“GUTS Final Report”

20 July 2007

GOCE User Toolbox Specifications (GUTS)

DEVELOPMENT OF ALGORITHMS FOR THE GENERATION OF A GOCE USER TOOLBOX AND AN ABSOLUTE DYNAMIC TOPOGRAPHY PRODUCT

Ref:
ESA/XGCE-DTEX-EOPS-SW-04-0001 "GOCE User Toolbox Specifications (GUTS)"
Coordinator: Per Knudsen (DNSC)

Deliverable: WP2000: "User Toolbox Requirements Document"
Responsible partner: DNSC
Prepared by Per Knudsen

DOCUMENT CHANGE LOG				
Rev.	Date	Sections modified	Comments	Changed by
0	27 April 2007	All	Draft	Per Knudsen / Toke Andersson
1	20 July 2007	All	Draft submitted to ESA	Per Knudsen

List of Contents

1 Executive summary of the GOCE User Toolbox Study

Following the recommendations explicitly expressed in several conferences and workshops such as the GOCINA final workshop (Proceedings of the GOCINA workshop, 2006) and the 2nd International GOCE User Workshop (Proceedings of the GOCE workshop, 2004) ESA reacted positively and set out to develop a GOCE User Toolbox. Indeed, no ocean circulation products are planned to be delivered as level-2 products as part of the GOCE project so that a strong need exists, for oceanographers, to further process the GOCE level-2 geoid and merge it with Radar Altimetry as well as basic tools to support all GOCE mission products user including the Solid Earth scientists. As the specifications for such a Toolbox were far from being known, ESA decided to develop the Toolbox in two phases: 1) a user requirement consolidation and trade-off study of the toolbox functionalities leading to a toolbox specification and 2) the actual implementation of the consensus toolbox specifications. GUTS, the GOCE User Toolbox Specification study is phase 1. Phase 2 is called GUT, the GOCE User Toolbox implementation and is the subject of the follow-on contract. The GUTS Team was founded by co-optation of the principal European geodesy and oceanography specialists in an open group structured around a core team and a group of observers taking part of the brainstorming progress meetings and reviewing the produced documentation.

The objective of the GOCE User Toolbox Study was to develop – in close collaboration with ESA's HPF effort – algorithms and input and output specification for the subsequent generation of a user toolbox that is required by the general science community for the exploitation of GOCE level 2 products and altimetric data.

In order to achieve this goal, the study was divided into different work packages including the Review of user requirements (WP2000), the Toolbox Functionality and Algorithm Specification (WP3000) and the Toolbox System Specification and Architectural Design (WP4000). The main results and recommendations from these three work packages are synthesized hereafter.

- The goal of the “Review of user requirements” work package was to produce a user requirements and an input/output specification document, which will be a reference to identify the main users together with Products and Functionalities they require from a toolbox.

In a first step, a review of all functionalities that optimally would be included in the toolbox, mostly in a qualitative way, was done, covering three main aspects, related to their use in geodetic, oceanographic and the solid Earth applications.

In a second step, all functionalities that could possibly be incorporated in the toolbox were described quantitatively. A detailed input/output definition document and an Algorithm Specification document were compiled, providing a general overview of the input data, ancillary data and output data.

At this stage, the decision was made to use the full expressions for the wanted quantities instead of the spherical approximations used in the products obtained from the HPF since using the non-approximated algorithms will give a higher accuracy.

- The objective of WP3000 was to define and describe the “Toolbox Functionality and Algorithm Specification” This work package entailed different tasks described hereafter including a scientific trade-off study whose aim was to select the best (in terms of accuracy and computational demands) algorithms to compute the variables listed in the user requirement document.

First, the User requirements from WP2000 were reviewed and sorted by priority order. It was recommended that the first release of the GOCE User Toolbox should prioritise those products and functions, identified in the User Requirement Document, that relate to the oceanographic community. This should bring immediate benefit to a community currently unfamiliar with the use of gravity and geodetic products whilst providing basic functionality in support of all science areas. In particular, it was recommended that the toolbox should focus on the generation of dynamic topography fields, through merging satellite altimeter and GOCE gravity model data, the estimation of the absolute dynamic topography and therefore the absolute geostrophic currents from altimetry being a key objective of the GOCE mission.

Also, seven different classes were defined: “basic” “essential”, “highly desirable”, “desirable but low priority”, “Functionality requiring extensive computing resources or not yet at consensus”, “nice added features”, ”Functionality Desirable at Future Date”.

The main element that was included in the category: “Functionality requiring extensive computing resources” concerns the handling of the full error variance-covariance matrix of the spherical harmonic coefficients. Although it was recognized that this is an important step in the use of GOCE data, it was not recommend that propagation of the full errors, from the spherical harmonic error covariances, should be carried out within the toolbox. This is primarily due to the high computational and storage space demands this would place on the toolbox. Further investigation were recommended into the possibility of allowing such calculations to be carried out using remote, possibly GRID, computing facilities, using parameters determined from the toolbox.

Part of the “Functionalities not yet at consensus” is the optimal filtering functionality. At the moment, all filters considered in GUTS are suboptimal and further investigations into the subject of filtering a geoid and sea surface height or the resulting dynamic topography are sorely required before we can consider including such functionality into the toolbox.

Then, the functionalities going into the first three categories (and some of the fourth category) were organized into 7 workflows. One workflow, the main workflow, includes the core functionality of the toolbox in a default processing mode and default input data designed for fast access of the main output fields and use in tutorials. It is recommended that the input data used in the main workflow should be included in the toolbox distribution; so that the main workflow is the ready-to-use playground for novices.

The convenient use of the toolbox including the application of the user’s own data and applications deviating from the main workflow is supported by the remaining 6 sub-workflows organized in the following six modules:

1. Geoid and gravity field computation

The goal of this workflow is to provide computation of: geoid heights, gravity anomalies and deflections of the vertical.

2. Sea surface height and a-priori dynamic topography selection

The objective of this workflow is, first, to read data from the archive and to calculate the mean for a specified period (averaging routine) and, second, to compute the output field on a grid or at specific points as convenient for the subsequent computations in the toolbox (adaptation routine).

3. Satellite dynamic topography computation

The objective of this workflow is to compute a Satellite-only MDT (MDTS) by subtracting a geoid model from an altimetric mean sea surface and further filtering the obtained field. Computation in both geographical and spectral space is possible.

4. Combined dynamic topography (MDTC) computation

The objective of this workflow is to compute a combined MDT where the short scales are provided by an a-priori (GUT default or user-supplied) MDT through a remove-restore technique (further details in section 3.2). Two variants of a remove-restore combined technique are included in GUT to be used for different purposes.

5. Dynamic topography-derived quantities

The objective of this workflow is to compute geostrophic velocities from dynamic topography on a grid or list of locations (mean or time-dependent).

6. Pre-viewing function

Visualisation functions and graphical user interface were initially identified as important features of the GOCE User toolbox. However, in a second stage, it was decided that the GUT project should concentrate on the toolbox "core", e.g. the different functionalities and workflows. It was therefore decided to rely on the existing BRAT toolbox for further visualisation (potentially via the BRAT GUI) of the GUT outputs. Furthermore, the BRAT toolbox provides additional functionalities that have been recognized as significant for the completeness of the GOCE User Toolbox (more sophisticated altimeter data access and manipulation facilities: such as generation of mean sea surfaces or generation of time series of absolute dynamic topography by merging mean dynamic topography and sea level anomaly). It is recommended that GUT be able to input altimeter products generated within BRAT, and provide output that be used within BRAT to provided these extended facilities.

The basic elements in the previously described six workflows are the GUT functions as well as input and output fields. The GUT functions are the lowest processing level that can be accessed by the user when running the toolbox. The computations within a function might contain more than one algorithm as specified in the User Requirement Document. An overview of the functionality in GUT was therefore also given in WP3000, providing the input and output definition of the functions and the link to the algorithm specification in the User Requirement Document.

As part of WP3000, a trade-off study was realized for which it was decided to concentrate on investigating the efficiency and accuracy of filtering techniques as this is an essential part of the generation of dynamic topography fields due to the differences in resolved spatial scales between the altimetric and geodetic data. This study concluded that filtering data in spectral space, particularly in conjunction with a remove-restore technique, provided the best option for a current toolbox implementation.

Finally, a number of fields were identified (MSSH, MDTS, a-priori MDT, geoid model) for which it is recommended that a default file is included within the toolbox package and provide the default for appropriate workflows. For instance, two candidates were identified as potential default MSSH file, the KMS04 solution (Andersen et al, 2005) and the CLS01 solution (Hernandez and Schaeffer, 2001). As new fields may (will) be available by the time the GOCE User Toolbox is constructed and delivered, this list of potential candidates may be modified later.

In addition to these default fields, a strong requirement is that users can easily ingest their own data into the toolbox: Consequently, it is recommended that as a minimum requirement, the toolbox documentation must provide the format specifications of the included input data files in order that the user can supply their own data in the same format. In addition, the toolbox should supply the ability for users to write their own input modules for expansion of the toolbox to cover the widest possible range on input data formats.

- The objective of WP4000 (system specification and architectural design) was to provide a logical description of the toolbox architecture and data flow. The aim was not to include any software design details but rather to provide a link between scientists in the GUTS team and the software engineers who will implement the toolbox.

A system specification and architectural design document was produced that contains the definition of all system external interfaces and formats, the description of a logical model of system functions as well as a complete specification of output content: data products, metadata, reports and logs.

The proposed design is based on C and Fortran core scientific codes managed using the Python programming language, which would also provide the main user interface to GUT.

Python is not the only way to implement GUT, and it is possible that the important features of the design specified in the WP4000 report could be implemented in another way. However, if alternative solutions are evaluated during the implementation phase, the following elements need to be considered as basic requirements:

Capability to re-use existing Fortran programs
Capability to be easily extensible by adding new C and Fortran based functionality. This is to allow the flexible use and development of the toolbox by the science community
Multi-platform capability, including Windows
Good command interface with scripting ability
API for advanced customisation
Protective User Environment
Facilitates good science and traceability of results

All through the study, a number of existing sets of software routines and packages were collected – mainly Fortran routines- for making a first GUT prototype and for input to the subsequent GUT construction. This includes the Harmonic_synth package of Simon A. Holmes and Nikolaos K. Pavlis as well as the GRAVSOFT package (Tscherning et al, 1992, 1994).

In conclusion, this first phase, the GOCE User Toolbox Specification study, has successfully set up the basis for the further implementation of phase 2, the GOCE User Toolbox construction. In addition, it was strongly recommended that two activities should be led in parallel to the toolbox construction as part of the phase 2 project. First, further research is needed into exploiting the full error variance-covariance matrix. Second, it is highly recommended that the GOCE User Toolbox is made available to the users with an updated set of auxiliary data. More particularly, a specific study should be done in order to provide a new altimetric Mean Sea Surface taking advantage of all improved processing of altimetric data, mainly in coastal areas. This new Mean Sea Surface should be provided with an error estimate.

2 User Toolbox Requirements

2.1 Foreword

This document will try to give user requirement for a toolbox designed around the GOCE mission. Because of the high accuracy that GOCE will hopefully attain, the data from this mission will be useful to geodesists, oceanographers, and solid earth scientists. Because geodetic data and handling of this may be new to oceanographers and solid earth scientists, this toolbox proposes to make available a simple way of dealing with the data processing that is needed to use the data from GOCE for applications within the two fields.

For an oceanographer the following would be a typical use-case. The user has hydrographic data along a cruise track providing geostrophic currents across the track.
The user wishes to compare these current values with those obtained from altimeter derived geostrophic velocities for the same transect and the same time period and place in the context of the time series of surface currents in the area.

Another oceanographic application where GUT might be useful is where the user needs a gridded MDT interpolated or derived on a specific model grid. The user will then add an altimetric anomaly to the MDT to get a gridded sea level field for a particular time instant. This will then be used to assimilate into the model at that time.

For solid earth applications the main required functionalities are covered by general functionalities described above. However, on land the quantities may be needed on the surface of the earth. Hence, a few additional functionalities are required:
Computation of geoid heights, gravity anomalies or deflections of the vertical at a given, user-specified, degree and order of the spherical harmonic expansion (i.e., at a given spatial resolution) at the surface of the terrain.

These use cases will be used in this document to determine which functionalities need to be included in the final toolbox, and furthermore algorithm specifications will be given for the functionalities that are to be included. These functionalities will then be collected into workflows in a later section of this document. However not all the functionalities included in the use cases will go into the toolbox. For some of them toolboxes already exist or the scientists know how to handle these cases.

2.2 Background

2.2.1 Toolbox rational

Data from the ESA GOCE mission are of fundamental importance to the oceanographic community. It is expected that in conjunction with altimetric observations, gravity data from the ESA GOCE Mission will - for the first time in history - allow access to the absolute ocean dynamic surface topography and to compute the absolute ocean surface geostrophic currents at spatial scales down to about 100 km. At the moment, only the variable part of the sea level, and thus the geostrophic currents, can be inferred from altimetric heights with sufficient accuracy.

Despite their importance for oceanographic studies, the processing and analysis of gravity mission data has proven to be complicated to the point that the lack of proper processing software is hampering progress in the use of those data. Success in the exploitation of GRACE data therefore seems to depend fundamentally on the proper knowledge of several steps of the detailed gravity data processing procedure in terms of spherical harmonics, their implicit consistent normalization factors, filtering and error data, among others. To facilitate the easy use of GOCE products for oceanographers and to support the needs of specific applications, the development of a user toolbox (GOCE user toolbox study = GUTS) was identified as an urgent step at the Second International GOCE User Workshop. Such a toolbox is required to guarantee optimal use of the existing and future gravity data acquired from GRACE and GOCE. In particular it is recognized and accepted that software packages are required that allow the gravity field data, in conjunction and consistent with any other auxiliary data sets, to be pre-processed by users not intimately familiar with gravity field processing procedure, for oceanographic and hydrologic application, regionally and globally.

From previous work, a preliminary idea about the scope of a GUTS toolbox exists already in the community, especially from experience gained through GRACE data processing. Any new effort should build directly on this insight and should expand on user needs in a flexible way. The now finished GOCINA project is another source of valuable information about needs and solutions, which will be consulted while defining GOCE user needs and toolboxes.

2.2.2 Applicable and Reference Documents

The applicable documents are:

- Statement of Work

Reference documents are:

- Algorithm specification standard/guideline (ENVISAT document)
- ENVISAT Product Handbook
- GOCE: Product specification for L2 Products and Auxiliary Data Products (GO-ID-HPF-GS-0041)
- GOCE: Product Data Handbook (GO-MA-HPF-GS-0110)
- GOCE: Standards (GO-TN-HPF-GS-0111)

2.2.3 Previous Experience from The GOCINA Project

The EU FP5 GOCINA project was undertaking a study to determine regional high accuracy gravimetric geoid and mean sea surface height fields over the Northeast Atlantic and Nordic Seas, bounded by 53ºN, 73ºN, 40ºW and 20ºE. Among the project goals was to determine a best possible mean dynamic sea surface topography and to provide specific recommendations for quality assessment of GOCE data. These recommendations will be based on the generation and dissemination of the three regional fields for mean sea surface, geoid and mean dynamic topography and their error characteristics. The input to these three fields are, respectively, satellite altimetry, high resolution marine and airborne gravimeter data blended with GRACE long wavelength data, and output from numerical ocean models.

During the last phase of the GOCINA project these three fields and their error characteristics were evaluated and assessed. In addition, the impact of the new regional geoid for ocean circulation studies were investigated using direct current observations from an array of bottom mounted ADCP moorings between Scotland and Greenland. In doing so, the processing and analyses scheme (Figure 1) first suggested in ESA SP –1233 (1) will allow us to examine and quantify how the new GOCINA geoid, and its error covariance, combined with radar altimetry enable more reliable estimation of absolute ocean topography.

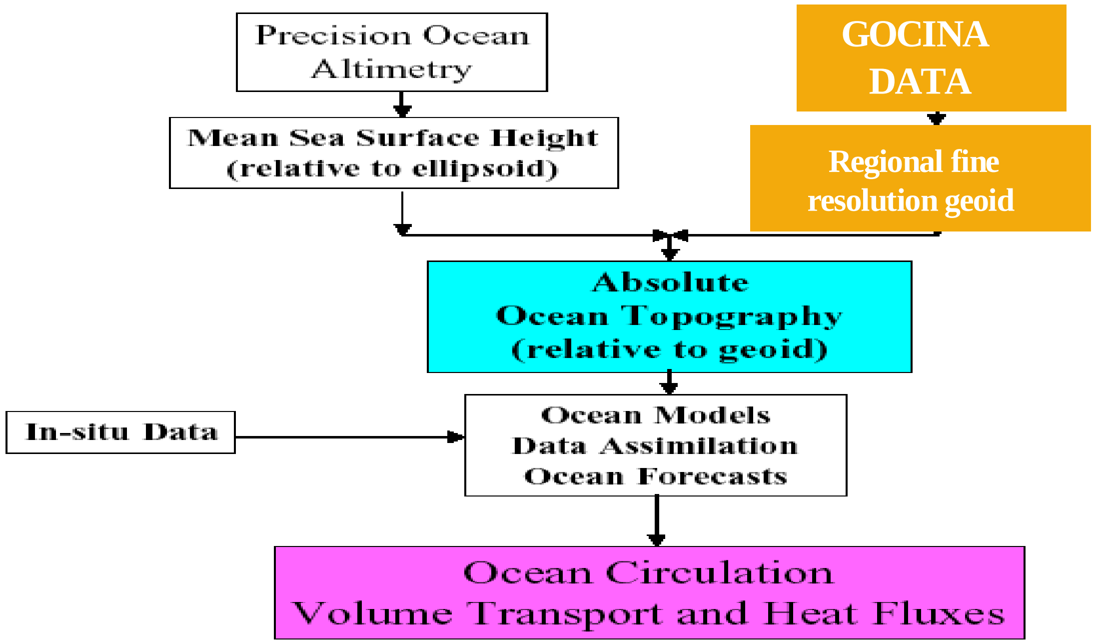


Figure 1: Schematic of the improvement in absolute ocean circulation studies from combination of the GOCE-produced marine geoid with precision altimetry (ESA-SP 1233 (1)).

Building on experience from this approach the GOCINA project delivered results that will be essential in the preparation for the GOCE mission, in the context of:

- Generation of best possible time-invariant regional gravity field and geoid model for the Northeast Atlantic and Nordic Seas that may be used for regional validation of GOCE mission products.
- Verification of the schematic analysis approach for integrated use of regional geoid and altimetry to compute the absolute dynamic topography and other relevant products such as volume and mass transport.
- Develop specific recommendations for integrating GOCE data in determination of geoid and mean dynamic topography models.

An important outcome of the GOCINA project has been that it is essential to describe proper (a-priori) statistical characteristics associated with the Mean Dynamic Topography. Those characteristics include a-priori variance and covariance functions, which in turn will be linked to statistical characteristics of the geostrophic surface currents through their associated a-priori power spectra. Those values should agree with the empirically derived values too, as the statistical characteristics of the gravity field should agree with its empirically derived values. Only by combining that information can consistent reliable error covariances of the Mean Dynamic Topography be expected.

2.2.4 Relation to the HPF Effort

As part of the GOCE Ground Segment, ESA is running a GOCE High-level Processing Facility (HPF) which is responsible for the generation of the final Level 2 products, and the generation of Quick-Look and External Calibration products. The final Level 2 products consist of: the precise orbit of GOCE, the gravity field in terms of spherical harmonic coefficients, the corrected, calibrated and geo-located gravity gradients, a map of the gravity anomalies and a map of the geoid. This effort will be an essential element in the entire GOCE data stream. However, no ocean circulation products are planned to be delivered as Level-2 products as part of the HPF project so that a strong need exists for GUTS to deliver oceanographers with additional information and tools that can turn the HFP products into application-dependent fields by further processing the GOCE Level-2 geoid and merging it with Radar Altimetry and other auxiliary data. Among the fields that are important for oceanographers is the geoid error covariance function. This field will be provided by HPF for spherical harmonic coefficients. For many oceanographic applications it is required in geographic coordinates or at specific locations. The toolbox therefore needs to perform this and other extra functions that are essential for using the GOCE geoid fields efficiently and for testing it in geoid validation studied.

To reach a maximum benefit from GUTS, the effort will work in close collaboration with the HFP project. This will be facilitated through joint memberships. In return, the HPF effort will be one of the first users interested in using GUT for their own geoid validation efforts. A mutual feedback and benefit and an optimum toolbox design is thereby almost automatic.

The GOCE HPF is lead by TUM. TUM participates in the GUTS consortium. Also UCPH participates in both the GOCE HPF and in GUTS. The main contact to the GOCE HPF will go through TUM who will participate actively at any stage of the work and ensure that knowledge about GOCE Level-2 products and specifications are available and utilized in GUTS and that information about the developments in GUTS are available to the HPF. To assure further consistency between GUTS and other GOCE related developments the participation of IFREMER, POL, NERSC, TUM, and UCPH as members of the GOCE Mission Advisory Group is important.

2.2.5 Study outline

In order to fulfil the GUTS objectives, a toolbox effort was proposed which aims at consolidating the user requirements, the algorithms needed to produce the required outputs, both globally and for selected areas, and system design of a toolbox. Although the development of a user toolbox was clearly recommended, the detailed community requirements for a toolbox need to be specified through a first action of a GUTS effort. In particular, all added-value products that could potentially be output from such a tool need to be better defined.

In more detail, the aim of a GOCE gravity field toolbox is to facilitate and ease the use, viewing and post-processing of GOCE mission and useful auxiliary data products for optimal use at higher levels than Level-2 products. Such a tool would be a basis for the computation and validation of Level-3 products for oceanographers and, indirectly, for the validation of Level-2 products. One functionality of such a tool should be to provide geoid heights at any point of the ocean as well as geoid height covariances and error covariances for any pair of points on the surface of the Earth.

This user requirements and an input/output specification document (GOCE User Requirements Report), which will be a reference to identify the main users together with Products and Functionalities they require from a toolbox. The User Requirement document will modify the list above and/or complete it with additional requirements so as to contain all what the GOCE User Toolbox shall hold to help the users in fully exploiting the GOCE level 2 data.

No ocean circulation products are planned to be delivered as level-2 products as part of the GOCE project so that a strong need exists, for oceanographers, to further process the GOCE level-2 geoid and merge it with Radar Altimetry. This need, and subsequent recommendations for optimal use of GOCE data by oceanographers, was explicitly expressed during the second International GOCE Workshop, which was held in ESRIN from 8-10 March 2004. A synthesis of the main recommendations has been published in the GOCE workshop proceedings. The primary requirement of oceanographers is to have access to a geoid and its error covariance at the highest spatial resolution and accuracy possible, although required resolution depends on application. These two requirements are consistent with the conclusions reached at last ISSI workshop on Earth Gravity Field from Space (March 2002, Bern) where the need for two gridded geoid Level-2 products was identified. For effective use of the geoid data, knowledge of the error covariance is mandatory so that the two previous geoid products have to be delivered with their corresponding covariance matrices. The possibility of having such a matrix delivered by the HPF as Level 2 products with standard resolution is planned in spherical harmonic space. However, no geoid or error covariance will be delivered in geographic space. Moreover, resolution depends on the application, as does the required geographic region of interest. It is therefore necessary to provide the user community with a general tool that takes HPF Level 2 information, merges it with other auxiliary data and provides users with the possibility to obtain required fields with self-specified resolution.

2.3 *Review of needs for existing and planned applications*

2.3.1 Input information

The toolbox will have several primary input fields. Most importantly, GOCE Level 2 products are required as starting point. This will be in form of spherical harmonic (SH) coefficients of the Earth Gravity field and geoid height at maximum resolution, and the error variance and covariance matrices of those SH coefficients. In addition, auxiliary (meta) data are required for geoid fields including the following information:

- Whether the mean (zero frequency) tide was removed and the spherical harmonic expansion used.
- Which tidal models were used, including statements about which tidal frequencies are included.
- The atmospheric correction used, including a statement of whether atmospheric tides are included (separated time mean and time varying atmosphere).
- Specification of any ocean corrections that were made, including a specification of any model that was involved.
- Specification of Love number corrections made, if any.
- Any polar motion correction, including, but not limited to the Chandler period.

Compliant with the requirements stated by Jakob Flurry and Reiner Rummel in their article on *Future satellite gravimetry and Earth Dynamics*, other important auxiliary data sets are required, some of which should be included as auxiliary data in the tool box. Others, not included, should be provided through links to data servers. Among those the most important ones are:

- Altimetric data sets:
    - along-track data sets.
    - altimetric Mean Sea Surfaces Height (MSSH) field(s). Most up to date versions of those products should be included as well as the option to exchange them with newer versions once they become available.
    - Also included should be error information about the above time mean and time-varying altimetric measurements.
- In-situ data through links to Argo, tide gauge and hydrographic datasets.
- Wind field and wind stress measurements, e.g., from scatterometers or NWP centres.
- In-situ and modelled MDT. The latter would include best possible fields available from several ocean models that were constrained by ocean data and should include information about their uncertainties as well.
- In-situ gravimetric data and in-situ local geoid estimations

Some altimetric fields should be included in the tool box for instant use, but a requirement is also that the users can add their own fields easily, as long as they are in the same format as that used for the included fields. This will allow the users to update the tool box with their own data and other readily available fields.

In addition to the data sets or links to them, tutorials in how to use the toolbox should provide methods of including these auxiliary datasets. In addition – links to other appropriate data service, such as the IAG's ACGEM should be specified, where a list of global earth models (http://icgem.gzf-potsdam.de/ICGEM/ICGEM.html) is being maintained.

2.3.2 Toolbox Functionality

Toolbox functionalities will be described qualitatively here and quantitatively later. As specified later, main functionalities of the toolbox would be to select data, make global, regional and local approximations of the gravity and geoid fields over the ocean, translate and adapt the spherical harmonic information into gridded information for any location in world, and to translate the errors associated with the scales of interest. The main aim, however, is to produce a geoid and geoid slopes that can be used for oceanographic purposes of ocean modelling, assimilation, and current estimation. This will require solutions valid over whole ocean basins starting from level 2 global solutions. Moreover, errors should be delivered in a form that they can be used for oceanography. As an example, nearly all the GRACE geoid products have correlated errors along the orbits in the meridional direction. These usually get removed by smoothing the entire geoid. An error description should capture the anisotropy in the errors and should allow the data to be combined with an a priori solution to allow the final solution to be isotropic and not display the banding.

Accordingly, toolbox requirements can be split into three main aspects, related to their use in geodetic, oceanographic and the solid Earth applications.

Geoid and gravity fields

In a variety of geophysical studies information associated with the Earth's gravity are used in form of geoid heights, gravity anomalies or deflections of the vertical. Those quantities may be represented in the nodes along a profile, in a grid or in discretely located points. Associated with such quantities error covariance information may be needed. Hence, the following functionalities are required:

1. Computation of global, gridded geoid heights or gravity anomalies at a given, user-specified, degree and order of the spherical harmonic expansion (i.e., at a given spatial resolution. The maximum resolution/degree is defined by the final GOCE products). User input would be the spatial resolution or the grid and the maximum degree order of the expansion. If several different geoid fields are available, input would also be the choice of geoid model or gravity field, if necessary). Output would be a gridded geoid. The gridded field should be available as both a single point representation and as area averages.
2. Computation of geoid heights at a given spatial resolution (as point-wise or area averages) and a given point or list of points (e.g. unstructured grid, transect). Input would be a list of user-specified geographic position parameters and the maximum degree order of the expansion. If several different geoid fields are available, input would also be the choice of geoid model.
3. Option to replace geoid heights by geoid slopes (deflection from the vertical) for either the gridded field or the unstructured positions.
4. Computation of geoid heights covariance for any pair of points on the sphere or the computation of a full covariance matrix for a given maximum degree and order of the spherical harmonic expansion.
5. Computation of geoid cumulative height errors and error covariances at a given spatial resolution on a global regular grid or for a list of points.
6. Covariance error matrix within chosen degree/order range for commission and omission error. The computation of omission errors maybe based on the original gravity gradients.
7. Option to include the omission errors for the GOCE gravity field.

