GREAT-IFCB Users Guide

Xingxing Li, Xin Li, Hongjie Zheng

Email: *xxli@sgg.whu.edu.cn,*
lixinsgg@whu.edu.cn,
hongjzheng@whu.edu.cn,

Contents

1 Overview

The GREAT (GNSS+ REsearch, Application and Teaching) software was designed and developed at the School of Geodesy and Geomatics of Wuhan University for precise orbit and clock determination, high-precision positioning and navigation applications. As one important module in GREAT software, GREAT-IFCB was developed for multi-GNSS inter-frequency clock bias (IFCB) estimation. The open-source GREAT-IFCB software is written in C++ 11 language following the Object-Oriented principle and can compile, run on several popular operating systems, such as Windows, Linux, and Macintosh. It consists of three portable program libraries named libIFCB, libMat and libGnut. The libIFCB library is mainly in charge of estimating IFCB, including the encoding, decoding, and storage of auxiliary data involved in the IFCB estimation, as well as the implementation of IFCB estimation algorithm. As for LibMat and libGnut, they are auxiliary libraries for the software. LibMat is an open-source C++ matrix library named Newmat(http://www.robertnz.net/nm_intro.htm), which offers standard matrix operations. LibGnut library comes from the open-source GNSS software G-Nut(Vaclavovic et al. 2013), including the decoding and storage of GNSS data as well as basic parameter configuration module. The features of GREAT-IFCB are:

(1) It supports multi-GNSS IFCB estimation for: GPS (G), Galileo (E), BeiDou (C) satellites.

(2) It supports IFCB estimation with multi-frequency specified observations.

(3) All of the executable binary APs (application programs) for Windows/Linux/Macintosh are included in the package as well as the whole source programs of the libraries and AP.

(4) Adopting the open-source, cross-platform compilation tool CMake, which is convenient for users to customize their own executable APs.

2 Environmental Requirements and License

2.1 Environmental Requirements

The executable CUI AP for Windows in the package was built by VS (Microsoft Visual Studio) 2017 on Windows 10 (64bit). All of the necessary dynamic link libraries are involved in the folder. Moreover, the CUI AP and shared libraries for Linux were built and tested on CentOS Linux release 7.7.1908 and x64 CPU. As for Macintosh, the CUI AP and dynamic libraries were built by AppleClang 11.0.3.11030032 on MacOS 10.15.3, in which kernel version is Darwin 19.3.0.

Also, the users can use the open-source, cross-platform compilation tool CMake to build executable binary AP on their own operating systems (Windows, Linux or Macintosh).

2.2 License

GREAT-IFCB is an open-source software, which is governed by the GNU General Public License (version 3) (https://www.gnu.org/licenses/gpl-3.0.html).

3 Compile and Installation

The software package can be accessed via the website (https://geodesy.noaa.gov/gps-toolbox). Extract the program package `GREAT-IFCB_<ver>.zip` to appropriate directory `<install_dir>` (`<ver>` indicates the version number). The GREAT-IFCB directory structure is as follows.

`GREAT-IFCB_<ver>`	
`./bin`	The executable binary APs for Windows/Linux/Macintosh *
`./src`	Source programs of GREAT-IFCB software *
`./app`	Main function of IFCB estimation *
`./LibIFCB`	Source programs of IFCB estimation library *
`./LibMat`	Source programs of the newmat library *
`./LibGnut`	Source programs of the G-Nut library *
`./sample data`	Sample data for AP *
`./IFCB_2021001`	Sample data for GREAT-IFCB AP *
`./util`	Utilities *
`./PythonScriptss`	Python scripts for IFCB estimation *
`./doc`	Document files *
`IFCB_CONFIG.xml`	Sample XML files for GREAT-IFCB *
`GREAT-IFCB_1.0.pdf`	User manual

3.1 Windows

To generate multi-frequency GNSS IFCB products, you can either use the existing program under the folder `<install_dir>/GREAT-IFCB_<ver>/bin/Windows`, or compile an executable program by yourself. The following instructions show how to build GREAT-IFCB executable program on Windows.

(1) Get CMake via the website (https://cmake.org/download/) and install it. Note that the minimum requirement of CMake version is 3.0.0.

(2) Execute the cmake-gui and then execute "Browse Source..." to select the folder `<install_dir>/GREAT-IFCB_<ver>`, or you can drag CMakeList.txt in the directory `<install_dir>/GREAT-IFCB_<ver>` to the interface of cmake-gui. Then modify attribute "Where to build the binaries" as `<install_dir>/GREAT-IFCB_<ver>/build`.

(3) Execute "Configure" and choose the Integrated Development Environment (IDE) for the project (appears only the first time you click "Configure" button). It is noticed that you can specify a specific IDE or other build tools especially when you don't want to use the default options. Similarly, the compiler path can also be specified.

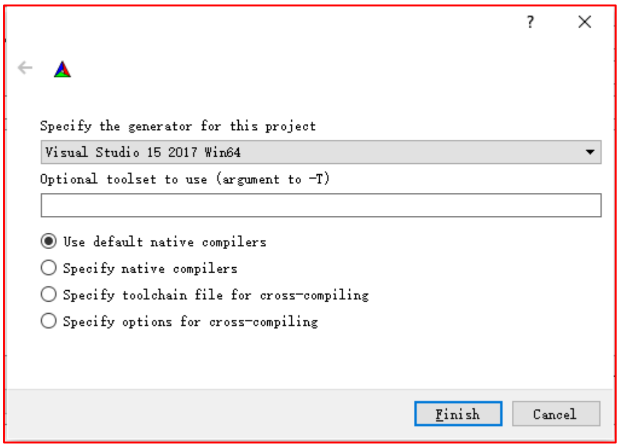

Figure 3.1-1 Example of IDE selection

(4) Execute "Generate" to write the build files to **<install_dir>/GREAT-IFCB_<ver>/build**.

(5) Execute "Open Project" and then compile source code in corresponding IDE.

