PPPx User Manual

Version 1.2.2

Yuanxin Pan

June 14, 2025

Contents

1 Introduction

1.1 Overview

PPPx is a software package developed for multi-GNSS data processing. Its main component pppx, dedicated to multi-GNSS positioning, is introduced in this manual. Its primary features are listed below:

1. Support four solution modes:
   (a) SPP: Single Point Positioning
   (b) PPP: Precise Point Positioning
   (c) RTK: short baseline processing
   (d) TDP: Time-Differenced Positioning
2. Support multi-GNSS: GPS, GLONASS, Galileo, Beidou-2/3, and QZSS
3. Support multiple solvers: Extended Kalman Filter (EKF), Factor Graph Optimization (FGO), and Least-Square Estimator (LSQ)
4. Support PPP-AR with Observable-specific Signal Bias (OSB) products
5. Flexible frequency selection (L1/L2/E1/E5a/...)
6. High-precision and efficiency: Capable of processing 2880 epochs within 1-2 s
7. Unified input/output for different solution modes
8. Support Linux, Windows, and macOS

1.2 License

This software is distributed under the GPL-3.0 license.

2 Getting started

In this section, the system requirements and installation steps for pppx on different platforms (Linux, macOS, and Windows) are introduced.

2.1 System requirements

Generally, there is no specific hardware requirement for pppx. To process a 24-hour GNSS observation file with a sampling rate of 30 s, pppx consumes around 150 MB of resident memory and 1 CPU core. However, resident memory consumed may change with the number of GNSS systems used and the data sampling rate.

Note that pppx only supports Macs with Apple silicon (i.e., ARM architecture) and an OS version higher than Monterey.

2.2 Installation

2.2.1 Linux

It is recommended to run pppx on Ubuntu 22.04, since the binary was built with gcc 11 on Ubuntu 22.04. However, the binary should work on most modern Linux systems with gcc 11 and libceres-dev (version >= 2.0.0) available. Note that this requirement only applies to the dynamically linked executable pppx. You can always use `pppx_static` on a Linux machine without the support for FGO.

2.2.1.1 With libceres-dev

The installation can be done with the following commands:

```
$ git clone git@github.com:YuanxinPan/PPPx_bin.git
$ sudo apt install libceres-dev

$ mkdir -p ${HOME}/.local/bin
$ cp PPPx_bin/bin/linux/pppx ${HOME}/.local/bin/
$ echo "export PATH=\${HOME}/.local/bin:\$PATH" >> ${HOME}/.bashrc
# Restart your terminal afterward
```

2.2.1.2 With .deb package

Download the .deb package from the latest release on GitHub. Then run the following command to install the pppx software:

```
$ sudo dpkg -i pppx_1.2.2_amb64.deb
```

2.2.2 Windows

The easiest way to run pppx on Windows is using the Windows Subsystem for Linux (WSL). Please follow the instructions in Section 2.2.1.

However, pppx can run natively on Windows since version 1.2.1.

1. Download DLLs via this link
2. Uncompress the zip file and move the DLLs to the folder "bin/windows/"
3. Add the absolute path of the folder "bin/windows" to the PATH environment variable
4. Open cmd.exe and type pppx.exe to test if it is correctly installed

2.2.3 macOS

If you do not have Homebrew on your Mac, please install it first.

Then run the following commands via the Terminal application:

```
# Install ceres-solver
$ brew install ceres-solver

# Install pppx
$ git clone git@github.com:YuanxinPan/PPPx_bin.git
$ sudo mkdir -p /usr/local/bin/
$ sudo cp PPPx_bin/bin/macos/pppx /usr/local/bin/

# Cehck if the installation is successful
$ pppx
```

Due to Apple's security settings, you might need to authorize the pppx software: Open a Mac app from an unknown developer.

3 Usage

In this section, the usage of the pppx software is introduced. The general steps to run pppx are:

1. Prepare input files, including a GNSS observation file in the Receiver Independent Exchange (RINEX) format and necessary products (either broadcast ephemeris or precise products from IGS)
2. Modify the configuration file *pppx.ini* if necessary
3. Execute pppx with command line:
   `pppx -c pppx.ini path-to-rnxobs [rnxobs-of-base]`
4. Check output files

Here are some example commands to run pppx:

```
# SPP/PPP/TDP
# (Note: TDP is suitable for high-frequency data, e.g., 1 Hz)
$ pppx -c pppx.ini rinex/ZIM200CHE_R_20221000000_01D_30S_MO.rnx

# RTK
# (ZIMM is the base station in this example)
$ pppx -c pppx.ini rinex/ZIM200CHE_R_20221000000_01D_30S_MO.rnx
↪  rinex/ZIMM00CHE_R_20221000000_01D_30S_MO.rnx
```

3.1 Input

The complete list of input files is provided below:

- GNSS observations in the RINEX format, specified by command line arguments
- Configuration file `pppx.ini`, specified by command line argument
- Satellite products (either brdc or precise), specified by `pppx.ini`
- Table files (already provided in the table/ directory), specified by `pppx.ini`

The configuration file and satellite products will be explained in detail in subsequent sections.

3.1.1 Configuration

The configuration file `pppx.ini` is the main interface of the software pppx. It is written in the `ini` format and is suitable for human reading. Users can specify general GNSS positioning strategies (Table 3.1) with it.

Some example configuration files are provided in the folder "example/pppx/". Most settings in `pppx.ini` are self-explained. Usually, users just need to modify the following

Table 3.1: Available sections of the configuration file pppx.ini.

