

A variable named cfgOpts, which stores all parameters in the configuration file.

*Introduction:*

This subroutine is used to read the configuration file defined by the user in a flexible form. In addition, the internal method is also called to determine whether the currently opened observation file meets the need of user in the configuration file. The internal method returns a logical variable, and if it is not satisfied, the observation file is closed and the relevant information is printed.

**FindFile.hpp/cpp**

*Input parameters:*

cfgOpts, “vector<string> files” or “string file”

*Output parameters:*

The aim files which type is “string” or “vector<string> files”.

*Introduction:*

This subroutine is used to find the files in the folder which is stored in the cfgOpts. This subroutine is mainly used to scan the file folder, obtain all observation files and record them in the corresponding variables; obtain all precise product files under the folder and record them in the corresponding variables. The corresponding product file is matched according to the observation time of the observation file.

**EstimateBias.h/c**

*Input parameters:*

The address of configuration file

*Output parameters:*

Formatted output DCB and Code OSB files.

*Introduction:*

According to the parameter settings of the EstimateBias module in the configuration file. This subroutine is used to estimate DCB and Code OSB. In this subroutine, firstly, the P4 observations are smoothed, and secondly, the quality control is performed to eliminate the observations with poor observation quality. Then, set the equation. Finally, the least squares algorithm is used to obtain the results and format the output.

**Rinex3ObsStream.hpp/cpp**

*Input parameters:*

Address of observation file

*Output parameters:*

A class object with storing observation data and information.

*Introduction:*

The observation file is read through the file input stream, and the header information of the observation file can be decoded.

**CommonTime.hpp/cpp**

*Input parameters:*

Year, Month, Day, Hour, Minute, Second

*Output parameters:*

Modified Julian Date (MJD)

*Note:*

The routine converts a calendar date with hour, minute and second to a Modified Julian date. At the same time, set the corresponding time system.

**Rinex3EphemerisStore.hpp/cpp**

*Input parameters:*

Broadcast ephemeris file address

*Output parameters:*

A class object with storing broadcast ephemeris data and information.

**SP3EphemerisStore.hpp/cpp**

*Input parameters:*

Precision clock and orbit file addresses

*Output parameters:*

A class object with storing precise clock and orbit ephemeris data and information.

**EOPDataStore.hpp/cpp**

*Input parameters:*

ERP (Earth Rotation Parameters) file addresses

*Output parameters:*

A class object with storing ERP data and information.

**BiasDataReader.hpp/cpp**

*Input parameter:*

DCB file or OSB file address.

*Output parameters:*

A class object with storing “bias” data and information.

*Note:*

This subroutine is used to read the “bias” file and store the data for subsequent program processing.

**IFCBDataReader.hpp/cpp**

*Input parameters:*

iGMAS GPS ifcb file address.

*Output parameters:*

A class object with storing GPS third pifcb bias data and information.

*Note:*

This subroutine is used to read data. In addition, in the PPP process, it uses its own internal methods to correct the observations.

**UPDDataReader.hpp/cpp**

*Input parameters:*

COD or WUM UPD (Uncalibrated Phase Delays) files (EWL, WL, NL) addresses

*Output parameters:*

A class object with storing the UPD information.

*Note:*

This subroutine is used to read data. In addition, for the PPP-AR process, it uses the internal methods to correct the observations.

**XYZStore.hpp/cpp**

*Input parameters:*

IGS weekly coordinate file or the “.crd” file which stores the observation station real position information containing the time, x, y, and z.

*Output parameters:*

A class object with storing the observation station real position information.

**DecimateData.hpp/cpp**

*Input parameters:*

The sampling interval, the first and last epoch time in the configuration file.

*Introduction:*

This subroutine is used to determine whether the current epoch is within the input start and end epoch. If the current epoch is not in the range, the program skips the current epoch data and does not execute the subsequent program.

**KeepSystems.hpp/cpp**

*Input parameter:*

The required satellite system set by the user in the configuration file.

*Introduction:*

This subroutine is used to judge and eliminate the data of the satellite system that does not belong to the user setting in the current epoch data.

**FilterCode.hpp/cpp**

*Input parameter:*

GNSS epoch-wise observation

*Note:*

This subroutine is mainly used to detect and eliminate gross errors.

**SPP.hpp/cpp**

*Input parameters:*

GNSS epoch-wise observation and broadcast ephemeris data.

*Output parameters:*

Receiver antenna approximate coordinate values, x, y, z and receiver clock.

*Introduction:*

This subroutine computes the initial position of receiver antenna and receiver clock using pseudorange C1 based on the single point positioning.

**RequiredObs.hpp/cpp**

*Input parameters:*

GNSS epoch-wise observation and cfgOpts.

