

PyHawk

An efficient gravity recovery solver for low–low satellite-to-satellite tracking gravity missions

WU, Yi ; Yang, Fan; Liu, Shuhao; Forootan, Ehsan

Published in:
Computers & Geosciences

DOI (link to publication from Publisher):
[10.1016/j.cageo.2025.105934](https://doi.org/10.1016/j.cageo.2025.105934)

Publication date:
2025

Document Version
Early version, also known as pre-print

[Link to publication from Aalborg University](#)

Citation for published version (APA):
WU, Y., Yang, F., Liu, S., & Forootan, E. (2025). PyHawk: An efficient gravity recovery solver for low–low satellite-to-satellite tracking gravity missions. *Computers & Geosciences*, 201, Article 105934.
<https://doi.org/10.1016/j.cageo.2025.105934>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

PyHawk: An efficient gravity recovery solver for low-low satellite-to-satellite tracking gravity missions

Yi Wu^a, Fan Yang^{a,b,*}, Shuhao Liu^a and Ehsan Forootan^b

^aHuazhong University of Science and Technology, School of Physics, Wuhan 430074, China

^bAalborg University, Geodesy Group, Department of Sustainability and Planning, Aalborg 9000, Denmark

ARTICLE INFO

Keywords:
GRACE(-FO)
Python toolbox
Gravity recovery
Orbit Determination
Level-2 gravity solutions
low-low satellite-to-satellite tracking

ABSTRACT


The low-low satellite-to-satellite tracking (ll-sst) gravity missions, such as the Gravity Recovery and Climate Experiment (GRACE) and its Follow-On (GRACE-FO), provide an important space-based Essential Climate Variable (ECV) over the last two decades. Due to the high-precision Global Navigation Satellite System (GNSS) receiver, accelerometers and inter-satellite ranging instrument, these ll-sst missions are able to sense extremely tiny perturbation on both the orbit and inter-satellite ranging, which can eventually project into the Earth's time-variable gravity fields. The measurement systems of these ll-sst missions are highly complex, thus, a data processing chain is required to exploit the potential of their high-precision measurements, which challenges both the general and expert users. In this study, we present an open-source, user-friendly, cross-platform and integrated toolbox "PyHawk", which is the first Python-based software in relevant field, to address the complete data processing chain of ll-sst missions including GRACE, GRACE-FO and likely the future gravity missions. This toolbox provides non-expert users an easy access to the payload data pre-processing, background force modeling, orbit integration, ranging calibration, as well as the ability for temporal gravity field recovery using ll-sst measurements. In addition, a series of high-standard benchmark tests have been provided to evaluate PyHawk, confirming its performance to be comparable with those are being used for providing the official Level-2 time-variable gravity field solutions of GRACE and GRACE-FO. Researchers working with the low-Earth-orbit space geodetic techniques, GNSS based orbit determination, and gravity field modeling can benefit from this toolbox.

CRedit authorship contribution statement

Yi Wu: Implemented the code, performed the experiments and wrote the original manuscript. **Fan Yang:** Conceived the software and wrote the original manuscript. **Shuhao Liu:** Discussed the results and reviewed the manuscript. **Ehsan Forootan:** Discussed the results and improved the manuscript.

1. Introduction

The low-low satellite-to-satellite tracking (ll-sst) technique has been one of the major advances of space geodesy in the last decades (Tapley et al., 2019). This technique, thanks to the high-precision on-board payloads, can detect extremely small perturbations in both the orbit and inter-satellite ranging (Kornfeld et al., 2019; Landerer et al., 2020). These measurements can then be inverted to reveal their source, which is the Earth's Time-Variable Gravity (TVG) field (Wahr et al., 2004). The ll-sst system was realized by the successful Gravity Recovery and Climate Experiment (GRACE) mission (2002-2017) and its successor mission (GRACE-FO, 2018-now). The monthly snapshots of TVG derived from these missions (Chen et al., 2021) are applied to compute global Terrestrial Water Storage (TWS), which

 fany@plan.aau.dk (Fan Yang)
ORCID(s):

is known as an Essential Climate Variable (ECV) to be used for understanding the Earth system processes (Eicker et al., 2016; Rodell et al., 2018; Scanlon et al., 2018; Chen et al., 2022; Yang et al., 2024a).

However, developing a reliable ll-sst data processing chain from Level-1b, i.e., processed instrument data, to Level-2, i.e., TVG product in terms of potential coefficients of spherical harmonics expansion, is computationally complex (Mayer-Gürr et al., 2021). Especially, it is expected to exploit full potential of the high-precision instruments on-board, such as the GPS receivers for precise orbit determination and the dual-frequency K/Ka band microwave and laser instruments used for tracking inter-satellite range changes (Ghobadi-Far et al., 2022). Non-expert users can hardly access this process, and even the standard of processing strategy is controversial among official producers (Meyer et al., 2019). Therefore, it would be beneficial to establish a friendly toolbox for broader user groups to address a complete but diverse ll-sst processing chain. This can not only accelerate the iteration and upgrade of Level-2 scientific products from the existing GRACE(-FO) missions, but also provide an opportunity to access and simulate future ll-sst missions, such as the Next Generation Gravity Mission (NGGM, Pail et al., 2019) and the Mass Change and Earth Science International Constellation mission (MAGIC, Daras et al., 2024). All these future missions conceptually consist of one or multiple pairs of ll-sst with the aim to resolve TVG with better orbital sampling and lower uncertainty (Purkhauser and Pail, 2019), which requires processing tools to be more accessible, extensible, compatible, and numerically efficient.

So far, the publicly available toolbox for addressing the complete processing chain of ll-sst missions includes Bernese (Meyer et al., 2016), GRACETOOLS (Darbeheshti et al., 2018), and GROOPS (Mayer-Gürr et al., 2021). The others, such as EPOS (Earth Parameters and Orbit System) developed by the GRACE(-FO) official producer (Dahle et al., 2019), are not open-source and unavailable. In addition, since Bernese does not make the module of gravity recovery available and GRACETOOLS primarily targets the simulation rather than real data processing, GROOPS remains the only option for public users. GROOPS is a high-standard and powerful toolbox with rich functionalities include (but not limited to) orbit determination, TVG recovery from GRACE(-FO) and regional gravity field modeling from terrestrial data (Mayer-Gürr et al., 2021), which make it popular across the geodesy community (Kvas et al., 2019; Eicker et al., 2020). Nevertheless, since GROOPS is developed in C++, a strong expertise with its compiling/installation/programming is required, which might still challenge the users.

In this study, we present an open-source and user-friendly toolbox, written in Python language, to target more general users. This toolbox, known here as "PyHawk", comprises a comprehensive data processing chain of ll-sst gravity missions, for example, the payload data preprocessing, background force modeling, orbit integration, inter-satellite ranging calibration, as well as the TVG recovery from GRACE(-FO) missions. PyHawk enables an easy and fast installation at multiple platforms, e.g., Windows, Linux or cluster. In addition, PyHawk is designed with advanced data exchange mechanism between user's configuration files and the core, assuring easy interaction without the need to modify any source code. Finally, particular optimization assures a computational efficiency comparable to Fortran and

C++, which are likely the most common options for the known toolbox for producing current GRACE(-FO) products.

In what follows, firstly the software architecture that consists of five functional modules is introduced in Section 2, and a novel multi-channel method that addresses the Ordinary Differential Equation (ODE) is proposed to enhance the numerical efficiency in Section 3. In Section 4, we evaluate the software's performance from various aspects through inter-comparisons against other reputed software or official Level-2 products. Then, in Section 5, a new sliding-window daily TVG product recovered from GRACE is demonstrated to present the flexibility/extension of this software and its potential for scientific applications. Finally, the conclusion and outlook are provided in Section 6.

2. Architecture

PyHawk is developed with a modular structure, which mainly consists of five independent (de-coupled) modules. One can see the overall structure in Fig. 1, and each module shall be introduced in what follows.