Section	Description
session	The desired sampling rate and start and end time for data processing
constellation	The GNSS systems to be used and specific satellites to be excluded
observation	Selection of signal frequency and observable priority
model	The tropospheric and ionospheric models
estimation	Various estimation strategies, including solution mode, positioning mode, solver type and etc.
product	The choice of satellite product source and the path to each product
table	The paths to table files
output	The output path and log level

settings and keep the rest as default:

```
[constellation]: system
[estimation]: sol_mode, pos_mode, solver
[product]: src, nav, sp3, clk
```

When a template configuration file is not available, you can run the following command to get solution mode-specific (including spp, ppp, rtk, and tdp) default configurations. Through redirecting to a file, you can modify the configuration, then it can be used for data processing with pppx.

```
# Output the default configurations for SPP and redirect them to a file
$ pppx -x spp > spp.ini
```

The following paragraphs demonstrate how to set the "key-value" pair for each section of the configuration file.

3.1.1.1 session

Here, "interval" specifies the required interval for data processing, it should be no smaller than the sampling rate of the RINEX file and should be integer number times of it. "date" denotes the year and day of year (DOY) for data processing, and it will be used to replace the placeholders in the "[product]" section. By default, it can be read from the first epoch of the RINEX file. "start" and "end" denote the start and end epoch for data processing within the day (closed interval).

A screenshot of the example settings is shown in Figure 3.1. Note that this figure shows how to correctly configure these fields. Usually, users can leave all the fields in this section blank. Then, pppx will process the whole RINEX file epoch by epoch.

```
[session]
interval = 300                    ; interval (sec)     [ RINEX ]
date     = 2020 100               ; year DOY           [ RINEX ]
start    = 02 15 30               ; hour min sec       [ none ]
end      = 23 45 00               ; hour min sec       [ none ]
```

Figure 3.1: The "session" section of configuration.

3.1.1.2 constellation

This section defines the GNSS systems to be used for data processing and the problematic satellites to be excluded. An example is shown in Figure 3.2. Note that the order of letters (i.e., GRECJ) does not matter, and there should be no space between them. The GPS system is not mandatory; for example, Galileo-only solutions are also feasible. The excluded PRNs should be separated by white space.

```
[constellation]
system  = GE                      ; opt: GRECJ         [ GRECJ ]
exclude = E14 E18 G04             ; excluded PRNs      [ none ]
```

Figure 3.2: The "constellation" section of configuration.

3.1.1.3 observation

This section is well explained in the configuration file. Note that the observation noise represents the noise level of the uncombined observations. For ionosphere-free combinations, observation noise will be automatically computed based on the uncombined ones. A screenshot of the example settings is provided in Figure 3.3.

```
[observation]
noise = 0.3 0.002                 ; observation noise of code/phase (m) [ 0.3 0.002 ]
; G/R/E/C/J = f1 f2 obs_priority: high -> low
G = 1 2 WPCLSXYMN                 ; default: [ 1 2 WPCLSXYMN ]
R = 1 2 PCIQX                     ; default: [ 1 2 PCIQX ]
E = 1 5 XCIQB                     ; default: [ 1 5 XCIQB ]
C = 2 6 IQX                       ; default: [ 2 6 IQX ]
J = 1 2 SLXCZ                     ; default: [ 1 2 SLXCZ ]
```

Figure 3.3: The "observation" section of configuration.

3.1.1.4 model

This section specifies the tropospheric and ionospheric models. An example is provided in Figure 3.4. Users should choose an option (case sensitive) from the values listed in the comment. Note that this section is not effective for RTK, since tropospheric and ionospheric delays are ignored for short baselines; "iono" can only be IF for PPP.

Currently, the tide model cannot be configured through `pppx.ini`. Solid earth tide, ocean tidal loading, and pole tide are applied for PPP whenever possible, but they are not applied for spp/rtk/tdp.

```
[model]
trop = GMF                                  ; opt: GMF/VMF1/GPT2w/none      [ GMF ]
iono = IF                                   ; opt: IF/brdc/IONEX/none       [ IF ]
```

Figure 3.4: The "model" section of configuration.

3.1.1.5 estimation

This is the most important section of the configuration file. Here, you can choose the solution mode (spp/ppp/rtk/tdp), positioning mode (kinematic/static/fixed) and solver (ekf/fgo/lsq) used for adjustment. An example is provided in Figure 3.5. Most settings are self-explained.

The fields "sol_mode", "pos_mode", and "solver" are well explained in the configuration file. However, spp and tdp can only support the kinematic positioning mode, and not all the solvers are supported for each solution mode (i.e., lsq and fgo for spp and tdp, ekf and fgo for ppp and rtk). The software will output an error message if a specific configuration is not supported.

"fgo_var" controls the output of variance information when fgo is specified as "solver". By default, variances are not compuated for fgo since it might be time-consuming.

"site_pos" specifies the approximate station coordinates in XYZ format (unit: meters). If not specified, the approximate coordinates will be automatically computed using SPP. This rule also applies for the field "base_pos" for rtk.

"ppp_ar" is a switch for PPP-AR. Ambiguity resolution can be enabled by setting it to "yes", and a valid phase-bias product must be provided using the field "[product] bia". The LAMBDA and rounding methods are used to resolve integer ambiguities if "solver" is set to "ekf" and "fgo", respectively. Note that GLONASS is disabled for AR and the GPS L5 signal is currently not supported for PPP-AR.

"sf_ppp" is designed to run single-frequency PPP with GIM products, that is, ionospheric delays are not estimated but fixed to GIM corrections. In this way, the accuracy of GIM products can be evaluated. However, dual-frequency observations are still required for cycle slip detection. "ppp_ar" should be set to "no" when sf_ppp is enabled.

"slip_det" specifies the methods used for cycle slip detection for ppp. The option "TDCP" represents the time-differenced carrier phase. You can set "slip_det" to "ALL" for most stations, but "MW" should not be used for stations with noisy pseudorange, such as smartphones. "LLI" could be problematic at some point, since some stations set a lot of wrong LLI flags, but it is recommended for processing smartphone GNSS data. Note that rtk and tdp always use LLI and GF for cycle slip detection, and currently it is not configurable.