*Introduction:*

This subroutine is used to judge and eliminate the observation satellite data that does not meet the conditions of constructing the equation in the current epoch. For example, in the IF123 mode, it is necessary to satisfy that the observation satellite has three frequency data, in which the program will eliminate the observation satellite data without three frequency data.

**ComputeSatPos.hpp/cpp**

*Input parameters:*

SP3EphemerisStore class object and GNSS epoch data.

*Introduction:*

This subroutine is used to calculate the position and velocity of the satellite by using the precise satellite ephemeris. At the same time, these result data are added to the GNSS epoch-wise observations.

**LinearCombinations.hpp/cpp**

*Note:*

This module is used to store the combination coefficient of the observation equation or other combination information.

**ComputeCombination.hpp/cpp**

*Introduction:*

This subroutine is used to calculate the combined equation related to the linear combination, such as IF, MW, and GF combination observations, etc.

**CombinationOptions.hpp/cpp**

*Input parameters:*

cfgOpts, requrieObs, ComputeCombination class objects

*Introduction:*

This subroutine is used to add LC type and coefficient to the relevant processing subroutine, such as the coefficient of IF combination observation equation.

**DetectCSMW.hpp/cpp, DetectCSGF.hpp/cpp, MarkArc.hpp/cpp**

*Input parameters:*

GNSS observations and linear combination coefficients.

*Introduction:*

These subroutines are used to handle cycle slip detection. DetectCSMW and DetectCSGF are used to detect cycle slips with MW combined observations and GF combined observations, respectively. If the data has a cycle slip, MarkArc further marks and resets the ambiguity.

**SolverPPPAll.hpp/cpp**

*Input parameters:*

GNSS observations and cfgOpt.

*Introduction:*

This subroutine is used to deal with PPP estimation. In this program, the observation equation is firstly listed; then, the transfer matrix of sequential least squares filtering is prepared, and the noise matrix and weight matrix of filtering process are prepared; the gain matrix of filtering process is calculated. Accordingly, sequential least squares filtering is performed; finally, the estimated parameter results required by the user are formatted.

**SovlerPPPAR.hpp/cpp**

*Input parameters:*

UPD data, PPP solutions and current epoch equations.

*Introduction:*

This subroutine is used to handle PPP-AR. It is an inter-satellite single-difference IF combination PPP-AR, which is mainly fixed by partial ambiguity search. It supports LAMBDA and ROUND to fix narrow-lane ambiguity. In this subroutine, quality control (QC) and checking the PPP results accuracy of current epoch are firstly conducted. Then, the ROUND method is used to fix the inter-satellite single-difference wide-lane ambiguity. In addition, the inter-satellite single-difference narrow-lane ambiguity is constructed. Thus, LAMBDA or ROUND method is used to fix the narrow lane ambiguity through QC. Finally, the fixed solution is solved by integer least squares, and the positioning results are updated and output.

Appendix

PPP Models

In this section, the fundamental GNSS observation equations are firstly introduced. Then the single- to five-frequency uncombined (UC) and ionosphere-free (IF) PPP model are derived in details. For convenience, we record the frequency order of GPS, Galileo and BDS respectively as shown in Table 1. Moreover, FiPPP can offer a new frequency order option for BDS-3 by consideration of new signals.

Table 1 The frequency order of GPS, Galileo and BDS systems

|  |  |  |
| --- | --- | --- |
| Constellation | Frequency order | Notes |
| GPS | L1/L2/L5 | Normal |
| Galileo | E1/E5a/E5b/E5/E6 | Normal |
| BDS-2 | B1I/B3I/B2I | Normal |
| BDS-3 | B1I/B3I/B1c/B2a/B2 | Normal |
| BDS-3 | B1c/B2a/ B1I/B3I/B2 | New option |

General Observation model

The raw observation equations of GNSS satellites can be written as[1]



where *L* and *P* are the carrier phase and code observations, respectively; *s*, *r* and *fi* denote the satellite, receiver and the *i*-th frequency; *ρ* is the geometrical distance between satellite and receiver, which can be also rewritten as the multiplying of direction and positions vectors, namely ***u****rs·****x***; moreover, d*tr* and d*ts* are the clock offset of receiver and satellite; *Tr* and *Ir*,1 are the tropospheric delay for receiver and slant ionospheric at the *f*1 frequency, where *κ* is the ionospheric factor; *N* represents the ambiguity parameters; *λ* is the wavelength; *b* and *d* mean the uncalibrated delays for phase (UPD) and code (UCD) observations; *ε* and *ζ* are the model noise of phase and code equations, respectively. It should be mentioned that the satellite clock offsets are estimated and issued by GNSS Analyst Center (AC), such as GBM products from German Research Centre for Geosciences (GFZ), the UCD of satellite is included in the clock products. For convenience, the coefficients of the IF combination are defined as