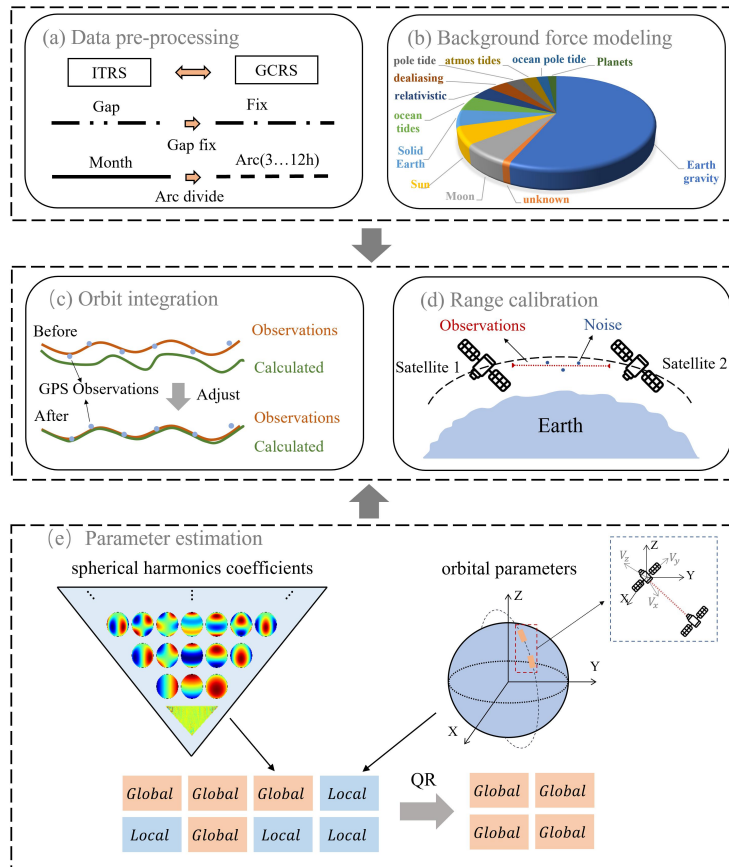


Figure 1: An overview of PyHawk structure and its workflow including: (a) Data pre-processing; (b) Background force modeling; (c) Orbit integration; (d) Range calibration; and (e) Parameter estimation.

2.1. Data pre-processing

This module is defined to have three submodules: (i) data interface, (ii) raw data pre-processing, and (iii) reference frame transformation for both time and coordinate, which we achieved by developing a Python interface to call the SOFA C++ library (Hohenkerk, 2017). Specifically, the data interface submodule is responsible for accepting and decoding the binary or ASCII data including GRACE(-FO) Level-1b data (Yang et al., 2022), de-aliasing product (Shihora et al., 2022), planet ephemerid, Earth orientation parameter and other auxiliary files (Park et al., 2021). The raw data pre-processing submodule includes (but not limited to) arc-wise data redistribution, outlier detection, gap interpolation/extrapolation, quality screening and down-sampling etc (Dahle et al., 2019).

2.2. Background force modeling

To obtain desired signals such as TWS, it is mandatory to remove other prior geophysical models beforehand, for example, solid Earth effects, polar tides, atmospheric and oceanic induced mass variations (see Yang et al., 2021). Besides, gravitational change from planets or relativistic effect or non-conservative forces must be corrected towards a precise orbit determination (Lasser et al., 2020). All these are considered herein as the background force modeling, and PyHawk pursues a high-precision modeling of them to make best of the high-precision measurements. In addition, PyHawk integrates a wide selection of the force models with flexible choices, see the supplementary information for more details.

2.3. Orbit integration

Ordinary Differential Equations (ODE) establishes the basis of dynamic orbit determination, which is also the core of II-SST missions. Existing numerical solutions for ODE are primarily divided into single-step ODE and multi-step ODE (Montenbruck et al., 2002). Single-step ODE can initiate itself accurately by taking advantage of 'small step', but meanwhile it is computationally costly. On the contrary, the multi-step ODE is less accurate but more efficient because of its 'large step'. In PyHawk, the implemented single-step ODE includes Runge-Kutta and Runge-Kutta-Nyström (RKN, Kosti et al., 2012); the multi-step ODE include Adams-Bashforth-Moulton and Gauss-Jackson (GJ, Berry and Healy, 2004). In practice, two methods are often combined together to compromise the accuracy and efficiency. For example, we suggest RKN for initialization and GJ for orbit propagation with the initial records obtained from RKN. In PyHawk, the ODE configuration is flexible, where its type, order and step length are all optional. It is worth mentioning that PyHawk develops a novel multi-channel ODE approach to gain a high computation efficiency, where details are provided in the supplementary information.

2.4. Ranging calibration

The inter-satellite along-track range rate, measured by the onboard ranging instrument like K-Band Ranging (KBR), is extremely sensible to even tiny TVG signals, so that its treatment must be careful (Flechtner et al., 2016). Therefore, a series of corrections (light-time correction and antenna phase centre correction etc) and calibrations (de-noising and outlier detection) are addressed by this module. In particular, it is relevant to eliminate the one Cycle-Per-Revolution (1-CPR) noise, which is a resonance to the orbital period (approximately 1.5 hours) of the GRACE satellites (Zhao et al., 2011; Yang et al., 2018).

2.5. Parameter estimation

This module is the basis of orbit/accelerometer calibration and TVG recovery, which mainly comprises three sub-modules: (1) parametrization, (2) observation equation, (3) large-scale matrix inversion. In submodule-1, the unknowns or the parameters to be solved, are explicitly defined and assigned with attribution of being either local or global. In submodule-2, the variational-equation method (Yang et al., 2017) is implemented to establish the observation equation to relate the previous parameters with the observations. Then, the submodule-3 is able to (i) transform the observation equation into the normal equation, and accumulate the normal equation via sequential least square solution, (ii) split/re-group the normal equation by parameter attribution, and (iii) estimate the parameters by solving ill-posed normal equations (Save et al., 2012).

3. A multi-channel ODE method

In addition to the orbit integration discussed in Sec. 2.3, ODE are also central to gravity recovery, so that it is the core of the software. Experiments have shown that executing ODE requires the most computational resources, making it essential to optimize this process in order to significantly improve the software's efficiency. Specifically, future gravity missions, which will involve multiple pairs of GRACE-like satellites, will face increasing computational burdens. As a result, there will be a more urgent need for ODE optimization compared to current gravity missions.

To address this computational challenge, we propose a novel multi-channel ODE method. Before presenting this method, we first describe the traditional single-satellite ODE, which we modify to create the new approach. Based on Newton's second law, the equation of motion for a single satellite in an inertial reference frame is given by:

$$\begin{cases} \ddot{r}(t) = f(t, r, \dot{r}) \\ r(t_0) = r_0 \\ \dot{r}(t_0) = \dot{r}_0, \end{cases} \quad (1)$$

where r , \dot{r} , \ddot{r} represent the position, velocity, and acceleration vectors of the satellite at time t , respectively. In this

context, r_0 and \dot{r}_0 denote the initial position and velocity of the satellite. The function f represents the forces acting on the satellite, including both conservative and non-conservative forces, as described in Section 2.2. Then, solving $r(t)$ and $\dot{r}(t)$ from Eq. (1) is a classic second-order ordinary differential problem, for which an analytical solution is rarely available. Instead, numerical integrators are often implemented, which are found to reach a good accuracy as the analytical solution, e.g., to compute Kepler orbit (Montenbruck et al., 2002).