"pos_pri", "clk_pri", "isb_pri", and "ztd_pri" specify a-prior uncertainties and process noises of station position, receiver clock, receiver inter-system bias (ISB), and tropospheric delay for EKF, respectively. Note that only one receiver clock is estimated for one GNSS system, and ISBs are estimated for other systems using EKF. When FGO is chosen as the solver, only "ztd_pri" is effective, and other "pri" settings are not used. Individual receiver clock errors (instead of ISB) are estimated as white noise for different constellations, and the station position is estimated as white noise for the kinematic mode.

"rtk_ar" is a switch for RTK-AR, with "yes" as its default value. The LAMBDA and rounding methods are used to resolve integer double-differenced (DD) ambiguities if "solver" is set to "ekf" and "fgo", respectively. "glonass_ar" can be set to "yes" for rtk if rover and base stations share the same receiver type. Otherwise, keep it as "no".

```
[estimation]
sol_mode    = ppp                    ; opt: spp/ppp/rtk/tdp          [ spp ]
pos_mode    = kinematic              ; opt: kinematic/static/fixed   [ kinematic ]
solver      = ekf                    ; opt: ekf/fgo/lsq              [ ekf ]
weight_opt  = elev                   ; opt: elev/snr                 [ elev ]
elev_mask   = 7                      ; elevation mask (°)            [ 10 ]
snr_mask    = 25                     ; SNR mask (dB-Hz)              [ 25 ]
site_pos    =                        ; initial position in xyz (m)   [ spp ]
; ppp only
ppp_ar      = no                     ; PPP ambiguity resolution      [ no ]
sf_ppp      = no                     ; single-frequency PPP          [ no ]
slip_det    = ALL                    ; opt: off GF MW LLI TDCP ALL   [ ALL ]
pos_pri     = 100 1                  ; uncertainty  process_noise    [ 1E+02 1 ]
clk_pri     = 100 100                ;      m         m/sqrt(s)      [ 1E+02 1E+02 ]
isb_pri     = 50  3.2E-04            ;                               [ 5E+01 3.2E-04 ]
ztd_pri     = 0.5 3E-05              ;                               [ 0.5 3E-05 ]
; rtk only
rtk_ar      = yes                    ; RTK ambiguity resolution      [ yes ]
glonass_ar  = no                     ; AR for GLONASS                [ no ]
base_pos    =                        ; base position in xyz (m)      [ spp ]
```

Figure 3.5: The "estimation" section of configuration.

3.1.1.6 product

This section specifies the paths to various products. If "src = brdc", then "nav" should be set to a valid path to a broadcast ephemeris. If "src = precise", then "sp3" should be set to a valid path to a SP3 product, and all other products are not necessary. If you use relative paths for products, the paths are relative to the directory where pppx is executed, not where pppx.ini is located. In addition, placeholders (see Figure 3.6) can be used to specify product names, making it easier to process GNSS data on multiple days. More details on the products are introduced in Section 3.1.2.

3.1.1.7 table

This section specifies the paths to the table files. An example is provided in Figure 3.7. More details can be found in Section 3.1.3. If you use relative paths for the tables, the paths in the configuration are relative to the directory where pppx is executed.

```
; available placeholder:
;   -YEAR-   : year
;   -YR-     : 2-digit year
;   -DOY-    : day of year
;   -WEEK-   : GPS week
;   -DOW-    : day of GPS week
[product]
src = precise                                       ; opt: brdc/precise
nav = rinex/BRDC00IGS_R_-YEAR--DOY-0000_01D_MN.rnx  ; broadcast ephemeris       (brdc)
sp3 = products/cod-WEEK--DOW-.sp3                   ; precise oribt             (precise)
clk = products/cod-WEEK--DOW-.clk                   ; precise clock             (precise)
erp = products/cod-WEEK--DOW-.erp                   ; earth rotation parameters (precise)
bia = products/cod-WEEK--DOW-.bia                   ; satellite bias            (precise)
obx = products/cod-WEEK--DOW-.obx                   ; satellite attitude        (precise)
ion = products/cod-WEEK--DOW-.ion                   ; global ionospheric map    (common)
vmf = products/VMFG_-YEAR--DOY-                     ; VMF1 grid                 (common)
```

Figure 3.6: The "product" section of configuration.

```
[table]
igsatx     = ../../table/igs14.atx
oceanload  = ../../table/oceanload
channel    = ../../table/glonass_chn
gpt2w      = ../../table/gpt2_1wA.grd
orography  = ../../table/orography_ell
```

Figure 3.7: The "table" section of configuration.

3.1.1.8 output

This section specifies the output path and log level. An example is provided in Figure 3.8. More details can be found in Section 3.2.

```
[output]
path  = ./                      ; the folder where results will be stored
level = info                    ; opt: off/critical/error/warn/info/debug/trace
```

Figure 3.8: The "output" section of configuration.

3.1.2 Product

There are two options of satellite products for each solution mode: "brdc" and "precise", controlled by the configuration field: "[product] src". The "brdc" option uses satellite orbits and clocks computed from broadcast ephemeris, whereas the "precise" option uses orbits and clocks from IGS precise products. Note that "[table] igsatx" is required if "[product] src" is set to "precise".

Here are some clarifications:

- All products labeled "precise" will not be used if "[product] src" is set to "brdc"
- Precise orbits and clocks can be used for SPP, but the Diffential Code Bias (DCB) issue is not handled for single-frequency SPP. In this case, it is recommended to set "[model] iono = IF"
- Only the SP3 file is necessary for the "precise" option. Other precise products are

optional. If the CLK product is missing, the clock corrections from the SP3 file will be used

- "ion" and "vmf" are only mandatory when the corresponding models are specified
- PPP specific:
  - Broadcast ephemeris can also be used for PPP. Precise products will not be used in this case
  - The nominal attitude will be used if the OBX product is missing
  - The BIA product will be used if "[estimation] ppp_ar" is set to "yes"
  - Pole tide corrections will not be applied if the ERP product is missing, but this is fine for most applications.