In general, the satellite clock products are generated by using the first and second frequency (e.g., GPS L1/L2, Galileo E1/E5a and BDS B1I/B3I) IF combination; thus, the satellite clocks absorb the IF satellite UCD; thus, the IF satellite clock error is defined as



It is reported that GPS and BDS-2 satellites have significant time-varying characteristics of UPD, of which the BDS-2 can be ignored in regarding BDS-2 and BDS-3 as BDS[2]. The GPS satellite clock products is defined as



In addition, the UPD parameters of GPS satellites are divided into constant and time-varying parts, which are defined as



whereandare the constant part and the time-varying parts of UPD, respectively.

Because of the time-varying part of GPS UPD, the bias will be occurred by using signals of different frequencies for the dual-frequency IF combination precise clock estimation, which is called the inter-frequency clock bias (IFCB)[3]. The bias consists of the receiver and satellite code hardware bias and phase hardware bias time-varying components, which causes the code-based IFCB (CIFCB) and PIFCB, respectively. In FiPPP software, we can use iGMAS GPS PIFCB products and set the unknown parameters to correct the GPS IFCB.

For convenience, we define the variables as



After the correction by precision products and related models, the parameters are rewritten; thus, the observation equation of the UC PPP model can be read as



where



where is the inter-frequency bias (IFB),is IFCB and exists only in GPS.

As mentioned above, FiPPP software can use the iGAMS GPS PIFCB products to correct the , of which the Equation (7) can be written as



In addition, FiPPP can parameterize the GPS IFCB using a random walk model for estimation. The equation of estimate GPS IFCB can be written as



where



The above model can be applied not only on GPS, but also for Galileo and BDS systems. However, it cannot be applied to IF combination of equation as the number of unknown parameters to be estimated larger than that of equations.

Therefore, for a geometry-based ionosphere-free (GBIF) model based on the combination of multi-frequency satellite observations, the original combination form of multi-frequency observations can be simply expressed as



where, *e* is the combinations coefficient of different frequencies. Equation is the combined function of multi-frequency observations with *n* frequencies. To meet the requirements of high-performance combination, the Equation will be set as constrained equations and solved by the Lagrange multiplier method.



where each system can independently estimate a group of *e* based on the received frequencies to construct an optimal IF combination. The estimated ***α****f* can be read as



where *n* is the total numbers of frequencies; and *m* = 2, 3, … *n*. The Equation is the expressions of multi-frequency combination coefficients[1]. For example, the coefficients of five-frequency combination can be written as Equations .



BDS-3 New Observation Combination Model

The frequency order B1c/B2a/B1I/B3I/B2 for BDS-3 option will be analyzed in this section. Not only the traditional PPP models but two new IF combination models, namely three dual‑frequency IF combined observables (B1c/B2a, B1c/B2, B1I/B3I, named CCI), single triple‑frequency and dual‑frequency IF combined observations (B1c/B2a/B2, B1I/B3I, named CI)[4] is discussed.

* General PPP model



where



We can obtain the traditional ionosphere-free combined PPP model according to the above method.

* CCI



where



The general observation model in the CCI model can be replaced by equation , which means that the CCI model can be corrected without DCB or OSB products.

* CI



where



Examples

* **Data Processing Strategies**

We select the GPS, Galileo and BDS observation of 20 MGEX stations in DOY 255, 2022 (sampling interval 30s), which is listed as Figure 32. Users can find the dataset and some examples under './test\_case/files'. The dataset includes required files to performance PPP solutions. The precise satellite orbit and clock products used in PPP experiments provided by the GFZ AC. The DCB product is provided by the CAS AC and the Observable-specific Signal Bias (OSB) product is provided by WHU AC. In experiments, static and kinematic PPP are tested. The coordinate parameters of the static PPP solution are constant estimates, while the coordinate parameters of the kinematics PPP are set as white noise model. Since the similar frequencies for B1I and B1C, and the large combined noise, FiPPP software adopts the combination of B3I/B1C and B2a/B1I in the IF PPP model. Strategies for PPP solution can refer to Table 2.