In analogy with Eq.(1) that solely indicates orbit integration, it is known that the acquisition of state transition matrix and the parameter sensitivity matrix (which are used to construct the design matrix to solve gravity field related parameters) also take advantage of the second-ordinary differential equation, while which we called variational equation instead. The variational equation can be summarized as

$$(\ddot{\Phi}_r, \ddot{S}_r) = \frac{\partial \ddot{r}}{\partial r}(\Phi_r, S_r) + \begin{pmatrix} 0_{3 \times 6}, & \frac{\partial \ddot{r}}{\partial p} \end{pmatrix}, \quad (2)$$

with Φ and S indicated by

$$\Phi = \begin{pmatrix} \Phi_r \\ \Phi_v \end{pmatrix} = \begin{pmatrix} \frac{\partial r(t)}{\partial(r_0, \dot{r}_0)} \\ \frac{\partial \dot{r}(t)}{\partial(r_0, \dot{r}_0)} \end{pmatrix}, \quad S = \begin{pmatrix} S_r \\ S_v \end{pmatrix} = \begin{pmatrix} \frac{\partial r(t)}{\partial p} \\ \frac{\partial \dot{r}(t)}{\partial p} \end{pmatrix}, \quad (3)$$

where, Φ is the state transition matrix, and S is the parameter sensitivity matrix, and p indicates the gravity field-related parameters (such as gravity field coefficients, accelerometer scale/bias parameters and so on). As one can see, the computation burden of solving Eq. (2) is much heavier than that of Eq. (1), since the dimension of parameters p is much larger, for instance, number of sole gravity field coefficients is close to ~ 8100 up to degree and order 90. This indicates that it is more urgent to optimize the ODE solver in variation equation than in the orbit integration.

Then, Eq. (1) and Eq. (2) can be generalized as

$$\frac{d^2 y}{dt^2} = \ddot{y} = f(t, y, \dot{y}), \quad (4)$$

with the initial conditions ignored here for brevity. Here in Eq. (4), y could be indicating either r or (Φ, S) depending on the need. To address the classical ODE problem as depicted by Eq. (4), we recommend GJ as the numerical solver, see Sec. 2.3 for more details. Specifically, GJ consists of an iterated predictor step and corrector step, called as PECE-algorithm (Berry and Healy, 2004) that yields (i) the predictor step:

$$y_{n+1} = h^2 \left(S_{n+1} + \sum_{k=-N/2}^{N/2} a_{N/2+1,k} \ddot{y}_{n+k-N/2} \right), \quad (5)$$

$$\dot{y}_{n+1} = h \left(s_n + \frac{\ddot{r}_n}{2} + \sum_{k=-N/2}^{N/2} b_{N/2+1,k} \ddot{y}_{n+k-N/2} \right), \quad (6)$$

158 and (ii) the corrector step:

$$y_n = h^2 \left(S_n + \sum_{k=-N/2}^{N/2} a_{N/2,k} \ddot{y}_{n+k-N/2} \right), \quad (7)$$

$$\dot{y}_n = h \left(s_n + \sum_{k=-N/2}^{N/2} b_{N/2,k} \ddot{y}_{n+k-N/2} \right), \quad (8)$$

159 where N is the selected order (e.g., 8 for this study) of GJ, n indicates the step forward from the initial time, h is the step
 160 size (5 seconds in our case); a and b are the integrator coefficients, and S and s are intermediate variables, please see
 161 Berry and Healy (2004) for their definitions. It can be observed from Eq. (5-8) that the PECE process involves intensive
 162 computation of second-order derivative function \ddot{y} (either the force or the second derivative of sensitivity/transition
 163 matrix), and therefore they should be given special care if one desires to optimize the ODE. For a traditional ODE
 164 method, a complete implementation from Eq. (5) to Eq. (8) is executed for each individual satellite. Consequently, in
 165 case of GRACE-like missions that consist of two satellite, the ODE has to be performed twice. However, it is worth
 166 mentioning that, for two simultaneous satellites, their ODEs have much in common, especially for the calculation of
 167 second-order derivative function \ddot{y} . In this sense, it would be obviously a waste of resources to repeat computing these
 168 shared variables multiple times.

169 Therefore, we propose the multi-channel ODE method to address the above issue. It is known that the second
 170 derivative, $\ddot{y} = f(t, y, \dot{y})$, is dependent on three inputs, i.e., the time epoch t , the position r and the velocity \dot{r}
 171 (for brevity, but they may also indicate sensitivity/transition matrix). Apparently, the position/velocity is satellite-
 172 dependent, whereas the time is common for all satellites. This fact makes it possible to separate the time-related
 173 operation from $\ddot{y} = f(t, y, \dot{y})$ to be shared by all satellites. Specifically, the second order differential \ddot{y} (i.e., $f(t, y, \dot{y})$)
 174 could be conceptually re-organized as

$$f(t, y, \dot{y}) = F(t) \otimes F(y, \dot{y}), \quad (9)$$

175 where the notation of \otimes does not have a physical meaning but indicates the decoupling of $f(t, y, \dot{y})$ into two independent
 176 operations; $F(t)$ indicates the collective manipulations that are solely dependent on time, while $F(y, \dot{y})$ indicates those

relevant to the state vectors. Subsequently, multiple (e.g., n) satellites are assembled together as below

$$\begin{cases} f(t, y^1, \dot{y}^1) \\ f(t, y^2, \dot{y}^2) \\ \dots \\ f(t, y^n, \dot{y}^n) \end{cases} = F(t) \otimes \begin{cases} F(y^1, \dot{y}^1) \\ F(y^2, \dot{y}^2) \\ \dots \\ F(y^n, \dot{y}^n) \end{cases} = F(t) \otimes F\left(\begin{bmatrix} y^1, \dot{y}^1 \\ y^2, \dot{y}^2 \\ \dots \\ y^n, \dot{y}^n \end{bmatrix}\right). \quad (10)$$

Equation (10) establishes the basis of 'multi-channel ODE', where each channel indicates one satellite. There are two major differences between the proposed method and the traditional one: (1) only once calculation of $F(t)$ is required, (2) a matrix is used to assemble the multiple channels (indicated by the second term of Eq. (10), which as a whole serves as the input so that only once ODE will be performed. Be aware that multi-channel ODE is not parallelized, and only one CPU core shall be used like the traditional single satellite ODE, despite a larger running memory. Figure 2 conceptually illustrates the differences between two methods.

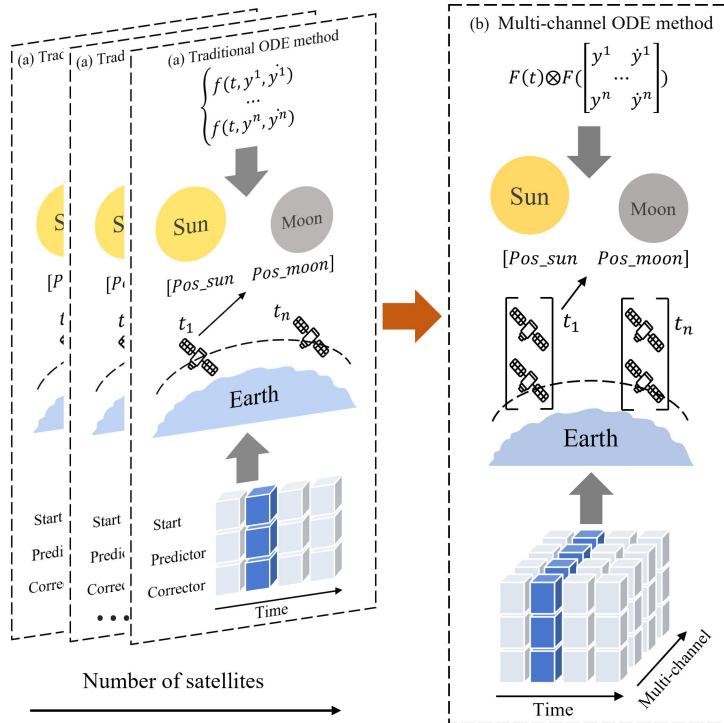


Figure 2: Schematic diagram of (a) traditional ODE and (b) multi-channel ODE method, in terms of GRACE-like (two satellites) mission. The time-relevant manipulation, e.g., acquisition of states of Sun and Moon, is calculated once for the new method.