8. Regional geoid solutions starting from the global products. This is not ideal for regional geoid implementation. At a later stage, Level 1 data might be required as the basis for an extended GOCE geoid, Concerning regional geoids, a combination of model solutions using both GOCE and in situ gravity data (see Haines et al., 2003, space science reviews, 108, p 205-217) could be useful and including such regional models for the European Seas would be a useful element of GUTS.

Additional functionalities would be required for geoid validation, to:

- Handle appropriate ancillary data, e.g., external MDT from in-situ/modelled, as well as local geoids, combined CHAMP/GRACE/GOCE geoids, among others.
- Compute differences / Root Mean Square differences / correlation coefficient / regression slopes between GOCE geoid and external geoids / 'GOCE' MDT and external MDT / between absolute altimetric dynamic topography and in-situ absolute dynamic topography / between absolute altimetric geostrophic velocities and in situ geostrophic velocities.
- Determine the parameters in a priori degree variance model for the modelling of the gravity field a priori spectrum (global and/or regional models).
- Derive a degree variance model for the MDT and determine statistical properties of the MDT and its associated geostrophic surface currents.
- Derive an optimal filter for the low pass filtering of the altimetric MSSH and/or the MDT derived from the altimetric MSSH and the GOCE geoid.

Dynamic Surface Topography

The primary oceanography variable of interest to be provided by the tool box is the dynamic topography resulting from the difference between altimetric measurements and the geoid model. Altimetric MSSH fields or time-varying SSH fields would be auxiliary input data set fields from which a consistently filtered mean dynamic topography (and associated mean geostrophic circulation) need to be computed by the toolbox. This requires that a consistent reference system be chosen for both the geoid and the (M)SSH (both surfaces expressed relative to the same reference ellipsoid) as well as a consistent permanent tide system (tide mean, zero tide or tide free system) or that supporting tools exist with which to move from one system to another. In this context, it would be useful to compute the magnitude of the aliasing parts of the MDT that are not resolved by the model/grid spacing. The three different tide systems will be briefly explained here. The mean tide contains the mean of all the different tide constituents (the permanent tide) in addition to the indirect effects on the Earth due to the elasticity of the earth. The zero tide system has the indirect effects but the permanent direct tide effects are removed. Finally the tide free system has neither indirect nor direct tide effects. Ideally the same tidal models or other corrections should have been used in the geoid and the altimetric fields. But as a requirement this would be too stringent. Instead a frequent update of both input field would be more useful. In addition enough information is required about the geophysical and environmental corrections applied to the altimetric data, such as atmospheric tide correction.

The possibility of defining the geoid in alternative "over the ocean-only" basis functions may allow the user to avoid problems with continental gaps that arise when using global spherical harmonics. The development of alternative basis functions is still a research topic and in that sense out of the scope of the toolbox. Nevertheless, it is a user requirement that eventually needs to be approached: Because the ocean is a not simply connected domain, the use of global basis functions is therefore not trivial or desirable. To avoid problems with continental gaps

(Gibbs effects) a representation of the geoid and MSSH in terms of ocean basis functions is desirable. The possibility of including ocean basis functions into the toolbox software would therefore be a worthwhile effort. Such functionality would be an alternative to sophisticated filter programs designed to cope with Gibbs effects at continental boundaries that arise when using global basic functions (spherical harmonics) to develop and smooth MDT fields. Concerning regional geoids, a combination of model solutions using both GOCE and in situ gravity data (see Haines et al., 2003, space science reviews, 108, p 205-217) could be useful and including such regional models for the European Seas would be a useful element of GUTS. Alternatively, it may be possible to work in spherical harmonic space and do MSSH-Geoid residuals in spherical harmonics. The field of MSSH-MDT which exists over the ocean only has a natural continuation over land as the Geoid itself. The use of a priori estimates of both MSSH and MDT could lead to a complete global a priori field of MSSH-MDT over the ocean which matches to a geoid over land. The goal then is to solve for corrections to this global field. It would be assumed that this correction over the ocean represents a correction to the a priori MDT rather than to the MSSH. This approach could avoid problems near the coasts.

Functionalities related to the dynamic sea surface topography aspects of the tool box would include:

- Interpolation of external MSSH on any regular grid or at given points (unstructured grid): This functionality should include procedures to translate and adapt the spherical harmonic information into geoid height information for any location in world (many of the oceanographic applications require geoid information on a global or regional geographical – often non-structured – grid), and to translate the errors associated with the scales of interest.
- Spatial filtering of MSSH consistent with a specific harmonic geoid height field expansion.
- Change of reference system for the geoid and/or MSSH (reference ellipsoid, tide system)
- Computation of a 'GOCE' MDT (MSSH-GOCE geoid) at a given spatial resolution and on a given structured or unstructured grid as the difference between the MSSH and the geoid.
- Computation of altimetric time-varying absolute dynamic topography as the difference between altimetric fields and a geoid model.
- Provision of a priori MSSH, MDT and Geoid data on a grid
- Provision of tools to produce a global description of a combination of these a priori gridded fields in spherical harmonics
- Computation of altimetric absolute geostrophic velocities from the spatial gradients of the geoid field.
- Covariance error matrix within chosen degree/order range for commission and omission error for mean dynamic surface topographies. This needs to include the auxiliary altimetric error covariances on top of the geoid model error covariance.
- Option to include the omission errors for the MDT and the associated geostrophic surface currents.

In addition, the computation of gridded time-mean geostrophic velocities with arbitrary resolution should be possible. Those computations should be possible from either pure data products (MDT) or from assimilation results. The latter results should be provided as auxiliary input from assimilation models.

Solid Earth Applications

In studies of the solid Earth the relations between different physical properties of different geological materials are utilized together with a series of observations that reflect those different physical properties. The relevant property in this perspective is the density of geological materials which may vary between significantly from materials such as rock salt, sediments, and bedrock. Hence, geological structures may be associated with changes in the density which are mirrored into the gravity field. By mapping the gravity field on the surface of the Earth information about the interior of the Earth is obtained. Low gravity reflects light materials and high gravity reflects heavy materials in the subsurface.

The commonly used quantity in solid earth studies is gravity anomalies and associated error information. Hence, the main required functionalities are covered by general functionalities described above. However, on land the quantities may be needed on the surface of the earth. Hence, a few additional functionalities are required:

- Computation of geoid heights, gravity anomalies or deflections of the vertical at a given, user-specified, degree and order of the spherical harmonic expansion (i.e., at a given spatial resolution) at the surface of the terrain. Output would be a grid or at single point representation and as area averages.
- Computation of geoid cumulative errors and error covariances associated with the above at a given spatial resolution on a global regular grid or for a list of points.
- Covariance error matrix within chosen degree/order range for commission and with the option of including the omission errors for the GOCE gravity field.

Pre-viewing Function

The toolbox functionalities should include

- Pre-viewing capabilities for results from the toolbox. This is not to be a full-fledged graphics representation of the data, but should rather be a way of easily checking the veracity of the processed data immediately after the processing is done.
- The previewing function should include the possibility of adding a latitude-longitude grid to the previewed data and should present rendered graphics.
- Data access and the possibility to transfer appropriate data from remote location to local machine.

Comments on the functionality

The previous sections have talked at some length about the functionalities that optimally would be included in the toolbox, mostly in a qualitative way. Some of the functionalities are essential, some are highly desirable, and others less so.

The following section will try to give a quantitative description of the functionalities that will possibly be incorporated in the toolbox, a first thinning of the functionalities.
There are a few exceptions here, functionalities that will need to be included but that will not be described in the following section. The ability for the user to provide external MSSH, geoid, MDT and other fields is vital to the toolbox, but this functionality will not be described in the following except for the most general handling of the fields. Additionally some fields will be

included in the toolbox, however these will be described later in the document together with the workflows that describe exactly how the functionalities described in the following section will be implemented. Note that as mentioned not all the functionalities required for the use-cases from the foreword will be implemented. Some of these will have to come from other sources.

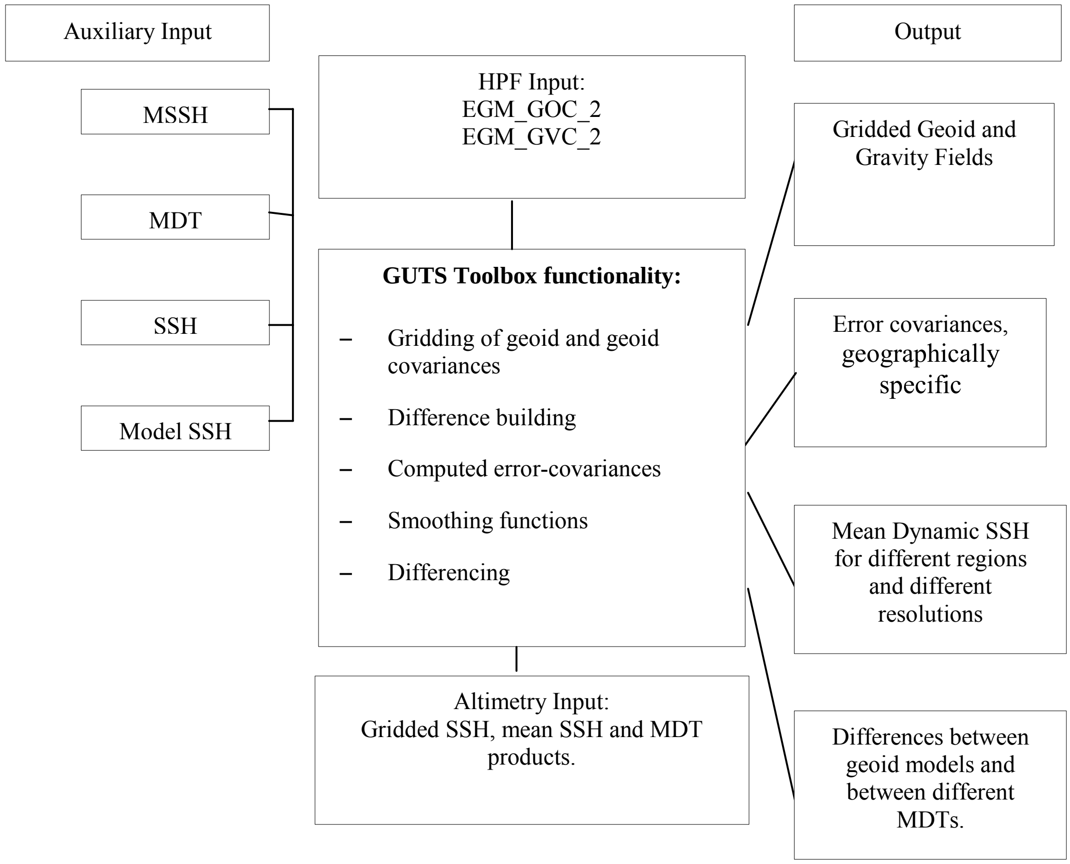


Figure 2 Schematic of a preliminary input/output lists and toolbox functionalities.

2.4 Required functionality, input and output parameters

2.4.1 Introduction

The task here is to compile an input/output definition document and an Algorithm Specification document. This will provide the source algorithms for providing the range of user requirements. The application of these algorithms will be defined later in this document, together with the specification of the toolbox functionality. The input and output for each algorithm is given in terms of the required parameters. The source file and output format definitions will also be supplied later.

The following provides a general overview of the input data, ancillary data and output data:

Input data

- GOCE Level-2 products:

  `EGM_GOC_2: Final GOCE gravity field model with error estimates and quality report.`

  This product has the following sub product files:

  ```
  EGM_GCF_2: Spherical harmonic series in ICGEM format
  EGM_GEO_2: Grid with geoid heights in Grid format
  EGM_GAN_2: Grid with gravity anomalies in Grid format
  EGM_GVE_2: Grid with east-west vertical deflections in Grid
  format
  EGM_GVN_2: Grid with north-south vertical deflections in Grid
  format
  EGM_GER_2: Grid with geoid height errors in Grid format
  EGM_GRP_2: Quality report in PDF format
  ```

  `and`

  `EGM_GVC_2: Variance-covariance matrix file of the spherical harmonics coefficients`

- Altimetric Mean Sea Surfaces
- Error information about time mean altimetric measurements.
- Digital terrain model

Software

Several software packages are available that will be collected here as the basis of an open source software package and that will be used as a base line in the trade-off studies of WP3000. Among those is the GRAVSOFT program GEOCOL that is being used in the ARCGICE effort. Another example is the spherical harmonic MATLAB toolbox (SHBUNDLE). The MATLAB toolbox can compute all relevant quantities from a spherical harmonic model such as geoid undulations, gravity anomalies, gravity disturbances, and the entire tensor of second order derivatives. This can be done in a grid or on given data points. The software collected in

WP3000 will be used and tested later as prototype toolbox. The software will be bundled later with a tutorial document to become a first prototype of toolbox.

A note on algorithms

It has been decided in the GUT consortium to use the full expressions for the wanted quantities instead of the spherical approximations used in the products obtained from the HPF since using the non-approximated algorithms will give a higher accuracy.

The HPF grids will also be supplied in the GUT and tools will be supplied to make interpolation to other user-defined grids possible.

A note on units

The units for the products will be as follows:

- Geoid height/height anomaly will be given in meters (m)
- Gravity anomalies will be given in milligal (mgal)
- Deflections of the vertical will be given in arcseconds.
- Dynamic topography will be in meters (m)
- Geostrophic currents will be in m/s

Handling of data

The requirements on the data input from the toolbox should be as low as possible to make it easier to include data supplied by the users. As such there should be only a demand on the format that this data should be in, and not on the naming of input data. Rather this name should be an input to the toolbox from the user. The fields that will be included in the toolbox should however, be named in a way that makes it easy to recognise what quantity they contain and which model the data comes from.

Regarding the output, this could easily be named GUT_XXX according to the functionalities described in the next section. With this document in hand the user could then easily see which product comes out.

2.4.2 Products, Input and Output Parameters

A list of products and functionalities are listed in Appendix A including documentation, formulas, and I/O parameter specifications.

3 Scientific Trade-off and Study and Algorithms

3.1 *Determination of toolbox functionality*

3.1.1 Review of User Toolbox requirements

The algorithm specification for GUT, as presented in this document, is based on the User Requirement consolidation, as described in [1] and above. However, determination of the functionality recommended for inclusion in the toolbox must also take into account the computational demands of the algorithms to fulfil the desired functionality and the accuracy that can be achieved by existing, available software, or algorithms that easily can be coded. Some of the algorithms needed for the desired functionality are subject to ongoing research and will not be part of the first release of the toolbox but might be included in later editions.
Since some of the required fields are necessary input for the computation of other products, while others are end products, the user requirements need also to be reviewed in terms of the logical structure of the computations in the toolbox before the provided functionality is determined.
A review of the user requirements has been carried out in order to rank the desired functionality and products. This will allow an evaluation of whether the first release of GUT meets the user requirements and where improvements will be needed.
In deciding on the functionality proposed for the first stage toolbox, we have used the priorities below in the first instance. In some cases, functionality that has been given a lower priority has still been included in the proposed functionality, as the practical decisions on how to incorporate the functionality have a very small overhead in terms of software development. For example, in some cases, existing software that provides high priority functions will also provide the additional functionality with no additional development.

Basic functionality

From the user requirements, as determined in the user requirements document, it is imperative, for all applications (geodesy, oceanography and solid earth applications) that the absolute minimal requirement of the toolbox includes:

- Computation of global, gridded geoid heights at a given, user-specified, degree and order of the spherical harmonic expansion (*i.e.*, at a given spatial resolution). User input would be the spatial resolution of the grid and the maximum degree and order of the expansion. Output would be a grid of single-point representations of the geoid.
- Computation of geoid heights at a given spatial resolution (*i.e.* specified degree and order of the spherical harmonic expansion) at a given point or list of points (*e.g.* unstructured grid, transect). Input would be a list of user-specified geographic position parameters and the maximum degree and order of the expansion. Output would be the geoid field interpolated to the specified positions.

This essential, basic functionality is required for application of the GOCE data to all scientific areas. At this early stage in the development of the toolbox, we expect that priority should be given to the needs of the oceanographic community in determining the algorithms to be included in the toolbox:

1) The oceanographic community have least experience in the use of gravity information and therefore have the greatest need for support in the use of these data.

2) The oceanographic community provide the largest potential group of users, who have well defined scientific questions and are organised to address these questions, and who would benefit from a toolbox, even of limited functionality.

In the following sections of the document, higher priority has been given to those toolbox functions and algorithms required by the oceanographic community, as identified in the User Requirement Document [1]. In order to provide a toolbox that will generate products that are more than the gridded level 2 products that are generated by the HPF, the toolbox must address at least one application of the data. We consider that the most widely applicable toolbox output is the generation of dynamic topography products to allow the exploration of absolute surface currents – a primary mission aim of the GOCE project.

Essential

In addition to the functionality given above, these user functions and products are considered essential outputs of any GOCE user Toolbox, without which, the toolbox would provide no significant advantage over the existing level 2 products.

- Provision of a priori MSSH, MDT and Geoid data on a grid
- Computation of a 'GOCE' MDT (MSSH-GOCE geoid) at a given spatial resolution and on a given structured or unstructured grid as the difference between the MSSH and the geoid
- Spatial filtering of MSSH consistent with a specific harmonic geoid height field expansion. This includes spectral space filtering of the MSSH.

Highly Desirable

In order for the toolbox to provide functionality that will support a wide range of users and cover the use-cases identified in the URD [1], it is *strongly recommended* that all of the functions and products identified in this section are implemented,

- Computation of global, gridded gravity anomalies at a given, user-specified, degree and order of the spherical harmonic expansion (*i.e.*, at a given spatial resolution). User input would be the spatial resolution or the grid and the maximum degree and order of the expansion. Output would be gridded gravity anomalies.
- Option to replace geoid heights by geoid slopes (deflection from the vertical) for either the gridded field or the unstructured positions
- The gridded fields should be available as both a single point representation and as area averages
- Change of reference system for the geoid and/or MSSH (reference ellipsoid, tide system)
- Computation of altimetric time-varying absolute dynamic topography as the difference between altimetric fields and a geoid model.
- Interpolation of external MSSH on any regular grid or at given points (unstructured grid, transects etc): This functionality should include procedures to translate (synthesize) and adapt the spherical harmonic information into geoid height information for any location in the world
- If several different geoid fields are available, either within the toolbox or by inclusion of external data fields, input to the calculation of both geoid and gravity anomalies would also be the choice of geoid model or gravity field.
- Handle appropriate ancillary data, *e.g.*, external MDT from ocean *in situ*/modelled, as well as local geoids, combined CHAMP/GRACE/GOCE geoids, among others
- Provision of tools to produce a global description of a combination of *a priori* gridded fields (*e.g.* provided or external MDT, provided or external geoids etc) in spherical harmonics
- Computation of geoid heights, gravity anomalies or deflections of the vertical at a given, user-specified, degree and order of the spherical harmonic expansion (*i.e.*, at a given spatial resolution) at the surface of the terrain. Output would be a grid or as single point representations, and as area averages
- Pre-viewing capabilities for results from the toolbox
- Must be able to select and display a 2D plot of any gridded field or transect calculated or provided within the toolbox (geoid height, MDT, MSSH etc).
- Should display at least 2 fields side by side – *e.g.* if remove-restore techniques used, select to display the replaced and smoothed residual fields
- Derive an optimal filter for the low pass filtering of the altimetric MSSH and/or the MDT derived from the altimetric MSSH and the GOCE geoid.

Desirable but lower priority

The following functions and products have been identified as providing useful additions to the toolbox, should it be possible to incorporate them. However, their exclusion would not pose a significant threat to the viability of the toolbox and its usage as they represent 'standard' procedures that can be carried out by users outside the toolbox.

- Compute differences / Root Mean Square differences / correlation coefficient / regression slopes between GOCE geoid and external geoids / 'GOCE' MDT and external MDT / between absolute altimetric dynamic topography and in-situ absolute dynamic topography / between absolute altimetric geostrophic velocities and *in situ* geostrophic velocities
- Computation of absolute geostrophic velocities from the spatial gradients of the dynamic topography field
- Extra viewing features
- select and display subsets and transects of data
- display statistical information – e.g. root mean square differences
- display spectral plots – will need to be calculated elsewhere

Functionality requiring extensive computing resources

- Computation of geoid heights covariance for any pair of points on the reference ellipsoid or the computation of a full covariance matrix for a given degree and order of the spherical harmonic expansion
- Computation of geoid cumulative height errors and error covariances at a given spatial resolution on a global regular grid or for a list of points.
- Computation of geoid cumulative errors and error covariances associated with geoid heights, gravity anomalies or deflections of the vertical at a given, user-specified, degree and order of the spherical harmonic expansion at a given spatial resolution on a global regular grid or for a list of points. Note: - the level 2 GOCE product, EGM_GER_2, will provide the geoid errors of the delivered gridded geoid heights, EGM_GEO_2. These will be an easy way of getting a priori errors for some of the products that GUT will deliver.
- Covariance error matrix within a chosen degree and order range for commission and with the option of including a model for the omission errors for the GOCE gravity field.
- Determine the parameters in a priori degree variance model for the modelling of the gravity field a priori spectrum (global and/or regional models)

Functionality not yet at consensus

- Derive a degree variance model for the MDT and determine statistical properties of the MDT and its associated geostrophic surface currents
- Covariance error matrix within chosen degree and order range for commission and omission error for mean dynamic surface topographies. This needs to include the auxiliary altimetric error covariances on top of the geoid model error covariance
- Option to include the omission errors for the MDT and the associated geostrophic surface currents.

Nice added features:

- data access and the possibility to transfer appropriate data from remote location to local machine

Must give user the instructions for how to do this if not included in the functionality of the toolbox

Functionality Desirable at Future Date

The User Requirement document suggested some potential toolbox functions that would, ideally, be included, but which at this stage are too immature in terms of application of the underlying science by the wider community.
Examples are:

- Regional geoid solutions starting from the global products. This is not ideal for regional geoid implementation. At a later stage, Level 1b data might be required as the basis for an extended GOCE geoid. Concerning regional geoids, a combination of model solutions using both GOCE and *in situ* gravity data [2] could be useful and including such regional models for the European Seas would be a useful element of GUTS.
- Methods for dealing with regional geoids and *in situ* gravity data are often based on a remove-restore algorithm and this would be a sensible extension of this aspect of the GUTS toolbox at a later date.

As updates to the GOCE geoid are made available, future releases should include the most recent gridded geoid product from GOCE.
The default MDT should be updated to be consistent with the GOCE geoid used by default in the toolbox.

3.2 Pilot applications; determination of filter algorithms

The scientific trade-off study focuses on the different filtering methods. The effect of different filters on the accuracy of the solution is studied along with the computational cost and the cost of implementation of the filters. Other factors in evaluating different filters include their general applicability and their priority, based on known qualities of the individual filters. The results of this study have been used to determine the default filtering and Dynamic Topography methods in the functionality and algorithm sections.

3.2.1 Introduction

Filter types and methods

For a first classification we differentiate between three types of filters:

geographical space filters: These filters involve a kernel $K(x,x')$ that is convolved with the original field $\eta_D(x')$ to give the filtered field $\eta_D(x) = \int K(x,x')\eta_D(x')dx'$. Different kernel functions are available, among them a spherical cap, a rectangular cap, a Gaussian or quasi-Gaussian cap, Hanning and Hamming windows.

spectral space filters: Here the field η_D is transformed into coefficients of a suitable expansion; in the current case only spherical harmonic functions $Y_n(x)$ of degree and order $n=(l,m)$ are considered. The spectral expansion $s_n(\eta_D)$ is then modified according to the filter kernel K_n and the filtered field is $\eta(x) = \sum_n K_n s_n Y_n(x)$. The most popular kernel functions are the Pellinen filter (equivalent to a spherical cap in geographical space), a quasi-Gaussian filter (the exact transform of the quasi-Gaussian in geographical space [3]), and a simple boxcar filter (Dirichlet window). The latter has undesirable properties in the geographical space domain.

optimal filters: Optimal interpolation or collocation techniques are used to find the filter kernel as the solution of minimizing an objective function of the type $J = (\eta_D - \eta)^T C_D^{-1} (\eta_D - \eta) + (\eta_B - \eta)^T C_B^{-1} (\eta_B - \eta)$. This solution is the average between the original field and a prior estimate η_B weighted by the inverses of the error covariances $C_{D/B}$ associated with the input fields: $\eta = (C_D^{-1} + C_B^{-1})^{-1} (C_D^{-1}\eta_D + C_B^{-1}\eta_B)$. Both C_B^{-1} and η_B can be zero, resulting in different properties of the solution.

When these linear filters are applied to the full fields directly, we will refer to the method as direct filtering.

In the second approach, the so-called Remove-Restore (RR) technique, the direct filtering is replaced by combining a prior guess η_B with a filtered residual: $\eta = \eta_B + F^T (\eta_D - \eta_B)$ with F^T being some linear filter. In the context of combining a geoid model with altimetric sea surface height (SSH), the residual will be taken between the geoid model and a to-be-constructed "background" geoid model.

Filtering dynamic topography

In the case of filtering dynamic topographies, these are the criteria, which are used to evaluate a filter

• accuracy (distance from a reference solution), formal error description (with or without omission error) and comparison with reference data (to be specified)

• universality (can the filter be used on irregular grids, individual point pairs, etc.?)

• computational cost (CPU time and memory requirements: difficult because these depend strongly on the implementation of the filter)

• cost of implementation (we can only roughly estimate that, *e.g.* this filter is simple to implement, this filter is available as a prototype, etc ...)

• simplicity in terms of usage (how many parameters are there, *e.g.*, just the maximum degree cut-off or a full error covariance matrix that needs to be specified by the user)

• necessity of having the filter (*e.g.*, do we need an ”optimal” filter?)

Each filter can get a “grade” in each category. The overall “grade” has to be a weighted average of the individual “grades”, where the weights reflect the importance of the individual categories (*e.g.* is cost of implementation more important than accuracy?).

Technically speaking, we construct a benchmark system which is simple enough, so that all filters can be run within it and complicated enough so that it represents enough features of a real ocean dynamic topography problem. Here we have the choice of using ”real” data, or purely synthetic data, that is, numerical model results that are modified to represent geoid and altimetry data. The advantage of the former is that we are immediately faced with all the difficulties that one normally has with real data and can take these difficulties into account while evaluating the filters. The disadvantage is the same. Using synthetic data and using identical twin experiments enables us to know the results a priori and we can control the biases. We choose the latter approach.

Error estimates of filtered dynamic topography

For any filtering method, the filtered field should be associated with an error estimate that is computed in a way consistent with the errors of the original fields and/or prior guesses. In principle this can be achieved by filtering the error covariances C_D associated with the input field η_D:

$$C = F^T C_D F \tag{1}$$

To take this a step further, the filtering result itself should implicitly include the prior error estimates. Only the optimal filters do this in a straightforward way (see above). For the optimal filter the posterior error can also be computed rigorously as $C = \left(C_D^{-1} + C_B^{-1}\right)^{-1}$, in principle. In practice, both the optimal filter and its error estimate are computationally extraordinarily expensive, and approximate methods may be used. Also, reliable estimates of the prior errors C_B are not available. Although, as the name implies, this method is “optimal”, finding efficient and accurate filters based on this approach is the subject of research and can only be considered as a future part of the GOCE User Toolbox.