3.1.3 Table

Several table files are required when specific models are used for data processing. These mainly include the antenna model, the tropospheric model, and the tide model. A complete list of the required table files is provided in Table 3.2. The instructions for downloading each table file are provided in Appendix A.3.

Table 3.2: List of table files.

Table	Description
channel	Frequency channel numbers for GLONASS satellites
gpt2w	Coefficients of the GPT2w tropospheric model
igsatx	Satellite and receiver antenna models in the IGS ANTEX format
oceanload	Ocean tide loading coefficients for stations
orography	Ellipsoidal station heights of the VMF1 grid points

Here are the clarifications:

- "channel" is only needed when GLONASS carrier phase measurements are used for data processing (i.e., ppp/rtk/tdp)
- "gpt2w" should be set when "[model] trop" is set to "GPT2w"
- "igsatx" is mandatory if "[product] src" is set to "precise". When "igstax" is specified, the satellite and station PCO corrections will be applied for all "sol_mode", but the satellite and station PCV corrections will only be applied for PPP and RTK.
- "oceanload" is optional. Ocean tide loading will not be corrected if coefficients for the processed station are missing
- "orography" is needed when "[model] trop" is set to "VMF1"

Although table files are not always required, it is recommended to properly set all table paths to simplify the process and avoid potential issues.

3.2 Output

This section introduces the output files generated by the pppx software. An overview of these files is provided in Table 3.3. The function and format of each file are described in detail in the following sections.

Table 3.3: List of output files.

Output	Description
pos file	Receiver position, clock (GPS) and ZTD estimates for each epoch
log file	Information for debugging, including cycle slip detection, a-prior and post-fit residuals
stat file	Post-fit residuals and various estimates in the RTKLIB stat format, intended for visualization with RTKLIB

3.2.1 pos file

The primary output of pppx is the pos file, which contains the main estimates for each processed epoch. These include station coordinates in the Earth-centered and Earth-fixed (ECEF) coordinate system, standard deviations of the coordinate estimates, receiver clock errors for the GPS system, and Zenith Total Delays (ZTD).

Figure 3.9 shows the beginning part of a sample pos file. The units used for coordinates, clock errors, and ZTD estimates are meters. Note that the coordinate system will align with the satellite orbit products used for data processing. By default, the standard deviations of coordinates are not included in the output when "fgo" is selected as the solver, because of the long computation time required.

mjd	sod	nsat	x	y	z	stdx	stdy	stdz	rck(m)	zhd	zwd	dzwd
59679	0.00	17	4331298.023	567538.192	4633133.548	0.795	0.278	0.797	-9.8	2.081	0.025	0.2127
59679	30.00	17	4331299.398	567538.409	4633134.059	0.494	0.196	0.471	-8.9	2.081	0.025	0.1388
59679	60.00	17	4331299.590	567538.000	4633134.399	0.370	0.159	0.339	-8.4	2.081	0.025	0.0210
59679	90.00	17	4331299.938	567538.075	4633134.310	0.298	0.137	0.265	-8.2	2.081	0.025	0.0189
59679	120.00	17	4331299.981	567538.018	4633134.309	0.253	0.122	0.221	-8.1	2.081	0.025	0.0125
59679	150.00	17	4331299.899	567537.991	4633134.164	0.219	0.111	0.187	-8.0	2.081	0.025	0.0160
59679	180.00	17	4331299.918	567538.003	4633134.082	0.191	0.103	0.160	-8.0	2.081	0.025	0.0129
59679	210.00	17	4331299.932	567538.003	4633134.040	0.169	0.095	0.139	-8.1	2.081	0.025	0.0147
59679	240.00	17	4331299.774	567537.826	4633134.008	0.151	0.090	0.121	-8.1	2.081	0.025	0.0187
59679	270.00	17	4331299.776	567537.830	4633133.938	0.136	0.085	0.107	-8.2	2.081	0.025	0.0248

Figure 3.9: The screenshot of a sample pos file.

3.2.2 log file

A log file will be generated if the "[output] level" in the configuration file is not set to "off". This file contains debugging information such as cycle slip detection, outlier detection, and a-prior and post-fit residuals. Each log message generally begins with a letter that indicates the message type: 'T' for Trace, 'D' for Debug, 'I' for Information, 'W' for Warning, 'E' for Error, and 'C' for critical.

Figure 3.10 shows the screenshot of a sample log file. For example, a cycle slip was detected at epoch 86370.0 (seconds of the day). Following the PRN number, the azimuth, elevation, and between-epoch difference of the geometry-free (GF) combination of the satellite are provided. The subsequent line indicates that a new ambiguity parameter (numbered 410) was introduced at that epoch. The following lines provide information for all satellites available at that epoch. These include the a-prior residuals of pseudorange and carrier phase, the corresponding post-fit residuals, ambiguity estimates in cycles, as well as the azimuth and elevation for each satellite, listed after its PRN and epoch time.

```
I ## 86370.00
I  GfJump: 86370.0 G30 -170.3 10.2  val: 16.07
I +++ G30 410 86370.00
I E04 86370.00   -0.432   -0.005    -0.423     0.004    187.118 -100.0   63.8
I E09 86370.00    3.216    0.011     3.222     0.016      0.179 -109.7   11.2
I E11 86370.00    0.231   -0.006     0.240     0.003     -4.188  -83.7   54.4
I E12 86370.00   -0.064   -0.009    -0.059    -0.004      1.739 -168.9   39.2
I E18 86370.00   -0.479   -0.016    -0.471    -0.008      3.720  -63.6   10.7
I E19 86370.00   -0.077   -0.011    -0.069    -0.003      7.753   59.7   52.5
I E21 86370.00    1.853    0.001     1.856     0.004     -2.511   93.7   14.4
I E27 86370.00    0.895   -0.015     0.901    -0.008     14.281   43.2   12.0
I E36 86370.00    0.604   -0.014     0.613    -0.005      5.998  -37.7   18.2
I G02 86370.00    1.937   -0.007     1.946     0.002     -2.125  -48.7   29.3
I G03 86370.00   -1.312    0.014    -1.309     0.017     47.787  106.7   26.2
I G04 86370.00   -0.193   -0.016    -0.185    -0.008    369.198   57.6   51.2
I G06 86370.00   -0.409   -0.012    -0.401    -0.004     54.960 -104.9   62.2
I G07 86370.00    0.904   -0.011     0.907    -0.008    -28.983  173.3   31.9
I G09 86370.00    0.800   -0.002     0.808     0.007    246.278  -20.0   86.5
I G11 86370.00   -1.175   -0.009    -1.166     0.000    -10.412  -49.6   30.7
I G19 86370.00    3.370   -0.005     3.373    -0.001     39.702 -129.2    9.7
I G26 86370.00   -1.855   -0.011    -1.850    -0.005     46.984   44.9    7.4
I G30 86370.00   -0.267   24.679    -0.266     0.000    354.394 -170.3   10.2
```