In practice, the time-relevant $F(t)$ involves several heavy manipulations, including but not limited to: (1) time-coordinate transformation, (2) Earth orientation parameter acquisition, (3) deriving celestial position from planetary ephemerids (such as the Sun and Moon), (4) calculating spherical harmonic coefficients for ocean tide, polar tide,

solid Earth tide, atmospheric tide, non-tidal atmospheric and ocean mass change, etc. Regardless of the number of satellites, these time-relevant manipulations need to be computed only once per epoch and afterwards can be shared across all satellites. This avoids redundant calculations and reduces computational time. In our tests, these time-relevant manipulations account for roughly 20% to 50% of the total computation time, and consequently, the proposed method should be practically useful.

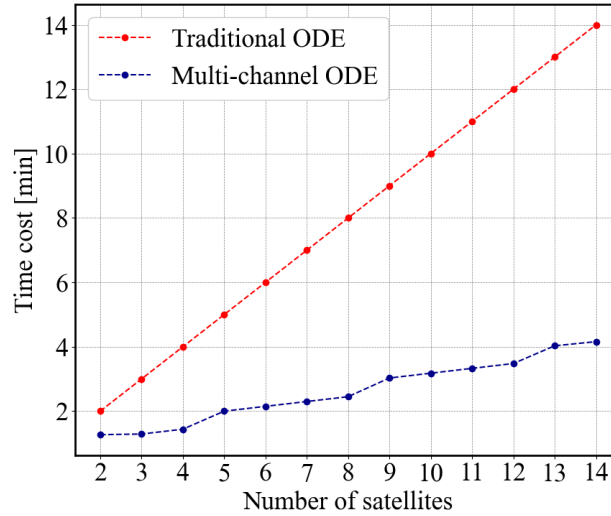


Figure 3: Computation time against the number of satellites: a comparison between traditional ODE and multi-channel ODE. Be aware that the time cost of traditional ODE method is proportional to the number of satellites because of the sequential implementation.

Another advantage of this method is the compatibility with future large-scale satellite mission, where the matrix of Eq. (10) is extensible regardless of how many pairs of GRACE-like satellites included. Moreover, theoretically, as the number of satellites increases, the advantages of this method relative to the traditional method should be more evident because of the nature of Eq. (10). To verify this, we designed an efficiency test to evaluate the performance. The experiment was conducted on a Windows system with an i7-10700K processor and 24 GB of RAM, using only one CPU core for the entire test. The ODE solver used was the GJ method of 8th order with a step size of 5 seconds, simulating an arc of 6 hours, where all the force models addressed in Sec. 2.2 have been used to be close to the reality. The computation time is recorded for each method against the number of satellites to simulate the scenario of present and future gravity mission, see Fig. 3. It has been found that our method can significantly reduce the time cost, compared to the traditional one. In the worst case like GRACE mission that has two satellite, our method improves the computational efficiency by about 36% (2 minutes versus 1.27 minutes). In a scenario of future mission (four satellites included), our method improves the computational efficiency by about 64% (4 minutes versus 1.44 minutes), which is significant enough to justify its value for future missions.

4. Performance evaluation

4.1. Benchmark test for background force models

High-precision background force models are essential for orbit determination, because LL-SST missions are not tolerable to the force modeling errors. Currently, several institutions have developed software for the modeling background forces, and these software packages vary in programming languages, modeling algorithms and processing workflows. In this context, Lasser et al. (2020) suggested a cross-software comparison based on the same benchmark data to test the precision of background force modeling. The benchmark data are generated from GROOPS toolbox (Mayer-Gürr et al., 2021) and the data quality has been fully confirmed to ensure a fair comparison, see Lasser et al. (2020) for more details.

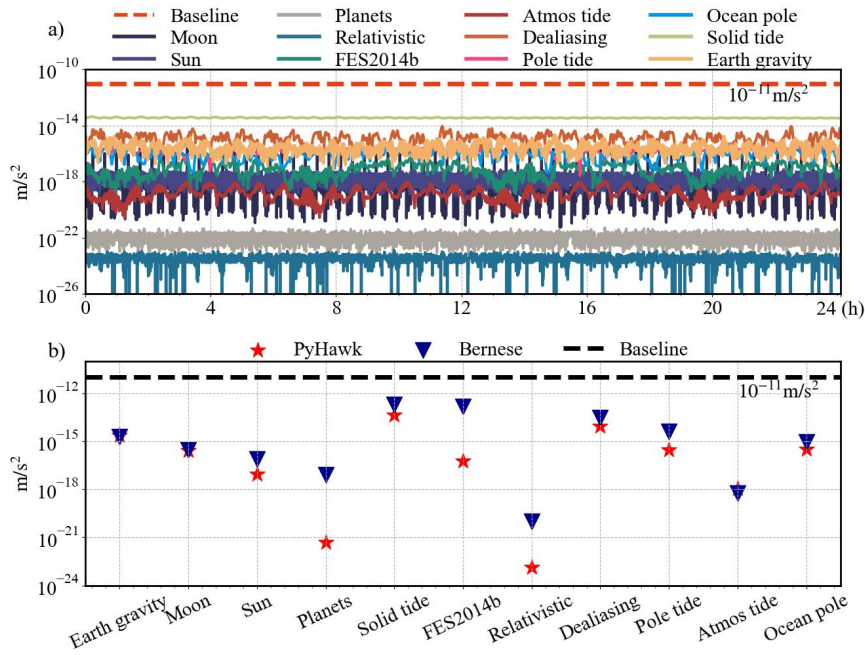


Figure 4: Benchmark test of PyHawk following Lasser et al. (2020): (a) PyHawk's force model deviation with the ground truth, (b) a comparison between PyHawk and Bernese in terms of the maximum deviation obtained from Fig. 4(a).

As suggested by Lasser et al. (2020), we take $1 \times 10^{-11} \text{ ms}^{-2}$ as the baseline accuracy, so that a force model with its error (deviation from the 'ground truth') below this threshold would be deemed as safe. Such threshold is chosen since it is at least one order of magnitude lower than the noise of the accelerometer instrument onboard of GRACE. Then, the acceleration of each force model at each epoch is computed in the inertial reference frame, and the norm of three-axis acceleration is compared to the benchmark data, see Fig. 4(a). One can see that the most significant error source of PyHawk comes from the solid Earth tide, whose overall magnitude is around $3.5 \times 10^{-14} \text{ ms}^{-2}$ that is much smaller than the baseline. The smallest error comes from the relativistic effects, which is down to $2 \times 10^{-24} \text{ ms}^{-2}$. Then, as many other reputed software have participated in the benchmark tests and published their results in

Lasser et al. (2020), we also take the chance to compare PyHawk and Bernese toolbox shown in Fig. 4(b). It is found that PyHawk's precision meets all the benchmark requirements, and some results even slightly outperform those of Bernese.

4.2. Prefit orbit/ranging residuals

The prefit orbit/ranging residuals, which are the differences between the nominal orbit/ranging values and the direct instrument measurements, serve as the pseudo observations. Here, the residuals rather than the measurements are used to reduce the non-linearity of the system and to enhance the numerical condition for the least-square solver. In this context, on one hand, the residuals are expected to be minimized, but on the other hand, the underlying physical signals should be well preserved. Clearly, this necessitates a high-precision computation of the nominal orbit/ranging, which in turn requires a high-quality calibration of the on-board accelerometers and the satellites' initial state vectors. In this experiment, we use PyHawk to perform the calibration, calculate the nominal orbit/ranging values, and obtain their residuals as indicators of the capability of PyHawk.