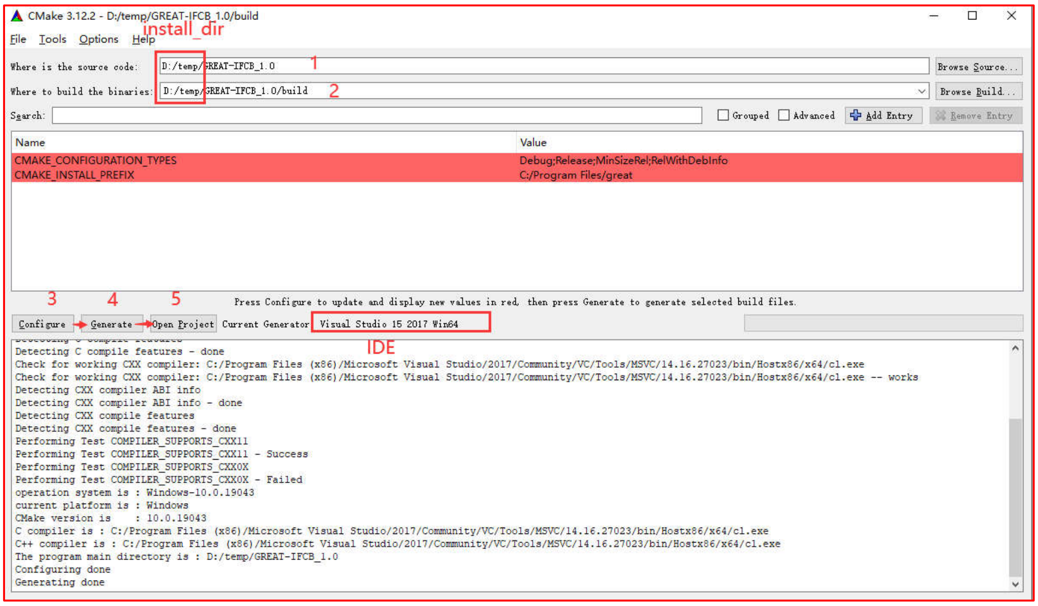

Figure 3.1-2 The compilation of GREAT-IFCB on Windows

(6) If you are used to use the command line prompt on Windows instead of the GUI operaiton, you should refer to the following instructions for Linux/Macintosh.

3.2 Linux/Macintosh

For Linux/Macintosh environment, you can use the existing program under the folder **<install_dir>/GREAT-IFCB_<ver>/bin/<platform> (Linux/ Macintosh)** to estimate IFCB. For Linux/Macintosh environment, before that, please enter
"**export LD_LIBRARY_PATH=<install_dir>/GREAT-IFCB_<ver>/bin/Linux**"
to load the relevant shared libraries in the current terminal. You can also rebuild GREAT-IFCB executable

program on Linux/Macintosh according to the following instructions.

(1) Get CMake via the website (https://cmake.org/download/) and install it. Note that the minimum requirement of CMake version is 3.0.0.

(2) Create "build" directory in the directory **<install_dir>/GREAT-IFCB_<ver>** and change the directory to "build".

>> **mkdir <install_dir>/GREAT-IFCB_<ver>/build**

>> **cd <install_dir>/GREAT-IFCB_<ver>/build**

(3) Execute "cmake ../" and then "make" to compile the source code. The executable GREAT-IFCB can be found in **<install_dir>/GREAT-IFCB_<ver>/build/Bin**.

>> **cmake ../**

>> **make**

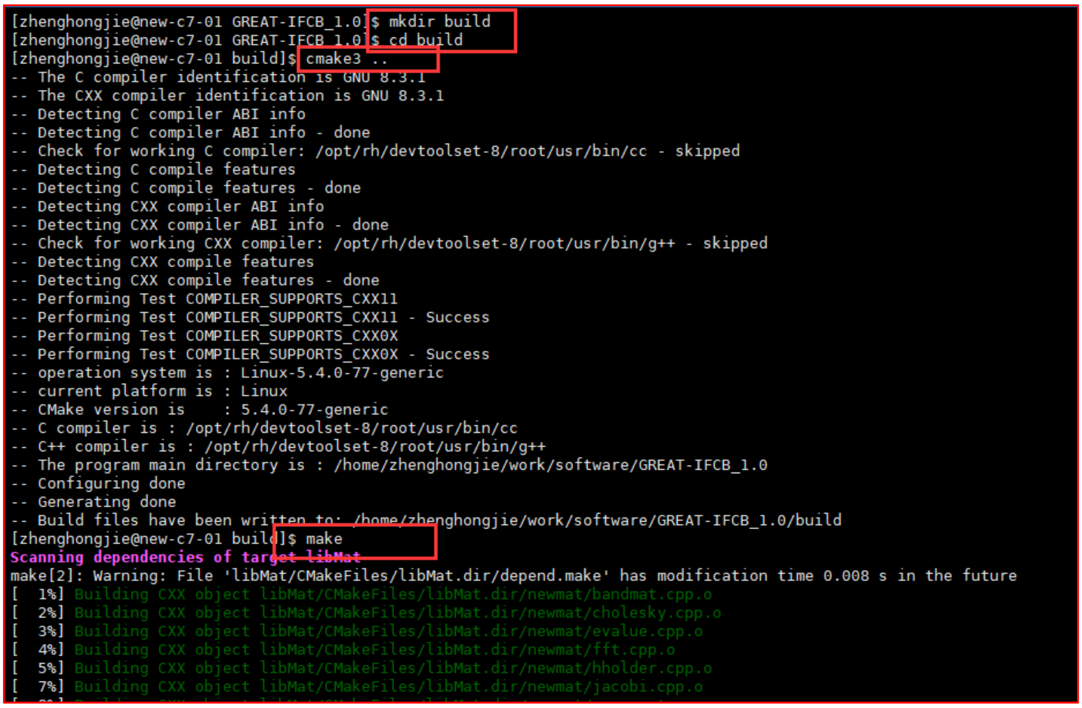

Figure 3.2-1 The compilation of GREAT-IFCB on Linux

Here, CMake will automatically find the default C/C++ compiler and its build tools on the system. If CMake reports a compiler not found error like “CMake Error: CMAKE_C_COMPILER not set”, you should specify the path to the C/C++ compiler according to the link(https://stackoverflow.com/questions/45933732/how-to-specify-a-compiler-in-cmake). Also, you can specify the build tools for CMake according to the link(https://cmake.org/cmake/help/latest/manual/cmake-generators.7.html).

4 Software and Tools Instructions

By convention, we have the following definitions firstly:
YYYY: 4-digit year; YY: 2-digit year; MM: 2-digit month; DD: 2-digit day;
DOY: 3-digit DOY (Day of Year).

4.0* Python virtual environment

It is noted that GREAT-IFCB provides some auxiliary python scripts and these scripts require Python 3.7 (or higher version) environment and some python packages. Here, we recommend that you can create a virtual environment (venv) to run these scripts. For detailed instructions of venv, please refer to the official documentation link(https://docs.python.org/3/tutorial/venv.html).