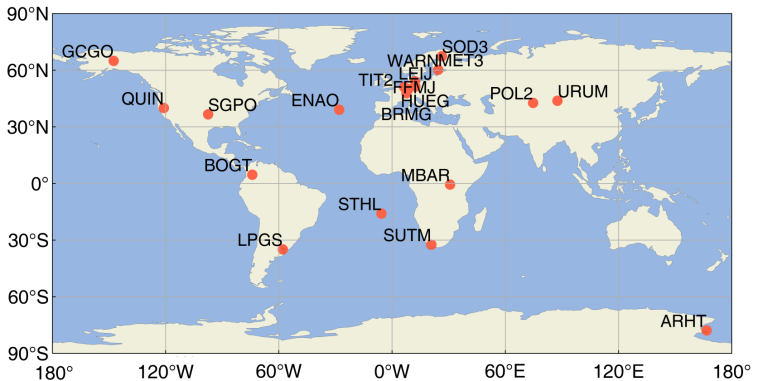


Figure 32 The stations distribution of PPP experiments

Table 2 Strategies used in PPP experiments

|  |  |
| --- | --- |
| Items | Strategy |
| Data source | MGEX stations observation data in DOY 255, 2022 (sampling interval 30s) |
| Satellite elevation mask | 10° |
| Estimator | Sequential Least Squares filtering algorithm |
| Weighting scheme | Elevation-dependent weight; 0.003m and 0.3m for GPS and Galileo raw phase and code, respectively; 0.006m and 0.6m for BDS raw phase and code, respectively. |
| PCO/PCV | igs14\_2196.atx |
| DCB correction | DCB products from CAS |
| OSB correction | OSB products from WUM |
| Satellite orbit and clock | Products from GFZ |
| Tropospheric delay | Zenith Hydrostatic Delays (ZHD) are corrected using the Saastamoinen model, and Zenith Wet Delays (ZWD) are estimated using random walk |
| Ionosphere delay | Estimated using random walk in UC PPP model |
| Tide effect | Solid Earth, pole and ocean tide |
| Station coordinates | Static: estimated as constants; kinematic: estimated using white noise process |
| Receiver clock | Estimated using white noise process |
| Inter-frequency bias | Estimated using random walk |
| Inter-frequency clock bias | Estimated using random walk |
| Ambiguity | Estimated as a constant |

* **DCB/Code OSB estimation**

FiPPP software can process the GPS frequencies of C1C/C1W/C2W/C5Q/C5X, the BDS frequencies of C2I/C6I/C7I/C1X/C5X/C7Z/C8X, and the Galileo frequencies of C1X/C5X/C7X/C8X/C6C. We compare the results estimated by FiPPP with the products of CAS/WUM based on multi-frequency DCB and code OSB. After the preparation of multi-GNSS and multi-frequency observations, examples of configuration file, running window, and DCB/code OSB result files can be seen in Figure 33.

**Estimate Bias parameters in configuration file:**



**Information in CUI:**



**DCB/OSB contents:**

|  |  |
| --- | --- |
|  |  |
| DCB | OSB |

Figure 33 Examples of configuration, running window, and DCB/ OSB result files

Furthermore, performances of multi-GNSS and multi-frequency DCB and code OSB are analyzed as Figures 34 and 35, respectively. In Figure 34, the horizontal axis represents the satellite PRN; and the vertical axis is the residuals between the FiPPP estimation and the precise AC values. In Figure 34, the accuracy of DCB parameter are referred to CAS AC. It is indicated that GPS DCB residuals are less than 1ns, except C1C-C2W of G15, in which most of DCB residuals are lower than 0.5ns; Galileo DCB residuals are less than 0.6ns, except C1X-C8X of E03, in which most of DCB residuals are lower than 0.3ns; BDS-2 DCB residuals of C1X-C8X are almost less than 2ns; and most of BDS-3 DCB residuals of C2I-C6I are less than 2ns. The observations used to estimate BDS-2 DCB are limited by selected stations, which may cause the lower accuracy of FiPPP estimation.

(a) GPS DCB residuals (b) Galileo DCB residuals

(c) BDS-2 DCB residuals (d) BDS-3 DCB residuals

Figure 34 The DCB residuals between FiPPP estimation and CAS AC products

And in Figure 35, the code OSB residuals of different frequencies are shown, which is the difference between the FiPPP estimation and WUM precise products. Similarly, it is indicated that some of GPS code OSB residuals are large than 20ns, which the C5Q and C5X is presented as a good accuracy; most of Galileo code OSB residuals are less than 4ns, except the estimated C8X and C6C Code OSB of E10, E11, E12, E18 satellites; all types of BDS-2 code OSB residuals are exceeded 8ns; and some of BDS-3 code OSB residuals for C1X and C5X exceeds 6ns. The reasons causing the gross errors of code OSB are the limitation of estimated stations and sub-optimal level of parameter constraints.

(a) GPS code OSB residuals (b) Galileo code OSB residuals

(c) BDS-2 code OSB residuals (d) BDS-3 code OSB residuals

Figure 35 The OSB residuals between FiPPP estimation and WUM products

In addition, to show the accuracy of estimated DCB/OSB, Figure 36 plots the averaged residuals of all satellites. It is found that the DCB parameters are good consistency with the publicly downloading products, while some of the OSB values deviates from the normal values. The constraint condition of OSB estimated will be further tested to increase the consistency of multi-frequency OSB.