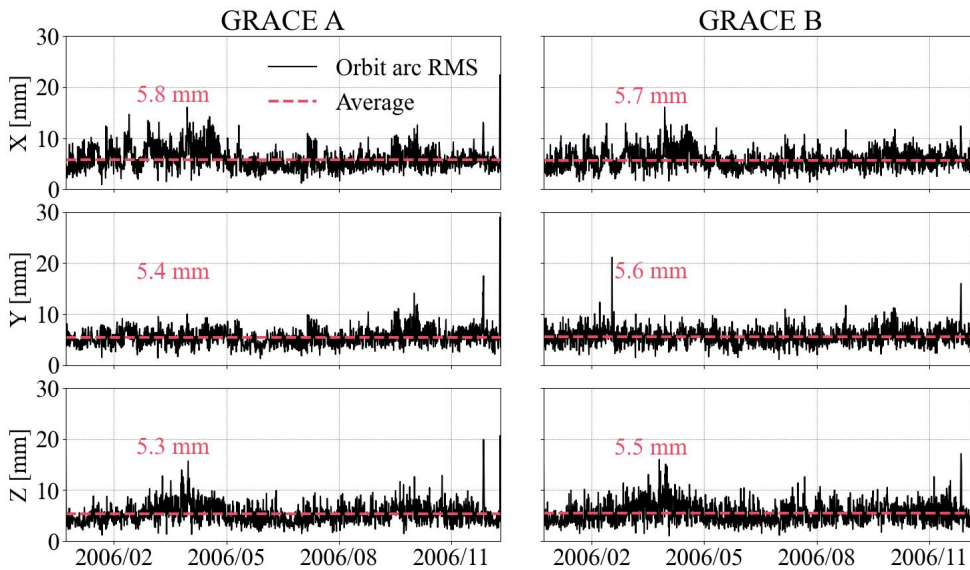


Figure 5: Arc-wise (per 6 hours) RMS (root mean square) of the three-axis orbit residuals (over year 2006) between the nominal (adjusted) orbit and the reduced dynamic orbit.

The orbit residuals of twin-satellites (-A, -B) of GRACE for one year (2006) are illustrated in Fig. 5 as example. It is found that the residuals are generally below the threshold of 20 mm, which is desirable since it is close to current precision of the GPS-based orbit determination (Montenbruck et al., 2005). Specifically, the mean RMS of the residuals of one year for both satellites and three axis are as small as < 6.0 mm, which is far beyond the accuracy of GPS.

Figure 6 illustrates a long record of inter-satellite along-track K-band ranging rate (KBRR) residuals, from which we

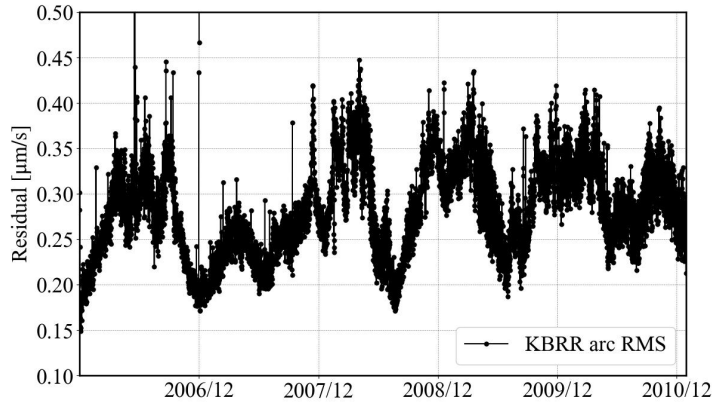


Figure 6: Arc-wise RMS of KBRR (K-band ranging rate) residuals of five years.

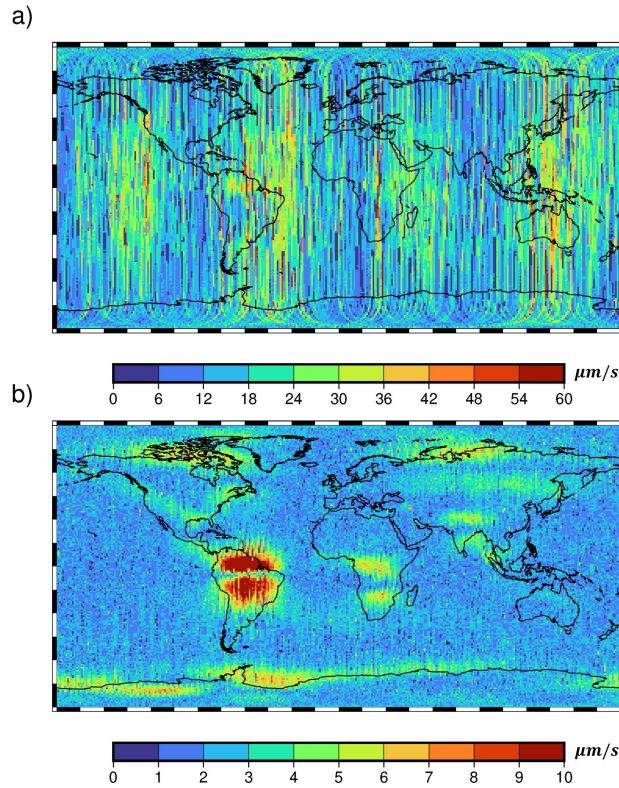


Figure 7: Spatial distribution of KBRR-residuals before (a) and after (b) calibration in May 2009.

also see a signal/noise level that stays within $0.15 \sim 0.45 \mu\text{m/s}$, which is slightly higher than the pre-launch expectation, i.e., $\sim 0.1 \mu\text{m/s}$ (Kim, 2000). However, the fluctuations of Fig. 6 are consistent with those of official products (Dahle et al., 2019), which indicate underlying physical signals. For verification, we project the range-rate residuals from time domain as Fig. 6 to the space domain in Fig. 7, following the ground track of the twin-satellites. It is evident that, before the calibration, the range-rate residuals exhibit severe and undesirable orbit-resonance noise, whereas the realistic TVG

signals appears after the calibration, such as the pronounced water change in Amazon. Combined with the previous experiment of orbit residuals, PyHawk has fulfilled the goal of eliminating noise while preserving geophysical signals well.

4.3. Inter-comparison of level-2 TVG products

Following the standard processing chain, we generate the level-2 monthly TVG products over period of 2006.01-2010.12, where GRACE has relatively stable performance. This enables a comparison against the latest official level-2 TVG products (available at <https://icgem.gfz-potsdam.de/home>). We select one-year data to simplify the visualization, see Fig. 8, where a common static gravity fields has been removed from the TVG products to highlight the discrepancy. From Fig. 8, PyHawk demonstrates nearly identical capability in preserving signals prior to degree-30 (excluding very low degrees that GRACE is not sensitive to), and this is exactly where TVG signals dominate. For example, the mean (across 12 months, from degree-2 to degree-30) correlation coefficients between PyHawk and CSR is 0.90. Despite the discrepancy increases after degree-30, we claim that these mainly reflect the noise content because of its dominant impact at shorter wavelengths. In this sense, PyHawk has achieved a comparable signal level and a moderate noise level, with respect to the official products.

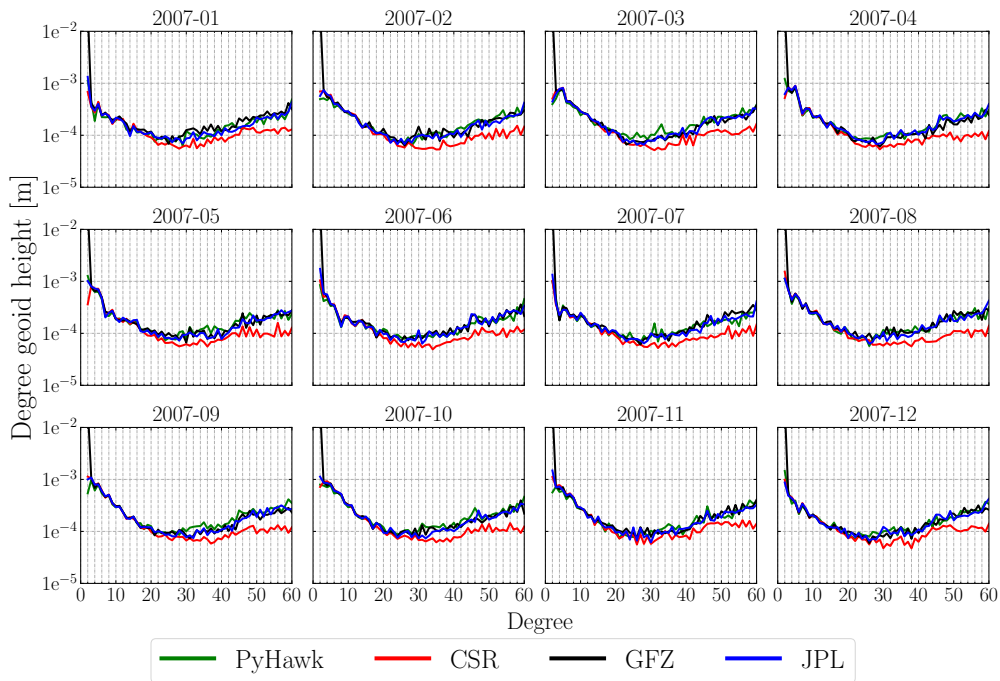


Figure 8: Inter-comparison of the official TVG products and PyHawk in terms of the geoid height up to degree/order 60. The latest (RL06) official products are respectively produced by CSR (Center for Space Research, University of Texas), GFZ (German Research Center for Geoscience, Potsdam), and JPL (Jet Propulsion Laboratory, NASA).