The file named "requirements.txt" list all the dependencies for these python scripts, which is under the folder **<install_dir>/GREAT-IFCB_<ver>/util/PythonScripts.** Users can type the following command line to install the dependency packages:

```
>> cd <install_dir>/GREAT-IFCB_<ver>/util/PythonScripts
>> pip3 install -r requirements.txt
```

4.1 Data Download tools

The necessary input files for generating multi-GNSS IFCB products include observation files, broadcast ephemeris files, differential code bias (DCB) files. The following instructions show more descriptions of these input files.

The GNSS observation data of global network can be downloaded from CDDIS. The number of GNSS observation data will affect the quality of IFCB estimation results, so it's not recommended to use observation data of fewer stations. The broadcast ephemeris is needed for detecting ambiguity cycle slips and BDS satellite-induced code bias correction. Therefore, broadcast ephemeris is used as the input file of IFCB estimation. The DCB file is needed to obtain the value of code-specific IFCB (CIFCB). The DCB products provided by Institute of Geodesy and Geophysics (IGG) include multi-GNSS and multi-frequency, so it's recommended to use on IFCB estimation.

GREAT-IFCB offers python scripts to download GNSS observation, navigation files and DCB files more conveniently, which are under the folder **<install_dir>/GREAT-IFCB_<ver>/util /PythonScripts**:

download_obs.py	Download mutli-GNSS observation files -ftp://gdc.cddis.eosdis.nasa.gov/pub/gps/data/daily/YYYY/DOY
download_nav.py	Download multi-GNSS broadcast ephemeris files. -ftp://gdc.cddis.eosdis.nasa.gov/pub/gps/data/campaign/mgex/daily/rinex3/YYYY/DOY/YYp
download_dcb.py	Download mutli-GNSS DCB files provided by IGG. -ftp://gdc.cddis.eosdis.nasa.gov/pub/gps/products/mgex/dcb/YYYY -ftp://igs.gnsswhu.cn/pub/gps/products/mgex/dcb/YYYY

To download multi-GNSS observation files, the users need to type command line like:

```
>> python download_obs.py <year> <doy> <length> <destination> <sitelist>
```

Here, the <sitelist > indicates the file that contains the station list to be download and the format of

<sitelist> is shown in Fig. 4.1-1.

Fig. 4.1-1 The format of <sitelist >

To download multi-GNSS broadcast ephemeris files, the users need to type command lines like:

>> python download_nav.py <year> <doy> <length> <destination>

To download multi-GNSS DCB files, the users need to type command lines like:

>> python download_dcb.py <year> <doy> <length> <destination>

All "download_*.py" files can add "-h" or "--help" command line option for more detailed usage information.

4.2 GREAT-IFCB software

Before performing IFCB estimation, the users need to generate configuration files in XML (Extensible Markup Language) format. The details of IFCB estimation configuration file setting is explained in A.1. The template configuration file named "**IFCB_CONFIG.xml**" is provided for easier reference and modification, which is under the folder **<install_dir>/GREAT-IFCB_<ver>/doc.**

To run "GREAT-IFCB" in a single project, the users only need to type command lines:

>> cd <path_to_config>

>> <path_to_executable>/GREAT-IFCB -x IFCB_CONFIG.xml

<path_to_config> and **<path_to_executable>** indicates the path to configuration file and the GREAT-IFCB executable program, respectively. The output of GREAT-IFCB will include the IFCB estimation results file, the format of IFCB estimation results file can be found in A.2.

Here, the python script named "**ifcb.py**" is provided for users to generate configuration files and make a batch processing more conveniently. The python script is under the folder **<install_dir>/GREAT-IFCB_<ver>/util/PythonScripts**

One can type "**ifcb.py** -h" or "**ifcb.py** --help" for usage information as shown in Fig 4.2-1.

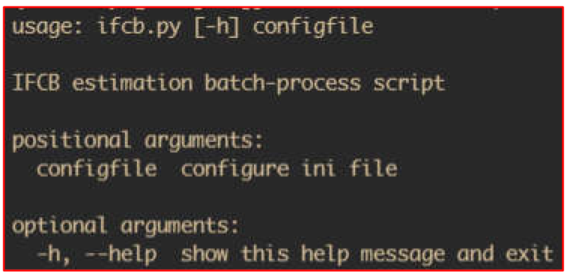

Figure 4.2-1 The help information of "ifcb.py"

To run this script, the users need to prepare an Initialization File (.ini). The Fig. 4.2-2 shows the format of the Initialization file for "**ifcb.py**". The template Initialization File named "**ifcb_example.ini**" have been prepared, which was placed in the same directory as file "**ifcb.py**".

```
; -YYYY- can be replaced by 4-digit year
; -DDD- can be replaced by 3-digit DOY (Day of Year)
[project]
; Begin Time: Year-Mon-Day
ymd_beg = 2021-01-03
; End Time  : Year-Mon-Day
ymd_end = 2021-01-03
; Begin Time in Every Day: Hour:Min:Sec
hms_beg = 00:00:00
; End Time in Every Day  : Hour:Min:Sec
hms_end = 23:55:00
; System  : G or C or E (Space separated)
satsys = G C E
; GPS BAND : (Space separated)
gps_band = 1 2 5
; GAL BAND : (Space separated)
gal_band = 1 5 7
; BDS BAND : (Space separated)
bds_band = 2 7 6
; SiteList: Site List Directory
sitelist = ./site_list
; Interval: Sample Interval in Seconds
interval = 30
; Working Directory
work_dir = ./
; Executable program Directory
software =  <executable_path>
; PreEdit: cutoff elevation [unit: degrees]
minimum_elev = 7.0
; PreEdit: MW combination limitation [unit: cycle; 0 Means not check mw]
mw_limit = 3.0
; PreEdit: GF combination limitation [unit: meter; 0 Means not check gf]
gf_limit = 0.10
; PreEdit: gapped epoch limitation [unit: epoch; 0 Means not check gap]
gap_limit = 20
; PreEdit: short epoch limitation [unit: epoch; 0 Means not check short]
short_limit = 10

[data]
; File Directory
dcb_dir = gnss
nav_dir = gnss
obs_dir = obs
```

Figure 4.2-2 The template Initialization File "**ifcb_example.ini**"

4.3 Results Plotting and Analysis Tools

GREAT-IFCB offers the python scripts to plot and analyze the IFCB estimation results, which are under the folder **<install_dir>/GREAT-IFCB_<ver>/util/PythonScripts**. The description of each script is as follows:

draw_ifcb.py	Draw IFCB estimation results for GPS, Galileo and BDS.
draw_rms_ifcb.py	Draw root-mean-square (RMS) statistics of IFCB estimation.