In constructing a (spectral) geoid model, one truncates the spectrum at a certain degree L, usually where the signal to noise ratio exceeds unity. For all degrees (and orders) $\leq$ L one has the coefficients of the model along with their error co-variances (the **commission** error) from the collocation process. The signal for degrees > L is not modelled, but it is identified as the omitted signal or **omission** error. Different models for the omission error exist, but they all have in common, that the omission error is *finite*, although it can be large depending on the cut-off degree L. The omission error poses a problem when one wishes to construct a dynamic

topography (DT) from SSH minus geoid model. It is vital that the scales that are suppressed by the chosen filtering algorithm and therefore become "omission" are matched identically in the SSH and in the Geoid. It is very easy to cause aliasing if for the SSH and the geoid model, the represented scales are slightly different (especially at low resolution), because these fields are usually computed with different "basis" functions (grid vs. spherical harmonic functions).
For computing the errors associated with a filtered mean dynamic topography (MDT), the omission error may become an issue that needs to be dealt with. Usually, the geoid model has a lower resolution than the SSH, so the geoid omission error is larger (reaches down to larger scales). The error spectrum of geoid models is blue, that is, errors are very small for small degrees and orders of magnitude larger for degrees near the cut-off degree L. For degrees > L the error is omission error and tapers off according to the chosen omission error model.
Previous research [4] has shown that a considerable portion of the omission error can leak into the commission error of the filtered signal, if the involved transformation involves dramatically different basis functions (Legendre function vs. trigonometric functions, spherical harmonics on the sphere vs. eigenfunctions of the error covariance along a hydrographic section).

3.2.2 Methods

Accuracy of filter method

First let us introduce some terminology: The dynamic topography η_D, mean or instantaneous, is the difference between the altimetric sea surface h, (mean or instantaneous) over a reference ellipsoid, and the geoid height N_D, over the same reference ellipsoid, so that

$$\eta_D = h - N_D \tag{2}$$

The index D stands for data (observations). Because h and N_D do not contain the same spatial scales it is common to introduce the (linear) filter (matrix) F^T:

$$\eta = F^T \left(h - N_D\right) \tag{3}$$

Eq. (3) comprises the first method (the direct method) of obtaining a dynamic topography η. The assumption is that the filtering must ensure that the omitted scales in h and N_D match each other very closely because these fields vary 100 times more than the residual η.
The second method is called the Remove-Restore (RR) method and is based on the idea of adding a correction to a prior guess of the dynamic topography η_B, where the index B stands for the prior background. This background can be a high-resolution ocean model or observed dynamic topography. From this we construct a synthetic (background) geoid by

$$N_B = \begin{cases} N_D & \text{over land} \\ h - \eta_B & \text{over the ocean} \end{cases}$$

The RR-method then adds a (filtered) residual to the background value:

$$\eta = \eta_B + F^T \left(\eta_D - \eta_B\right) = \eta_B + F^T \left(N_B - N_D\right) \tag{4}$$

Depending on the quality of the prior dynamic topography η_B we expect the RR-method to perform better than direct filtering, as additional independent information is used. It is again critical that the filtering ensure a close match of the omitted scales in the two Geoids. Using the field $h - N_D$ as prior η_B, results (correctly) in no increment at all. Note that this is still not optimal unless information about the accuracy of the measurements and prior guess enter the algorithm through the filter definition.

With this machinery, we choose a η_T to be the "truth" and a high-resolution "true" geoid N_T that also represents the background geoid N_B over the ocean. In fact, N_T does not have to be very accurate on the short scales, so the EGM96 ($L = 360$, [5]) is sufficient. The "true" dynamic topography is a mean taken from a ¼˚ ocean model with data assimilation (OCCAM, Ocean Circulation and Climate Advanced Modelling Project [6, 7]).
The background dynamic topography is taken from a different ocean model with data assimilation (ECCO [8]) to take into account possible errors and biases is the prior during the RR-method. From Eq. (2) we estimate the altimetric measurement $h = \eta_T + N_T$. To generate an "observed" geoid we remove short scales from N_T by applying an appropriate filter G (see also Figure 1):

$$N_D = G^T N_T$$

The measure of accuracy of the filter F is expressed as

$$\left\| \eta_T - F^T \left(h - N_D \right) \right\| \text{ and } \left\| \eta_T - \eta_B - F^T \left(N_B - N_D \right) \right\| \quad (5)$$

for the direct method (Eq. 3) and the Remove-Restore technique (Eq. 4), respectively. $\| \cdot \|$ is an appropriate $L2$-norm, for example the root-mean-square (rms).
For testing purposes it is sufficient to use coarse resolution ocean model data (*e.g.* on a 1 degree grid), and use a "cut-off " for the coarse-grained geoid N_D of $L = 70$, which corresponds to a resolution (half-wavelength) of 2.5˚ (286 km) spherical distance. In two different cases the filter F is supposed to suppress scales above $L = 45$ and $L = 25$, corresponding to scales (half-wavelengths) of 445 km and 800 km, respectively.
For the Remove-Restore technique, one needs to take into account the error of the prior or background dynamic topography η_B. These errors can be substantial. As an example, we compare the mean dynamic topography η_B of the ECCO2 product and the "truth" η_T of the OCCAM product. The mean dynamic topographies differ by as much as 1 m in the regions of strong topography gradients, that is, strong surface currents (see Figure 2). The rms-difference is 11 cm.

Computational cost

Computational cost is estimated based on the used Matlab-code, which is designed for small problems, so that execution time can be favoured over memory requirements. For large problems the algorithms may have to be modified in order to reduce memory requirements at the cost of computational speed. On the other hand, the algorithms used for the geographical filters leave some room for optimization for speed.

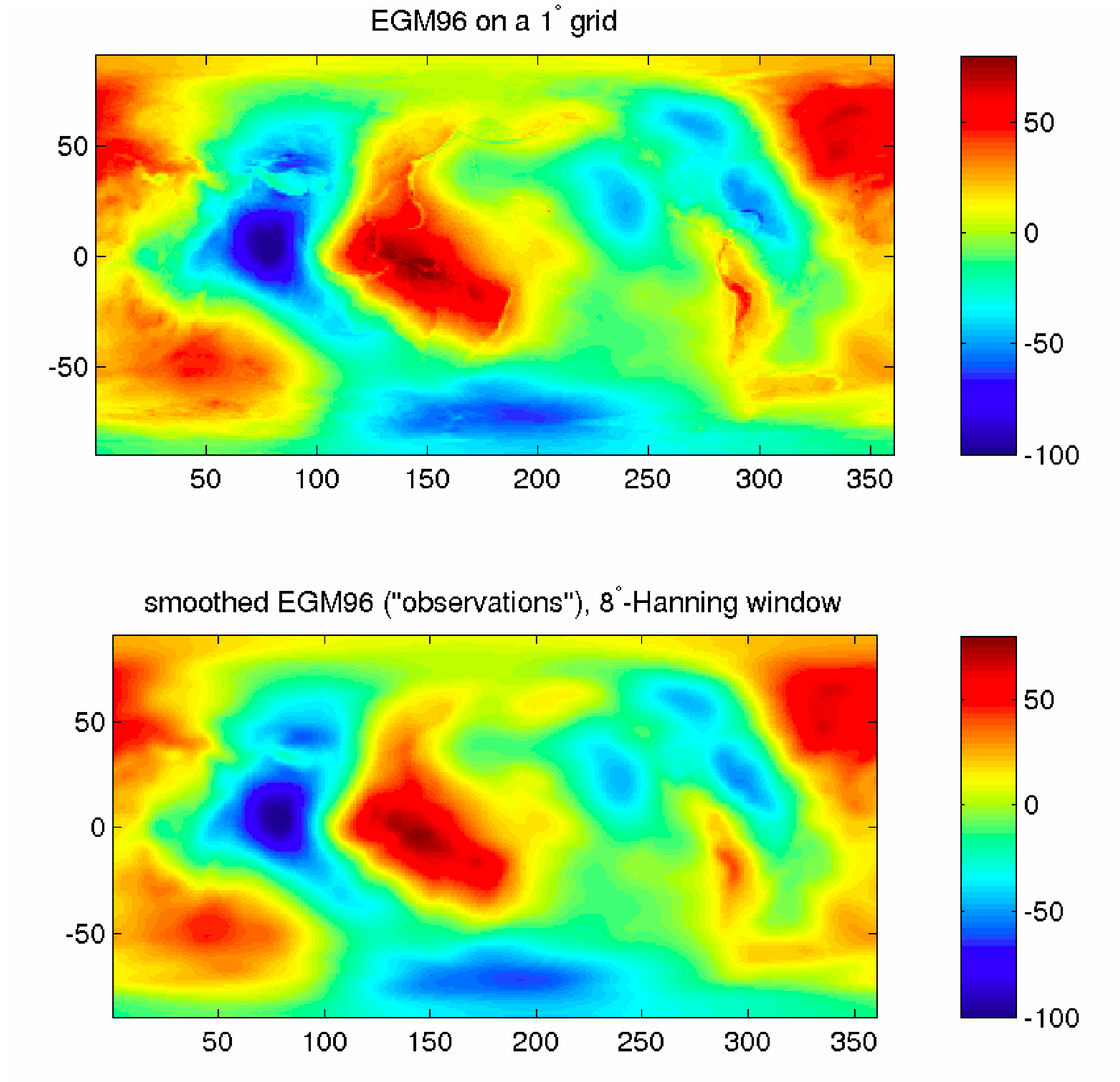


Figure 1: EGM96 geoid undulations on a 1° grid. Bottom: Smoothed geoid, pseudo-observations

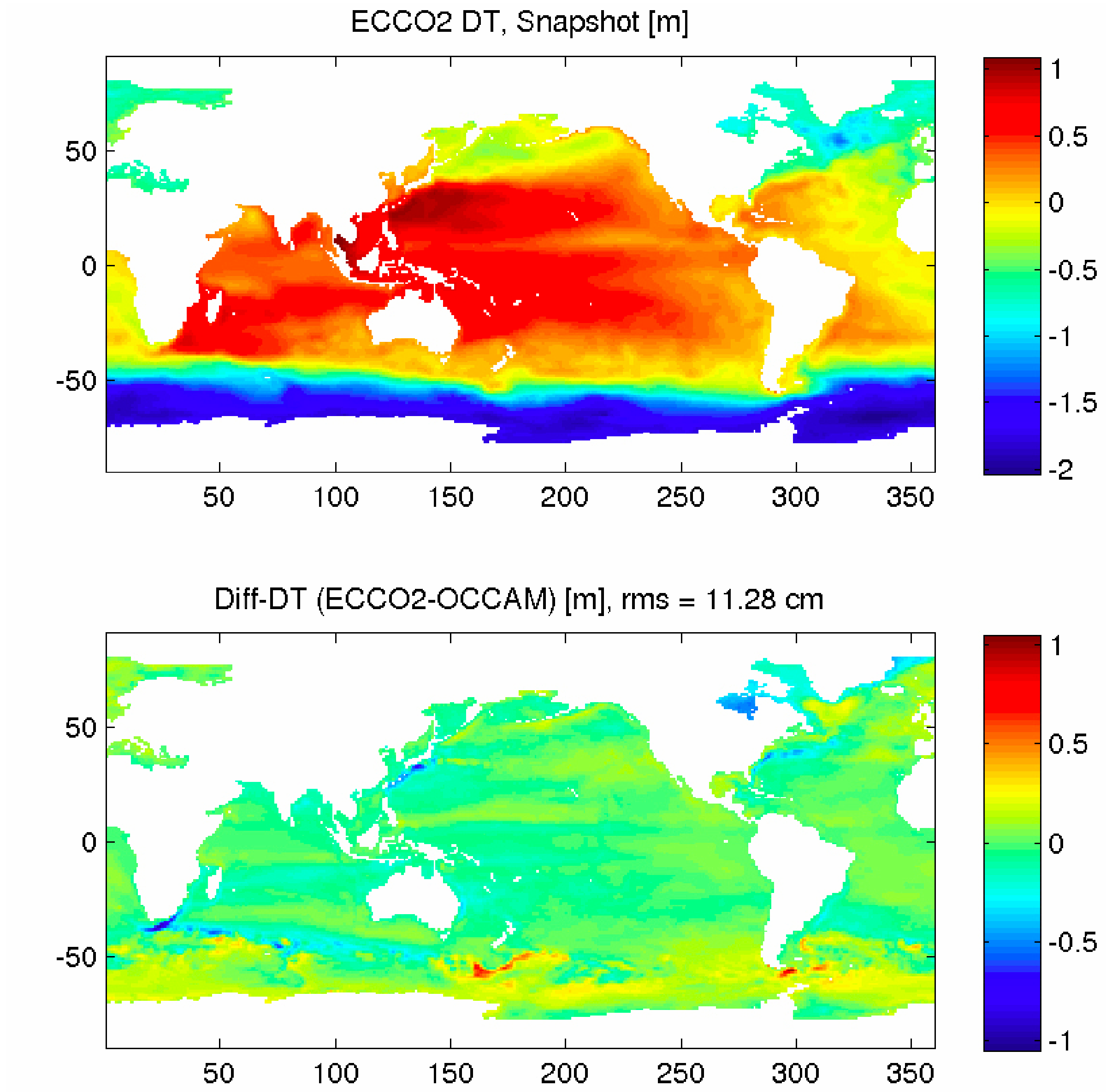


Figure 2: Top: Prior background dynamic topography used in this study. Bottom: Difference of mean dynamic topography of two different ocean models

Omission error

Filtering the error covariance, provided that there is one, is very expensive and time consuming. Therefore, we use an even coarser resolution (4˚, corresponding to 445 km) to test the effect of the omission error on the resulting filtered errors, bearing in mind that the issues with the omission error may increase with decreasing resolution. For geographical space filters it is important that the resulting errors are computed over the ocean only. For the spectral space filters the omission error of the geoid is partially suppressed, if the filter scale is larger than scales associated with the cut-off degree L, however the geoid omission errors created by spectral filtering will not exactly match those of a real space filter.
The total error covariance of the dynamic topography is the sum of the commission error covariance C_L, the omission error C_{om} and the error of the SSH C_{SSH}. C_{SSH} is taken to be diagonal for simplicity assuming a constant 5 cm error; one could introduce horizontal correlations, but for the current application with a maximum resolution of 445 km we consider

the SSH error uncorrelated. For the geoid model we use a preliminary error covariance of the CHAMP mission [9], which is available to degree and order 60, corresponding to 335 km resolution (wavelength), so that our 4˚ grid can just not resolve the full geoid errors. (We can equally well use a covariance model such as that of [10]). We choose a commission cut-off of $L = 20$, corresponding to 1008 km half-wavelength, which can be resolved by the 4˚ grid. The error information on degrees $L + 1$ to 180/4 = 45 is then considered "omission" error, because our low-resolution geoid is truncated at L, but the computational (ocean model) grid resolves degree 45. This "omission" error has a cumulative degree variance corresponding to 13 cm. (For a realistic case with an expected resolution of GOCE of $L = 200$ corresponding to 100 km half wavelength, the target resolution of an ocean model could be 1/4˚, which corresponds to $L = 720$. The omission error from $l = 201$ to 720 can contribute to large scales mainly because its cumulative variance is on the order of 20 cm.) In our approach we neglect all omission error beyond the resolution of the computational grid. This error, which is still large, could be taken into account by evaluating the omission error on an even higher resolution grid and averaging it onto the target resolution. However, we assume, somewhat optimistically, that the omission errors of the geoid model and the errors due to grid resolution cancel.
Now we consider two cases: inclusion and exclusion of the omission error before filtering according to Eq. (1). If the filters remove short scales efficiently, the omission error should have little effect on the solution. However, if including the omission error leads to larger errors, particularly of large scale quantities such as the global mean dynamics topography, leakage from short geoid scales to large geographical scales is a problem and needs to be addressed. For the geographical space filters, leakage stems from evaluating the geoid model over the ocean only by applying a realistic land mask to the filter. For spectral space filters the omission error of the geoid can be filtered efficiently (in principle by a box car filter, which unfortunately has bad properties in geographical space), but in order to apply a spectral space filter, the SSH error needs to be transformed into spectral space according to:

$$C^{SSH}_{n,n'} = \iint Y_{n'}(x') C^{SSH}(x',x) Y_n(x) dx dx'$$

and here again the land mask may cause leakage.

3.2.3 Results

Commission: Filtering the signal

Table 1 summarizes the trade-off study. In general, spectral space filters are much faster (factor 16 in our case) than grid-space filters, because the latter involve as many convolutions with the filter kernel as there are grid points, whereas the former involves only one. With increasing resolution, this difference in efficiency may even increase. All filters are simple to implement. Optimization for speed may involve a little more sophistication.

Table 1: Summary: Grading of filters

Filter type	accuracy [cm]	universality	Computational cost [s]	cost of implementation	simplicity	necessity
Direct filters						
a. geographical space filters						
spherical cap	44	good	823	small	good	yes
lat-lon box	45	good	862	small	good	no

Gaussian	36	good	893	small	good	yes
quasi-Gaussian	37	good	1009	small	good	yes
Hanning	39	good	1006	small	good	yes
Hamming	39	good	1018	small	good	yes
b. spectral space filters						
box car	49	medium	59	medium	medium	no
Pellinen filter	37	medium	60	medium	medium	yes
quasi-Gaussian	34	medium	60	medium	medium	yes
c. "optimal" filters, not evaluated						
Remove-Restore techniques						
a. geographical space filters						
spherical cap	43	good	823	small	good	yes
lat-lon box	44	good	862	small	good	no
Gaussian	33	good	893	small	good	yes
quasi-Gaussian	35	good	1009	small	good	yes
Hanning	38	good	1006	small	good	yes
Hamming	37	good	1018	small	good	yes
b. spectral space filters						
box car	48	medium	59	medium	medium	no
Pellinen filter	35	medium	60	medium	medium	yes
quasi-Gaussian	30	medium	60	medium	medium	yes
c. "optimal" filters, not evaluated						

The general advantage of the grid-space filters is that they are universally applicable, while the spectral space filters strictly speaking can only be applied to truly global fields. For computing regional fields spectral space filters may be inefficient, because they still require filtering a global field. Also it is not intuitive why local features should have a global effect, as they do with spectral space filters with insufficient resolution.

The difference between the constructed sea-surface height "from altimetry" h and the "observed" geoid $N_D = G^T N_T$ is shown in Figure 3. This "raw dynamic topography" has many short scale features near coasts and near steep topographic gradients, where the filter G has removed the short scales from the geoid, but not from the altimeter signal.

Figure 4 summarizes the filter performance in terms of rms-difference (Eq. 5).

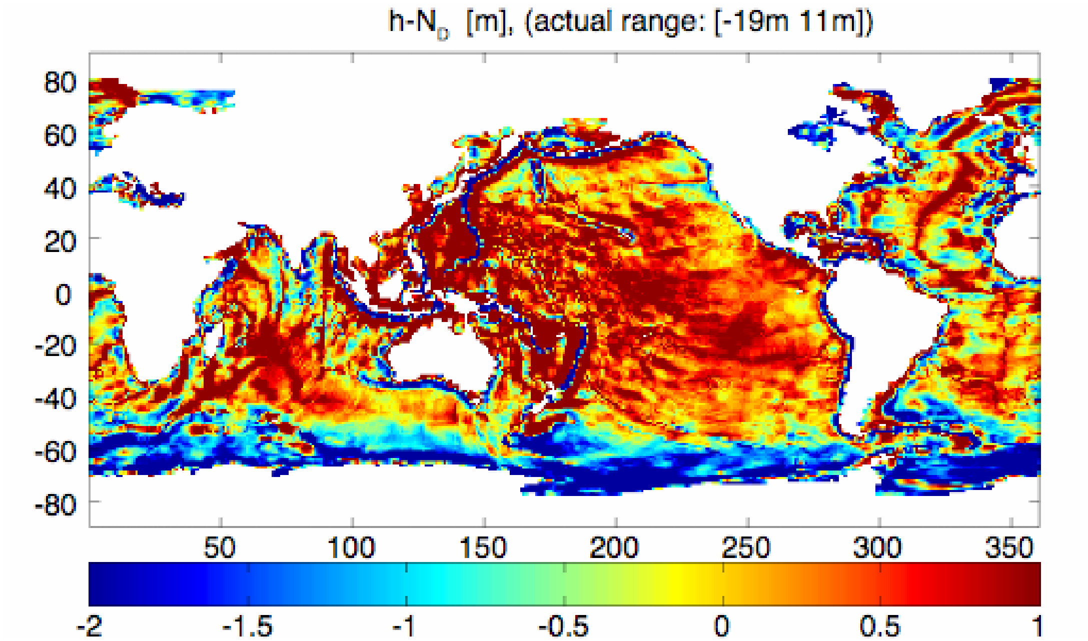


Figure 3: Difference between "altimetric" sea surface height and observed geoid. Note that the difference reaches more than 10 m. The colour range is restricted to make the figure comparable to following figures of smoothed dynamic topography

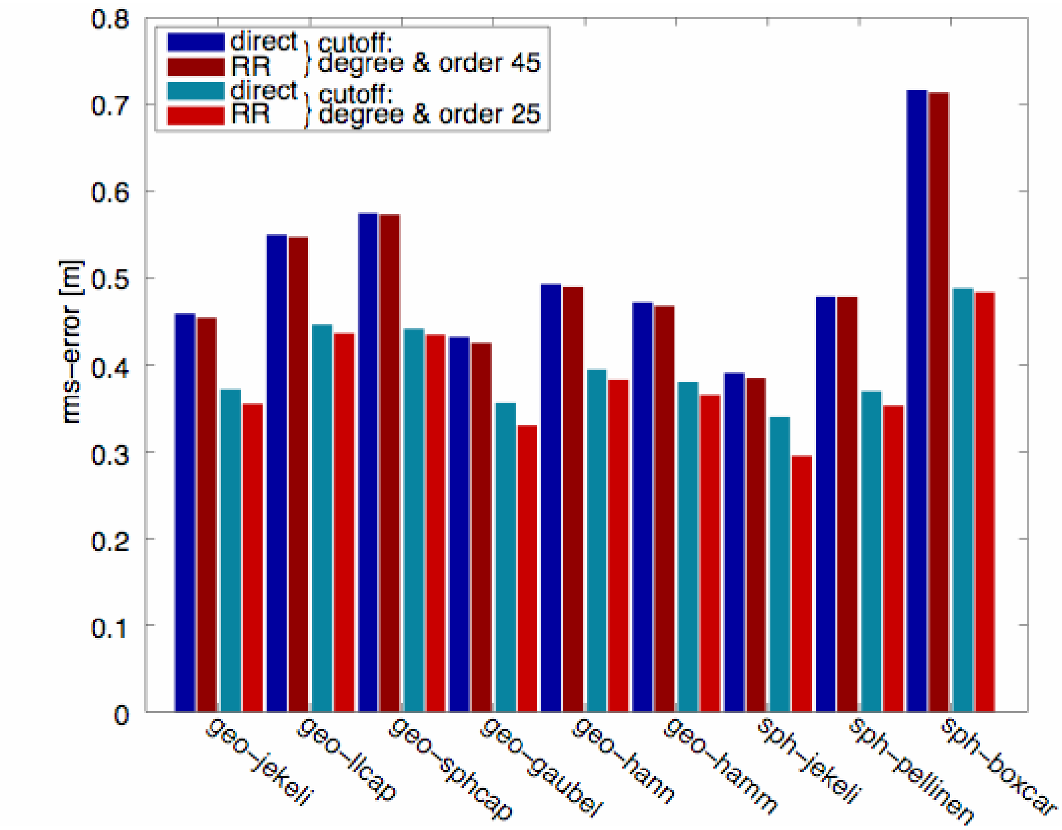


Figure 4: Comparison of rms-difference between filtered fields and "truth", for the default cut-off degree L=25 and for a higher cut-off degree of L=45. Note that with increasing resolution the positive effect of the RR-method decreases

The Gaussian and quasi-Gaussian filters lead to the smallest rms-differences. As expected, the filter with the Dirichlet-window in spectral space is the worst in terms of rms-difference.
As an example, Figure 5 & Figure 6 show the dynamic topographies of Figure 3 after filtering with a quasi-Gaussian [11] and a spherical cap/Pellinen filter, respectively. Each figure contains the results of 4 different methods: Direct filtering with a geographical (grid-) space filter (top left), with a spectral (spherical harmonics) space filter (top right), and with the Remove-Restore method (bottom left and right). With the Remove-Restore technique the resulting dynamic topographies contain more small scales (bottom panels) than with the direct method.
There appears to be an advantage, possibly fortuitous, with using a spectral space filter in the representation of marginal and shelf seas, such as the Hudson Bay or the Mediterranean Sea where the "raw" signal is very much depressed. The grid space filters, due to the lack of information over land, cannot reduce this depression, whereas the spectral space filters by construction use (false) geoid information over land and thus can better correct this problem.
The quasi-Gaussian filter appears to be more efficient in removing short scales associated with topographic features than the spherical cap/Pellinen filters, which is reflected in a smaller rms-difference to the truth (cf. Figure 4).

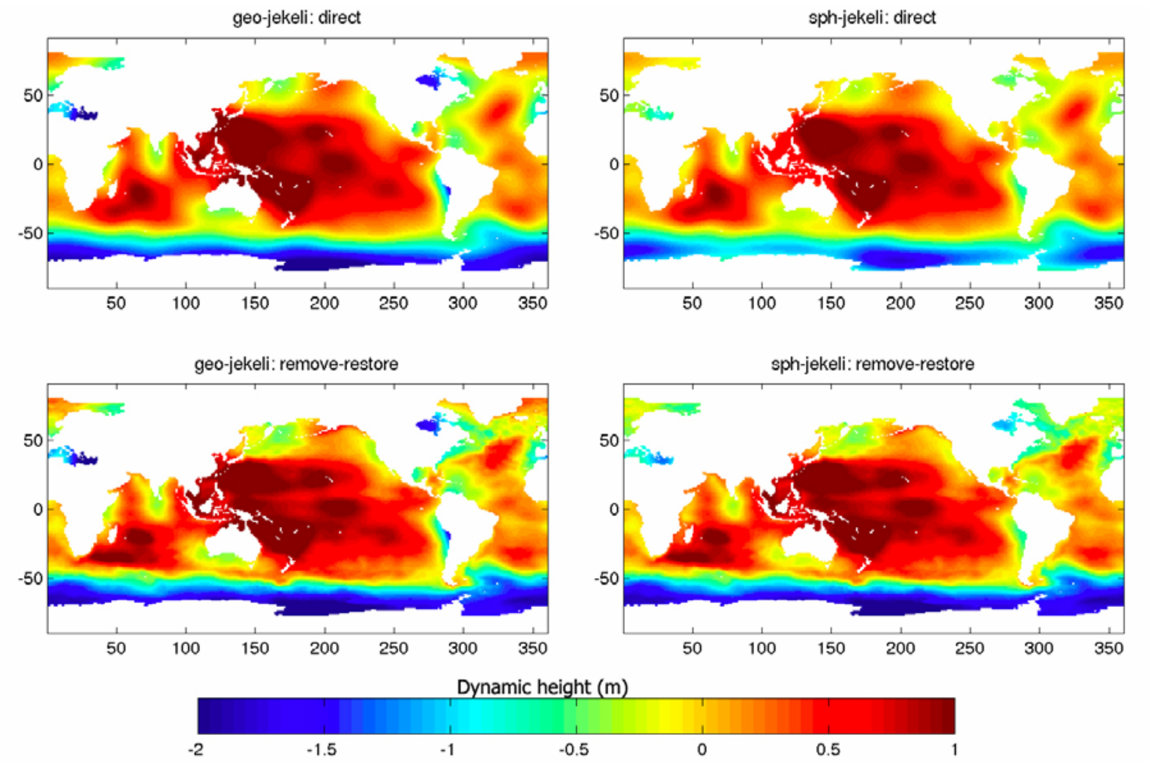


Figure 5: Dynamic topography after filtering with a quasi-Gaussian (Jekeli) filter. Top: direct method, bottom: RR-method, left: geographical (grid) space filter, right: spectral (spherical harmonics) space.