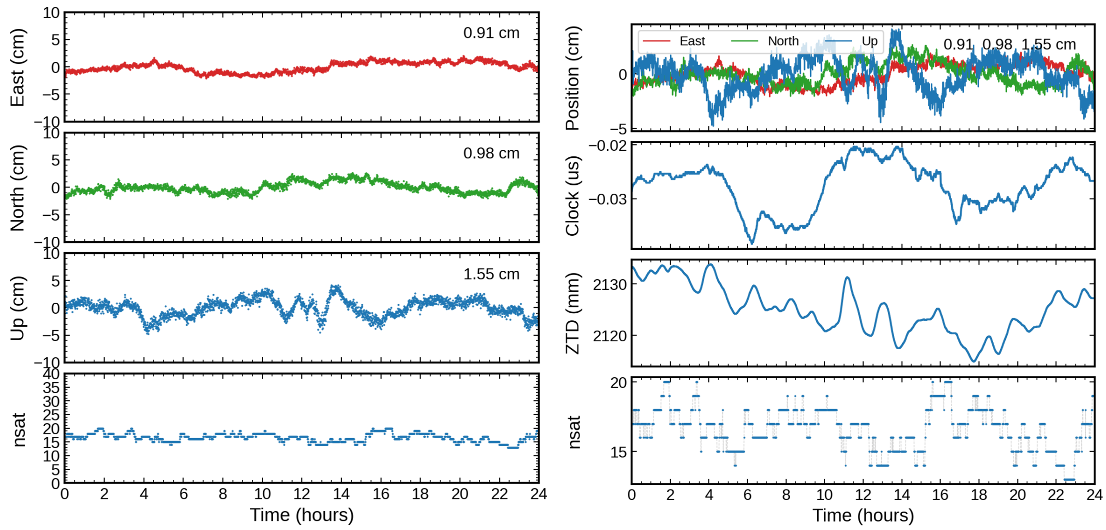
Figure 3.10: The screenshot of a sample log file.

3.2.3 stat file

A stat file is always generated for visualization with the RTKLIB software. For detailed information on the format, you can refer to the RTKLIB source code. Currently, pppx only outputs messages with the keywords "$POS", "$CLK", and "$SAT". The only difference is that the "$CLK" values are expressed in meters (instead of nanoseconds) and represent absolute clock errors for each GNSS system (rather than inter-system biases). Figure 3.11 shows a screenshot of a sample stat file.

```
$POS,2205,0.000,6,4331298.0228,567538.1921,4633133.5484,0.7955,0.2775,0.7969
$CLK,2205,0.000,6,-9.811,0.000,1.618,0.000,0.000
$SAT,2205,0.000,E03,1,-108.9,30.2,0.8635,0.0000,0,45.300,0,0,0,0,0,0
$SAT,2205,0.000,E05,1,-62.4,83.0,-0.2401,0.0000,0,52.100,0,0,0,0,0,0
$SAT,2205,0.000,E09,1,62.2,39.7,0.2750,0.0000,0,47.500,0,0,0,0,0,0
$SAT,2205,0.000,E15,1,-49.4,34.3,-0.5920,0.0000,0,47.300,0,0,0,0,0,0
$SAT,2205,0.000,E24,1,119.5,7.2,0.8040,0.0000,0,38.900,0,0,0,0,0,0
$SAT,2205,0.000,E31,1,74.2,15.2,-0.4149,0.0000,0,41.800,0,0,0,0,0,0
$SAT,2205,0.000,E36,1,177.0,24.0,-0.3656,0.0000,0,43.300,0,0,0,0,0,0
$SAT,2205,0.000,G02,1,-47.8,28.1,1.0852,0.0000,0,44.700,0,0,0,0,0,0
```

Figure 3.11: The screenshot of a sample stat file.



(a) Plotted with option "-s"

(b) Plotted with option "-a"

Figure 3.12: Visualization of an FGO-based PPP solution with the Python script.

3.3 Visualization

This section explains how to visualize the solutions generated by pppx.

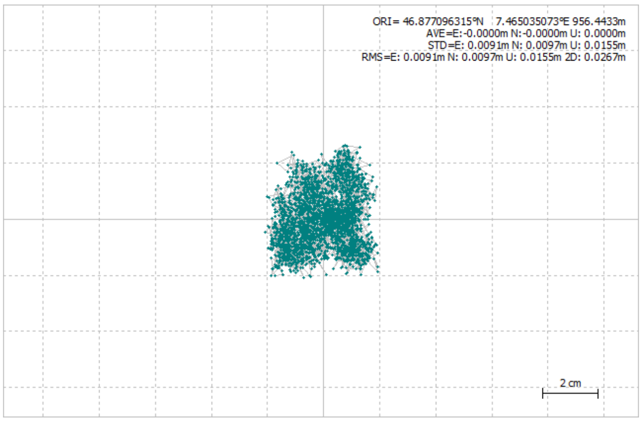
3.3.1 With Python

A Python script named "plot_ppppos.py" is included in the software package and is located in the "scripts/" folder. This script can generate plots of either (1) the positioning time series in the east, north, and up components (relative to the mean position) or (2) additional time series of the receiver clock and ZTD estimates. The general usage of the script is demonstrated with the commands below, and the corresponding figures are shown in Figure 3.12:

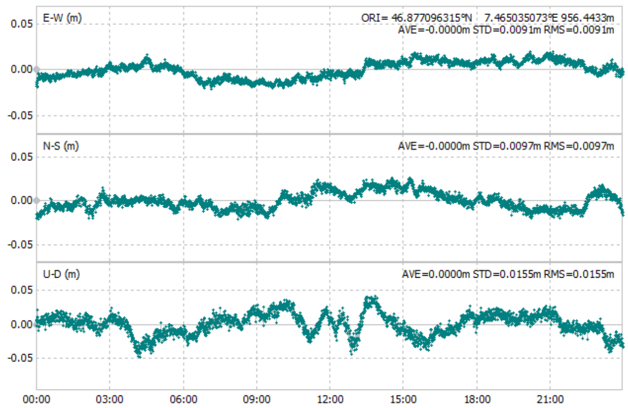
```
# (1) plot position estimates only
$ ../../scripts/plot_ppppos.py pos_file -s

# (2) plot position, receiver closk, and ZTD estimates
$ ../../scripts/plot_ppppos.py pos_file -a

# Options:
# -a  Plot position, receiver clock, and ZTD estimates
# -i  Interactive mode
# -s  Fixed scale for the y-axis; otherwise, the scale is set automatically
```

3.3.2 With RTKLIB

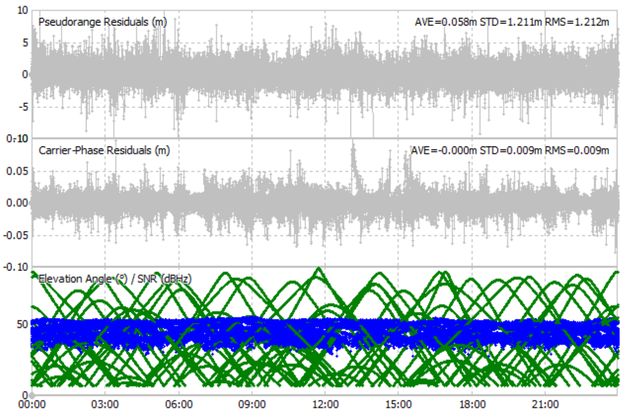
The RTKLIB software is recommended for an enhanced interactive view of various plots and visualization of post-fit residuals. However, a limitation is that the author of



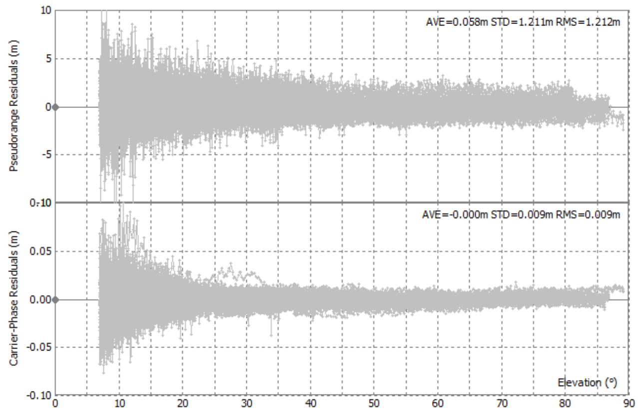
(a) Ground track plot.



(b) Positioning time-series plot.



(c) Residual time-series plot.



(d) Residual-elevation plot.

Figure 3.13: Visulization of an FGO-based PPP solution with RTKLIB.

RTKLIB provides executables only for Windows users. Linux users need to compile the GUI applications themselves using the Qt library.

Using RTKLIB for visualization is straightforward:

1. Open the application "rtkplot.exe"
2. Click the menu "File" > "Open Solution-1" > Select the generated stat file
3. Interactively view the various plots

Figure 3.13 shows some example visualizations. A useful tip: you can open two solutions within the same window for easy comparison.

4 Examples

This section provides some examples of the use of pppx for various types of data processing. The corresponding folder is ‘example/pppx/‘. Specifically, two RINEX files and some products are provided in "example/pppx/rinex/" and "example/pppx/products/", respectively. Basic information on the observation and product files can be found in Tables 4.1 and Table 4.2, respectively. Note that the VMF1 grid product is a spliced file of the epoch 00 h, 06 h, 12 h, and 18 h of the current day and the 00 h of the next day. By default, GPS and Galileo observations are used for each example, except that the RTK example is GPS only.

Table 4.1: GNSS data used in examples.

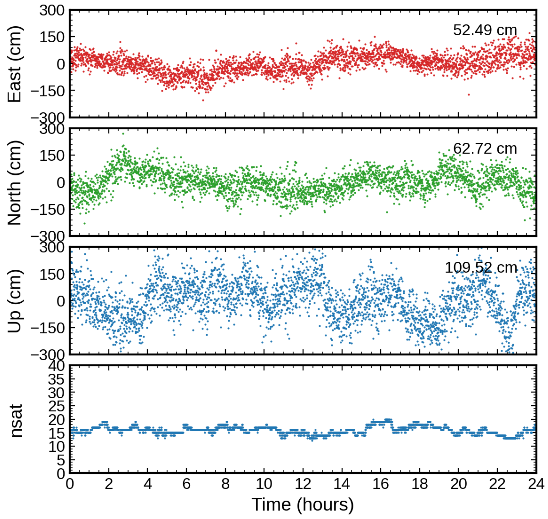
Data	Description
ZIM200CHE_R_20221000000_01D_30S_MO.rnx	Rover station (GREC)
ZIMM00CHE_R_20221000000_01D_30S_MO.rnx	Reference station (G)
ALGO00CAN_R_20221000000_15M_01S_MO.rnx	High-rate data

Table 4.2: Products used in examples.