`draw_ifcb.py:` One can type "**draw_ifcb.py** -h" or "**draw_ifcb.py** --help" for usage information, as shown in Fig 4.3-1.

```
usage: draw_ifcb.py [-h] [--metdata METDATA] beg_year beg_doy days system data_dir

draw IFCB series script

positional arguments:
  beg_year           beg year of IFCB products. (e.g. 2021)
  beg_doy            beg doy of IFCB products. (e.g. 001)
  days               days of IFCB products. (e.g. 10)
  system             satellite system of IFCB products. (e.g. G or GE or GCE)
  data_dir           directory of IFCB products. (-YYYY- can be repleaced by 4-digit year; -DDD-
                     can be repleaced by 3-digit doy; e.g. /works/-YYYY--DDD-/ifcb_-YYYY--DDD-_G)

optional arguments:
  -h, --help         show this help message and exit
  --metdata METDATA  absolute directory of satellite metdata file. Default is using the info from
                     https://files.igs.org/pub/station/general/igs_satellite_metadata.snx
```

Figure 4.3-1 The help information of "draw_ifcb.py"

Take the GPS satellites as an example. Fig. 4.3-2 generated by "**draw_ifcb.py**" shows the estimated IFCB series from DOY 001 to 006 of 2021 for GPS satellites.

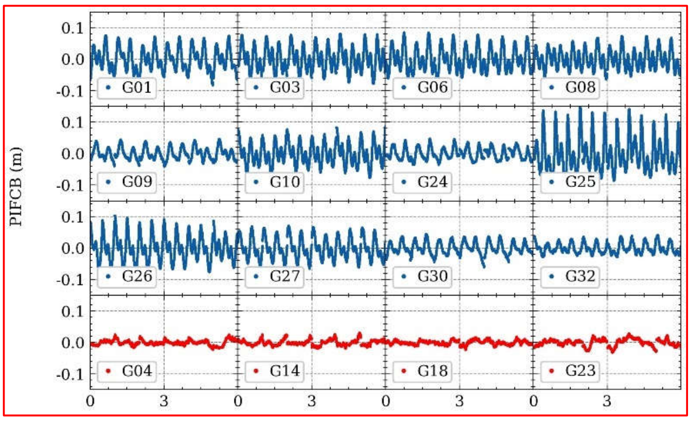

Figure 4.3-2 IFCB estimation results for GPS satellites from DOY 001 to 007 of 2021. The blue dots denote GPS Block IIF satellites and the red dots denote GPS III satellites

draw_rms_ifcb.py: One can type "**draw_rms_ifcb.py** -h" or "**draw_rms_ifcb.py** --help" for usage information, as shown in Fig 4.3-3.

```
usage: draw_rms_ifcb.py [-h] [--metdata METDATA] beg_year beg_doy days system data_dir

draw RMS of IFCB series script

positional arguments:
  beg_year           beg year of IFCB products. (e.g. 2021)
  beg_doy            beg doy of IFCB products. (e.g. 001)
  days               days of IFCB products. (e.g. 10)
  system             satellite system of IFCB products. (e.g. G or GE or GCE)
  data_dir           directory of IFCB products. (-YYYY- can be repleaced by 4-digit year; -DDD-
                     can be repleaced by 3-digit doy; e.g. /works/-YYYY--DDD-/ifcb_-YYYY--DDD-_G)

optional arguments:
  -h, --help         show this help message and exit
  --metdata METDATA  absolute directory of satellite metdata file. Default is using the info from
                     https://files.igs.org/pub/station/general/igs_satellite_metadata.snx
```

Figure 4.3-3 The help information of "draw_rms_ifcb.py"

Take the GPS satellites as an example. Fig. 4.3-4 generated by "**draw_rms_ifcb.py** " shows the RMS statistics of the estimated IFCBs from DOY 001 to 006 of year 2021 for GPS satellites

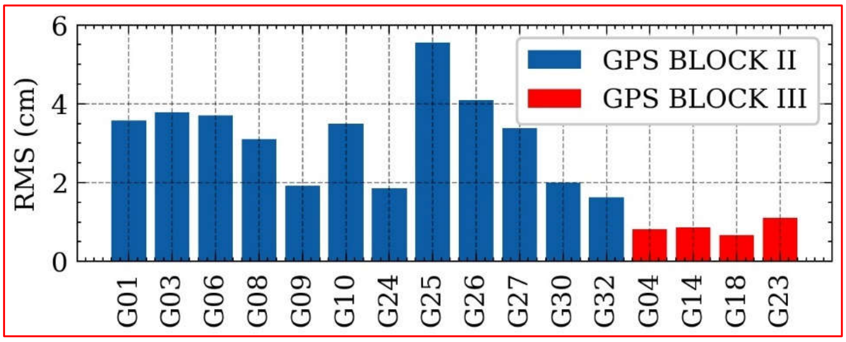

Figure 4.3-4 RMS statistics of the estimated IFCBs for GPS satellites

5 Sample Data Processing Instructions

The Sample Data is provided under the folder **<install_dir>/GREAT-IFCB_<ver>/sample_data.** Before using the python scripts to process sample data, users should install the dependency packages follow the section 4.0.

5.1 Data Preparation

All necessary input files already have been downloaded under the folder **<install_dir>/GREAT-IFCB_<ver>/sample_data/IFCB_2021003/.** The **obs** directory contains observations files and **gnss** directory contains broadcast ephemeris file and DCB file. The file named "site_list" shows the station list used by the sample data. Users can also download all these input files themselves by the python scripts mentioned on section 4.1.

To download these observation files, the users need to type command line like:

```
>> cd <install_dir>/GREAT-IFCB_<ver>/util/PythonScripts
>> python download_obs.py 2021 3 1 <dest> <install_dir>/GREAT-IFCB_ver/sample_data/IFCB_2021003/site_list
```

To download this broadcast ephemeris file, the users need to type command lines like:

```
>> cd <install_dir>/GREAT-IFCB_<ver>/util/PythonScripts
>> python download_nav.py 2021 3 1 <dest>
```

To download this DCB files, the users need to type command lines like:

```
>> cd <install_dir>/GREAT-IFCB_<ver>/util/PythonScripts
>> python download_dcb.py 2021 3 1 <dest>
```

Here **<dest>** indicates the path where the downloaded files are saved.