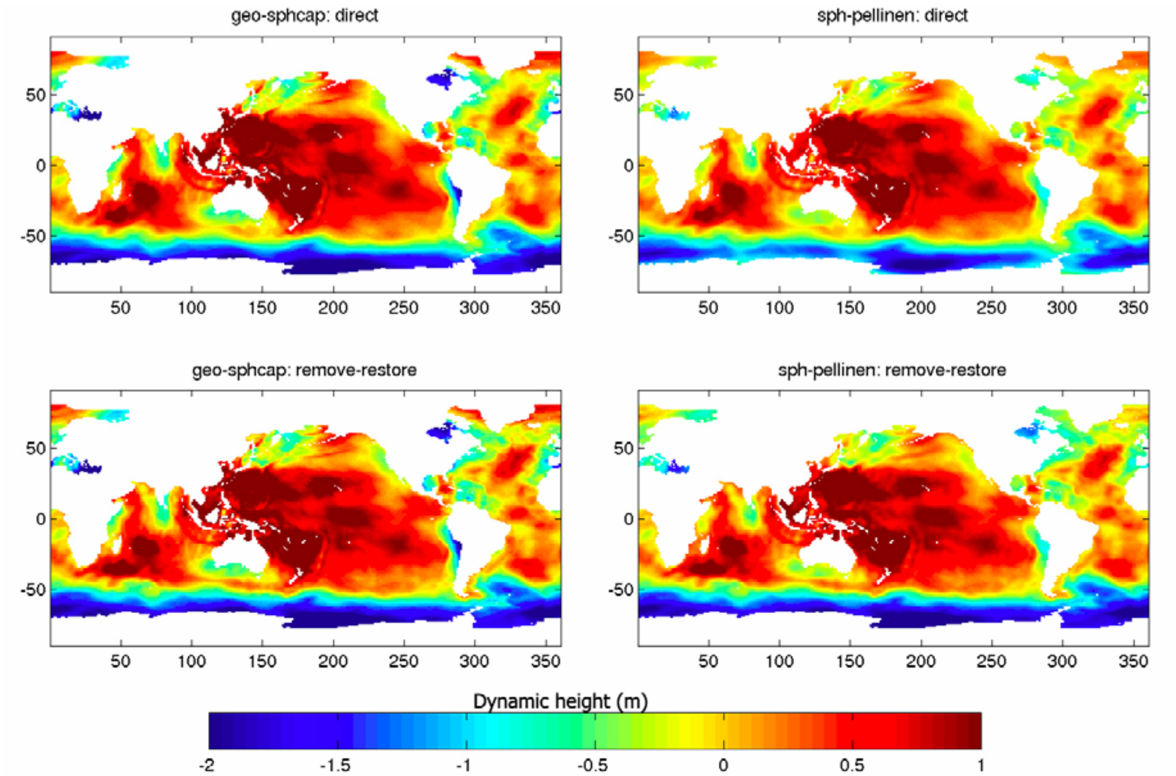


Figure 6: Dynamic topography after filtering with a spherical cap/Pellinen filter. Top: direct method, bottom: RR-method, left: geographical (grid) space filter, right: spectral (spherical harmonics) space

The difference between the Remove-Restore and direct method in Figure 7 shows the features of the dynamic topography that are omitted in the direct method. However, these features are completely associated with the prior dynamic topography η_B and may contain large errors.

Effects of the omission error

For brevity, we consider the effects of the omission error mainly by the example of the quasi-Gaussian (Jekeli) filter. From Figure 8 and Figure 9 it is clear that the spectral space filter reduces the estimated errors more efficiently than the grid-space filter. In the present example the effects of the omission error are small (two orders of magnitude smaller than the error signal) for the spectral space filter, but not negligible for the grid-space filter (Figure 10). In the latter case the magnitude of the omission error effect is about a factor 6 smaller than the error signal. As an example for a large scale feature of the dynamic topography, where all short scale effects, and thus the omission error, is expected to drop out, the error of the global mean of the dynamic topography is estimated (Figure 11). While the mean error is small (and irrelevant for any dynamical considerations), including the omission error increases the mean error by approximately 10% in the case of the grid-space filters. The spectral space filter leads to even smaller errors and they can effectively suppress any omission error effect (by construction).

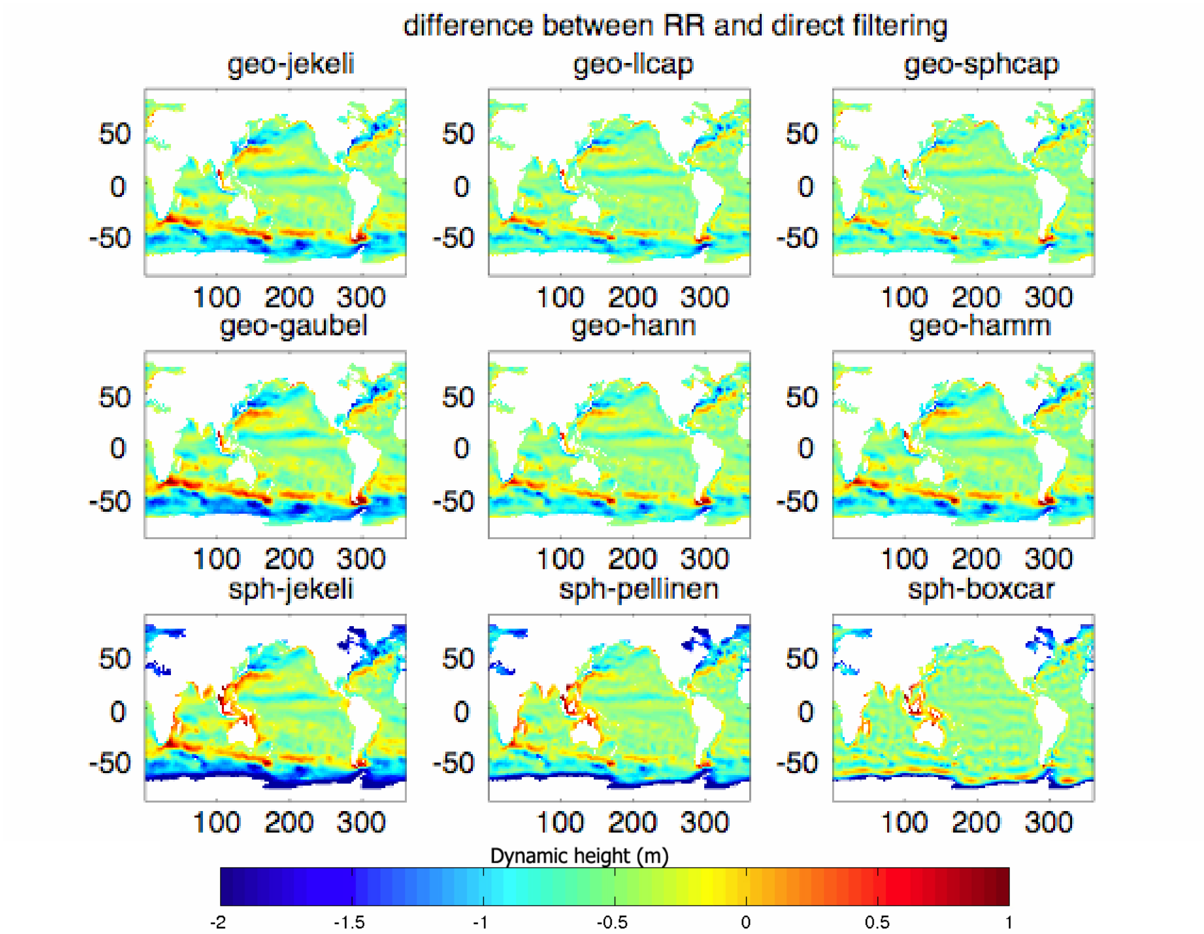


Figure 7: Comparison between the RR-method and the direct method. The difference shows the features of the dynamic topography that are omitted in the direct method. However, these features are completely associated with the prior dynamic topography η_B and may contain large errors

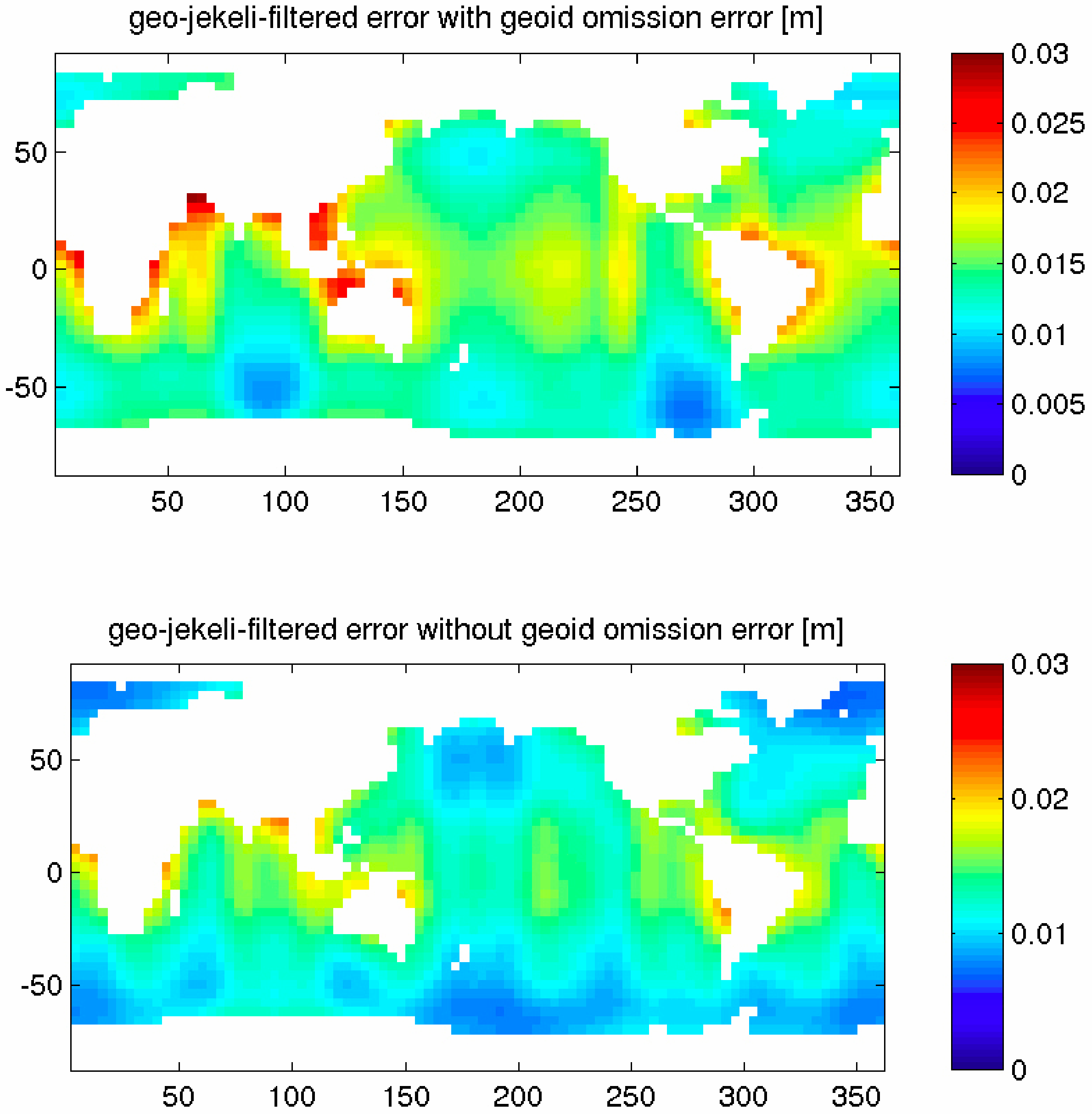


Figure 8: Dynamic topography error after filtering with the quasi-Gaussian (Jekeli) grid-space filter. Top: omission error included, bottom: omission error excluded.

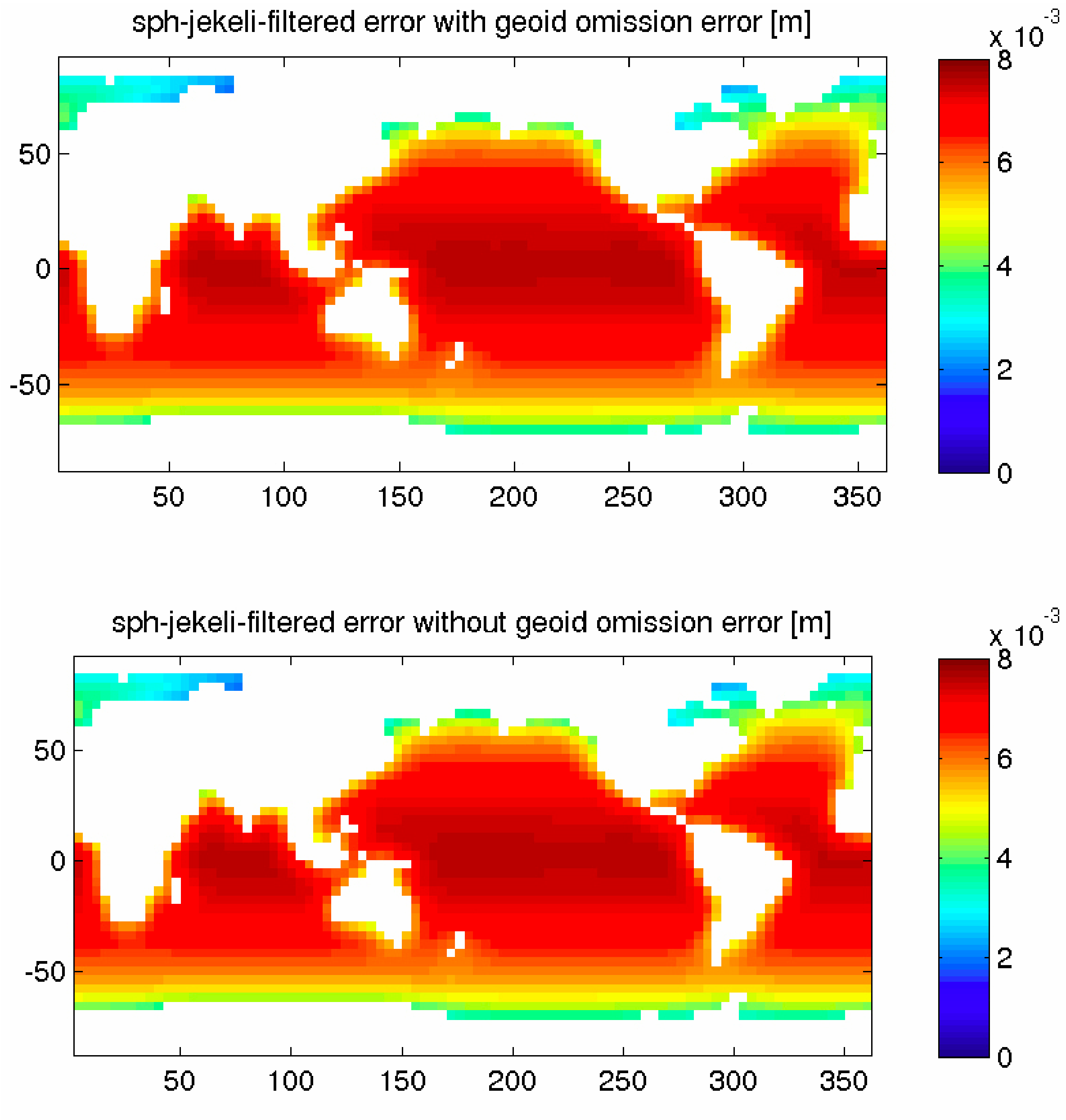


Figure 9: Dynamic topography error after filtering with the quasi-Gaussian (Jekeli) spectral space filter. Top: omission error included, bottom: omission error excluded

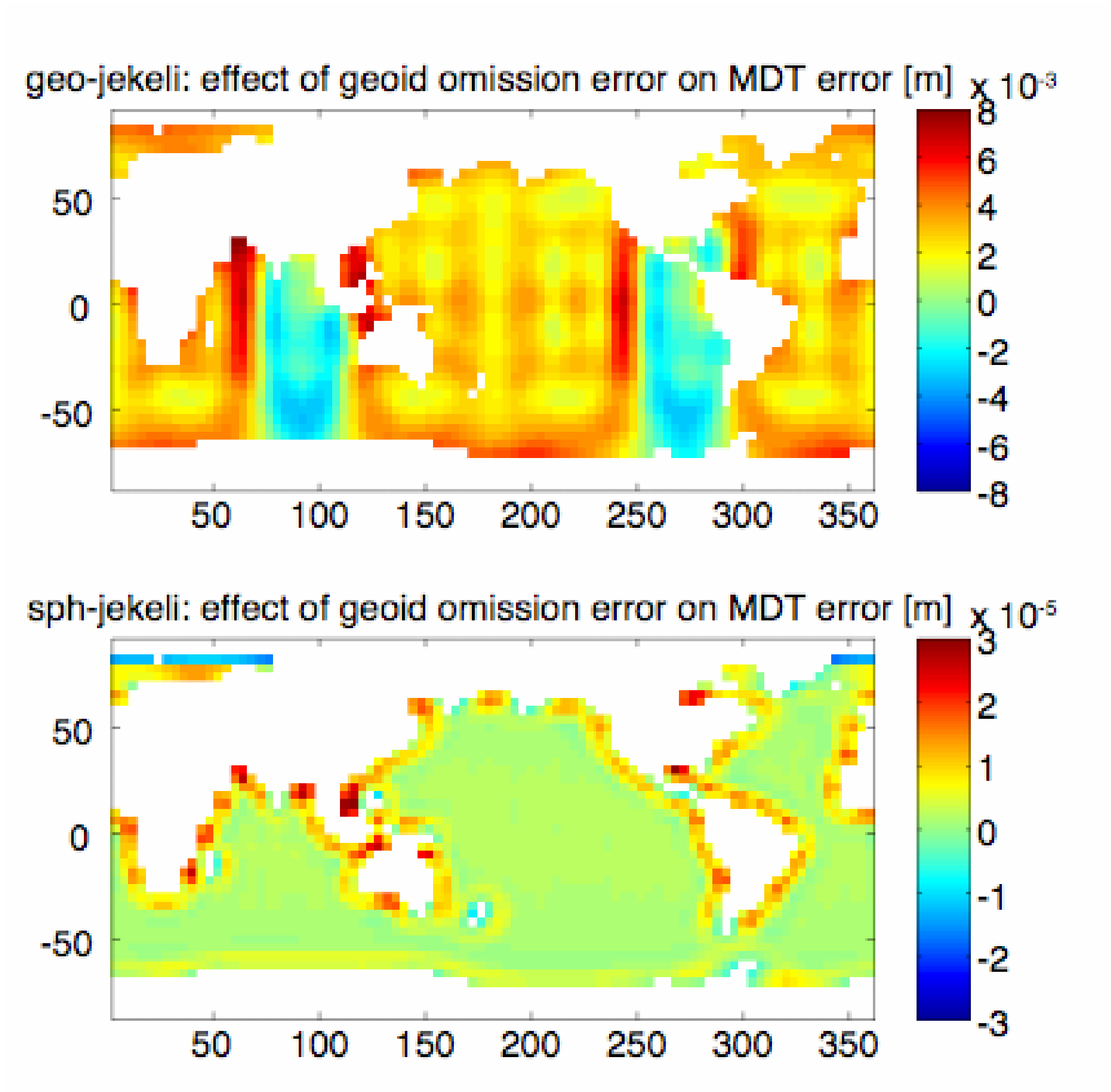


Figure 10: Difference in quasi-Gaussian-filtered error due to inclusion of moderate omission error. Top: grid-space filter, bottom: spectral space filter

3.2.4 Discussion and Conclusion

In this section we evaluate the results and give a rough recommendation for choosing an appropriate filter, based on section 3.2.3. In our testbed examples, the Remove-Restore technique emerges as superior to the direct filtering method, because it yields the smallest overall rms-differences to the "truth" (not the prior dynamic topography!). The requirement of a prior dynamic topography is not regarded as a restriction, **as long as such a prior guess is provided as an integral part of the toolbox** along with a simple method to replace the default prior guess with one provided by the user. However, estimating the error of this prior guess remains a difficult issue.

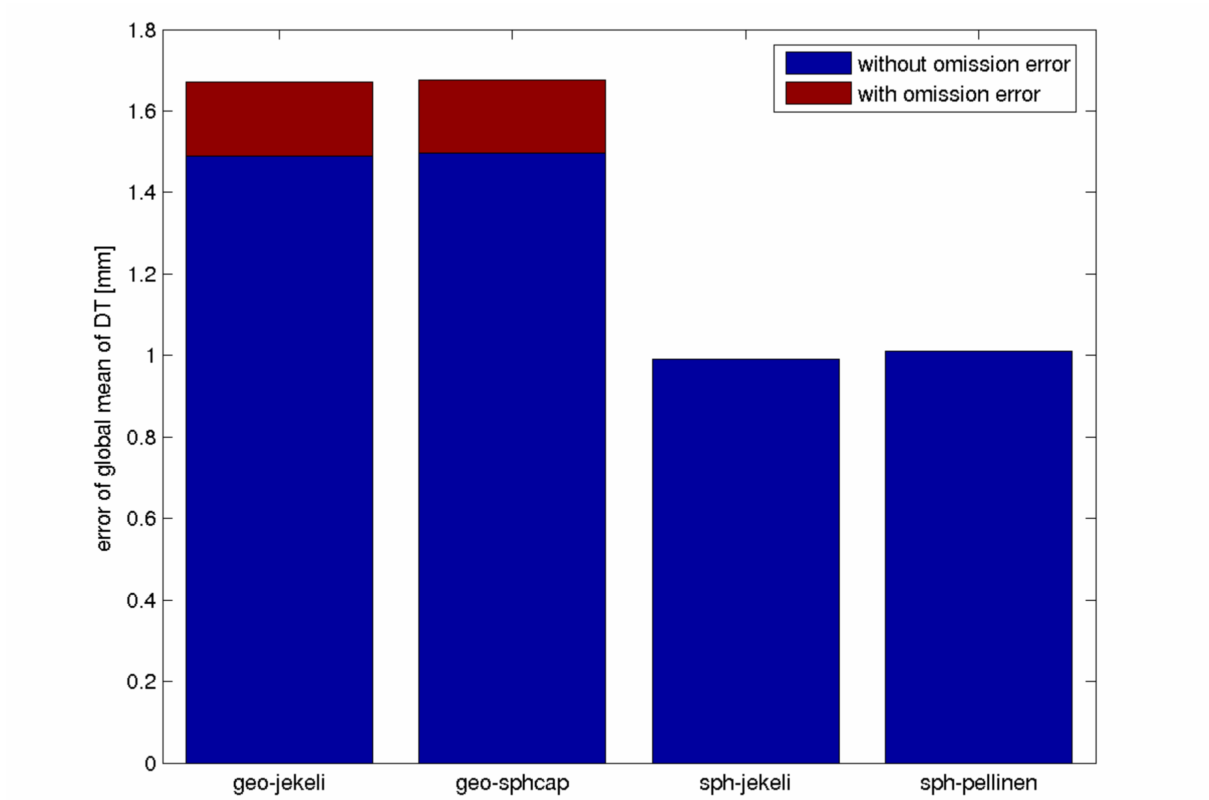


Figure 11: Effect of including the omission error on the error estimate of the global mean of the dynamic topography. Note that for the spectral space filters, the effect is not visible in this plot

Many filter kernels of this study perform reasonably well in smoothing the unfiltered difference between sea-surface height and observed geoid. Based on the global rms-difference between the filtered topography and the truth, the simplest filters, that is the grid-space rectangular (lat-lon) cap, spherical cap, and the spectral space boxcar (Dirichlet-window) filter are not recommended. These grid-space filters have a spectral response with negative side lobes and a boxcar filter in spectral space leads to Gibbs fringes in geographical space.
On the other hand, grid-space filters with a shape that resembles the Gaussian bell curve, such as the quasi-Gaussian kernel of [11], a true Gaussian kernel, and the Hanning and Hamming type windows give the smallest rms-difference between filtered dynamic topography and "truth".
The spectral versions of the quasi-Gaussian and spherical cap (Pellinen) filters also give small rms-differences. These small differences to the "truth" can be attributed to the good representation of enclosed seas, such as the Hudson Bay and the Mediterranean Sea. The filtering of the geoid, which is implicit in the spectral representation of any geoid model leads to a removal of short scales. However, these scales are present in the sea-surface height data so that the dynamic topography is grossly wrong. In the case of enclosed seas, or near coastlines where the geoid gradients are large (*e.g.*, along South America's West coast), where grid-space filters fail due to the lack of information on the true geoid gradients, the originally undesired property of a spectral space filter, namely that it uses information of an undefined and therefore arbitrary "sea surface over land", tends to alleviate the problems of grid-space filters. Therefore the spectral space filters appear to be more accurate than the grid-space filters. However, with increasing resolution and required accuracy, this advantage is believed to vanish, as the spectral space filters always use arbitrary information over land. We stress that the "omission" error of the sea surface height is not considered. This "omission" error (unresolved signal) is difficult to

estimate, but we expect, that its effect will be opposite to that of the geoid omission error, that is, larger with spectral space filters than with grid space filters.
Spectral space filters, when applied to the error covariance of the dynamic topography to estimate the errors of the smoothed dynamic topography, lead to smaller and smoother error estimates than grid-space filters. Additionally, by construction, they can suppress the omission error efficiently by matching these errors between the Geoid and the MSS. In contrast, filtering the error covariance in grid space leaves a residual of the omission error that contributes to even the larges scales of the dynamic topography.
In this study the error propagation is treated only very approximately and results can only serve as a rough guideline. Only “optimal” filters treat the formal errors rigorously. We have excluded the “optimal” filters from this study, as they may be very expensive computationally; and conceptually, they are still subject to research. In the future, filters based on optimal interpolation or collocation techniques and in combination with the Remove-Restore technique are expected to provide more reliable estimates of the dynamic topography along with an error estimate.
In conclusion, none of the studied filters entirely satisfy the requirements of producing a reliable dynamic topography. All rms differences exceed 35 cm and are thus too large. However, compared to grid-space filters, spectral (spherical harmonics) space filters generally appear to be more accurate in this study. Filters with a Gaussian-like roll-off give more accurate results than those with sharp cutoffs in either grid-space or spectral space. Spectral space filters are also much faster than grid space filters. Spectral space filters efficiently suppress the geoid omission error (but probably not the sea surface height omission error which is difficult to assess). The major issue of spectral space filters are discontinuities at the land-sea boundary. Therefore we recommend spectral space filters for filtering of global dynamic topography fields, in particular with remove-restore techniques that are designed to reduce this discontinuity. For regional dynamic topography applications, grid-space fields are likely to be more efficient and accurate than spectral (spherical harmonics) filters.
All filters of this study are suboptimal and further investigations into the subject of filtering a geoid and sea surface height or the resulting dynamic topography are sorely required.

3.3 Structuring functionality; defining workflows

The functionality as determined by the following evaluation of the user requirements has to be structured in order to provide fast computation of the output fields without reducing functionality and good visibility of the logical structure of all operations accomplishable within the toolbox.

The structure is provided by organizing the functionality of the toolbox into 7 workflows. One workflow, the main workflow, includes the core functionality of the toolbox in a default processing mode and default input data designed for fast access of the main output fields and use in tutorials. The input data used in the main workflow should be included in the toolbox distribution; thus, the main workflow is the ready-to-use playground for novices.