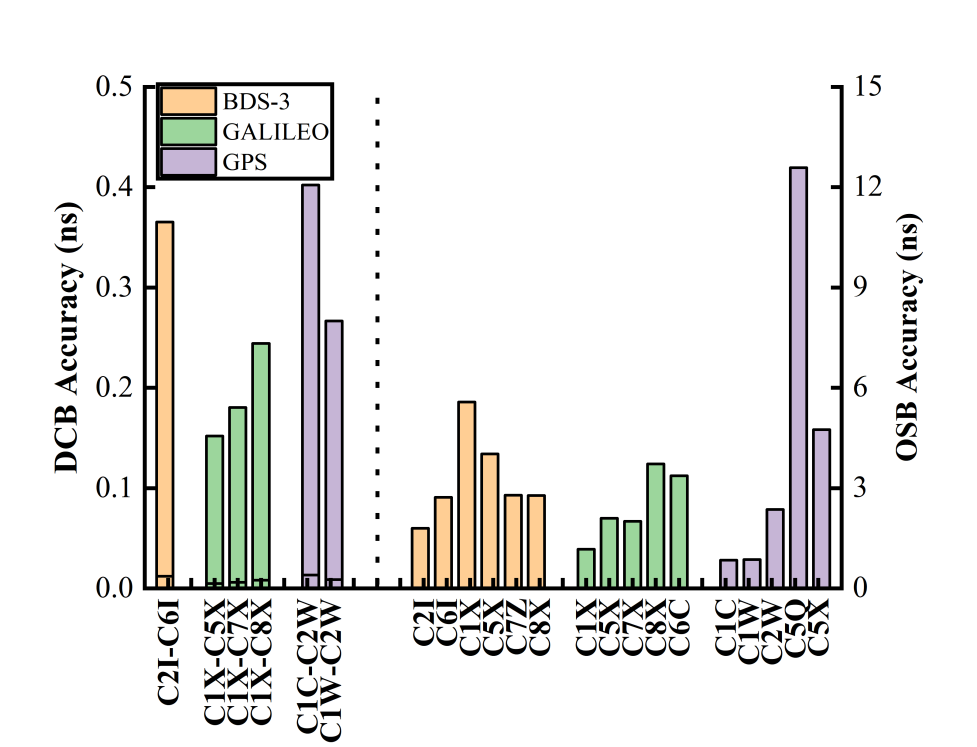


Figure 36 The averaged DCB and code OSB residuals for different GNSS systems

To assess the model of biases processing, two methods for biases correction are compared in FiPPP, namely estimation and precise products. The results of dual- and five-frequency GNSS PPP are shown in Figure 37 and 38. It should be noted that the EST denotes the estimation strategy for five-frequency PPP solution. According to the results, the EST strategy can reduce the positioning errors and convergence time compared with the using of precise products for five-frequency IF and UC PPP. However, the similar performances can be obtained with dual-frequency and five-frequency PPP, which means that some unmodelled errors in multi-frequency equations should be further processed. The FiPPP software provides a useful tool to conduct the system- and frequency-wide researches.