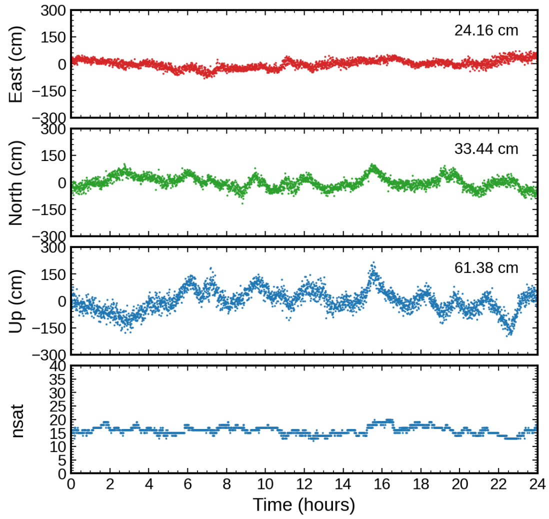
Product	Description
BRDC00IGS_R_20220010000_01D_MN.rnx	Broadcast ephemeris for multi-GNSS
COD0MGXFIN_20221000000_01D_05M_ORB.SP3	Precise orbit from CODE
COD0MGXFIN_20221000000_01D_30S_CLK.CLK	Precise clock from CODE
COD0MGXFIN_20221000000_01D_15M_ATT.OBX	Satellite attitude from CODE
COD0MGXFIN_20221000000_01D_01D_OSB.BIA	Satellite bias from CODE
COD0MGXFIN_20221000000_03D_12H_ERP.ERP	ERP from CODE
codg1000.22i	GIM from CODE
VMFG_2022100	spliced VMF1 grid from TU Vienna

4.1 SPP: Single Point Positioning

The pppx software supports multi-GNSS SPP, including GPS, GLONASS, Galileo, BeiDou, and QZSS, with broadcast ephemeris or precise satellite products. Various tropospheric and ionospheric models are supported. However, only kinematic mode is currently supported with either LSQ or FGO as a solver. Note that LSQ- and FGO-based SPP are both epoch-wise solutions (i.e., no constraints between epochs), and the only difference is that FGO utilize the Huber loss function (2 m threshold) to improve robustness. Here, the LSQ solver is used, as the GNSS data used in the examples are from geodetic-grade receivers.



(a) IF solution



(b) SF solution

Figure 4.1: Positioning results of SPP solutions.

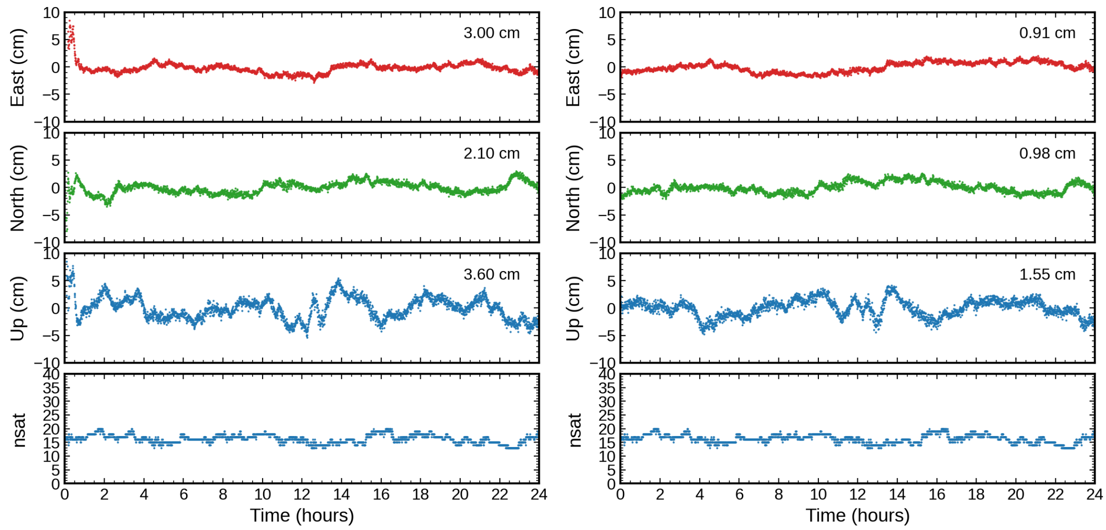
Here, SPP solutions with IF combination and single-frequency observations are demonstrated. Both examples use broadcast ephemeris, which is the most common case for SPP users. Specifically, the GMF model and the IF combination are used to correct tropospheric and ionospheric delays for the IF solution, and GMF and a GIM product and are used for the single-frequency solution. The other settings are kept as defaults. The corresponding configuration files can be found as "example/pppx/00_spp_if.ini" and "example/pppx/01_spp_sf.ini", respectively.

The positioning results are shown in Figure 4.1. It is found that the single-frequency solution is better then IF solution. This is attributed to the fact that the IF combination amplifies the noise of pseudorange about three times, and the ionospheric delays can be well corrected with the GIM product for the station ZIM2, which is located at middle altitude.

4.2 PPP: Precise Point Positioning

Here, PPP solutions are demonstrated with the Kalman filter or FGO as the solver. The corresponding configuration files can be found as "example/pppx/02_ppp_ekf.ini" and "example/pppx/03_ppp_fgo.ini", respectively. The Kalman-filter-based solution simulates the real-time scenario and outputs a forward only solution. The FGO-based solution is dedicated to post-processing and output a batch processing solution, i.e., without a convergence period. In addition, FGO utilize the Huber loss function to improve robustness, i.e., thresholds of 2 m and 0.02 m for pseudorange and carrier phase observations, respectively.