The convenient use of the toolbox including the application of the user's own data and applications deviating from the main workflow is supported by the remaining 6 sub-workflows organized in six modules:

7. Geoid and gravity field computation
8. Sea surface height and a-priori dynamic topography selection
9. Satellite Dynamic Topography computation
10. Combined Dynamic Topography computation
11. Dynamic Topography-derived quantities
12. Pre-viewing function

The basic elements in the workflows are the *GUT functions* (boxes in the workflow diagrams below) as well as input and output fields (rounded boxes in the workflow diagrams). The *GUT functions* are the lowest processing level that can be accessed by the user when running the toolbox. The computations within a function might contain more than one algorithm as specified in the User Requirement Document. The selection of the algorithm used in a function is in some cases controlled by options provided as input to the function or might be a chain of algorithms, or both in combination.

In this section an overview of the functionality in GUT is presented. The input and output definition of the functions and the link to the algorithm specification in the URD is provided in section 4.

3.3.1 Main workflow; supported input data

GUT comprises a set of functions that can be independently accessed by users. But in addition, pre-defined workflows support convenient use of the toolbox for most applications.
Figure 12 displays the main workflow of the GUT including in- and output fields. This workflow comprises the main functionalities of the toolbox by using input data provided by GUT. No additional data have to be provided by the user. This workflow gives an overview of the functionality of GUT and serves as a basis for the definition of tutorials that are part of the GUT distribution.
Each GUT function defined here is actually a chain of functions that again defines one of the workflows explained in the subsections below but with some restrictions in the options setting and the permitted input data. Different options settings ('test cases') are possible and subject to the tutorials set-up.
Computation of geodetic quantities (geoid height, deflections of the vertical and gravity anomaly) is performed on a regional or global grid or at a list of points specified by the user. Each of the three quantities is also provided on a pre-defined grid as part of the Level-2 products.

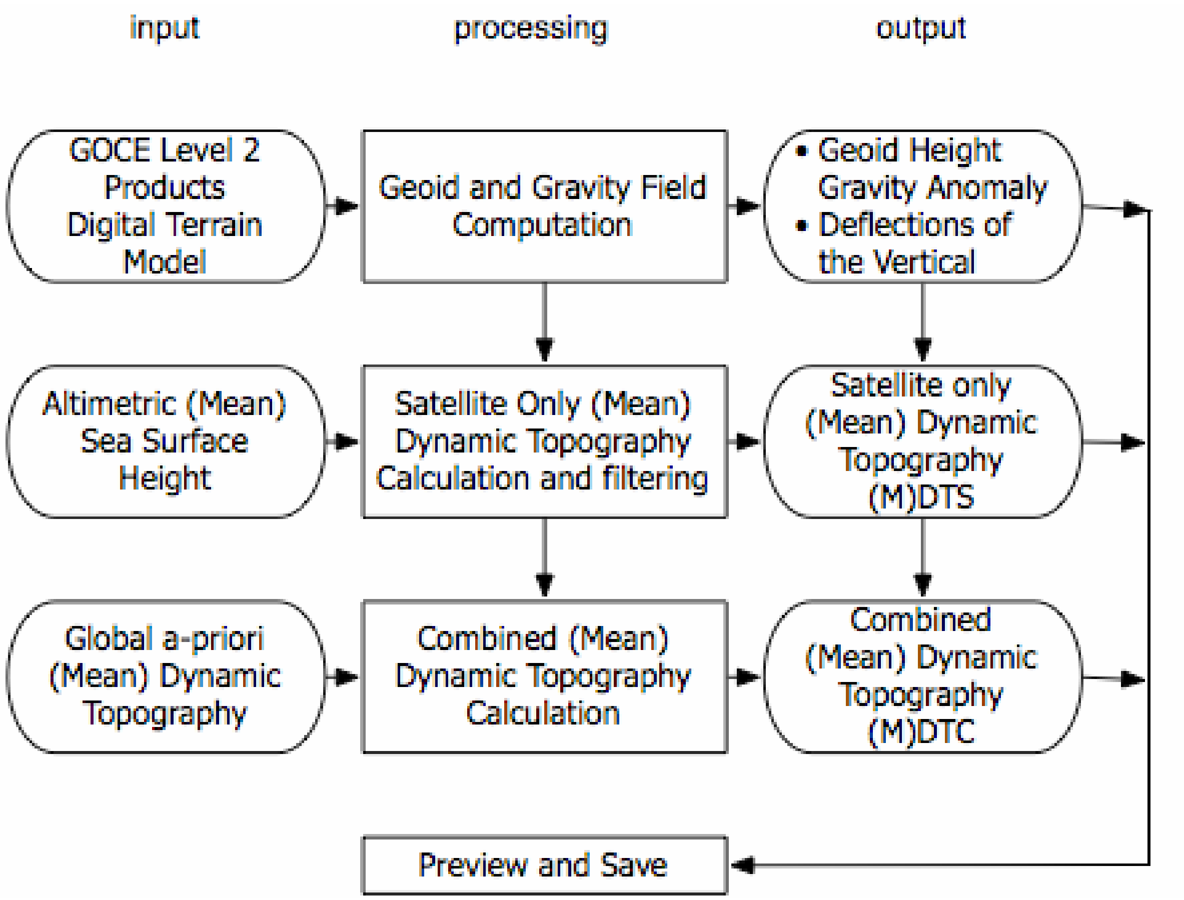


Figure 12: Main workflow for GOCE User Toolbox.

A Mean Dynamic Topography (MDT) is provided, as derived from the geoid height and the Mean Sea Surface Height (MSSH) implemented in the toolbox. The MDT is provided in two qualities:

1. as a geodetic or 'satellite only' version (MDTS): the difference of MSSH and geodetic geoid height, filtered using a linear operator.
2. in a combined version (MDTC): where in addition to the MDTS, small scale features are restored by using a remove-restore technique.

The notation of satellite only MDT (MDTS) and Combined MDT (MDTC) follows the use of the S and C products from the GRACE mission where the MDTC product includes additional information (in this case from oceanography).

The computation of the MDTC uses information about the small-scale variability in the MDT provided by the a-priori MDT included in the toolbox.

The Dynamic Topography calculation methods make no distinction between mean or instantaneous fields. Although the following descriptions refer to Mean SSH (MSSH) and Mean Dynamic Topography (MDT), the methods are equally applicable to instantaneous fields.

MDT slopes and surface geostrophic currents are computed and all input and output fields can be displayed by a previewing functionality.

In addition to the implemented input fields the users can provide their own data for each of the input fields, including a gridded or spherical harmonic representation of the geoid height field, using pre-defined formats with full support of functionality.

User input data supported also includes:

1. Gridded Sea Level Anomaly (SLA) or time series of Sea Surface Height (SSH)
2. altimetric along-track SSH relative to specified ellipsoid
3. a regional or global a-priori MDT.

3.3.2 Geoid and gravity field computation

Geoid and gravity field computation is performed in GUT by default as indicated in Figure 13. It is proposed that the toolbox provides computation of:

- Geoid height
- Gravity anomaly
- Deflections of the vertical

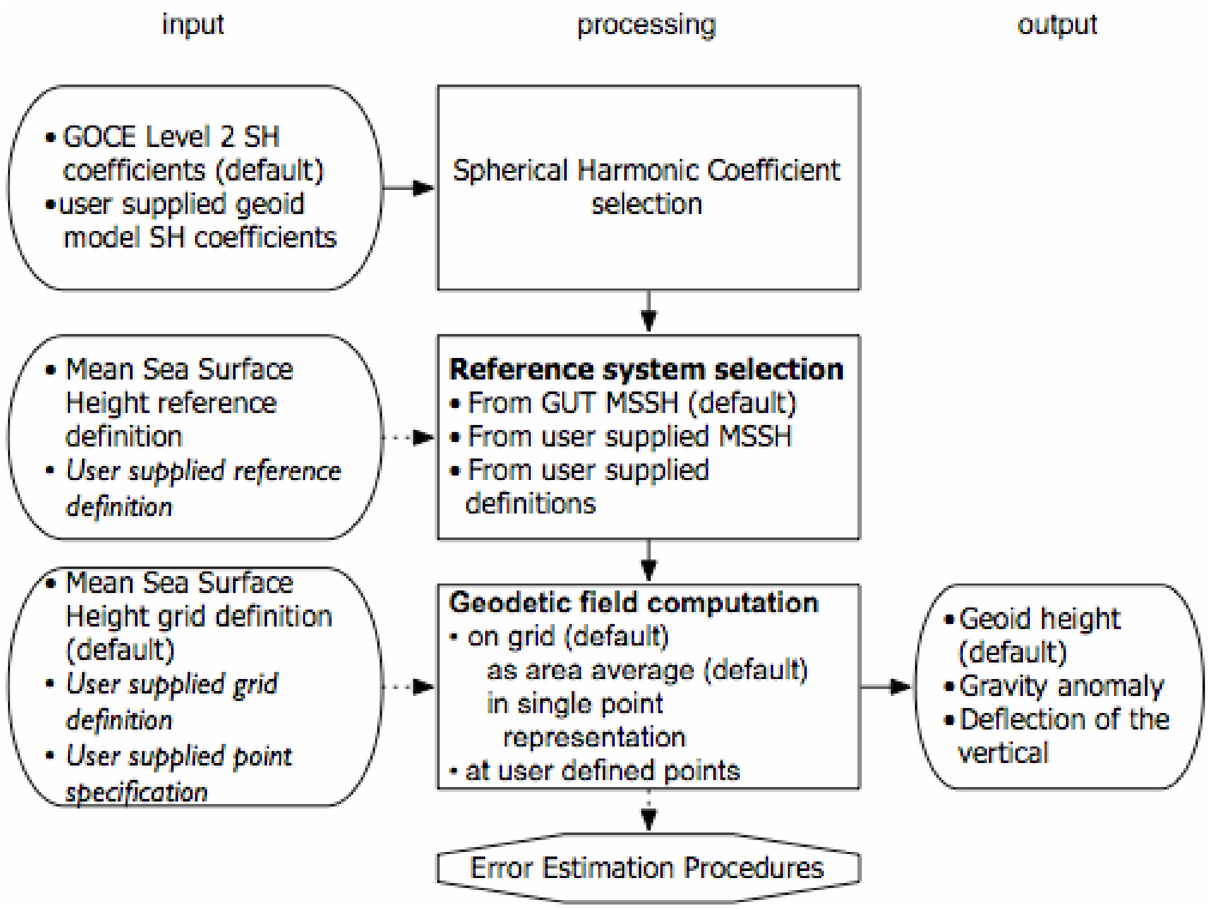


Figure 13: Workflow 1a: Geoid and gravity field computation

The fields are by default calculated from the GOCE Level-2 spherical harmonic (SH) coefficients (EGM_GCF_2). Ideally, the associated error fields would be calculated from the level 2 SH variance-covariance matrices (EGM_GVC_2). However, the computational requirements for propagation of error variances form the SH to grid space are too high to be included in a distributed toolbox. This part of the workflow could be developed at a later stage to utilise external computing facilities (local or ESA GRID, for example) using parameters provided from within the toolbox. Instead, the error variance for the default GOCE gravity model will be provided, for maximum degree and order, from the level 2 gridded products and interpolated to the required input grid or point series.
The highest degree and order (DO) used in the spherical harmonic expansion is specified by the user. Above the selected maximum DO SH coefficients might be selected from a geoid model included in GUT. User supplied geoid models are also supported.

The fields listed above are provided for a user-specified grid or a list of points. For irregular gridded fields the specified quantity can be determined as either a single point representation or an area average. The fields are evaluated on the reference ellipsoid or on the surface of the terrain by using the implemented or a user-defined digital terrain model.
By default, the commission error variance is provided. The omission error can optionally be added to gain a full error estimate. For the omission error, it has been suggested that a homogenous, isotropic function can be provided for a list of distances specified by the user. In practise, the variance of the omission error will vary and should be calculated regionally.
The reference system (reference ellipsoid and tide system) for the geoid height can be chosen by the user. By default the same system as used for the MSSH is adopted. Thus, the reference system has to be specified in the documentation of the MSSH.
Spatial filters for 2D fields are provided to account for the different omission error scales in geoid height and altimetric SSH for calculation of MDTS. The filters are described in the MDT section.

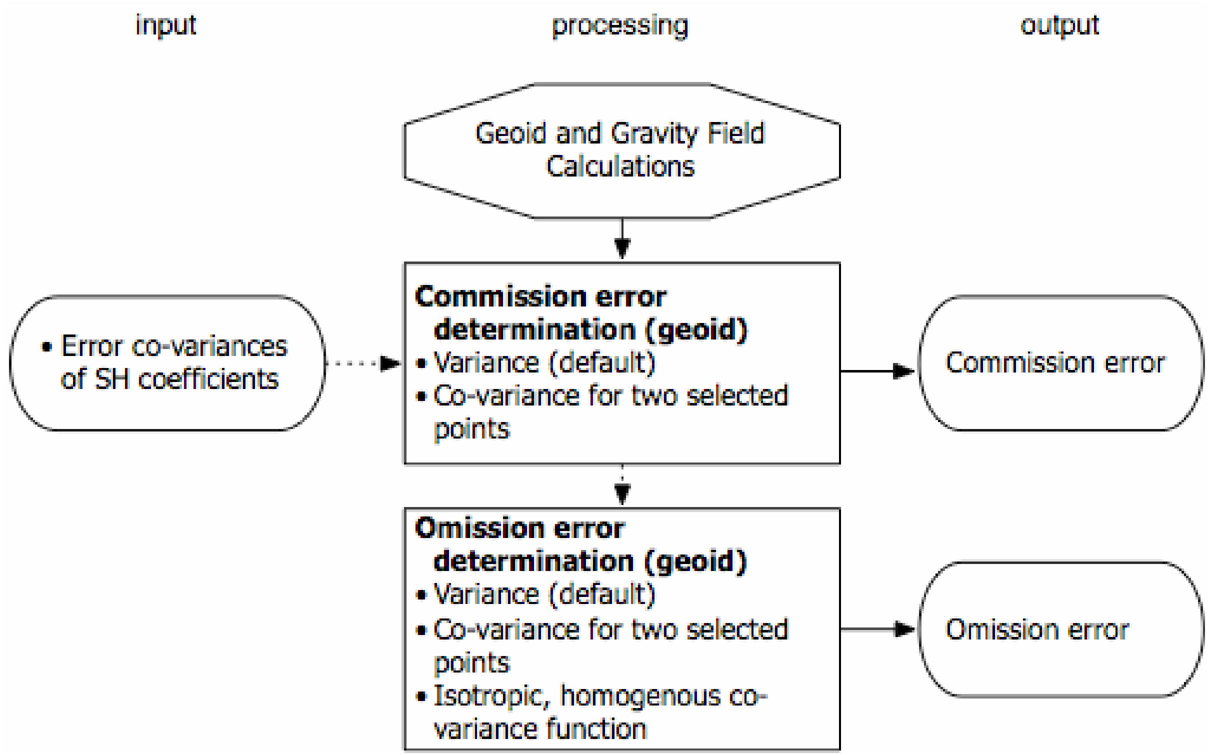


Figure 14: Workflow 1b: Error computation for geoid and gravity field: Note covariance calculations are not expected to be implemented in an initial toolbox

3.3.3 Sea surface height and a-priori dynamic topography selection

The Dynamic Topography as a main GUT product is time dependent and specification of the reference period, or computation of time dependent Dynamic Topography, is frequently needed. It is suggested that GUT provides two algorithms as links between an archive of SLA, MSSH and Dynamic Topography fields on one side and the computation of MDT within the toolbox on the other side (Figure 15). The averaging routine reads data from the archive and calculates the mean for a specified period. The output field can than be gridded or computed at specific points as convenient for the subsequent computations in the toolbox. Data in the archive might be given as weekly, monthly and annual fields. Coastal regions could introduce large errors into resultant fields when grid adaptation is carried out (from a source grid to the required output grid occurs). This toolbox will not attempt to resolve these issues, which are dealt with by more specific software tools, such as the Basic Radar Altimeter Toolbox (BRAT). This workflow has a low priority in the toolbox. A practical alternative, recommended for the initial toolbox release, is to take advantage of the Basic Radar Altimeter Toolbox to provide the necessary functionality in manipulating altimeter data prior to ingestion in to GUT.

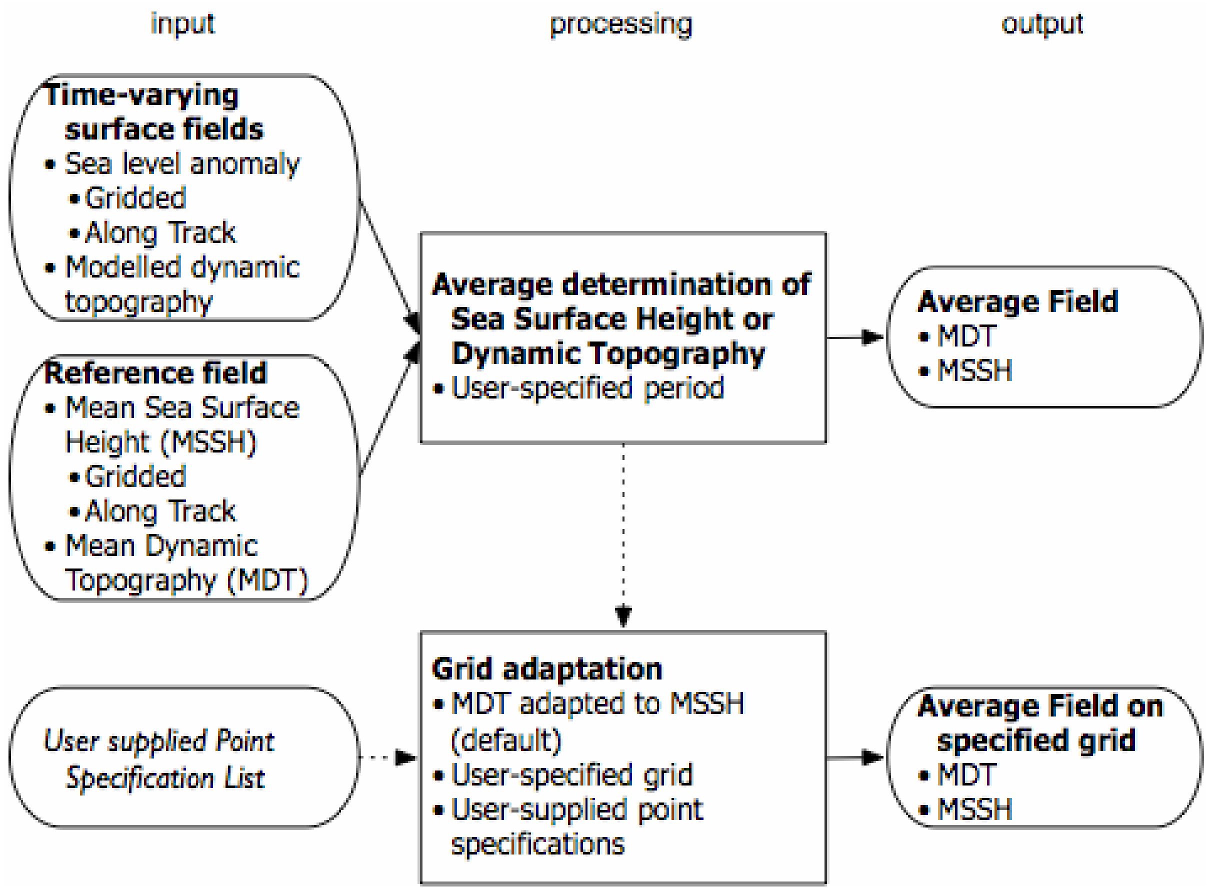


Figure 15: Workflow 2: Sea surface height and a-priori MDT selection

3.3.4 Satellite Dynamic Topography computation

Calculation of the Dynamic Topography (Figure 16) consists of two consecutive steps: First, the difference of altimetric SSH and the geoid provided upstream in the workflow (see section 3.3.2) is calculated with the aid of linear filters. This produces the satellite only product: MDTS. In a second step, small-scale structure is added using a remove-restore technique and an a-priori MDT. This provides the combined MDTC. The computation of the linear filtered satellite only MDTS is described here, the combined technique is described in section 3.3.5. Two different methods are provided to compute the MDTS. The first is using the geoid height as provided as output from workflow 1 (section 3.3.2) and computes the difference compared to a MSSH in geographical (grid-) space. The difference is filtered, using a linear filter, (workflow 3a, see Figure 16). The second method uses expansion of the MSSH into spherical harmonic coefficients and the MDTS is then determined from the difference of MSSH and geoid in spectral space, filtered using a linear filter and transformed to physical space (workflow 3b, see Figure 17). For global grids, the outcome of WP3300 suggests that the spectral (spherical harmonic filtering) option should be the default workflow.

Spatial MDTS (default for regional grids and user defined points)

In most cases a geoid height is provided upstream in the *Geoid and Gravity field computation* module and thus MSSH and geoid height are adapted to a common grid and reference system. However, both fields might also be provided by the user. Thus a consistency check is done. By default, both grid and reference system of the MSSH field are applied for both fields.
SSH is only provided as gridded MSSH within GUT. The user can provide SSH fields for specific dates or periods as well as along-track data. The *Surface height and a-priori dynamic topography selection* module can be used to average the SSH data if required.
The geoid height is subtracted from the MSSH field before performing the linear filter. A number of spatial linear filters are available:

- spherical cap
- lat-lon box
- Gaussian/quasi-Gaussian (=Jekeli)
- Hanning/Hamming filters

The user has to define the filter function, the filter width and the output grid. If subsequently the *combined technique* is used to improve the small scale structure it is recommended that the filtered MDT is gridded according to the a-priori MDT used in the combined method.
Recommendations from the *Pilot applications* report are that quasi-Gaussian filters provide the best geographical filter characteristics and should be presented as the default option.

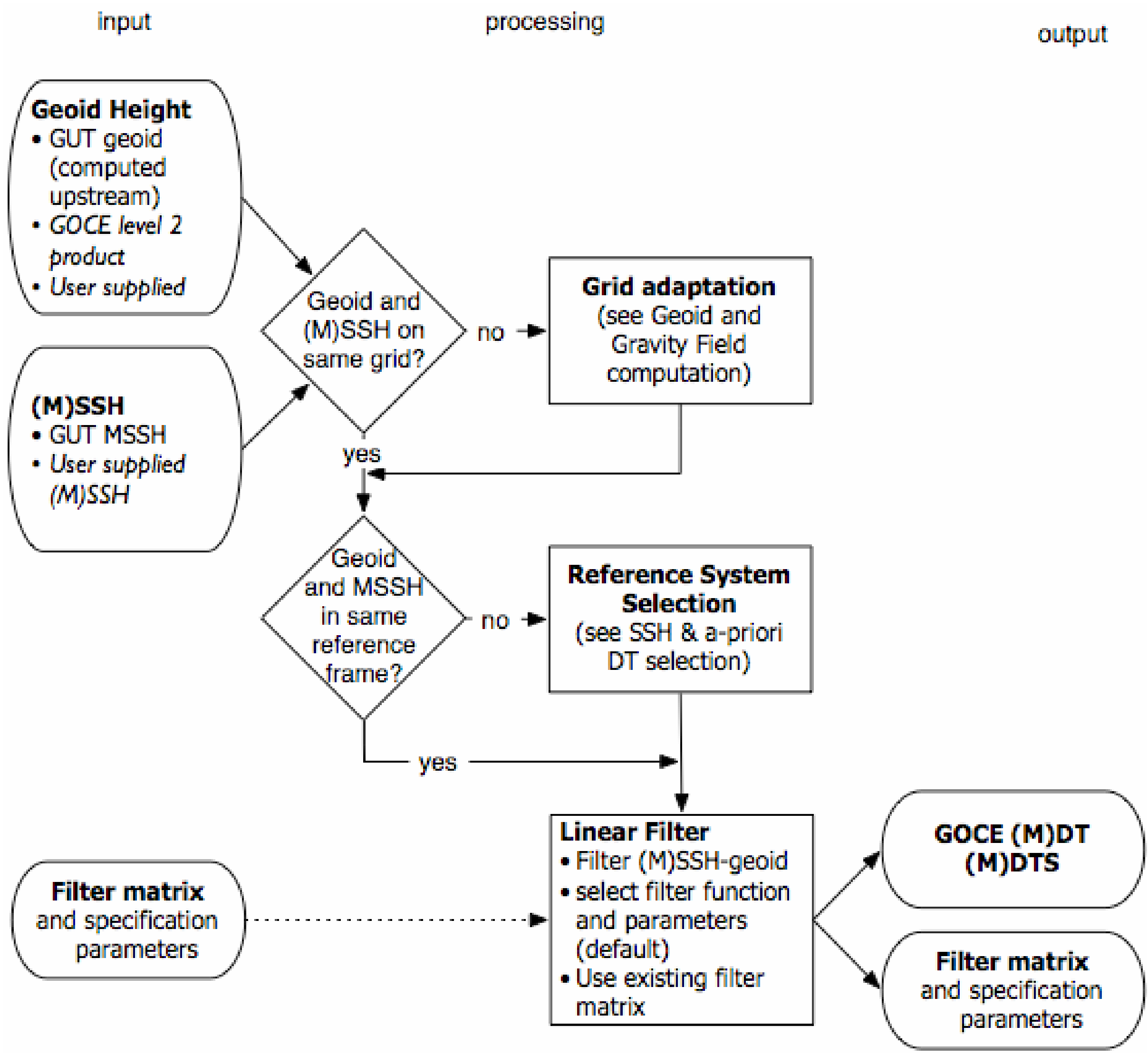


Figure 16: Workflow 3a: Satellite Dynamic Topography computation in geographical space

The output of the filter algorithm includes not only the filtered MDTS but also the provision of the filter matrix, which can be used for subsequent computations on the same grid and is especially convenient when time series of Dynamic Topography are calculated. However, for high resolution, global fields, memory requirements might exceed available resources. Thus, a system-dependent parameter is needed that specifies the upper limit of the matrix dimensions to allow the saving on disk.

Spectral MDTS (default for global regular grids)

To use the spectral method the MSSH field has to be provided as a global field. Gaps over land have to be filled before invoking the transformation to SH coefficients. A default global geoid field is provided by GUT for that purpose. The analysis is then performed depending on the selection of maximum degree and order of the spherical harmonics expansion. The determination of the (unfiltered) MDTS is then computed as the difference of spherical harmonic coefficients of the MSSH and the geoid.
The MDTS can then be filtered using either one of the spatial linear filters listed in section 0 above, or one of the supplied spectral linear filters:

- Pellini filter (spherical cap in ”geographical space”)

- Gaussian/quasi-Gaussian (=Jekeli)

If a spatial filter is selected, the MDTS is transformed into physical space before filtering while the spectral filters operate on the SH coefficients and gridding is done afterwards.