5.2 IFCB estimation

The configuration files that can work with the sample data has been prepared under the **<install_dir>/GREAT-IFCB_<ver>/sample_data/IFCB_2021003/.** The prepared configuration files are named as "**IFCB_<platform>.xml**", and **<platform>** includes "windows" and "linux". The only difference between the different platform configuration files is the file path separator.

To execute IFCB estimation, the users only need to type command line like:

```
>> cd <install_dir>/GREAT-IFCB_<ver>/sample_data/IFCB_2021003/
>> <executable_path>/GREAT-IFCB -x IFCB_<platform>.xml
```

<executable_path> indicates the path to GREAT-IFCB executable program. Users can either use the existing program under the folder **<install_dir>/GREAT-IFCB_<ver>/bin/<platform>** or compile an executable program by themselves. If the program runs successfully, the IFCB estimation results file named "ifcb_2021003_GCE" will be generated in the current directory.

Users can also use the python script "**ifcb.py**" mentioned on section 4.2 to automatically generate the IFCB configuration file and execute IFCB estimation. The template initialization file needed by the script has been prepared under the **<install_dir>/GREAT-IFCB_<ver>/sample_data/IFCB_2021003/.** The prepared template initialization file is named "**ifcb.ini**". Users only need to edit "**ifcb.ini**" to set the software option to the path where GREAT-IFCB executable program exist. Figure 5.2-1 marks the contents of "**ifcb.ini**" that need to be modified with red circles. Then users can type command line like:

```
>> cd <install_dir>/GREAT-IFCB_<ver>/sample_data/IFCB_2021003/
```

>> python <install_dir>/GREAT-IFCB_<ver>/util/PythonScripts/ifcb.py ifcb.ini

```
; GAL BAND : (Space separated)
gal_band = 1 5 7
; BDS BAND : (Space separated)
bds_band = 2 7 6
; SiteList: Site List Directory
sitelist = ./site_list
; Interval: Sample Interval in Seconds
interval = 30
; Working Directory
work_dir = ./
; Executable program Directory
software = <executable_path>
; PreEdit: cutoff elevation [unit: degrees]
minimum_elev = 7.0
; PreEdit: MW combination limitation [unit: cycle; 0 Means not check mw]
mw_limit = 3.0
; PreEdit: GF combination limitation [unit: meter; 0 Means not check gf]
gf_limit = 0.10
; PreEdit: gapped epoch limitation [unit: epoch; 0 Means not check gap]
gap_limit = 20
; PreEdit: short epoch limitation [unit: epoch; 0 Means not check short]
```

Figure 5.2-1 The help information of "draw_rms_ifcb.py"

Finally, the IFCB configuration file named "**ifcb_2021003.xml**" and the IFCB estimation results file named "**ifcb_2021003_GCE**" will be generated in the current directory.

It is noted that the results directory under the **<install_dir>/GREAT-IFCB_<ver>/sample_data/IFCB_2021003/** contains the reference file for the IFCB estimation results, which is named as "**ifcb_20201003_GCE_ref**". The reference file should be identical to the IFCB estimation results file if users follow the above instructions exactly.

5.3 Results Plotting

The python scripts "**draw_*.py**" mentioned on section 4.3 can be used to plot and analyze the IFCB estimation results.

To draw the estimated IFCB series from the "**IFCB_2021003_GCE**" obtained in the previous section, users need to type the command link:

>> cd <install_dir>/GREAT-IFCB_<ver>/sample_data/IFCB_2021003/

>> python <install_dir>/GREAT-IFCB_<ver>/util/PythonScripts/draw_ifcb.py 2021 3 1 GCE ifcb_2021003_GCE

Then the result plots named "**ifcb_result_<sys>.jpg**" will be generated in current directory. Here, <sys> includes the "GPS", "GAL" and "BDS".

To draw RMS statistics of IFCB estimation results from the "**IFCB_2021003_GCE**", users need to type the command link

>> cd <install_dir>/GREAT-IFCB_<ver>/sample_data/IFCB_2021003/

>> python <install_dir>/GREAT-IFCB_<ver>/util/PythonScripts/draw_rms_ifcb.py 2021 3 1 GCE ifcb_2021003_GCE

Then the result plots named "**ifcb_rms_<sys>.jpg**" will be generated in current directory. Here, <sys> includes the "GPS", "GAL" and "BDS".

Similarly, the results directory under the **<install_dir>/GREAT-IFCB_<ver>/sample_data/IFCB_2021003/** contains the reference files to the results plots, which are named as "**ifcb_result_<sys>_ref.jpg**" and "**ifcb_rms_<sys>_ref.jpg**". The reference files should be identical to the user-generated plots if users follow the above instructions exactly.

6 Support

Any suggestions, corrections, and comments about GREAT-IFCB are sincerely welcomed and could be sent to:

Xingxing Li
Email: lxlq109121@gmail.com, xxli@sgg.whu.edu.cn
Address: Room 421, School of Geodesy and Geomatics, Wuhan University, 129 Luoyu Road, 430079, Wuhan, China

Hongjie Zheng
Email: hjzheng@whu.edu.cn
Address: School of Geodesy and Geomatics, Wuhan University, 129 Luoyu Road, 430079, Wuhan, China

It is recommended to acknowledge GREAT-IFCB when you find it useful!

References

Bock H, Dach R, Jäggi A, Beutler G (2009) High-rate GPS clock corrections from CODE: support of 1 Hz applications. J Geod 83, 1083. doi: 10.1007/s00190-009-0326-1

Fan L, Shi C, Li M, Wang C, Zheng F, Jing G, Zhang J (2019) GPS satellite inter-frequency clock bias estimation using triple-frequency raw observations. J Geod 93, 2465–2479. doi: 10.1007/s00190-019-01310-5

Ge M, Chen J, Douša J, Gendt G, Wickert J (2012) A computationally efficient approach for estimating high-rate satellite clock corrections in realtime. GPS Solut 16, 9–17. doi: 10.1007/s10291-011-0206-z

Hauschild A, Montenbruck O (2009) Kalman-filter-based GPS clock estimation for near real-time positioning. GPS Solut 13, 173–182. doi: 10.1007/s10291-008-0110-3

Kouba J (2009) A guide to using international GNSS service (IGS) products. http://igscb.jpl.nasa.gov/igscb/resource/pubs/UsingIGSProductsVer21.pdf