Since there is evident noise in the level-2 TVG products as illustrated by Fig. 8, it is common in practice to apply a

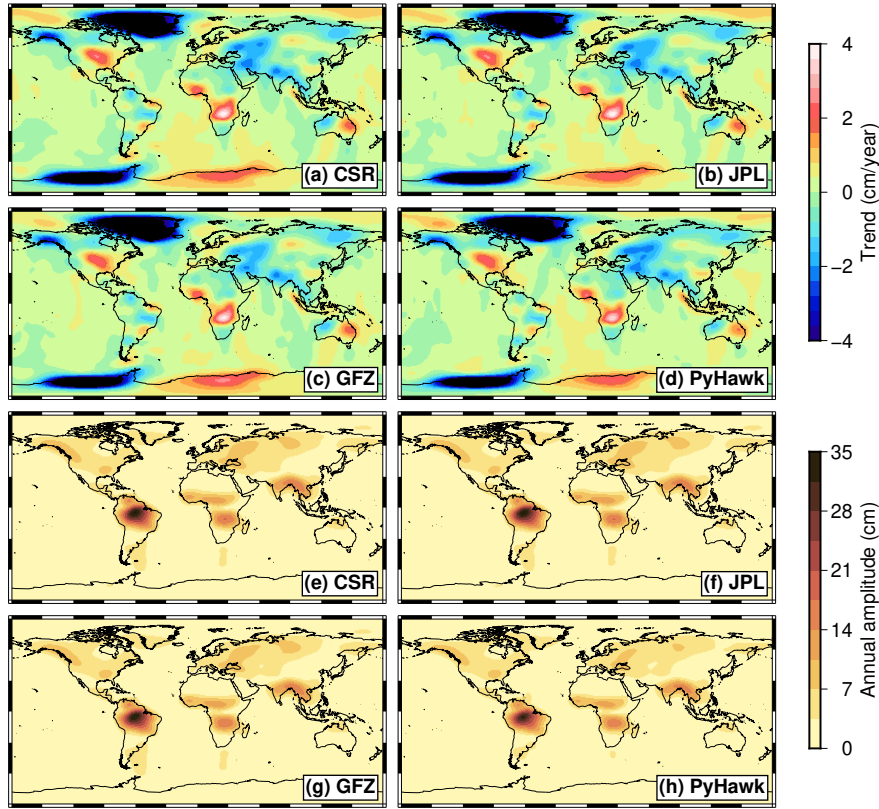


Figure 9: Inter-comparison of PyHawk and three official TVG products (2006.01-2010.12) in terms of trend and annual amplitude.

low-pass filtering to acquire higher signal-to-noise ratio (Kusche, 2007; Yi and Sneeuw, 2022; Yang et al., 2024b). At this experiment, we smooth the TVG with a popular Gaussian filtering with radius of 300 km, and project the filtered TVG from spectral domain to the EWH (Equivalent Water Height) in the spatial domain. Subsequently, we analyze the time-series (5-years from 2006-2010) and extract the secular trend and the annual amplitude as illustrated by Fig. 9. All these post-processing manipulations are realized through an open-source Python toolbox called SaGEA (Liu et al., 2025). It is found that both trend and annual amplitude of PyHawk exhibit fairly good agreement with the official products, particularly at regions such as glaciers (Greenland, Antarctica, and Alaska etc) and rain-rich river basins (Amazon, Ganges Delta etc), where geophysical signals are all plausible and similar. Moreover, the spatial correlation coefficients between PyHawk and other products are all found above 0.98, whether for the trend or annual amplitude, confirming that PyHawk product achieves a comparable quality.

5. Case study of a variant and non-standard product

Alternative to the standard processing chain, the PyHawk toolbox, due to its flexible and extensible structure, can also allow to derive several variants of the processing to satisfy specific research need. For example, the 'sliding' TVG

product, which is calculated with specific data length (window) of N days and sliding length of M days, has been the interest of the community (Sakumura et al., 2016). In Figure 10, we showcase that PyHawk is flexible to generate such sliding TVG product, e.g., with $N = 31$ days and $M = 1$ day. Several major river basins are selected to perform the experiment, where one can see that the daily sliding product reveals much more high-frequency dynamics than the standard products at all selected basins. Such sliding product is likely more beneficial to capture fast evolving events like flooding (Rateb et al., 2024). Moreover, PyHawk also supports flexible settings for inversion time scales, such as 7 days or 15 days. Some studies have already focused on these sub-monthly time-varying gravity fields (Bruinsma et al., 2010). Therefore, resolving TVG at multiple time scales and steps as illustrated by this case study can be an important supplement to standard product for making best of GRACE(-FO), and this is likely more relevant for future ll-sst missions that expects a much higher spatial-temporal resolution. These variants are optional, which constitute the flexibility of PyHawk.

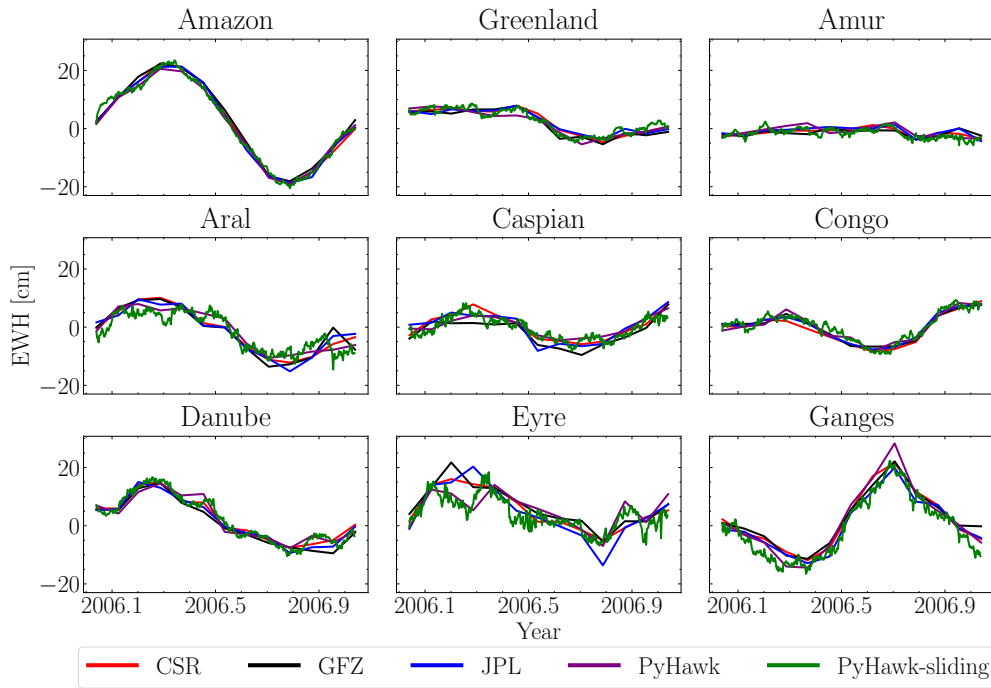


Figure 10: Basin-scale water changes revealed by regular monthly products and PyHawk-sliding product over one year.