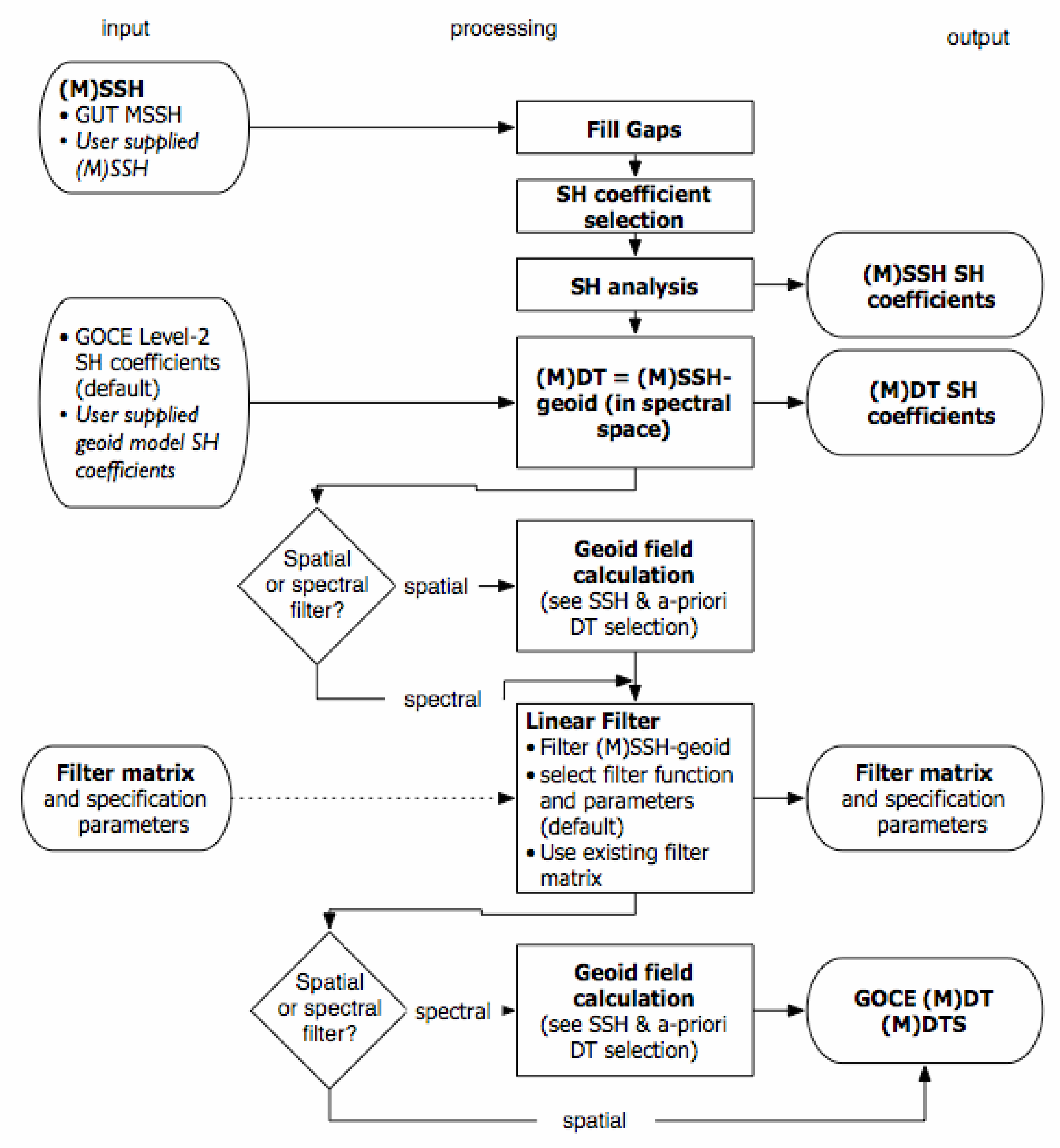


Figure 17: Workflow 3b: Satellite Dynamic Topography computation in spectral space

The output of the filter algorithm includes not only the filtered MDTS but also the filter matrix, which can be used for subsequent computations on the same grid and is especially convenient when time series of Dynamic Topography are calculated.

3.3.5 Remove-Restore "Combined" techniques

Two variants of a Remove-Restore "Combined" technique are included in GUT. The first utilizes a high-resolution a-priori MDT, e.g. from hydrodynamic modelling or observations, to restore the small-scale structure in the 'satellite' MDTS. This can be defined as:

$$\eta_C = \bar{\eta}_S + (\eta_a - \bar{\eta}_a)$$

Taking the notation that a is the a-priori, C is the combined solution and S is the satellite solution, with the overbar denoting a filtered solution. The filtered satellite solution here will be the output of the previous MDTS calculation and the filtering can be spatial or spectral – but the filtering of the a-priori MDT must be carried out in the same way.
The second variant takes the a-priori MDT as the basis and restores the large-scale structure by comparing the spectral equivalents of an a-priori geoid (based on the filtered difference of MSSH and a-priori MDT) and the GOCE geoid.

$$\eta_C = \eta_a + \left(\overline{\eta_G - \eta_a}\right)$$

This requires that we use the *unfiltered* version of MDTS (*i.e.* direct difference of MSSH – Geoid).
For both variants, as for the MDTS calculation, the filtering required can be carried out spatially or spectrally. For practical purposes, it is recommended that user remains within the same filtering space. Particularly for the spectral options, this will mean that the SH analysis of the MSSH field will already exist.
These two variants can be used for different purposes. Method A puts higher priority on the MDTS fields and assumes the high resolution features of the a-priori MDT are consistent with MDTS. The second method puts higher priority on the a-priori MDT and would be appropriate (*e.g.*) when using an ocean model for the a-priori to provide an improved model surface suitable for data assimilation fields, that was consistent with the ocean model dynamics and the GOCE geoid.

Remove-Restore Combined technique A

To improve the small-scale structure in the MDTS, a global a-priori MDT is included in GUT. The users might provide their own, global or regional, fields for their specific applications.
The MDT correction is determined as the difference of the filtered a-priori MDT from the unfiltered field. The filter should have the same specifications (filter function and filter width) as the one taken for the GOCE MDTS. If the GOCE MDTS grid diverges from the a-priori grid, the grids have to be adapted and the filter matrix has to be determined within the filter algorithm. Otherwise, the available filter matrix used when computing the GOCE MDTS (see section 3.3.4) is applied. If grid adaptation is necessary, the GOCE MDTS grid is applied for the a-priori MDT by default. If one or both MDT fields are regional, the user has to specify the region over which to apply the MDT correction.
The MDT correction is added to the GOCE MDTS resulting in the Combined MDTC.
The workflow for this variant using spatial filtering is given in Figure 18 and that using spectral filtering is given in Figure 19.

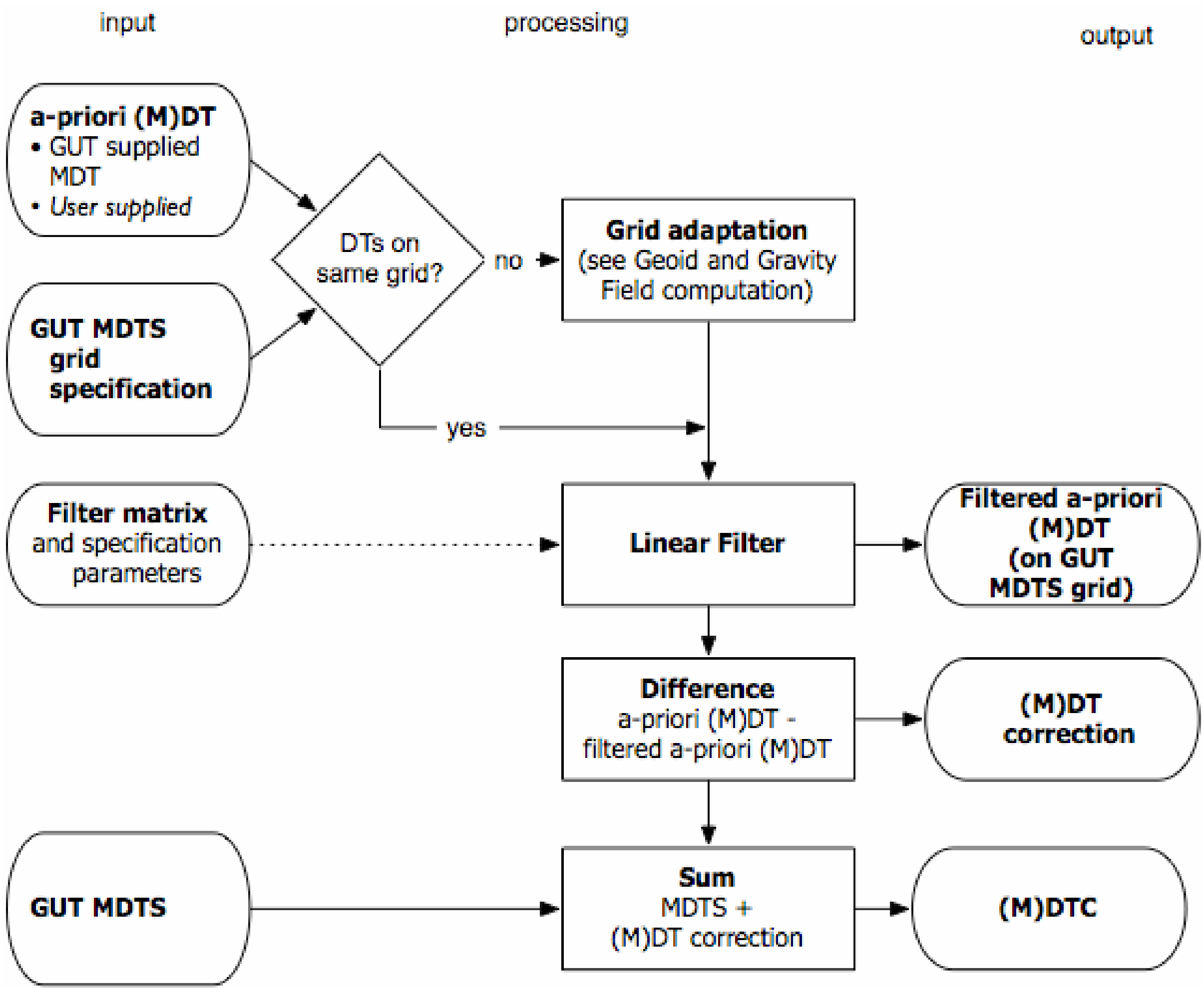


Figure 18: Workflow 4a: Remove-Restore combined technique A: spatial filtering

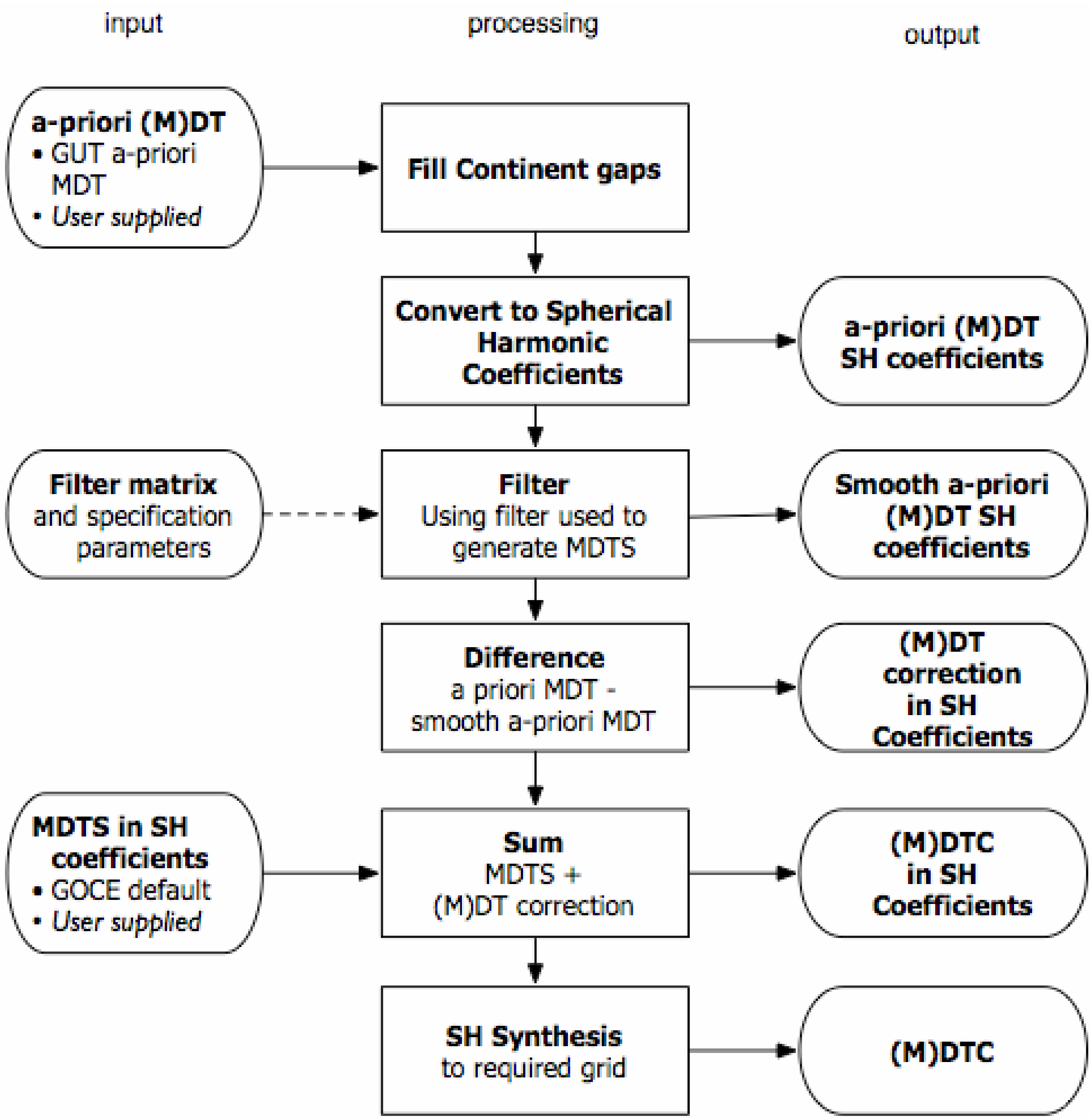


Figure 19: Workflow 4b: Remove-Restore combined technique A: spectral filtering

Remove-Restore Combined technique B

The second variant of the Remove-Restore technique is based on the comparison of an a-priori geoid with the GOCE geoid to improve the large-scale structure in the a-priori MDT. The a-priori geoid is the filtered difference of MSSH and the a-priori MDT. The method is not using the GOCE MDTS as computed in section 3.3.4 and is, in that sense, an independent way of computing a Combined MDT.

Because of the expansion into spherical harmonic functions, to use the spectral version of this technique not only the MSSH but also the a-priori MDT (or rather a combination of the two) has to be provided as a global field and gaps on land are filled automatically. The effects of coastal discrepancies between the ocean fields and the fields used to fill the continent areas can cause errors that propagate into the ocean interior and how to merge the fields at the coast is still an open science question. The remove restore techniques help to alleviate some of the resulting errors.

For the spatial filtering version (Figure 20), in most cases the MSSH and the a-priori MDT will not be provided on the same grid. The user has to specify a common grid. By default, the MSSH grid is taken. The a-priori MDT is then subtracted from the MSSH to result in the a-priori geoid over the oceans. The geoid is then also subtracted to give the difference between the GOCE and "a-priori" fields. The filter is needed to suppress the small scales in the MSSH that are not present in the a-priori MDT but has to retain the large-scale variability up to wave numbers where the GOCE geoid can be used to improve the a-priori MDT. Recommendations are given on the length scale and the filter function.

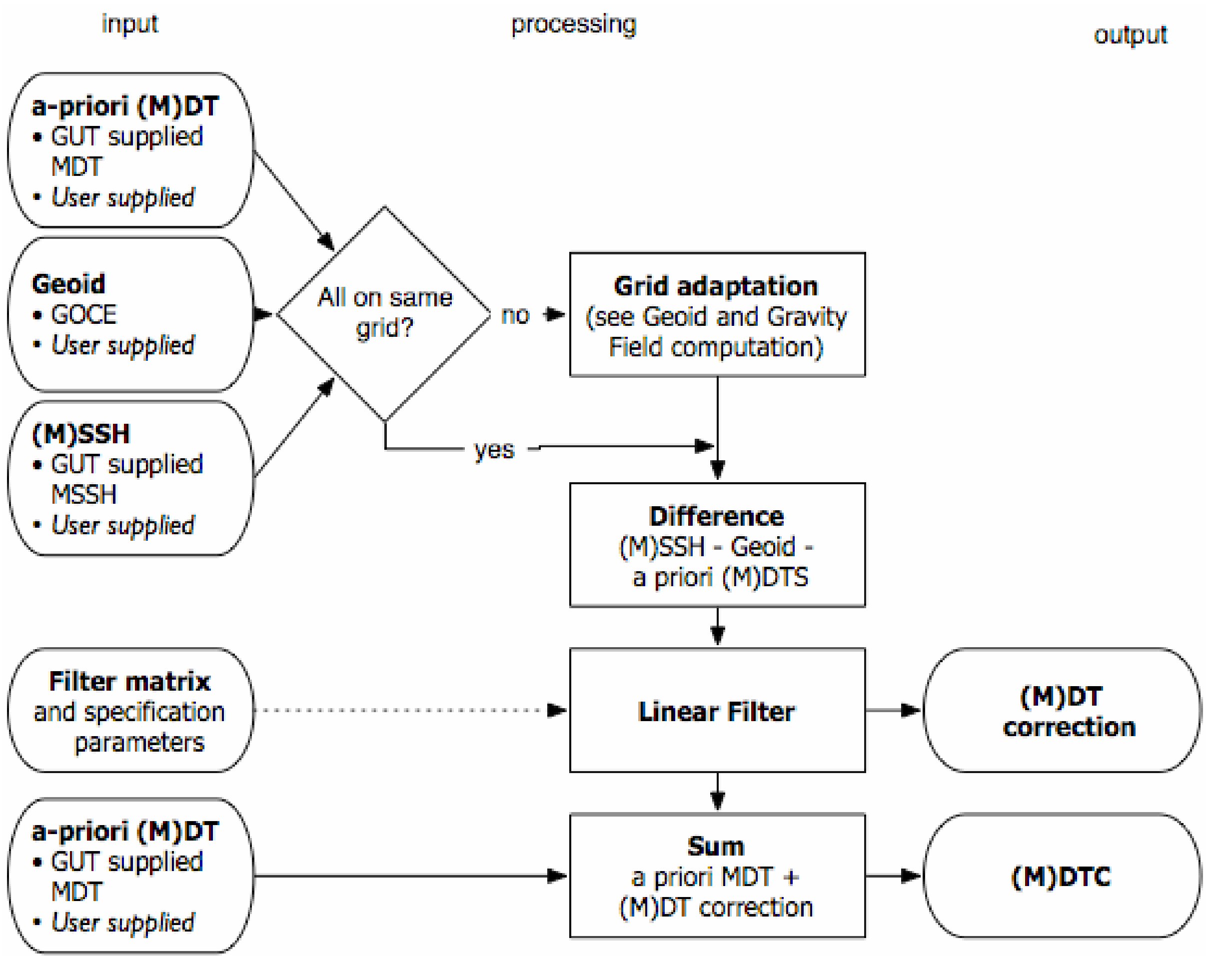


Figure 20: Workflow 5a: Remove-Restore combined technique B: spatial filtering

In the spectral version, (Figure 21), the a-priori MSSH and MDT are combined and then expanded into spherical harmonic functions. The difference of the MSSH and MDT in SH coefficients are subtracted from the Geoid SH coefficients (the GOCE Level 2 product, or user provided coefficients). The maximum degree and order, up to which the correction should be used, should be specified by the user. The correction can be transformed to physical space and added to the gridded a-priori MDT resulting in the Combined MDTC. Alternatively, the correction is added to the SH representation of the a-priori MDT and the result is transformed to physical space.

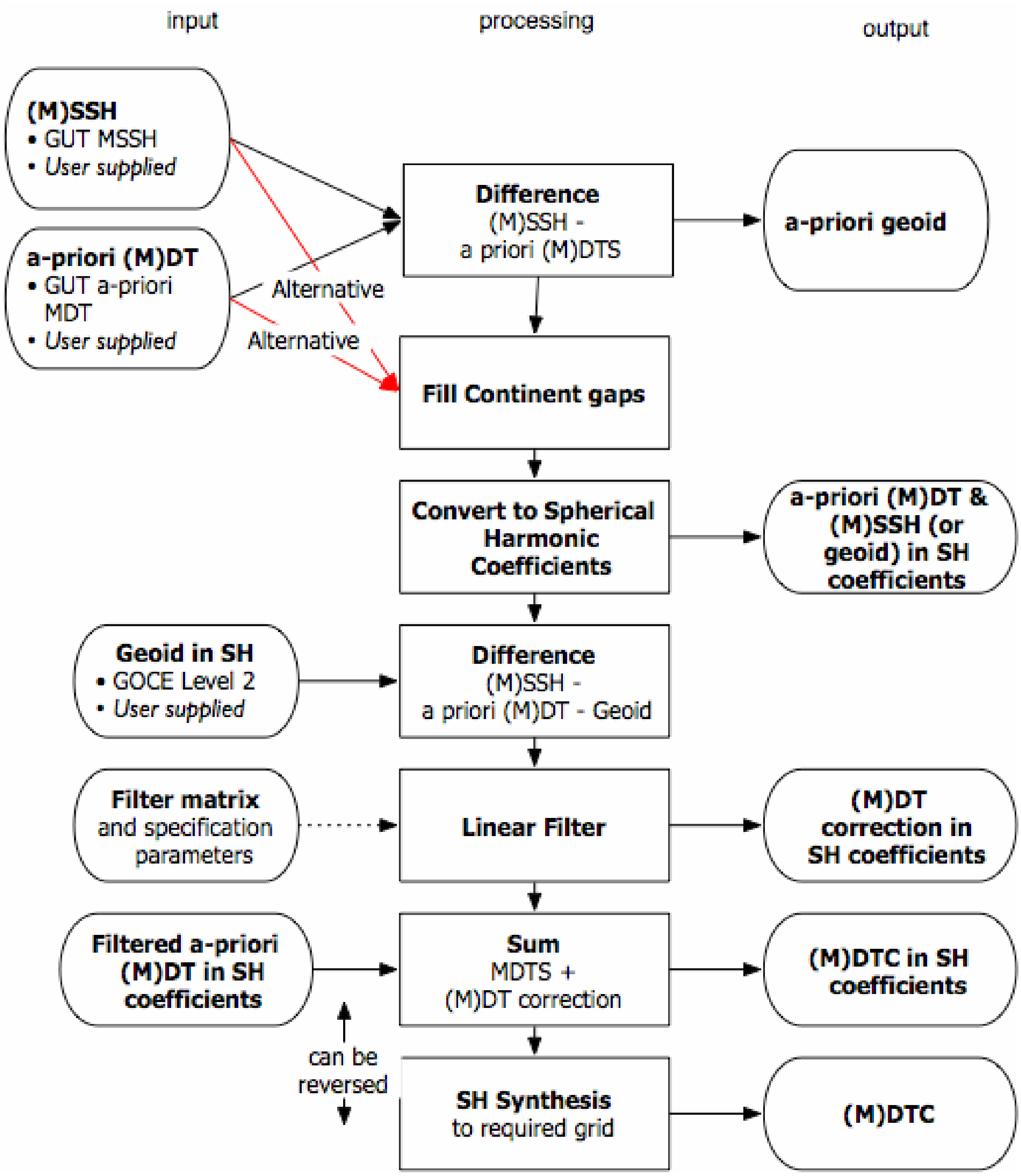


Figure 21: Workflow 5b: Remove-Restore combined technique B: spectral filtering

3.3.6 Dynamic Topography-derived quantities

The calculation of surface geostrophic currents is done in two steps. First the slope of the surface is determined. The slope calculation can actually be used for all 2D fields that are provided on a grid. The second step is then the calculation of the surface geostrophic currents from the slope of the MDT.

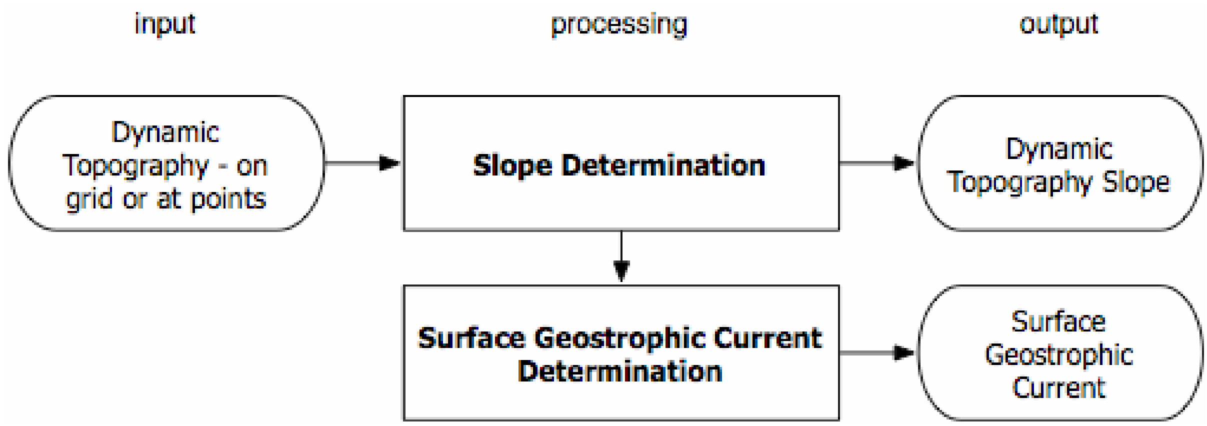


Figure 22: Workflow: Dynamic Topography-derived quantities

Additional use of the dynamic topography outputs will be facilitated by ingestion of GUT output into the BRAT toolbox.

3.3.7 Pre-viewing function

The pre-viewing functionality is intended as a means of rapidly assessing if a result from the toolbox is suitable, or to derive an approximate view of the difference caused by changes in the processing through the toolbox. As such, the functionality is expected to be limited to simple viewing of 2-dimensional fields as contour and surface plots and 2 dimensional graph plots.
It is anticipated that the necessary functionality can be obtained using the VTK package in combination with additional libraries developed for previous ESA toolboxes.
It is anticipated that the user will export results from the toolbox to make use of more sophisticated viewing and printing options available within packages such as MATLAB or GMT.
A global land-sea mask should be provided as part of the previewing function.

3.4 Algorithm specification

The algorithms as defined in this section are based on the fundamental algorithms and formulae defined in the User Requirements Document (URD) [1], composited to be consistent with the toolbox functionality as defined above. Each algorithm corresponds to a "processing" block in the workflow definitions. The basic algorithms to be implemented within each defined function are given. Where existing software exists that provides the required functionality, the software is also defined.
For each function, the input is divided into *options*, *input data* and *output data*. Some of the required input data are to be provided with the toolbox as default data sets, such as the GOCE level 2 data sets and a global a-priori MDT. Those data sets are indicated as *GUT default data*. By choosing the default options the functionality as defined in the main workflow given in (Figure 12) can be fully accessed by the user without providing additional data. When using a different option setting, the user may be required to provide additional data. However, each data set provided with GUT can be replaced by the user's own data using a format supported by the toolbox.
The potential sources of these parameters and the final output file format specifications are given in the Input/Output Definitions.
The algorithms are grouped by type. The first section: "Algorithms for Preference Selection" are required to provide parameters to other toolbox functions but do not include basic algorithms defined in the URD. The second section, "Computational Algorithms", generate the primary toolbox products, using the functionality of the toolbox as defined in Appendix B.

4 System Specifications and Architectura Design

4.1 Overview of GUT Design

This section gives an overview of the toolbox design, with justifications for the key design decisions. The core scientific algorithms are implemented as C and Fortran functions, mostly derived from existing software with the minimum amount of modification. All the existing software being considered for use in GUT at the time of writing is written in Fortran, but the capability for incorporating C will also be maintained. Both languages are appropriate for this purpose because they are free from software licensing costs and are widely used by scientists, maximising access for members of the scientific community involved in oceanography and solid earth studies. The C and Fortran scientific codes are managed using Python, which also provides the main user interface to GUT. The GUI provides access to some or all of the GUT functionality available through the Python interface. Detailed GUI design will be carried out during the implementation phase of GUT development.

There are several advantages to using Python for GUT implementation, as described below.

- Runs on many platforms, including Linux, Windows, MacOS and Solaris
- Provides a powerful, user friendly command interface with scripting ability
- Easily extensible by adding new C and Fortran based functionality
- Provides powerful mathematical capabilities [30]
- Considerable user interaction and programming capability will be required for GUT, more so than for previous ESA toolboxes. Python is highly suitable for such requirements

Python has many useful features but execution speed is not one of them. To achieve acceptable performance, all of the computationally expensive aspects of GUT processing will take place in C and Fortran routines rather than in Python. This approach has been used successfully in other scientific software toolboxes. Two popular Python-based toolboxes are described briefly below.