Li H, Zhou X, Wu B, Wang J (2012) Estimation of the inter-frequency clock bias for the satellites of PRN25 and PRN01. Sci. China Phys. Mech. Astron. 55, 2186–2193. doi: 10.1007/s11433-012-4897-0

Li H, Zhou X, Wu B (2013) Fast estimation and analysis of the inter-frequency clock bias for Block IIF satellites. GPS Solut 17, 347–355. doi: 10.1007/s10291-012-0283-7

Li X, Ge M, Dai X, Ren X, Fritsche M, Wickert J, Schuh H (2015) Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo. J Geod 89, 607–635. doi: 10.1007/s00190-015-0802-8

Montenbruck O, Hugentobler U, Dach R, et al. (2012) Apparent clock variations of the Block IIF-1 (SVN62) GPS satellite. GPS Solut 16, 303–313. doi: 10.1007/s10291-011-0232-x

Pan L, Zhang X, Li X, Liu J, Li X (2017) Characteristics of inter-frequency clock bias for Block IIF satellites and its effect on triple-frequency GPS precise point positioning. GPS Solut 21, 811–822. doi: 10.1007/s10291-016-0571-8

Pan L, Zhang X, Guo F, Liu J (2019) GPS inter-frequency clock bias estimation for both uncombined and ionospheric-free combined triple-frequency precise point positioning. J Geod 93, 473–487. doi: 10.1007/s00190-018-1176-5

Petit G, Luzum B, (eds.) (2010) IERS Conventions (2010), IERS Technical Note 36, Verlagdes Bundesamts für Kartographie und Geodäsie, Frankfurt am Main, Germany. http://tai.bipm.org/iers/conv2010

Vaclavovic P, Dousa J, Gyori G (2013) G-Nut software library - state of development and first results. Acta Geodyn Geomater 10(4):431–436. doi:10.13168/AGG.2013.0042

Wanninger L, Beer S (2015) BeiDou satellite-induced code pseudorange variations: diagnosis and therapy. GPS Solut 19, 639–648. doi: 10.1007/s10291-014-0423-3

Zhang B, Teunissen P, Yuan Y (2017) On the short-term temporal variations of GNSS receiver differential phase biases. J Geod 91, 563–572. doi: 10.1007/s00190-016-0983-9

Appendix A File Formats

By convention, we have the following definitions firstly:

YYYY: 4-digit year; YY: 2-digit year; MM: 2-digit month; DD: 2-digit day;

DOY: 3-digit DOY (Day of Year);

hh: 2-digit hour; mm: 2-digit minute; ss: 2-digit second

SITE: upper format of site name; site: lower format of site name

A.1 Configuration XML for GREAT-IFCB

A configuration file containing processing options, solution options and file options. It is expressed in XML format, which contains the "Keyword = Value" form records indicating the various options. The texts starting with "<!--" and ending with "-->" in a line are treated as comments. The following table shows the format of the XML file.

Item	Descriptions	Element in XML File
General Settings for IFCB Estimation (First Level Element)		**<gen>**
Begin Time	Set begin time in form of GPS time. The format is, "YYYY-MM-DD hh:mm:ss".	<beg>
End Time	Set end time in form of GPS time. The format is, "YYYY-MM-DD hh:mm:ss".	<end>
Station List	Set station list for IFCB estimation.	<rec>
Satellite System	Set satellite system for IFCB estimation. Note that only one system is allowed here.	<sys>
Sampling Interval	Set sampling interval of IFCB estimation.	<int>
Excluded Satellites	Set the excluded satellites for IFCB estimation. Fill in the PRN numbers of the satellites separated by spaces.	<sat_rm>
Input Files Settings for IFCB Estimation (First Level Element)		**<inputs>**
RINEX OBS File	RINEX observation file used for IFCB estimation. Note that it supports RINEX 2.10, 2.11, 2.12, 3.00, 3.01, 3.02, 3.03, 3.04 OBS.	<rinexo>
RINEX NAV File	RINEX navigation file used for IFCB estimation. Note that it supports RINEX 2.10, 2.11, 2.12, 3.00, 3.01, 3.02, 3.03, 3.04 NAV.	<rinexn>
DCB File	DCB file used for IFCB estimation. It is in IGG format.	<biabern>
Satellite Systems Settings for IFCB Estimation (First Level Element)		**<gps> or <bds> or <gal>**
Band	band used in each system	<band>
Freq	order of using band in combined observations	<freq>
Preedit Settings for IFCB Estimation (First Level Element)		**<preedit>**
Minimum Elevation	Minimum elevation in detecting cycle slip	<minimum_elev>

Item	Descriptions	Element in XML File
MW Combination Limitation	MW combination limitation in detecting cycle slip. The unit of <mw_limit> is cycles and <valid> indicates whether checking MW combination.	<check_mw>
GF Combination Limitation	GF combination limitation in detecting cycle slip. The unit of <gf_limit> is meters and <valid> indicates whether checking GF combination.	<check_gf>
Gapped Epoch Limitation	Gapped epoch arc limitation in detecting cycle slip. The unit of <gap_limit> is epoch and <valid> indicates whether checking gapped epoch.	<check_gap>
Short Epoch Limitation	Short epoch arc limitation in detecting cycle slip. The unit of <short_limit> is epoch and <valid> indicates whether checking short epoch arc.	<check_short>
Output Files Settings for IFCB Estimation (First Level Element)		**<outputs>**
Append	Whether to rewrite the log in original log file, the corresponding attribute is "append". The value of "append" is optional: - true: append - false: not append	
Verb	Set log file output level, the corresponding attribute is "verb". The value of "verb" is optional: - 0/1/2/3/4/5 (5: highest > 0: lowest)	
Log File	Set output log file.	<log>
IFCB File	Set output IFCB file. For the format, please refer A.2.	<ifcb>