6. Conclusions

With the significant development of ll-sst gravity missions over last two decades and the planned future gravity missions, there is an increasing demand for acquisition of accurate Level-2 gravity products. This study presents a Python-based and user-friendly toolbox, PyHawk, to address the complete and sophisticated processing chain from Level-1b raw data to the Level-2 scientific products, with the hope to advance ll-SST gravity missions from geodesy

community to potentially broader user groups.

In particular, to address the challenge of numerical efficiency for present and future II-sst missions that consist of multiple satellites, we propose a novel multi-channel ODE method to avoid repeated computation for shared variables across all satellites. Compared to the traditional method: it saves 36% of the computational time for present GRACE(-FO) mission, and the number improves to 64% for future mission (four satellites included). Furthermore, advanced code optimization helps to improve PyHawk's numerical efficiency as well. Experiment demonstrates that a complete processing for one-month GRACE(-FO) TVG recovery costs approximately 2 hours at a single-node cluster equipped with 56 CPUs and a memory of 240 GB, which should be comparable to the performance of Fortran in our practice.

In addition to the desired efficiency, PyHawk has confirmed to have an accuracy close to the other reputed software like GROOPS and Bernese in terms of the key module, i.e., force models. The eventual Earth's TVG products from PyHawk, are also found to be among the range of latest official products. Therefore, PyHawk can be an alternative but efficient and reliable II-sst gravity recovery solver for general users to handle present and even future II-sst gravity missions. Nevertheless, PyHawk still face challenges, especially when the data processing strategy is fast evolving these days, so that a timely update is anticipated to keep up with the most recent advancement of TVG modeling.

Code availability

GRACE level-1b data are the major input of PyHawk, and one can access the data via https://podaac.jpl.nasa.gov/dataset/GRACE_L1B_GRAV_JPL_RL03. The source code and the user manual of our software are publicly available via the Github (<https://github.com/NCSGgroup/PyHawk.git>). In addition to the code, we also provide sample data and all the necessary auxiliary data to support users for a stand-alone running of PyHawk at an open data repository, see <https://doi.org/10.5281/zenodo.14205243>.

Declarations

The authors have no competing interests to declare that are relevant to the content of this article

Acknowledgments

We acknowledge financial supports through the national Natural Science Foundation of China (Grant No. 42274112 and No. 41804016). This work is also supported by the Danmarks Frie Forskningsfond [10.46540/2035-00247B] through the DAnSk-LSM project and ESA SING project.

References

- Berry, M.M., Healy, L.M., 2004. Implementation of Gauss-Jackson integration for orbit propagation. *The Journal of the Astronautical Sciences* 52, 331–357. doi:10.1007/BF03546367.
- Bruinsma, S., Lemoine, J.M., Biancale, R., Valès, N., 2010. CNES/GRGS 10-day gravity field models (release 2) and their evaluation. *Advances in Space Research* 45, 587–601. doi:10.1016/j.asr.2009.10.012.
- Chen, J., Cazenave, A., Dahle, C., Llovel, W., Panet, I., Pfeffer, J., Moreira, L., 2022. Applications and challenges of GRACE and GRACE follow-on satellite gravimetry. *Surveys in Geophysics* 43, 305–345. doi:10.1007/s10712-021-09685-x.
- Chen, Q., Shen, Y., Kusche, J., Chen, W., Chen, T., Zhang, X., 2021. High-resolution GRACE monthly spherical harmonic solutions. *Journal of Geophysical Research: Solid Earth* 126, e2019JB018892. doi:10.1029/2019JB018892.
- Dahle, C., Murböck, M., Flechtner, F., Dobsław, H., Michalak, G., Neumayer, K.H., Abrykosov, O., Reinhold, A., König, R., Sulzbach, R., et al., 2019. The GFZ GRACE RL06 monthly gravity field time series: Processing details and quality assessment. *Remote Sensing* 11, 2116. doi:10.3390/rs11182116.
- Daras, I., March, G., Pail, R., Hughes, C., Braitenberg, C., Güntner, A., Eicker, A., Wouters, B., Heller-Kaikov, B., Pivetta, T., et al., 2024. Mass-change And Geosciences International Constellation (MAGIC) expected impact on science and applications. *Geophysical Journal International* 236, 1288–1308. doi:10.1093/gji/ggad472.
- Darbeheshti, N., Wöske, F., Weigelt, M., McCullough, C., Wu, H., 2018. GRACETOOLS—GRACE Gravity Field Recovery Tools. *Geosciences* 8, 350. doi:10.3390/geosciences8090350.
- Eicker, A., Forootan, E., Springer, A., Longuevergne, L., Kusche, J., 2016. Does GRACE see the terrestrial water cycle “intensifying”? *Journal of Geophysical Research: Atmospheres* 121, 733–745. doi:10.1002/2015JD023808.
- Eicker, A., Jensen, L., Wöhnke, V., Dobsław, H., Kvas, A., Mayer-Gürr, T., Dill, R., 2020. Daily GRACE satellite data evaluate short-term hydro-meteorological fluxes from global atmospheric reanalyses. *Scientific Reports* 10, 4504. doi:10.1038/s41598-020-61166-0.
- Flechtner, F., Neumayer, K.H., Dahle, C., Dobsław, H., Fagiolini, E., Raimondo, J.C., Güntner, A., 2016. What can be expected from the GRACE-FO laser ranging interferometer for earth science applications? *Remote sensing and water resources*, 263–280. doi:10.1007/s10712-015-9338-y.
- Ghobadi-Far, K., Han, S.C., McCullough, C.M., Wiese, D.N., Ray, R.D., Sauber, J., Shihora, L., Dobsław, H., 2022. Along-orbit analysis of GRACE follow-on inter-satellite laser ranging measurements for sub-monthly surface mass variations. *Journal of Geophysical Research: Solid Earth* 127, e2021JB022983. doi:10.1029/2021JB022983.
- Hohenkerk, C., 2017. IAU standards of fundamental astronomy (SOFA): Time and date, in: *The Science of Time 2016: Time in Astronomy & Society, Past, Present and Future*, Springer. pp. 159–163. doi:10.1007/978-3-319-59909-0_21.
- Kim, J., 2000. Simulation study of a low-low satellite-to-satellite tracking mission. *The University of Texas at Austin*.
- Kornfeld, R.P., Arnold, B.W., Gross, M.A., Dahya, N.T., Klipstein, W.M., Gath, P.F., Bettadpur, S., 2019. GRACE-FO: the gravity recovery and climate experiment follow-on mission. *Journal of spacecraft and rockets* 56, 931–951. doi:10.2514/1.A34326.
- Kosti, A., Anastassi, Z.A., Simos, T.E., 2012. An optimized explicit Runge–Kutta–Nyström method for the numerical solution of orbital and related periodical initial value problems. *Computer Physics Communications* 183, 470–479. doi:10.1016/j.cpc.2011.11.002.
- Kusche, J., 2007. Approximate decorrelation and non-isotropic smoothing of time-variable GRACE-type gravity field models. *Journal of Geodesy* 81, 733–749. doi:10.1007/s00190-007-0143-3.
- Kvas, A., Behzadpour, S., Ellmer, M., Klinger, B., Strasser, S., Zehentner, N., Mayer-Gürr, T., 2019. ITSG-Grace2018: Overview and evaluation of a new GRACE-only gravity field time series. *Journal of Geophysical Research: Solid Earth* 124, 9332–9344. doi:10.1029/2019JB017415.
- Landerer, F.W., Flechtner, F.M., Save, H., Webb, F.H., Bandikova, T., Bertiger, W.I., Bettadpur, S.V., Byun, S.H., Dahle, C., Dobsław, H., et al.,