- VISAN [3,7], the GUI for the Basic Envisat Atmospheric Toolbox (BEAT). According to the developers, the BEAT project "aims to provide scientists with tools for ingesting, processing, and analysing atmospheric remote sensing data." VISAN is used for browsing and plotting data, and incorporates a Python command prompt providing access to the full range of BEAT functionality via the Python application program interface (API).
- Climate Data Analysis Tool (CDAT). CDAT [33] is used for analysis of gridded data in a variety of different formats. Typical operations include plotting, aggregation of multiple files, differencing data sets, calculating departures from a climatology, re-gridding and calculating the mean variance and covariance of a variable.

VISAN is described in its documentation as a "visualization and analysis application for atmospheric data". Access to the atmospheric data is provided by BEAT, and the analysis capability is provided largely by Python, which allows the user to write their own data analysis and processing scripts. In GUT, the emphasis is on the data processing rather than the data access and visualisation, though GUT will have to be able to do these things as well. In other words, GUT users will not be expected to write their own Python scripts for using the GOCE products, because these scripts will be an important part of GUT itself. Of course, users will be able to write scripts and programs in Python and other languages to augment GUT capabilities.

Python is not the only way to implement GUT, and it is possible that the important features of the design specified in this report could be implemented in another way. However, if alternative solutions are evaluated during the implementation phase it is important to consider the key features of the protective *user environment*, which facilitates good science and improves the traceability of results. The user environment consists of the following elements:

- Command interface
- Relationships between *logical data structures*, which include algorithm parameters, spatial and spectral fields, filter matrices and error covariance matrices
- Rules governing execution of scientific work-flows
- Session management
- Internal Data Store (IDS)

All the elements of the user environment are described in more detail in the following sections.

4.2 Toolbox Components

4.2.1 Computationally Expensive Tasks

The parts of GUT that process large data structures and carry out mathematical calculations are written in Fortran and C. These computationally expensive tasks are described by a series of Processing Units, which are defined in Section 8 ("Logical Model of System Functionality"). The Processing Units include data pre-processing tasks and scientific calculations. Existing software will be re-used in the implementation of the Processing Units wherever possible. The scientific software being considered for re-use in GUT is described in the report from WP3000 [26]. All the software under consideration at present is written in Fortran, but the facility to incorporate C/C++ into GUT will be retained to ensure flexibility in the future. Software re-use is discussed in more detail in Section 4 ("Re-use of Existing Software").

4.2.2 Management of Internal data store

GUT maintains its own internal data store (IDS) on disk. The mathematical functions involved in the scientific work-flows take their input from, and write their output to the IDS. All the spatial data in the IDS is represented on the same set of spatial positions (e.g. a grid), and the reference system is common to the spatial and spectral representations of the data. Where spatial and spectral representations of the same field are both present (as occurs during execution of some of the scientific work-flows) they are consistent with each other. Data structures for the parameters, input data, intermediate stages and final output from scientific work-flow calculations are part of the IDS, which is separate from the external files where the data originates. This may seem like unnecessary duplication of spatial data, but the IDS is necessary because imported data may undergo transformations to make it consistent with data already present. Keeping the transformed fields permanently avoids lengthy transformation tasks being repeated each time a scientific work-flow is executed. The error covariance matrix of the SH coefficients of gravity potential is not part of the IDS; no transformations need to be performed on this large matrix, so there is no reason to create a copy in NetCDF format.

The supported formats for data import and export are discussed in Section 6 ("Specification of Input and Output"), which includes details of existing data handling software that may be suitable for re-use in GUT. Low level data import and export routines from some existing software packages can be re-used in GUT for dealing with a range of file formats.

4.2.3 Management of High Level Workflows

The high-level scientific work-flows are managed by Python software. Work-flows are executed by means of a sequence of calls to non-Python mathematical subroutines, which interact independently with the internal data store. These non-Python subroutines will be used to implement the Processing Units defined in Section 8 ("Logical Model of System Functionality"). Experts' scientific knowledge is encapsulated by the rules governing the sequences of events, and in the relationships between the logical data structures involved in the work-flows. Subroutines from other languages can be incorporated in the Python language by creating Python extensions [29]. It may not be practical to incorporate all existing software in this way. In cases where this is not practical, Python can also be used to manage binary executables in a way that is similar to a conventional shell scripting language. The Processing Units must be able to access data on the hard disk themselves, because it may not be possible to store all the data they need in memory simultaneously on all computers where GUT will be running. The default grid has 1/30 degree resolution, and some operations, such as grid differencing, have to deal with two gridded fields at the same time. Spatial filter matrices can also be very large.

4.2.4 Command Interface

Python is suitable as the basis for a command interface to GUT because it allows Python statements to be issued interactively at a command prompt. The command interface is described in Section 5.1 ("Command Interface").

4.2.5 Graphical Components

A comprehensive GUI can be built on top of the Python application using Python wrappers for wxWidgets [35], an Open Source suite of application programmers' tools that formed a key component of the BEAT and BRAT toolbox GUIs. The GUT visualisation capabilities should be based on VISAN or BRAT, both of which use the Visualisation Toolkit [37] (VTK) for producing images.

4.3 Re-Use Of Existing Software

This section discusses how existing software will be incorporated into the Python application. The two main areas of GUT where existing software can be incorporated are data handling (mainly import and export) and scientific algorithms. Existing software for data handling is discussed in Section 6 ("Specification of Input and Output").

As discussed earlier, the computationally expensive tasks described by the Processing Units defined in Section 8 ("Logical Model of System Functionality") will be implemented as Fortran or C subroutines. Several programs containing Fortran code that can be re-used for this purpose have been identified in the WP3000 report. Details of the Fortran programs (if there are any) relevant to each Processing Unit can be found in the WP3000 report in the discussion of the corresponding Functional Algorithm. The numbering of the Processing Units is aligned with the numbering of the Functional Algorithms. WP3000 has established that there is re-usable Fortran code suitable for implementing many of the Processing Units, but there are still a lot of gaps that will need to be filled by new software written in C and Fortran.

Two different approaches to re-using software have been suggested for GUT:

1. Re-use large portions of existing programs or whole programs
2. Re-use of some of the *algorithms* from existing software, possibly taking copies of small subroutines or loops into GUT.

The second approach would involve re-writing all the scientific software in Fortran or C. This is the best solution from the software engineering perspective, and also allows greater flexibility for extending the capabilities of GUT in the future. Consideration of software at the level of small subroutines is a task that will take place during the software design stage of GUT implementation. Therefore, it is beyond the scope of this document and will not be discussed further. The rest of this section will outline a solution involving the re-use of large portions of existing software or whole programs.

Re-using large portions of existing programs or whole programs is a tempting approach because many of the re-usable Fortran programs identified in the WP3000 report perform tasks that fit in well with the tasks performed by individual Processing Units, particularly PU04. Some existing programs provide the functionality of more than one Processing Unit. All existing software should be incorporated into GUT as Python extensions that interact with the internal data store independently, as discussed earlier in Section 3.1 ("Computationally Expensive Tasks"). This will mean that all existing software will need to be modified to a certain extent. All the existing input-output routines will need to be re-written in order to exchange data with the internal data store instead of their own file formats. In cases where a section of an existing program is taken without self-contained input and output functionality, input and output functions will need to be written from scratch. Python code should replace any functionality contained in operating system shell scripts that accompany the existing software, and code modifications may be required during the process of creating Python extensions.

4.4 User Interfaces

4.4.1 Command Interface

Introduction

The GUT command interface is an element of the user *environment,* which contains the input, output and parameters associated with scientific work-flows. The basic usage procedure involves three main stages.

1. Set up the parameters governing the operation of a particular scientific work-flow
2. Import the necessary data
3. Enter a command to calculate the desired output product

This integrated user environment is a little bit like having your input files, output files and parameter files all in the same directory, but the GUT user environment is more sophisticated than this conventional approach. The most important feature of the GUT environment is that the input, output and algorithm parameters are consistent with each other at all times. This means that the input data files and algorithm parameters do not have to be specified in GUT commands, and all the output can be reproduced exactly without knowing the command that was entered by the user to initiate the calculation. In other words, as a GUT user you can always see where your results came from if the internal data store is retained. Other features of the GUT user interface are described below.

- Users do not interact directly with data and algorithm parameters. Instead, GUT functions called *methods* are used. This ensures that the data and parameters are kept consistent, as described above, and enables the user to be informed about the consequences of important changes they make. Methods are described later in this section.
- Data associated with intermediate stages are saved during execution of work-flows
- A path through a particular work-flow can be executed in full with a single command, or executed in stages with a series of commands.
- The user environment can be saved to disk, and loaded again at a later time.
- The amount of typing is minimised by the use of a name completion facility, which is analogous to name completion in Unix and Linux shell environments. The user can select from a list of all possible commands matching any partially entered command.
- Experts' scientific knowledge is encapsulated in the constraints placed on input data and parameters and the relationships between them.

The Python language can be used to provide a user interface with the above features. One of the unusual features of the language is that it allows statements to be entered interactively at a command prompt, in the same way as operating system shell languages such as the C Shell or the Bash Shell. This makes python suitable as a command interface for applications such as GUT. Using GUT is not like writing a program (or as little like writing a program as possible), but users must have a basic knowledge of programming to be able to understand the Python command syntax. In the following sections, no knowledge of *object oriented* programming methods is assumed, but some object oriented programming terminology is used where appropriate. In the remainder of this report the "`Courier New`" font is used to highlight elements of GUT commands.

Python language elements used in GUT

This section describes the main Python language elements in GUT. The high level scientific work-flows are controlled by manipulating Python language elements called *objects*. An object in GUT consists of the following things:

- *Attributes*, which describe the object and its contents
- *Data*, including input data, results from calculations and parameters for algorithms
- *Methods*. These are things that the object can do

An Object in Python is somewhat similar to a compound variable in a procedural programming language like C. A C *structure* is an example of a compound variable; it is a container for two or more variables, which may be instances of monotonic data types such as integers, arrays, strings or even other structures. The Python object extends this concept by allowing members of the compound variable to be *methods*, which are known as functions or subroutines in other programming languages. The user interacts with GUT by entering commands to execute methods. The methods are used to manipulate, or perform calculations involving the logical data structures belonging to the Objects. The way in which the user interacts with the Objects is described in the Appendix ("Using the command interface"). The five types of Object in GUT are described below.

Main Objects

The most important feature of GUT is called the *Main Object*. A Main Object is a container for everything associated with the scientific work-flows, including the input and output data, algorithm parameters and the algorithms themselves. The only reason why GUT users have to know about Main Objects is that there can be more than one of them present in the user environment during a user session. There are many advantages to this approach. For example, it makes it easy to compare the same output field calculated in different ways or on different grids. However, not all users will find this facility useful, and it can be safely ignored. It is sufficient to understand the following three things about Main Objects in GUT user sessions:

- There must be at least one Main Object.
- It must have a name given to it by the user.
- It can be saved to disk, and loaded again during a GUT session at a later date.

At the start of a GUT session the user must either *create* a new Main Object or *load* a saved one from disk. All the information necessary to repeat any results stored in the logical data structures in the Object is contained within the internal data store belonging to the Object. This allows the relevant calculations to be repeated exactly at a later date if necessary. At the end of a user session the Main Object and its internal data can be saved to disk as a whole, and data from individual fields can be exported to files. The attributes and data are consistent with each other at all times. In addition to its attributes and methods, a GUT Main Object contains four sub-objects, named `Spectral`, `Covar, Filter` and `Spatial.` The five object types represent logical groupings of data structures, parameters and functionality, designed to make the user interface easier to use. Generally speaking, the attributes and methods that belong to the Main Object are concerned with the user environment as a whole. The sub-objects are described below. The full specification of their logical data structures can be found in Section 7 ("Specification of Logical Data Structures").

Spatial Objects

A Spatial Object concerns data in geographical space. Data are presented to the user as a series of one dimensional (1-D) vectors. There are two position vectors (one for latitude values and one for longitude values) and vectors for each of the spatial fields involved in the scientific work-flows. Alternatively, the data can equally well be imagined as one big 2-D array or matrix, with each row being a record representing one position on the globe and each column representing a spatial field. This record structure is similar in concept to the GRAVSOFT file record described in the WP2000 report. It does not matter which view of the data is preferred because the user does not interact directly with the logical data structures in the Objects, which exist only to simplify the understanding of the GUT user environment. The actual data structures in the GUT internal data store are likely to be different to either of these conceptual views, but the user does not need to know about what goes on in there. In this report the spatial fields are described as being stored or located in vectors, to make it easier to understand the link between Spatial Object vectors and Spectral Object arrays of the same name. However, the Spatial Object vectors could equally well be described as fields in the GRAVSOFT-like record structure described above.

Spectral Objects

Spectral Objects deal with data in the spectral domain. The Spectral Object contains a set of SH coefficients representing the gravity potential (GOCE L2 product by default) and the spectral representation of all the spatial fields defined in the Spatial Object. The various sets of SH coefficients appear to the user as separate arrays within the Spectral Object. Most of these arrays will not be populated with data during a typical GUT session, because they are not all involved with all the scientific work-flows. The names of the SH arrays are the same as the names of the 1-D vectors in the Spatial Object.

Covar Objects

A Covar Object deals with error covariances. Error covariance matrices can be generated for the geoid height, deflection and gravity anomaly output fields in the Spatial Object. The Covar Object is used for specifying a list of geographical positions to be used for error covariance calculations, and for storing the resulting error covariance matrices. Omission error covariance information is also produced in the form of the evaluation of an isotropic, homogenous error function over a range of distances.

Filter Objects

A Filter Object concerns the filtering of spatial and spectral data. Its local data structures include the filter method specifier, the filter scale parameter and the filter matrix, which is populated the first time a particular filter is used. There are five types of spatial filter and two types of spectral filter methods provided. Only one type of filter can be used during any work-flow calculation, i.e. once the filter method has been chosen and its parameters defined, it remains the same for all the calculations involved in the work-flow, right through to the final work-flow product. If a spatial filter type is selected, the filter matrix is valid for the current grid defined in the Spatial Object and will be removed if that grid is changed, subject to user confirmation. Similarly, if one of the spectral filter types is selected the filter matrix is valid only for the current maximum degree and order of the surface SH coefficients in the Spectral Object.

A Python command in GUT is constructed from several separate components. Most GUT methods are specified with respect to a particular, named Main Object, to allow for the possibility of having more than one Main Object present in a GUT user session. The only exceptions are the methods that are used to create a new Main Object, or to load a saved Main Object from disk. These methods are specified with respect to a Python language element known as a *package*, which could be described as a *module* in some other programming languages. The components that make up a command are described below.

- Name of Python package
- Name of Main Object. This is chosen by the user when creating a new Main Object.
- Name of sub-object, if any. There are four sub-objects in a Main Object, whose fixed names are Spatial, Spectral, Covar and Filter
- Name of method
- Parameters. If a method requires extra information to control its operation, these can be supplied in the form of comma separated parameters inside the parentheses that form part of the method name. If no parameters are supplied, the default action of the method is executed

There are three types of Python command in GUT, depending on which of the above components are used. The formats of these three types of command are described below. Each format description is shown with N parameters, where N can be any number including zero. In other words, the format of commands where no parameters are given is not explicitly shown.

- PackageName.MethodName(Parameter1, Parameter2, …, ParameterN)
- MainObjectName.MethodName(Parameter1, Parameter2, …, ParameterN)
- MainObjectName.SubObjectName.MethodName(Parameter1, Parameter2, …, ParameterN)

A number of pre-defined constants are available to GUT users for use as parameters in commands. In addition to GUT constants, one or more parameters can be specified as `None`, Python's null value. This indicates to GUT that the parameter in question should be ignored. This facility is useful when defining a new grid or reference ellipsoid, because different ways of specifying these involve different numbers of parameters. For example, when defining an irregular grid it is necessary to specify `None` as the value for grid cell width and height because these parameters are undefined for an irregular grid.

In this report, complete method names are always written with trailing empty parentheses, `View()` for example. Users will be able to issue GUT commands interactively at a command prompt, one at a time. Alternatively, command sequences can be saved in a text file and run like an executable program, inside or outside the GUT environment. These command sequences are known as *scripts*. GUT itself will be made up primarily of a series of Python scripts, plus a set of components written in other languages. Further explanation of the use of the command interface is given in the Appendix, with examples of commands relating to the execution of the scientific work-flows.

Full specification of command interface

This section specifies the Python methods used to interact with the logical data structures and control the execution of the scientific work-flows. The methods associated with the GUT package and each type of Object are specified below. The GUT package is not associated with any particular Main Object; its methods are used to bring Main Objects into the user environment by creating a new Main Object or loading a saved Main Object from a directory on disk. When a new Main Object is created, its internal data store is populated by the SH coefficients of the gravity potential and the default MSSH data set, which defines the default grid and reference system. All the attributes are set to their default values. A new Main Object is immediately ready for use with work-flows 1a and 1b, but other work-flows require additional data to be imported first.

In the following tables, the "Example of Usage" column shows one or more typical examples of Python commands involving each method. Pre-defined GUT constants are written in upper case letters. Methods that normally expect parameters in the parentheses can also be used with no parameters; this results in the default behaviour of the method. In the tables below, "default behaviour" of a method means the outcome if no parameters are specified.

GUT Package

Name	Example of Usage	Description
`New`	`New("/home/GUT/main_ob1")`	Creates new Main Object with default attributes and data in directory main_ob1

Load	Load("/home/GUT/main_ob1")	Loads a previously saved Main Object and its internal data store from directory “main_ob1”

Main Object

Most of the Main Object methods are for performing calculations relating to the scientific work-flows defined in WP3000. The names of these methods are derived from the names of the logical data structures provided for storing the intermediate stages and final output from the work-flows. The output from the work-flows is always provided in geographical space, and the results can be found in Spatial Object vectors. Every work-flow product has a unique name that identifies the method that was used to produce it. In the case of work-flows involving intermediate stages in geographical space, the intermediate stages can be found in Spatial Object vectors with names that include an additional identifier beginning with “-Inter”. In the case of work-flows involving data in spectral space, intermediate stages and some final output can also be found in arrays located in the Spectral Object. There is a `Calc` method corresponding to every field in the Spatial Object that is not an input field. Input fields are identified with names beginning with “Input”. It is important to note that all the `Calc` methods in the Main Object refer to fields in the Spatial Object, even though intermediate stages and final output from some of the work-flows also ends up in the Spectral Object. `Calc` methods for "-Inter" fields are provided to allow the user to experiment with one particular part of a work-flow, without incurring the computational cost of going right through to the end product each time. Some `Calc` methods result in output to more than one field. `CalcVelocity()` and `CalcDeflection()` calculate both the northward and eastward components of geostrophic current and gravity field deflection. The methods for calculating error covariances each produce a covariance matrix for a pair of points and the evaluation of an error function for a range of distances.

The table below specifies the methods associated with the Main Object. Most of the methods shown in the table refer to the Main Object attributes. The `Calc` methods for intermediate stages and final output from the scientific work-flows are not listed, to avoid repeating all the data structure names defined in Section 7 (“Specification of Logical Data Structures”). The only `Calc` methods that need mentioning specifically are those involved with Remove-Restore work-flows 4a, 4b, 4c and 4d. The 4a and 4c work-flows are only valid if a spatial filter is specified in the Filter Object, and the 4b and 4d work-flows are only valid if a spectral filter is specified. Attempting to perform calculations involving these work-flows when the wrong type of filter is specified results in an error message.

Name	Example of Usage	Description
Annotation	Annotation("Some text")	Defines ASCII string of user's comments. Default behaviour: Display string
ImportAnnotation	ImportAnnotation("file.txt")	Imports annotations from a text file, including new-line characters. Default behaviour: Try to import from default path.

`Info`	`Info("Ellips")`	Displays the value of the specified Main Object attribute Default: Displays values of all attributes
`Ellips`	`Ellips(GRS80)`	Changes all ellipsoid parameters. Parameters can be specified explicitly by providing them in the following order: $GM, a, \gamma a, f, J_2, \omega$ One or more of them can be None (undefined). Alternatively, one of three constants can be given as a single argument: GRS80, TOPEX or ENVISAT Default Behaviour: The default ellipsoid is used (TOPEX). Adapts internal data dependent on these values if possible and discards data that can not be adapted
`TideSys`	`TideSys(`TIDE_FREE)	Changes value of attribute. Adapts internal data dependent on this value if possible and discards data that can not be adapted
`Save`	`Save("main_path")`	Saves Main Object attributes and data in directory specified by user
`Load`	`Load("main_path")`	Loads a previously saved Main Object from a directory specified by user.
`Calc`	`Calc()`	Starts user dialog session in which user selects desired calculation from a list.

Spatial Object

In addition to methods for changing the values of attributes, the Spatial Object also provides sets of methods with names beginning with "`Convert`", "`Filter`", "`View`", "`Import`", "`Export`" and "`Delete`". These sets of methods are described below.

- The `Filter` methods are used for filtering spatial fields according to the filter parameters defined in the Filter Object. These methods are not used for controlling any of the scientific work-flows, and they are not shown in the table below. The `Filter()` method launches a wizard that allows the user to select the desired field from a numbered list.
- The `View` methods are used to display 2D plots of the spatial fields, and the `View()` method launches a wizard that allows the user to select the field and parameters controlling the type of plot required.
- The `Convert` methods are used for converting spatial data to spectral space. The output of a Spatial Object `Convert` method is an array with the same name as the converted Spatial Object vector. For example, `ConvertMDTC_B-Spectral()` converts the

gridded data in the MDTC_B-Spectral vector to SH coefficients by SH analysis, the result of this conversion going into Spectral Object array MDTC_B-Spectral. The maximum degree and order of coefficients used for SH analysis is governed by Spectral Object attribute MaxDegOrdSurface. The `Convert` methods are not used for controlling the scientific work-flows. The names of the `Convert` methods are derived from the names of the Spatial Object vectors, which are specified in Section 7 ("Specification of Logical Data Structures"). The `Convert` methods are not specified in this table, except for the `Convert()` method (with no reference to any particular spatial field), which launches a command based wizard.

- There is an `Import` method for the `Annotation` attribute and every spatial field vector. These methods are used for importing data from external data sources into the internal data store of the Main Object. These methods are not shown in the table below, to avoid repeating information from Section 7. The `Import()` method launches a wizard. Importing data is discussed in more detail in Section 6 ("Specification of Input and Output").
- There is an `Export` method and a `Delete` method corresponding to every spatial field in the Spatial Object. These methods are not shown in the table, to avoid repeating information. The `Export` methods are used to write spatial data from the internal data store to external files, and the `Export()` method launches a wizard. Data export is described in more detail in Section 6 ("Specification of Input and Output"). The `Delete` methods are provided to allow the user to manage the internal data store, to allow a Main Object to be used for calculations involving more than one work-flow without retaining all the input, output and intermediate stages from previous work-flow calculations. Deleting unnecessary fields may be desirable in situations where disk space is limited. The `Delete()` method launches a field deletion wizard

Name	Example of Usage	Description
`Annotation`	`Annotation("Some text")`	Defines ASCII string of user's comments. Default behaviour: Display string
`ImportAnnotation`	`ImportAnnotation("file.txt")`	Imports annotations from a text file, including new-line characters. Default behaviour: Try to import from default path.
`Info`	`Info("Ellips")`	Displays the value of the specified attribute Default: Displays values of all attributes
`GridCalc`	`GridCalc(POINT)`	Changes value of attribute. Adapts internal data dependent on this value if possible and discards data that can not be adapted

`DefineGrid`	`DefineGrid(lat1, lon1,….)`	Specifies grid parameters. Adapts internal data dependent on these values if possible and discards data that can not be adapted
`New`	`New()`	Overwrites Spatial Object with default attributes and data
`Blank`	`Blank()`	Overwrites Spatial Object with a blank object containing no data, with all attributes set to `None`
`MaxDegOrdPotential`	`MaxDegOrdPotential(200)`	Changes value of `MaxDegOrdPotential`. User specifies DO value. Adapts internal data dependent on this value if possible and discards data that can not be adapted
`MaxDegOrdSurface`	`MaxDegOrdSurface(200)`	Changes value of `MaxDegOrdSurface`. User specifies DO value. Adapts internal data dependent on this value if possible and discards data that can not be adapted
`MaxResKMPotential`	`MaxResKMPotential(100)`	Changes value of `MaxDegOrdPotential`. User specifies resolution in Km. Adapts internal data dependent on this value if possible and discards data that can not be adapted
`MaxResDegreesPotential`	`MaxResDegreesPotential(0.5)`	Changes value of `MaxDegOrdPotential`. User specifies resolution in degrees. Adapts internal data dependent on this value if possible and discards data that can not be adapted
`MaxResKMSurface`	`MaxResKMSurface(100)`	Changes value of `MaxDegOrdSurface`. User specifies resolution in KM. Adapts internal data dependent on this value if possible and discards data that can not be adapted

MaxResDegreesSurface	MaxResDegreesSurface(0.5)	Changes value of MaxDegOrdSurface. User specifies resolution in degrees. Adapts internal data dependent on this value if possible and discards data that can not be adapted
Filter	Filter()	Launches a wizard for filtering a spatial field
View	View()	Launches a wizard for producing a 1D or 2D plot of a spatial field
Convert	Convert()	Launches wizard for converting to spectral space
Import	Import()	Launches data import wizard
Export	Export()	Launches data export wizard
Delete	Delete()	Launches a wizard for deleting unused spatial fields in the internal data store

Spectral Object

Any of the 2D arrays of SH coefficients in the Spectral Object can be converted to geographical space using a `Convert` method. The names of the Convert methods are derived from the names of the 2D arrays in the Spatial Object. For example, `ConvertMDTC_B-Spectral()` converts the SH coefficients in the `MDTC_B-Spectral` array to geographical space by SH synthesis, the result of this conversion going into Spatial Object vector `MDTC_B-Spectral`. The maximum degree and order of coefficients used for SH synthesis is governed by Spatial Object attribute `MaxDegOrdSurface`.

There is an `Import` method corresponding to the Annotation attribute, and every array of SH coefficients. This allows the user to supply their own data in spectral form as well as spatial data. The `Import()` method launched a wizard. A set of `Export` methods are provided to allow the user to export data from any of the spectral fields. There is an `Export` method corresponding to all the Spectral Object arrays except for InputPotential. The `Export()` method launches a wizard. Data import and export are discussed in more detail in Section 6 ("Specification of Input and Output"). A set of `Delete` methods are provided to allow the user to delete unused spectral fields in order to save disk space. The `Delete()` method launches a wizard. To avoid repeating all of the field names defined in Section 7 ("Specification of Logical Data Structures"), the `Convert, Import, Export` and `Delete` methods are not shown in the table below.

Convert	Convert()	Launches wizard for conversion of data to geographical space
Annotation	Annotation("Some text")	Defines ASCII string of user's comments. Default behaviour: Display string

ImportAnnotation	ImportAnnotation("file.txt")	Imports annotations from a text file, including new-line characters. Default behaviour: Try to import from default path.
Info	Info("MaxDegOrderSurface")	Displays the value of the specified attribute Default: Displays values of all attributes
MaxDegOrdPotential	MaxDegOrdPotential(200)	Changes value of MaxDegOrdPotential. User specifies DO value. Adapts internal data dependent on this value if possible and discards data that can not be adapted
MaxDegOrdSurface	MaxDegOrdSurface(200)	Changes value of MaxDegOrdSurface. User specifies DO value. Adapts internal data dependent on this value if possible and discards data that can not be adapted
MaxResKMPotential	MaxResKMPotential(100)	Changes value of MaxDegOrdPotential. User specifies resolution in Km. Adapts internal data dependent on this value if possible and discards data that can not be adapted
MaxResDegreesPotential	MaxResDegreesPotential(0.5)	Changes value of MaxDegOrdPotential. User specifies resolution in degrees. Adapts internal data dependent on this value if possible and discards data that can not be adapted
MaxResKMSurface	MaxResKMSurface(100)	Changes value of MaxDegOrdSurface. User specifies resolution in Km. Adapts internal data dependent on this value if possible and discards data that can not be adapted
MaxResDegreesSurface	MaxResDegreesSurface(0.5)	Changes value of MaxDegOrdSurface. User specifies resolution in degrees. Adapts internal data dependent on this value if possible and discards data that can not be adapted
Import	Import()	Launches data import wizard
Export	Export()	Launches data export wizard
Delete	Delete()	Launches a wizard for deleting unused spectral fields in the internal data store

Covar Object

In addition to the methods defined in the table below, the Covar Object also provides a set of `Export` methods for exporting data from any of the data structures in the Object. The names of these `Export` methods are derived from the names of the data structures, which are defined in Section 7 ("Specification of Logical Data Structures"). The `Export()` method launches a wizard.