The positioning results are shown in Figure 4.2. It is observed that there is no convergence period for the FGO-based solution, but the EKF solution takes approximately 15 min to



(a) EKF solution

(b) FGO solution

Figure 4.2: Positioning results of PPP solutions.

converge. This is attributed to the fact that all the GNSS observations are used to form a large LSQ problem and all the parameters are solved together with FGO, while the parameters are solved epoch-by-epoch with EKF.

4.3 RTK: Real-Time Kinematic

The pppx software only supports short baseline solutions. Thus, "[model] trop" and "[model] iono" are always set to "none". Both EKF and FGO are supported as solvers for RTK. Similarly, the EKF solution simulates real-time scenarios and outputs a forward only solution; while the FGO solution is dedicated to post-processing and outputs a batch processing solution without a convergence period. The same Huber loss function as PPP is adopted for FGO-based RTK processing. Here, we show the RTK solution computed with EKF as an example. The corresponding configuration file can be found as "example/pppx/06_rtk.ini".

The positioning result is shown in Figure 4.4. The precision is 0.24 cm, 0.28 cm, and 0.48 cm for the east, north, and up components, respectively. It is typical precision for short baseline solutions with ambiguity resolved.

4.4 TDP: Time-Differenced Positioning

TDP is useful for earthquake studies since it does not have a convergence time and achieves high precision during a short time span. Ambiguities are eliminated by time-differencing, and station displacements between adjacent epochs are estimated. The displacements are then cumulatively added to the initial station position, and the absolute positions are outputted in the pos file for each epoch.

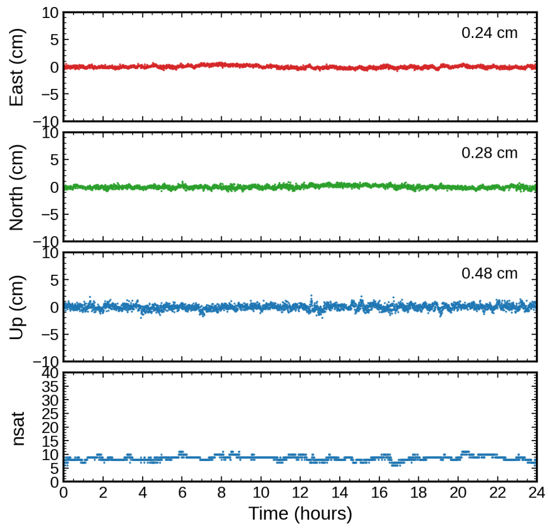


Figure 4.3: Positioning results of a short baseline solution between ZIM2 and ZIMM.

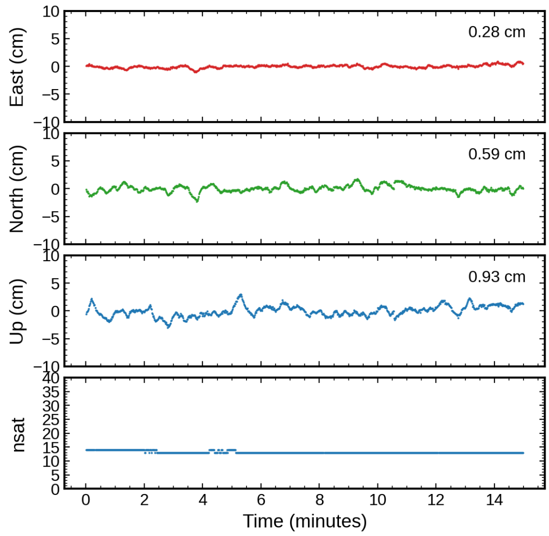


Figure 4.4: TDP positioning results for the ALGO station.

An example is provided as "07_tdp", where the 1-Hz GNSS data collected at the ALGO station are processed with TDP. Specially, precise satellite products are used instead of broadcast ephemeris. In this way, the drift of positioning time series caused by error accumulation becomes much smaller. As ALGO is a static station, the ground truth displacement is zero. The TDP positioning precisions over 15 min are 0.28 cm, 0.59 cm, and 0.93 cm in the east, north, and up components, respectively.

5 Troubleshooting

Text

6 FAQs

Text

A File Download

A.1 Download GNSS observations

1. CDDIS: https://cddis.nasa.gov/archive/gnss/data
2. EPN: ftp://ftp.epncb.oma.be/pub/obs
3. PBO: https://data.unavco.org/archive/gnss/
4. GA: https://data.gnss.ga.gov.au/docs/home/gnss-data.html
5. Curtin: http://saegnss2.curtin.edu/ldc
6. Hongkong: https://rinex.geodetic.gov.hk

A.2 Download IGS Products

Available HTTP sites for downloading GNSS products

1. CDDIS: https://cddis.nasa.gov/archive/gnss/
2. CODE: http://ftp.aiub.unibe.ch/CODE_MGEX/CODE/
3. ESA: http://navigation-office.esa.int/products/gnss-products/
4. CNES: http://www.ppp-wizard.net/daily.html

FTP sites:

1. CDDIS: ftps://gdc.cddis.eosdis.nasa.gov/gnss/
2. CODE: ftp://ftp.aiub.unibe.ch/CODE_MGEX/CODE/
3. WHU: ftp://igs.gnsswhu.cn/pub/
4. IGN: ftp://igs.ign.fr/pub/igs

Other products:

1. VMF1: http://vmf.geo.tuwien.ac.at/trop_products/GRID/2.5x2/VMF1/VMF1_OP/

A.3 Download Table Files

A.3.0.1 Ocean tide loading

Old portal: http://holt.oso.chalmers.se/loading/index-aside-2404271219.html

New portal: `https://barre.oso.chalmers.se/loading/l.php`

Here, we use the old portal to download oceanload coefficients.

1. Ocean tide model -> select **"FES2004"**
2. Keep other options as default values
3. Enter the station list: either XYZ or BLH
4. Enter and email address and click **"Submit"**

The required oceanload coefficients will be sent to you by email, and then you need to append the coefficients to the table file "table/oceanload".

A.3.0.2 ANTEX

`https://files.igs.org/pub/station/general`