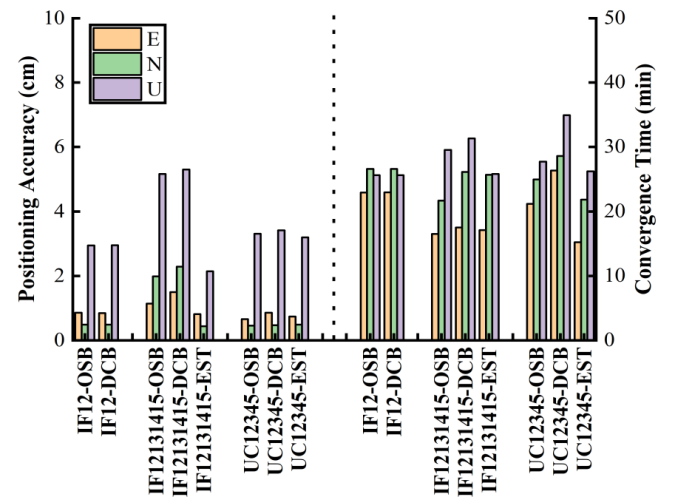


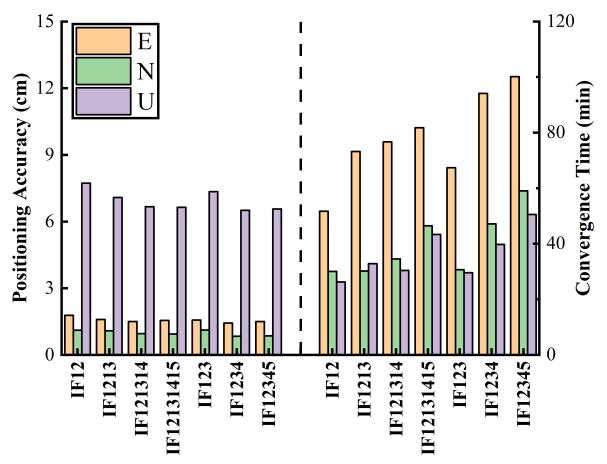
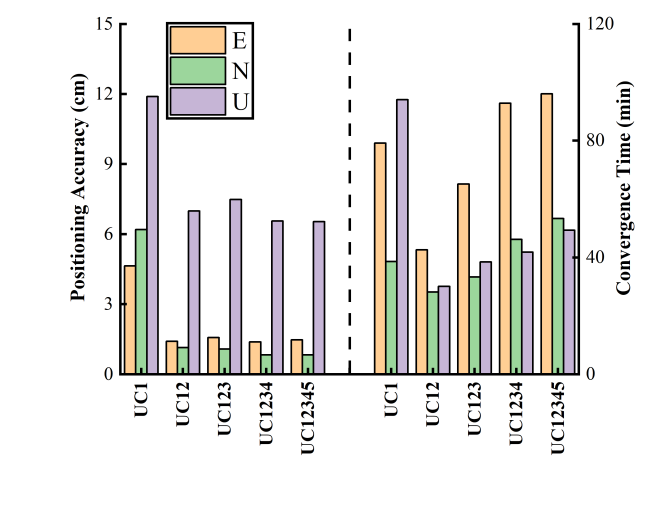
Figure 37 The positioning accuracy and convergence time of PPP solution based on different biases correction methods

|  |  |
| --- | --- |
|  |  |
| (a) IF PPP models positioning residuals at station ARHT and STHL | |
|  |  |
| (b) UC PPP models positioning residuals at station ARHT and STHL | |
| Figure 38 Positioning errors with IF-PPP (top) and UC-PPP (bottom) with different model in processing bias at station ARHT and STHL, respectively | |

* **Multi-GNSS and multi-frequency static PPP analyses**

Firstly, because of the ability of single- to five-frequency PPP, the advantages of multi-frequency observations are tested by FiPPP software as:

Scheme 1: the five-frequency observations of BDS-3 are used to test the performance of FiPPP software, in which the single- to five-frequency positioning is conducted with IF combination and UC mode, respectively. The positioning accuracy and the corresponding convergence time of BDS-3 single- to five-frequency static PPP are presented in Figure 39 and 40. It is shown that the similar centimeter-level positioning accuracy for most of experiments can be obtained, which indirectly verifies the validity and reliability of FiPPP. As the increasing of frequency used, the positioning accuracy presents an improved trend, while the convergence time is slightly worse as the equation noise amplification with multi-frequency observations, especially in E direction, which will be further solved by the partial PPP-AR algorithm.

(a) BDS-3 IF combination PPP (b) BDS-3 UC mode PPP

Figure 39 The averaged positioning accuracy and convergence time of BDS-3 single- to five-frequency static PPP solutions based on IF combination and UC mode

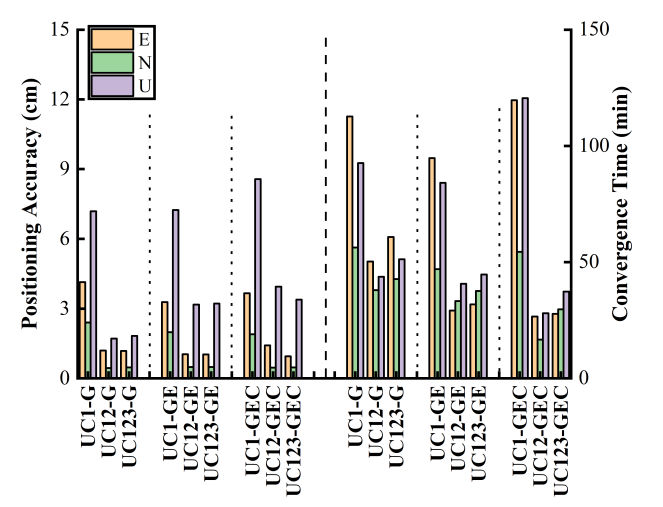
|  |  |
| --- | --- |
|  |  |
| (a) IF PPP models positioning residuals at station ARHT and STHL | |
|  |  |
| (b) UC PPP models positioning residuals at station ARHT and STHL | |
| Figure 40 Positioning errors with IF- (top) and UC-PPP (bottom) models for BDS-3-only at station ARHT and STHL | |