EXAMPLE

```xml
<config>
<!-- general descriptions -->
<gen>
   <beg> 2021-01-03 00:00:00 </beg> <!-- beg time -->
   <end> 2021-01-03 23:55:00 </end> <!-- end time -->
   <sys> GPS BDS GAL </sys>         <!-- GNSS system: GPS/BDS/GAL -->
   <rec> SOLO          </rec>         <!-- processing sites -->
   <int> 30            </int>         <!-- sampling interval -->
   <sat_rm> </sat_rm>           <!-- exclude certain satellites during IFCB estimation -->
</gen>
<!-- input descriptions -->
<inputs>
   <rinexn> brdm0030.21p </rinexn> <!-- obs RINEX decoder -->
   <rinexo> solo0030.21o </rinexo> <!-- nav RINEX decoder -->
   <biabern>CAS0MGXRAP_20210030000_01D_01D_DCB.BSX</biabern> <!-- code bias decoder -->
</inputs>
<!-- system descriptions -->
<gps>
   <band>1 2 5</band>  <!-- band used in GPS observations -->
   <freq>1 2 3</freq>  <!-- order of using GPS band in combined observations -->
</gps>
<bds>
   <band>2 7 6</band>  <!-- band used in BDS observations -->
   <freq>1 2 3</freq>  <!-- order of using BDS band in combined observations -->
</bds>
<gal>
   <band>1 5 7</band>  <!-- band used in GAL observations -->
   <freq>1 2 3</freq>  <!-- order of using GAL band in combined observations -->
</gal>
<!-- preedit setting descriptions-->
<preedit>
   <minimum_elev> 7.0 </minimum_elev>           <!-- cutoff satellite elevation [unit:degrees] -->
   <check_mw mw_limit="3.0" valid="true"/>      <!-- MW combination limitation [unit:cycles] -->
   <check_gf gf_limit="0.1" valid="true"/>      <!-- GF combination limitation [unit:meters] -->
   <check_gap gap_limit="20" valid="true"/>     <!-- gapped epoch limitation [unit:epoch] -->
   <check_short short_limit="10" valid="true"/> <!-- short epoch limitation [unit:epoch] -->
</preedit>
<!-- output descriptions  -->
<outputs append="false" verb="0">
   <log> LOGRT.log </log>          <!-- log encoder -->
   <ifcb> ifcb_2021003_GCE </ifcb> <!-- upd encoder -->
</outputs>
</config>
```

Figure A.2-1 Example of XML for IFCB estimation

A.2 IFCB File

DESCRIPTION

IFCB file is the output of GREAT-IFCB and it contains the IFCB records. A line indicates an IFCB record of specified satellite. The following table shows the format of the IFCB file.

No	Record/Field	Description	Formats
1	IFCB messages	A line contains IFCB information, which consists of the following fields.	
(1)	Epoch time	It indicates the valid time of IFCB. The format is showed as follows. - " EPOCH-TIME"	 1X,A

No	Record/Field	Description	Formats
		- Modified Julian Day	2X,I6
		- Time of a day in seconds	2X,F8.1
(2)	IFCB messages	A line contains value and standard deviation of IFCB for specified satellite.	
		- Availability indicator (" " => available, "x" => unavailable)	A1
		- PRN number	A3
		- IFCB value, unit: meter	8X,F10.3
		- standard deviation of IFCB	F10.3
		- Number of stations used for specified satellite IFCB estimation	I5

EXAMPLE

```
 EPOCH-TIME   58689   86040.0
 G01            -0.011     0.005   54
xG02             0.000 10000.000    0
 G03            -0.019     0.007   43
xG04             0.000 10000.000    0
xG05             0.000 10000.000    0
 G06             0.012     0.032   28
xG07             0.000 10000.000    0
 G08             0.017     0.008   72
 G09             0.001     0.009   59
 G10            -0.011     0.007   30
 EPOCH-TIME   58689   86070.0
 G01            -0.013     0.006   54
xG02             0.000 10000.000    0
 G03            -0.020     0.008   42
xG04             0.000 10000.000    0
xG05             0.000 10000.000    0
 G06             0.011     0.005   28
xG07             0.000 10000.000    0
 G08             0.016     0.008   72
 G09             0.001     0.006   59
 G10
```

Figure A.6-1 Example of IFCB file format

Appendix B Default Priorities of Multiple Signals

If input observation data contains multiple signals in a frequency, GREAT-IFCB will selects a signal for processing by the following default priorities.

System	Frequency.	Signal Priority (1: highest > 10: lowest) *							
		1	2	3	4	5	6	7	8
GPS	L1	1M	1Y	1W	1P	1X	1L	1S	1C
	L2	2M	2Y	2W	2P	2X	2L	2D	2C
	L5	5X	5Q	5I					
Galileo	E1	1Z	1X	1C	1B	1A			
	E5a	5X	5Q	5I					
	E5b	7X	7Q	7I					
BeiDou	B1	2X	2Q	2I					
	B2	7X	7I	7Q					
	B3	6X	6Q	6I					
	B2a	5X	5P	5D					
	B1C	1X	1P	1D					

Appendix C Mathematical method

This appendix introduces the IFCB estimation method adopted by the GREAT-IFCB software.

GREAT-IFCB can generate multi-GNSS IFCB products based on the ED approach with multi-frequency observations. The mathematical methods used on the GREAT-IFCB software are introduced in this section. The IFCB representation is first introduced in detail, and the IFCB estimation method based on the ED approach is then described.

C.1 IFCB representation

Here, we use the frequency type of L1, L2 and L3 to indicate uniformly three different frequencies on satellite system (i.e. GPS L1/L2/L5, Galileo E1/E5a/E5b, BDS B1/B2/B3). Then the IF combined pseudorange and phase observations can be written as (Li et al. 2015; Montenbruck et al. 2012; Zhang et al. 2017):

$$P_{r,IFmn}^{s} = \rho_{r}^{s} + dt_{r} - dt^{s} + T_{r}^{s} + d_{r,IFmn} + d_{IFmn}^{s} + e_{r,IFmn}^{s} \tag{1}$$

$$L_{r,IFmn}^{s} = \rho_{r}^{s} + dt_{r} - dt^{s} + T_{r}^{s} + N_{r,IFmn}^{s} + \overline{B}_{r,IFmn} + \delta B_{r,IFmn} + \overline{B}_{IFmn}^{s} + \delta B_{IFmn}^{s} + \varepsilon_{r,IFmn}^{s} \tag{2}$$

where the subscripts s and r indicate satellite and receiver, respectively; m and n denote the frequency type; ρ_r^s denotes the geometric distance from satellite phase centers to receiver antennas in meters; dt_r and dt^s represent the absolute clock offsets of receiver and satellite in meters; T_r^s denotes to the tropospheric delay in meters; $N_{r,IFmn}^{s}$ is the IF combined carrier phase ambiguity in meters; $d_{r,IFmn}$ and d_{IFmn}^{s} denote the IF combined code-specific hardware delay at receiver and satellite sides in meters, respectively; $\overline{B}_{r,IFmn}$ and $\overline{B}_{IFmn}^{s}$ denote the time-invariant part of the IF combined phase-specific hardware delay at receiver and satellite sides in meters, respectively; $\delta B_{r,IFmn}$ and δB_{IFmn}^{s} denote the time-variant part of the IF combined phase-specific hardware delay at receiver and satellite sides in meters, respectively; $e_{r,IFmn}^{s}$ and $\varepsilon_{r,IFmn}^{s}$ are the IF combined un-modeled bias of the code and carrier phase observations in meters, respectively. It is worth nothing that antenna phase center offsets (PCOs) and phase center variations (PCVs), phase wind up, tidal effects, relativistic effects and tropospheric delay should be considered in ρ_r^s (Kouba 2009). In addition, the Shapiro delay (Petit and Luzum 2010) and BDS-2 satellite-included code bias (Wanninger and Beer 2015) should also be corrected.