2020. Extending the global mass change data record: GRACE Follow-On instrument and science data performance. *Geophysical Research Letters* 47, e2020GL088306. doi:10.1029/2020GL088306.
- Lasser, M., Meyer, U., Jäggi, A., Mayer-Gürr, T., Kvas, A., Neumayer, K.H., Dahle, C., Flechtner, F., Lemoine, J.M., Koch, I., et al., 2020. Benchmark data for verifying background model implementations in orbit and gravity field determination software. *Advances in Geosciences* 55, 1–11. doi:10.5194/adgeo-55-1-2020.
- Liu, S., Yang, F., Forootan, E., 2025. SAGEA: A toolbox for comprehensive error assessment of GRACE and GRACE-FO based mass changes. *Computers & Geosciences* 196, 105825. doi:10.1016/j.cageo.2024.105825.
- Mayer-Gürr, T., Behzadpour, S., Eicker, A., Ellmer, M., Koch, B., Krauss, S., Pock, C., Rieser, D., Strasser, S., Süsser-Rechberger, B., et al., 2021. GROOPS: A software toolkit for gravity field recovery and GNSS processing. *Computers & geosciences* 155, 104864. doi:10.1002/essoar.10505041.1.
- Meyer, U., Jäggi, A., Jean, Y., Beutler, G., 2016. AIUB-RL02: an improved time-series of monthly gravity fields from GRACE data. *Geophysical Journal International* 205, 1196–1207. doi:10.1093/gji/ggw081.
- Meyer, U., Jean, Y., Kvas, A., Dahle, C., Lemoine, J.M., Jäggi, A., 2019. Combination of GRACE monthly gravity fields on the normal equation level. *Journal of geodesy* 93, 1645–1658. doi:10.1007/s00190-019-01274-6.
- Montenbruck, O., Gill, E., Lütze, F., 2002. Satellite orbits: models, methods, and applications. *Appl. Mech. Rev.* 55, B27–B28. doi:10.1007/978-3-642-58351-3.
- Montenbruck, O., Van Helleputte, T., Kroes, R., Gill, E., 2005. Reduced dynamic orbit determination using GPS code and carrier measurements. *Aerospace Science and Technology* 9, 261–271. doi:10.1016/j.ast.2005.01.003.
- Pail, R., Yeh, H.C., Feng, W., Hauk, M., Purkhauer, A., Wang, C., Zhong, M., Shen, Y., Chen, Q., Luo, Z., et al., 2019. Next-generation gravity missions: sino-European numerical simulation comparison exercise. *Remote Sensing* 11, 2654. doi:10.3390/rs11222654.
- Park, R.S., Folkner, W.M., Williams, J.G., Boggs, D.H., 2021. The JPL planetary and lunar ephemerides DE440 and DE441. *The Astronomical Journal* 161, 105. doi:10.3847/1538-3881/abd414.
- Purkhauer, A.F., Pail, R., 2019. Next generation gravity missions: Near-real time gravity field retrieval strategy. *Geophysical Journal International* 217, 1314–1333. doi:10.1093/gji/ggz084.
- Rateb, A., Save, H., Sun, A.Y., Scanlon, B.R., 2024. Rapid mapping of global flood precursors and impacts using novel five-day grace solutions. *Scientific Reports* 14, 13841. doi:10.1038/s41598-024-64491-w.
- Rodell, M., Famiglietti, J.S., Wiese, D.N., Reager, J., Beaulieu, H.K., Landerer, F.W., Lo, M.H., 2018. Emerging trends in global freshwater availability. *Nature* 557, 651–659. doi:10.1038/s41586-018-0123-1.
- Sakumura, C., Bettadpur, S., Save, H., McCullough, C., 2016. High-frequency terrestrial water storage signal capture via a regularized sliding window mascon product from GRACE. *Journal of Geophysical Research: Solid Earth* 121, 4014–4030. doi:10.1002/2016JB012843.
- Save, H., Bettadpur, S., Tapley, B.D., 2012. Reducing errors in the GRACE gravity solutions using regularization. *Journal of Geodesy* 86, 695–711. doi:10.1007/s00190-012-0548-5.
- Scanlon, B.R., Zhang, Z., Save, H., Sun, A.Y., Müller Schmied, H., Van Beek, L.P., Wiese, D.N., Wada, Y., Long, D., Reedy, R.C., et al., 2018. Global models underestimate large decadal declining and rising water storage trends relative to GRACE satellite data. *Proceedings of the National Academy of Sciences* 115, E1080–E1089. doi:10.1073/pnas.1704665115.
- Shihora, L., Balidakis, K., Dill, R., Dahle, C., Ghobadi-Far, K., Bonin, J., Dobslaw, H., 2022. Non-tidal background modeling for satellite gravimetry based on operational ECWMF and ERA5 reanalysis data: AOD1B RL07. *Journal of Geophysical Research: Solid Earth* 127, e2022JB024360. doi:10.1029/2022JB024360.

- 390 Tapley, B.D., Watkins, M.M., Flechtner, F., Reigber, C., Bettadpur, S., Rodell, M., Sasgen, I., Famiglietti, J.S., Landerer, F.W., Chambers, D.P., et al.,
 391 2019. Contributions of GRACE to understanding climate change. *Nature climate change* 9, 358–369. doi:10.1038/s41558-019-0456-2.
- 392 Wahr, J., Swenson, S., Zlotnicki, V., Velicogna, I., 2004. Time-variable gravity from GRACE: First results. *Geophysical Research Letters* 31.
 393 doi:10.1029/2004GL019779.
- 394 Yang, F., Forootan, E., Liu, S., Schumacher, M., 2024a. A Monte Carlo Propagation of the Full Variance-Covariance of GRACE-Like Level-2
 395 Data With Applications in Hydrological Data Assimilation and Sea-Level Budget Studies. *Water Resources Research* 60, e2023WR036764.
 396 doi:10.1029/2023WR036764.
- 397 Yang, F., Forootan, E., Schumacher, M., Shum, C., Zhong, M., 2018. Evaluating non-tidal atmospheric products by measuring GRACE K-band
 398 range rate residuals. *Geophysical Journal International* 215, 1132–1147. doi:10.1093/gji/ggy340.
- 399 Yang, F., Forootan, E., Wang, C., Kusche, J., Luo, Z., 2021. A New 1-Hourly ERA5-Based Atmosphere De-Aliasing Product for GRACE, GRACE-
 400 FO, and Future Gravity Missions. *Journal of Geophysical Research: Solid Earth* 126, e2021JB021926. doi:10.1029/2021JB021926.
- 401 Yang, F., Kusche, J., Forootan, E., Rietbroek, R., 2017. Passive-ocean radial basis function approach to improve temporal gravity recovery from
 402 GRACE observations. *Journal of Geophysical Research: Solid Earth* 122, 6875–6892. doi:10.1002/2016JB013633.
- 403 Yang, F., Liang, L., Wang, C., Luo, Z., 2022. Attitude Determination for GRACE-FO: Reprocessing the Level-1A SC and IMU Data. *Remote
 404 Sensing* 14. doi:10.3390/rs14010126.
- 405 Yang, F., Liu, S., Forootan, E., 2024b. A spatial-varying non-isotropic Gaussian-based convolution filter for smoothing GRACE-like temporal
 406 gravity fields. *Journal of Geodesy* 98, 66. doi:10.1007/s00190-024-01875-w.
- 407 Yi, S., Sneeuw, N., 2022. A novel spatial filter to reduce north south striping noise in GRACE spherical harmonic coefficients. *Journal of Geodesy*
 408 96, 23. doi:10.1007/s00190-022-01614-z.
- 409 Zhao, Q., Guo, J., Hu, Z., Shi, C., Liu, J., Cai, H., Liu, X., 2011. GRACE gravity field modeling with an investigation on correlation between
 410 nuisance parameters and gravity field coefficients. *Advances in Space Research* 47, 1833–1850. doi:10.1016/j.asr.2010.11.041.