Scheme 2: apart from the multi-frequency positioning, the benefits of multi-GNSS observation are also verified. The combinations of different systems are analyzed as G, G+E, G+C+E with single- to triple-frequency observations, respectively. Similarly, the IF combination and UC mode of PPP is conducted, where two forms of triple-frequency IF, namely IF123 and IF1213, are further analyzed. Figure 41 shows the positioning errors of stations ARHT and STHL. According to Figure 41, it can be seen that the result of the UC1 PPP model under single GPS is poor, but it is improved when Galileo and BDS are added, but it is not completely corrected, because the current version of the QC algorithm has not been perfected. Table 3 and Figure 42 summarize the averaged positioning residuals and convergence time for E, N and U directions. According to the results in Table 3 and Figure 42, it is suggested that the increasing of GNSS system can reduce the positioning errors and convergence time. However, some of PPP performances are slightly worse than that of GPS-only solution, especially for IF combinations, in which the ISB and receiver clock parameters will be redefined.

|  |  |
| --- | --- |
|  |  |
| (a) IF PPP positioning residuals for GPS-only at station ARHT and STHL | |
|  |  |
| (b) IF PPP positioning residuals for GPS and Galileo at station ARHT and STHL | |
|  |  |
| (c) IF PPP positioning residuals for GPS, Galileo and BDS at station ARHT and STHL | |
|  |  |
| (d) UC PPP positioning residuals for GPS-only at station ARHT and STHL | |
|  |  |
| (e) UC PPP positioning residuals for GPS and Galileo at station ARHT and STHL | |
|  |  |
| (f) UC PPP positioning residuals for GPS, Galileo and BDS at station ARHT and STHL | |
| Figure 41 Positioning errors with IF and UC-PPP models for different satellite system at station ARHT and STHL | |

Table 3 The averaged positioning accuracy and convergence time of multi-GNSS PPP

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PPP strategies | Positioning Accuracy(cm) | | | Convergence Time (min) | | |
| E | N | U | E | N | U |
| UC1-G | 4.136 | 2.399 | 7.192 | 112.600 | 56.300 | 92.650 |
| UC12-G | 1.195 | 0.446 | 1.713 | 50.238 | 37.976 | 43.667 |
| UC123-G | 1.181 | 0.469 | 1.824 | 60.810 | 42.714 | 51.214 |
| UC1-GE | 3.281 | 1.984 | 7.244 | 94.750 | 47.000 | 84.125 |
| UC12-GE | 1.035 | 0.491 | 3.173 | 29.214 | 33.286 | 40.690 |
| UC123-GE | 1.027 | 0.493 | 3.217 | 31.857 | 37.571 | 44.667 |
| UC1-GEC | 3.661 | 1.900 | 8.569 | 119.625 | 54.438 | 120.563 |
| UC12-GEC | 1.416 | 0.464 | 3.943 | 26.600 | 16.700 | 28.050 |
| UC123-GEC | 0.950 | 0.472 | 3.384 | 27.722 | 29.722 | 37.361 |
| IF12-G | 1.199 | 0.451 | 1.805 | 49.048 | 26.452 | 37.310 |
| IF123-G | 2.431 | 0.741 | 2.345 | 205.559 | 128.912 | 123.412 |
| IF1213-G | 1.517 | 0.600 | 2.104 | 58.476 | 35.595 | 43.238 |
| IF12-GE | 0.975 | 0.486 | 2.587 | 31.833 | 24.238 | 32.786 |
| IF123-GE | 2.475 | 4.253 | 12.013 | 68.421 | 62.316 | 61.053 |
| IF1213-GE | 1.972 | 3.336 | 7.299 | 33.762 | 30.190 | 50.952 |
| IF12-GEC | 0.843 | 0.486 | 2.948 | 22.972 | 26.611 | 25.639 |
| IF123-GEC | 2.371 | 2.838 | 10.882 | 71.967 | 45.500 | 41.867 |
| IF1213-GEC | 1.895 | 2.947 | 6.555 | 27.306 | 27.833 | 43.722 |