The satellite code-specific and time-variant phase-specific hardware delays will be grouped in satellite clock products on PCE (Hauschild and Montenbruck 2009; Bock et al. 2009; Ge et al. 2012). Hence, the difference of satellite clock products based on L1/L2 and L1/L3 IF combinations can be expressed as:

$$
\begin{aligned}
IFCB_{123}^{s} &= dt_{_{IF12}}^{s} - dt_{_{IF13}}^{s} \\
&= dt^{s} + d_{IF12}^{s} + \delta B_{IF12}^{s} - (dt^{s} + d_{IF13}^{s} + \delta B_{IF13}^{s}) \\
&= CIFCB_{123}^{s} + PIFCB_{123}^{s} \\
CIFCB_{123}^{s} &= d_{IF12}^{s} - d_{IF13}^{s} \\
PIFCB_{123}^{s} &= \delta B_{IF12}^{s} - \delta B_{IF13}^{s}
\end{aligned}
\tag{3}
$$

where $CIFCB$ and $PIFCB$ refer to the code-specific part and phase-specific part of IFCB (Fan et al. 2019), respectively; digital subscripts denote the carrier frequency band. Since the code-specific IFCB (CIFCB) can be obtained from the linear combination of existing Different Code Bias (DCB) products (Pan et al. 2019), here we mainly focus on the value of phase-specific IFCB (PIFCB).

C.2 Phase-specific IFCB estimation

The GFIF combination is usually defined as the difference between L1/L2 and L1/L3 IF combined carrier phase observations. Since the GFIF combination can eliminate the frequency-independent components (i.e., the geometric distance, the absolute clock offsets of both receiver and satellite, and the tropospheric delay), it is used for IFCB estimation. From Eq. (1) and Eq. (2), the GFIF combination equations can be expressed as:

$$
\begin{aligned}
GFIF_{r,123}^{s} &= L_{r,IF12}^{s} - L_{r,IF13}^{s} \\
&= N_{r,GFIF}^{s} + \overline{B}_{r,GFIF} + \delta B_{r,GFIF} + \overline{B}_{GFIF}^{s} + \delta B_{GFIF}^{s} + \varepsilon_{r,GFIF}^{s}
\end{aligned}
\tag{4}
$$

with

$$
\begin{cases}
N_{r,GFIF}^{s} = N_{r,IF12}^{s} - N_{r,IF13}^{s} \\
\overline{B}_{r,GFIF} = \overline{B}_{r,IF12} - \overline{B}_{r,IF13} \\
\delta B_{r,GFIF} = \delta B_{r,IF12} - \delta B_{r,IF13} \\
\overline{B}_{GFIF}^{s} = \overline{B}_{IF12}^{s} - \overline{B}_{IF13}^{s} \\
\delta B_{GFIF}^{s} = \delta B_{IF12}^{s} - \delta B_{IF13}^{s} \\
\varepsilon_{r,GFIF}^{s} = \varepsilon_{r,IF12}^{s} - \varepsilon_{r,IF13}^{s}
\end{cases}
\tag{5}
$$

Then, from Eq. (3), Eq. (4) and Eq. (5), PIFCB can be derived from the GFIF combination:

$$
PIFCB_{r,123}^{s} = GFIF_{r,123}^{s} - (N_{r,GFIF}^{s} + \overline{B}_{r,GFIF} + \delta B_{r,GFIF} + \overline{B}_{GFIF}^{s}) \tag{6}
$$

Here PIFCB replaces the δB_{GFIF}^{s} on account of the same definition.

Generally, the ED approach can be used to estimate PIFCB (Li et al. 2013; Pan et al. 2017). If there are no cycle slips between adjacent epochs, the difference between GFIF combination and PIFCB are time-invariant. In addition, the time-variant part of receiver phase-specific hardware delays is small enough to be ignored (Li et al. 2012; Pan et al. 2017). Hence, the ED PIFCB can be obtained from the ED GFIF combination, that is:

$$
\Delta PIFCB_{r,123}^{s}(t,t+1) = GFIF_{r,123}^{s}(t+1) - GFIF_{r,123}^{s}(t) \tag{7}
$$

where $\Delta PIFCB_{r,123}^{s}(t,t+1)$ refers to the ED PIFCB between the epoch t and $t+1$. Based on Eq. (3), it can be noticed that the ED PIFCB is only related to the phase-specific hardware delays at satellite sides. Therefore, assume that there are x stations observing satellite s at epoch t and $t+1$, the average ED PIFCB values can be formulated as:

$$\Delta PIFCB_{123}^{s}(t,t+1)=\sum_{r=1}^{x}\Delta PIFCB_{r,123}^{s}(t,t+1)/x \tag{8}$$

Then each epoch PIFCB values can be computed by cumulative method, that is:

$$PIFCB_{123}^{s}(t)=PIFCB_{123}^{s}(t_0)+\sum_{j=0}^{t-1}\Delta PIFCB_{123}^{s}(j,j+1) \tag{9}$$

where $PIFCB_{123}^{s}(t)$ refers to the undifferenced PIFCB value of epoch t; t_0 refers to the reference epoch, and is usually set to the first epoch of each day. Here, GREAT-IFCB uses the zero-mean constraint over the whole day in order to determine the value of $PIFCB_{123}^{s}(t_0)$. Assume that there are k epochs, the value of $PIFCB_{123}^{s}(t_0)$ can be expressed as:

$$\begin{aligned}&\sum_{j=0}^{k-1}PIFCB_{123}^{s}(j)=0\\&PIFCB_{123}^{s}(t_0)=[-\sum_{j=1}^{k-1}(k-j)\Delta PIFCB_{123}^{s}(j-1,j)]/k\end{aligned} \tag{10}$$

And the PIFCB values of other epochs can be determined by (9).