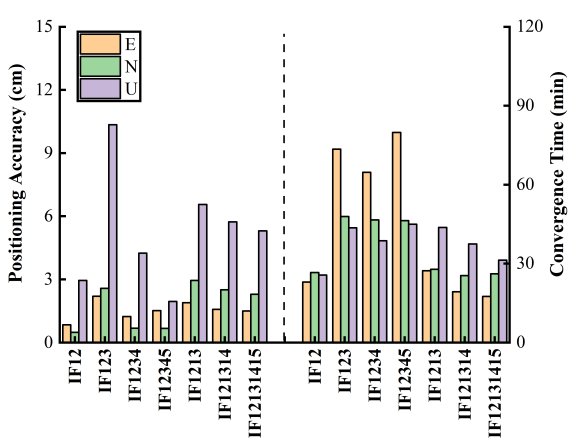
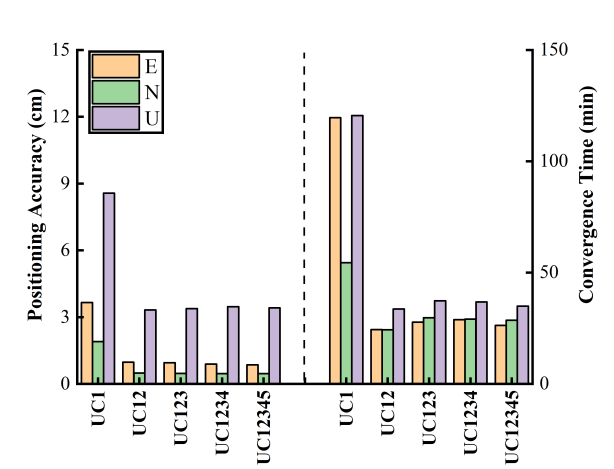
(a) GNSS IF combination PPP (b) GNSS UC mode PPP

Figure 42 The averaged positioning accuracy and convergence time of multi-GNSS PPP with G, G+E and G+C+E from single- to triple-frequency observations

Secondly, compared with the traditional multi-GNSS and multi-frequency PPP solutions, FiPPP software can fully use the received observations. Therefore, the PPP performances of system- and frequency-wide observations are tested as following:

Scheme 3: the traditional dual-frequency GNSS PPP including GPS, Galileo and BDS-3 observations is tested, where the B1I+B3I frequencies are used in BDS-3. Additionally, the received observations are tested from single- to five-frequency PPP based on IF and UC mode. The positioning residuals of ARHT and STHL based on different modes are shown in Figure 43. The PPP results of positioning accuracy and convergence time are shown in Figure 44. It is indicated that the similar results of multi-frequency UC modes can be obtained as the increasing of signals used. However, centimeter-level results of IF combinations are presented, where the whole received observations can not improve the PPP performances. The algorithms of PPP-AR, QC test and the optimal combination will be further researched based on the current version.

|  |  |
| --- | --- |
|  |  |
| (a) IF PPP models positioning residuals at station ARHT and STHL | |
|  |  |
| (b) UC PPP models positioning residuals at station ARHT and STHL | |
| Figure 43 Positioning errors with IF-PPP (top) and UC-PPP (bottom) models for GPS, Galileo and BDS-3 at station ARHT and STHL | |

(a) GNSS IF combination PPP (b) GNSS UC mode PPP

Figure 44 The positioning accuracy and convergence time of system- and frequency-wide GNSS PPP from single- to five-frequency observations

Acknowledgement

The open-source RTKLIB, GAMP, MG-PPP, PPPLib and GPSTK are referred in FiPPP. We pay tribute to the Prof. Shoujian Zhang, author of GNSSBOX software from Wuhan University. The structure of FiPPP is refer to GNSSBOX and GPSTK. Thanks to the authors of above software.

Contact author

Authors: Chao Hu, Ruiguang Wang

E-mail: chaohu@cumt.edu.cn, ragwang@cumt.edu.cn

References

1. Hu, C.; Wang, Q.; Wu, Z.; Guo, Z. A Mixed Multi-Frequency Precise Point Positioning Strategy Based on the Combination of BDS-3 and GNSS Multi-Frequency Observations. *Meas. Sci. Technol.* **2023**, *34*, 025008, doi:10.1088/1361-6501/ac9a62.

2. Wang, Z.; Wang, R.; Wang, Y.; Hu, C.; Liu, B. Modelling and Assessment of a New Triple-Frequency IF1213 PPP with BDS/GPS. *Remote Sens.* **2022**, *14*, 4509, doi:10.3390/rs14184509.

3. Pan, L.; Zhang, X.; Guo, F.; Liu, J. GPS Inter-Frequency Clock Bias Estimation for Both Uncombined and Ionospheric-Free Combined Triple-Frequency Precise Point Positioning. *J. Geod.* **2019**, *93*, 473–487, doi:10.1007/s00190-018-1176-5.

4. Wu, Z.; Wang, Q.; Hu, C.; Yu, Z.; Wu, W. Modeling and Assessment of Five-Frequency BDS Precise Point Positioning. *Satell. Navig.* **2022**, *3*, 8, doi:10.1186/s43020-022-00069-z.