Name	Example of Usage	Description
`Annotation`	`Annotation("Some text")`	Defines ASCII string of user's comments. Default behaviour: Display string
`ImportAnnotation`	`ImportAnnotation("file.txt")`	Imports annotations from a text file, including new-line characters. Default behaviour: Try to import from default path.
`Info`	`Info("InputOmFunctionStart_1")`	Displays the value of the specified attribute Default: Displays values of all attributes
`AddPos`	`AddPos(lat, lon)`	Adds a position to the `Lon` and `Lat` vectors, used for calculation of error covariances
`DelPos`	`AddPos(lat, lon)`	Deletes a spatial position from `Lon` and `Lat`
`Export`	`Export()`	Launches a wizard
`ErrorFunctionStart`		Changes value of attribute
`ErrorFunctionEnd`		Changes value of attribute
`ErrorFunctionInterval`		Changes value of attribute
`InputPotentialCovComPath`	`InputPotentialCovComPath( "/data/goce/covariances.dat")`	Allows user to specify path to error covariance matrix

Filter Object

The following methods are provided for changing Filter Object attributes.

Name	Example of Usage	Description
`Annotation`	`Annotation("Some text")`	Defines ASCII string of user's comments. Default behaviour: Display string

ImportAnnotation	ImportAnnotation("file.txt")	Imports annotations from a text file, including new-line characters. Default behaviour: Try to import from default path.
Info	Info("Scale")	Displays the value of the specified attribute Default: Displays values of all attributes
Type	Type(SPATIAL_CAP)	Changes value of attribute. Discards internal data dependent on this value
Scale		Changes value of attribute. Discards internal data dependent on this value

4.4.2 Graphical User Interface (GUI)

Detailed design of the GUI will not be performed as part of the GUTS project, but it is important to establish how it might be implemented and what its capabilities could be. Ideally, the GUT GUI would allow the user to do everything that can be done using the Python command interface, but it would require a lot of effort to design a clear, well laid out, user-friendly GUI that was capable of performing all these functions in accordance with the high level workflows in the WP3000 report [26]. One possible approach would be to adapt the BRAT GUI, which provides the following facilities for altimetry data.

- Data extraction and viewing
- Data processing
- Calculating statistics
- Re-gridding
- Computing basic altimeter parameters
- Calendar and date conversion
- Format conversion

The extra functionality required for GUT could be incorporated by allowing the user to select and then manipulate GUT Objects by editing attributes and executing methods. It is possible to imagine a GUI that allows the user to select the desired object from a list, and then select from a list of the attributes and methods belonging to that object. The method naming convention described earlier would be beneficial in this respect, for the same reason that it will make things easier for command interface users. It might be sensible to restrict GUI users to having only one Main Object in a user session. To avoid the GUT GUI becoming too cluttered and difficult to use, it may be better to leave the calculations involving the high level workflows out of the GUI altogether. It has been suggested that this part of GUT should be a completely separate application, accessible only through Python. Some desirable, non-essential GUI features have also been discussed. These include the management of data downloads from servers selected from a list, and the management of shared work-spaces for GUT objects. A visual display linking the toolbox commands with the scientific work-flow diagrams would be very instructive for novice users.

4.4.3 Application Program Interface (API)

The API enables programmers to build elements of GUT into their own software applications. The implementation of GUT will be accompanied by detailed documentation of the software architecture, including full specification of the Python language elements that are not normally exposed to the user. This will allow Python programmers to utilise any part of GUT, from the low level algorithms to the high level workflows.

Ideally, the API would also provide access to GUT functionality from other programming languages and proprietary scientific scripting languages such as IDL and MATLAB. However, for the first release of GUT at least, the non-Python API will be restricted to the software used to implement the Processing Units defined in Section 8 ("Logical Model of System Functionality"). The non-Python API will consist of a set of Fortran subroutines that each take input from a NetCDF file and send output to a NetCDF file. Functions for controlling the scientific work-flows will not be part of the non-Python API.

4.5 Specification Of Input And Output

4.5.1 Introduction

When a Main Object is created during a GUT user session, its internal data files are created in a directory specified by the user. Later, the `Save` method can be used to save the Main Object itself in that directory. The Main Object, with all its attributes and sub-objects is saved by Python in a process known as *pickling*. The saved Main Object then has a permanent presence on disk, in the directory containing the pickled Python objects and the internal data files. This Main Object can be loaded into another GUT session at a later date by using the `Load` method. Note that the logical data structures in the GUT user environment can not be loaded and saved individually by the user; the internal data store belonging to a Main Object is loaded and saved as a whole. The rest of this section concerns the `Import` and `Export` methods associated with the Spatial, Spectral and Covar objects. These methods deal with *external* data files, which are not part of the internal data store belonging to each Main Object. The `Import` methods read data from external files into the internal data store, and the `Export` methods write data from the internal store to external files.

4.5.2 Data exports

The user can export data from the Spatial, Spectral and Covar objects, including intermediate stages of scientific work-flows in spatial or spectral form. There is an `Export` method for every vector and array associated with these objects, except for the SH coefficients representing the gravity potential and their error covariance matrix. `Export` method names are identified using the names of the data structures specified in Section 7. The default exported file names are also derived from these data structure names. The default file name is used if the user does not specify a file name. The following file formats can be used for data exports from GUT:

- NetCDF. This is a binary format that is used for each Main Object's internal data store. This format is normally associated with gridded data or point lists, but can be used to store any collection of multidimensional arrays. Therefore, all data structures in the Spatial,

Spectral and Covar objects can be exported in this format. Open Source NetCDF software libraries are available for Python, Fortran and C, plus other languages not used in GUT.
- GRAVSOFT. This is an ASCII format, which is described in the WP2000 report. GRAVSOFT is suitable for export of real-space data on grids and in point lists, and surface SH coefficients. Components of the GRAVSOFT software, which is written in Fortran, can be re-used in GUT for the data export routines. A separate metadata file in an ASCII format is required for exports in GRAVSOFT format. The metadata file has the same file name stem as the GRAVSOFT file.

The format of GUT command names is described in Section 5.1 ("Command Interface"). `Export` method names are derived from the data structure names, as described in that section. Therefore, the `Export` method name in the command specifies the data structure to be exported. Export from input fields is allowed, to enable GUT to be used for the simple task of grid adaptation, where external data on a different grid can be imported, adapted to the current Spatial Object grid and then exported. The basic format of an export command is given below.

MainObjectName.SubObjectName.ExportMethodName(FilenameString, FieldIdentifier, FormatSpecifier)

The FormatSpecifier parameter can be either `"NETCDF"` or `"GRAVSOFT"`. These specifiers are Python constants, as described in Section 5.1. An error results if an `Export` command specifies a file format that is not supported for the data structure in question. The following types of data export are supported.

NetCDF and GRAVSOFT formats are suitable for the export of spatial field data from the Spatial Object. Fields can only be exported one at a time, but the same output file can be specified in more than one Export command. GUT exports from the Spatial Object contain the following metadata:
- Attributes from Spatial Object of origin
- Attributes from Main Object of origin
- Data structure identifier (see Section 7 - "Specification of Logical Data Structures"), from which the user can infer details of the origin of the data, such as what type of calculation was used to produce the results.

NetCDF files can store metadata themselves, but the GRAVSOFT files require a separate metadata file to store this information.

Surface SH coefficients from the Spectral Object can be exported in NetCDF or GRAVSOFT format. One GRAVSOFT file contains one array of SH coefficients, but a NetCDF file can contain more than one array. The SH coefficients of the gravity potential can not be exported from the Spectral Object. It does not make sense to provide export facilities for this array because GUT does not modify them in any way. Exports of SH coefficients contain the following metadata:
- Attributes from Spectral Object or origin
- Attributes from Main Object of origin
- Data structure identifier (see Section 7 - "Specification of Logical Data Structures"), from which the user can infer details of the origin of the data, such as what type of calculation was used to produce the results.

A separate metadata file is required for each exported GRAVSOFT file.

Error covariance matrices and covariance function evaluation vectors from the Covar Object can be exported only in NetCDF format. Each output file contains one error covariance matrix or function evaluation vector. These exports include the following metadata:
- Attributes from Covar Object of origin
- Attributes from Main Object of origin
- Data structure identifier (see Section 7 - "Specification of Logical Data Structures"), from which the user can infer details of the origin of the data, such as what type of calculation was used to produce the results.

4.5.3 Data imports

The subject of importing data into GUT is more complicated than exporting for several reasons:
- The importing routines in GUT deal with adapting data to the grid and/or reference system specified by the attributes in the Main Object and Spatial Object.
- The importing routines need to be supplied with certain parameters that describe the imported data, such as the grid and reference system specifications
- More file formats are supported for importing than for exporting
- There are restrictions on which data structures can receive imported data

These issues are discussed in more detail below.

The following file formats are supported for data import to Spatial Object vectors.
- ICGEM
- NetCDF
- GRAVSOFT
- AVISO. Some AVISO data is distributed in NetCDF format but some data sets are only available in Geophysical Data Record (GDR) format.
- GMT. This is the format used by the General Mapping Tool (GMT). The default land surface height data set will be supplied in this format.

The following file formats are supported for data import to Spectral Object SH coefficient arrays (potential and surface)
- ICGEM
- GRAVSOFT
- NetCDF

The default data sets to be supplied with GUT are specified in WP3000. It does not matter what formats these are supplied in from the point of view of the software design, as long as the formats are supported by GUT.

Data can be imported into any spatial or spectral field, but most of the time it will only be necessary for the user to import into fields used for input to the scientific work-flows. If data imported in spectral form is required in spatial form by one of the work-flows, the user must convert the data to spatial form before using the work-flow. `Convert` methods in the Spectral Object are provided for this purpose. The tables in Section 7 ("Specification of Logical Data Structures") specify the default data set associated with each of the "Input" fields in the Spatial and Spectral Objects. There is an `Import` method associated with each spatial and spectral data structure specified in Section 7 ("Specification of Logical Data Structures"). Additionally, all object types in GUT have an `ImportAnnotation()` method for importing ASCII text into the Annotation attribute. There is no Import method for the error covariance matrix of the

GOCE coefficients, because this matrix is not part of the Internal Data Store. The name of each `Import` method is derived from the name of the data structure in question. For example, the `ImportInputGeoidHeight()` method imports data into the InputGeoidHeight field. The format of an `Import` method command is described below:

MainObjectName.SubObjectName.ImportMethodName(FilenameString, FieldIdentifier, FormatSpecifier)

The FieldIdentifier parameter is for specifying the field or data structure of the required data in the external file. The FormatSpecifier parameter can be either NETCDF, `GMT`, `AVISO`, `ICGEM` or `GRAVSOFT`. These specifiers are Python constants, as described in Section 5.1 ("Command Interface").

The default MSSH data set defines the default grid and reference system. When a new Main Object is created, two `Import` methods are executed automatically: `Spatial.ImportInputMSSH()` and `Spectral.ImportPotential()`. These import data from the default MSSH data set and the default gravity potential coefficients, which establishes the default grid and reference system and readies the new Main Object for calculations involving work-flows1a and 1b. The user must execute other `Import` methods before using any of the other work-flows.

All the default data sets are supplied with GUT-specific metadata, as described in the previous section ("Data Exports"). The ICGEM files have some information in the header but an additional metadata file is required for GUT-specific information. The NetCDF files store all the required metadata internally. When GUT-specific metadata are supplied with external data files, the Import methods know how to find the important parameters describing the data being imported, and can associate them with the attributes of the GUT Objects. The `Import` methods always look for GUT-specific information in the metadata. If found, the import procedure takes place automatically, the values of the relevant attributes being read from the file header or metadata file. This feature is useful for importing data from the default data sets and for transferring data between two GUT work-flows, where the output from one work-flow is exported, then imported to become the input for another work-flow. If the metadata are not found, as is likely to be the case when importing data from external sources, the user is prompted to enter the relevant parameters describing the imported data, including the grid and reference system specification parameters.

The low level data ingestion routines in the GRAVSOFT software can be re-used in GUT for importing data in GRAVSOFT format. The low level data ingestion routines in BRAT might be useful for GUT imports of altimetry data in NetCDF and other formats. Software written in Perl that is currently being developed by the GOCE HPF team may be suitable for re-use in GUT for importing ICGEM files. Tools for handling GDR (AVISO) data are available from AVISO.

The rules for adapting imported spatial data to a different grid specification are described below. In general, imported data is always adapted to the grid described by the Spatial Object attributes. There are several possible scenarios that need particular mention.

- *Scenario*: The imported data is on a grid and the Spatial Object attributes specify a grid. *Action*: Imported data is adapted to the grid specified by the Spatial Object attributes
- *Scenario*: The imported data is on a grid and the Spatial Object attributes specify a point list. *Action*: Imported data is interpolated to the point list specified by the Spatial Object

- *Scenario*: The imported data is specified for a point list and the Spatial Object attributes specify a grid. *Action*: The import procedure can not be performed according to the general rule, but the user is given the option of interpolating all the data in the Spatial Object to the point list specified in the imported data file.
- *Scenario*: The imported data and the Spatial Object attributes specify two, overlapping regional grids. *Action*: Only data in the overlapping region is taken from the file, and existing data outside the overlapping region is discarded from the Spatial Object, subject to user confirmation. The new Spatial Object grid covers the overlapping region only.
- *Scenario*: The imported data and the Spatial Object attributes specify two regional grids that do not overlap. *Action*: The import operation is not allowed, and an error is reported to the user.
- *Scenario*: The imported data and Spatial Object attributes both specify point lists. *Action*: The import operation is not allowed, and an error is reported to the user.

4.5.4 Reports and logs

A record of GUT user sessions is stored by each Main Object. The Python methods that form part of the command interface record their use in a log file, along with any warnings and errors reported to the user. The log file is part of the internal data store belonging to the Main Object.

4.6 Specification Of Logical Data Structures

This section specifies and labels all the logical data structures that form the conceptual view of the data and parameters presented to the user. There are physical data structures in the internal data store files corresponding to each logical data structure. The data structures are grouped according to the Python objects used to interact with them in the command interface described earlier. These data structures form the basis of the logical model of system functionality described in Section 8. Pre-defined Python constants are written in upper case letters. Unspecified values are shown using Python's null value, `None`.

Main Object

Attributes

Name	Type	Possible values	Default value	Description
`Annotation`	String	ASCII chars, including '\n'	"Default "	Description of the object
`Ellips_GM`	Float		GOCE default ($3.986004415 \cdot 10^{14}$)	*GM*
`Ellips_a`	Float		TOPEX value (6378136.3)	*a*
`Ellips_gamma_a`	Float		None	γa
`Ellips_f`	Float		TOPEX value (1/ 298.257)	*f*
`Ellips_J2`	Float		None	J_2

Ellips_omega	Float		None	ω
TideSys	Flag	Python constants: TIDE_FREE, ZERO_TIDE, MEAN_TIDE	As for GOCE L2 products	Tide system
GeodeticCalc	Flag	Python constants: HEIGHT, ANOMALY or DEFLECTION	Python constant: HEIGHT	Controls which geodetic quantities are calculated in work-flows 1a and 1b. Set by choice of Calc method by user

Data

Name	Type	Description
Spatial	Python object	Data in the spatial domain
Spectral	Python object	Data in the spectral domain
Covar	Python object	Data associated with error covariance calculations for geodetic quantities
Filter	Python	Parameters associated with the filtering algorithms

Spatial Object

Attributes

Name	Type	Possible values	Default value	Description
GridType	Flag	Python constants: REGULAR, UNSTRUCTURED or LIST	Python constant: REGULAR	Determines whether or not the Spatial Object holds a regular or unstructured grid or a point list

`GridCalc`	Flag	Python constants: `POINT` or `AVERAGE`	Python constant: `POINT`	Specifies whether values at grid points represent only those points or area averages for the grid cells. Area averages are only allowed if the Spatial Object holds a grid (as opposed to a list of points). Area average calculations are only allowed for geodeditic output fields. Executing `Calc` methods for other fields while area averages are specified results in an error
`MaxDegOrdPotential`	Integer	In range 1 to `Spectral. MaxDegOrdPotential`	`Spectral. MaxDegOrdPotential`	Maximum degree and order for SH synthesis from SH coefficients of gravity potential
`MaxDegOrdSurface`	Integer	In range 1 to `Spectral. MaxDegOrdSurface`	`Spectral. MaxDegOrdPotential`	Maximum degree and order for SH synthesis from surface SH coefficients
`LatMin`	Integer	In range -90 to 90	-80	Grid starting latitude (degrees North)
`LatMax`	Integer	In range -90 to 90	82	Grid ending latitude (degrees North)
`LonMin`	Integer	In range -180 to 180	-180	Grid starting longitude (degrees East)
`LonMax`	Integer	In range -180 to 180	180	Grid ending longitude (degrees East)

LatCell	Float		1/30	Regular grid cell height (degrees)
LonCell	Float		1/30	Regular grid cell width (degrees)
InputMSSH_Start	Integer	4-digit year	1993	Start date for mean
InputMSSH_End	Integer	4-digit year	1999	End date for mean
InputMDT_Start	Integer	4-digit year	According to default data set	Start date for mean
InputMDT_End	Integer	4-digit year	According to default data set	End date for mean
Annotation	String	ASCII chars, including '\n'	"Default "	Description of the object

Data

Each spatial field vector has a unique name. There are three categories of vector: input, intermediate and output. Vectors are assigned to one of these categories depending on whether they are an input to a scientific work-flow, a final output or an intermediate stage. Input vector names all begin with "Input". The output vectors are the official GUT work-flow products. They each represent the final output from a scientific work-flow, and are highlighted in bold type in the table below. Vectors storing intermediate stages in a work-flow have names that begin with the name of the final output, plus an extra identifier starting with "-Inter". For example, the final output of Work-flow 4a is stored in a vector called MDTC-A_Spatial, and the two intermediate stages have vectors called MDTC_A-Spatial-InterAprioriMDT and MDTC_A-Spatial- InterMDTCorrection. The names of these data structures may need to be reviewed in the context of their use in the command interface described in Section 5.1 ("Command Interface"), because many of the GUT method names are derived from the names of the data structures they act on. This leads to some long and cumbersome method names such as `ExportMDTC_A-Spatial-InterMDTCorrection()`, although the name completion facility in Python would minimise the amount of typing that would be required in order to enter such a command. The idea of using Work-flow Objects rather than a single Main Object, as discussed in the Appendix ("Using the Command Interface"), may be an appropriate solution to the problem of rationalising the method names used in commands.

Filtering takes place during the execution of some of the scientific work-flows. GUT work-flow products can either be filtered or unfiltered, depending on their position in the work-flow. These products must be clearly separate from the results of "manual" filtering operations, which can be initiated by the use of the Spatial Object `Filter` methods. The user can manually filter any of the input, intermediate or output fields. To avoid confusion with GUT work-flow products, this manual filtering operation results in a new field with the same name plus a preceding "Filtered_". These fields are not shown here. They do not play a part in any of the work-flows, and although they can be exported like any of the other fields, they are not GUT work-flow products.

Type	Name	Description
Vector	Lat	Latitude in degrees Default: Positions in grid of default InputMSSH
Vector	Lon	Longitude in degrees Default: Positions in grid of default InputMSSH

Vector	InputPotential	Gravity potential Default: None. SH coefficients are used in work-flows
Vector	InputLandHeight	Height of land above reference ellipsoid Default: ETOPO2v2 digital elevation model
Vector	InputGeoidHeight	Geoid height Default: EGM_GEO_2
Vector	InputGravityAnomaly	Not part of any work-flow. Import facility provided just to allow grid adaptation followed by export. This facility provides GUT_002 Default: EGM_GAN_2.
Vector	InputEWDeflection	Not part of any work-flow. Import facility provided just to allow grid adaptation followed by export. This facility provides GUT_003 Default: EGM_GVE_2
Vector	InputNSDeflection	Not part of any work-flow. Import facility provided just to allow grid adaptation followed by export. This facility provides GUT_004 Default: EGM_GVN_2
Vector	InputGeoidHeightVarCom	Commission error variances. Not part of any work-flow. Import facility provided just to allow grid adaptation followed by export Default: EGM_GER_2
Vector	InputGravityAnomalyVarCom	Commission error variances. Not part of any work-flow. Import facility provided just to allow grid adaptation followed by export Default: EGM_GER_2
Vector	InputEWDeflectionVarCom	Commission error variances. Not part of any work-flow. Import facility provided just to allow grid adaptation followed by export Default: EGM_GER_2
Vector	InputNSDeflectionVarCom	Commission error variances. Not part of any work-flow. Import facility provided just to allow grid adaptation followed by export Default: EGM_GER_2
Vector	InputMSSH	MSSH for MDT calculations Default: CLS01
Vector	InputAverageSLA	Average SLA for a period taken from a SLA time-series Default time series: ?
Vector	InputAverageADT	Average ADT for a period taken from a ADT time-series Default time series: ?
Vector	InputMDT	MDT Default: OCCAM model at 1/12 degree
Vector	InputMDTS	Satellite MDT Default: None (Should be output from WF3a or WF3b)
Vector	**GeoidHeight**	Geoid Height

Vector	**EWDeflection**	Gravity deflection from the vertical, E-W direction
Vector	**NSDeflection**	Gravity deflection from the vertical, N-S direction
Vector	**GravityAnomaly**	Gravity anomaly
Vector	**GeoidHeightVarCom**	Commission error variances of Geoid Height. This will not be included in the first release of GUT
Vector	**EWDeflectionVarCom**	Commission error variances of Gravity deflection from the vertical, E-W direction. This will not be included in the first release of GUT
Vector	**NSDeflectionVarCom**	Commission error variances of Gravity deflection from the vertical, N-S direction. This will not be included in the first release of GUT
Vector	**GravityAnomalyVarCom**	Commission error variances of Gravity anomaly. This will not be included in the first release of GUT
Vector	**GeoidHeightVarOm**	Omission error variances of Geoid Height
Vector	**EWDeflectionVarOm**	Omission error variances of Gravity deflection from the vertical, E-W direction
Vector	**NSDeflectionVarOm**	Omission error variances of Gravity deflection from the vertical, N-S direction
Vector	**GravityAnomalyVarOm**	Omission error variances of Gravity anomaly
Vector	**MSSH_Average**	MSSH calculated using reference MSSH and average SLA
Vector	**MDT_Average**	MDT calculated using reference MDT and average SLA
Vector	**MDTS_Spatial**	MDTS from WF3a
Vector	MDTS_Spatial-InterMDTS	Unfiltered MDTS
Vector	**MDTS_Spectral**	MDTS from WF3b
Vector	MDTS_Spectral-InterMSSH	Intermediate MSSH. Spatial field not created in work-flow
Vector	MDTS_Spectral-InterGeoidHeight	Intermediate geoid height. Spatial field not created in work-flow
Vector	MDTS_Spectral-InterMDTS	Intermediate MDTS.
Vector	**MDTC_A-Spatial**	MDTC, RR method A in spatial domain
Vector	MDTC_A-Spatial-InterAprioriMDT	A-priori MDT
Vector	MDTC_A-Spatial-InterMDTCorrection	MDT correction
Vector	**MDTC_A-Spectral**	MDTC, RR method A in spectral domain
Vector	MDTC_A-Spectral-InterAprioriMDT	A-priori MDT. The spectral version of this field is calculated in the WF
Vector	MDTC_A-Spectral-InterSmoothAprioriMDT	Smooth A-priori MDT. The spectral version of this field is calculated in the WF

Vector	MDTC_A-Spectral-InterMDTCorrection	MDT correction. The spectral version of this field is calculated in the WF
Vector	**MDTC_B-Spatial**	MTDC, RR method B in spatial domain
Vector	MDTC_B-Spatial-InterUnfilteredMDTCorrection	MSSH – geoid – MDT
Vector	MDTC_B-Spatial-InterMDTCorrection	Filtered {MSSH – geoid –MDT}
Vector	**MDTC_B-Spectral**	MTDC, RR method B in spectral domain
Vector	MDTC_B-Spectral-InterAprioriGeoidHeight	MSSH – MDT, with gaps filled with geoid
Vector	MDTC_B-Spectral-InterGeoidHeight	Intermediate geoid. The spectral version of this field is calculated in the WF
Vector	MDTC_B-Spectral-InterUnfilteredMDTCorrection	A-priori-geoid - geoid. The spectral version of this field is calculated in the WF
Vector	MDTC_B-Spectral-InterMDTCorrection	Filtered {MSSH – geoid. – MDT}. The spectral version of this field is calculated in the WF
Vector	MDTC_B-Spectral-InterAprioriMDT	Filtered a-priori MDT. The spectral version of this field is calculated in the WF
Vector	**NVelocity**	Northwards component of geostrophic current
Vector	**EVelocity**	Eastwards component of geostrophic current

Spectral Object

Attributes

Name	Type	Possible values	Default value	Description
`MaxDegOrdPotential`	Integer	In range 1 to 250	250	Maximum degree & order of SH coefficients representing the gravity potential
`MaxDegOrdSurface`	Integer	In range 1 to default for `MaxDegOrdPotential`	Default for `MaxDegOrdPotential`	Maximum degree & order for SH analysis of spatial fields, i.e. for converting from spatial to spectral space
`Annotation`	String	ASCII chars, including '\n'	"Default "	Description of the object

Data

The Spectral Object has a 2D array equivalent to every vector in the Spatial Object. The names of the arrays in the Spectral Object are the same as the names of the vectors in the Spatial Object. Spatial fields can be converted to an array of SH coefficients during the course of a scientific work-flow. Any spatial field in the Spatial Object can also be manually converted to spectral space using a Spatial Object `Convert` method. The result of these conversions is stored as an array in the Spectral Object. The only data that are supplied in spectral form by default in GUT are the SH coefficients of the gravity potential. To avoid repeating most of the information in the Spatial Object data table above, the only two Spectral Object array that is shown in the table below is for the gravity potential coefficients, because this is the only Spectral Object array that does not have a corresponding vector in the Spatial Object. All the other arrays are undefined until a spatial field is converted to spectral space during a user session.

Type	Name	Description
2D Array	InputPotential	SH coefficients representing the gravitational potential. Default: EGM_GCF_2

Covar Object

Attributes

Name	Type	Possible values	Default value	Description
ErrorFunctionStart	Float	?	None	Starting distance for evaluation of omission error covariance function
ErrorFunctionEnd	Float	?	None	Ending distance for evaluation of omission error covariance function
ErrorFunctionInterval	Float	?	None	Interval for evaluation of omission error covariance function
Annotation	String	ASCII chars, including '\n'	"Default"	Description of the object
InputPotentialCovComPath	String		None	Path to error covariance matrix of gravity potential SH coefficients

Data

Name	Type	Description
Lon	Vector	Longitudes for positions for error covariance calculations
Lat	Vector	Latitudes for positions for error covariance calculations
GeoidHeightCovOm	2D Array	Omission error variance-covariance matrix
GeoidHeightCovOmFunction	Vector	Evaluation of omission error covariance function of distance
GravityAnomalyCovOm	2D Array	Omission error variance-covariance matrix
GravityAnomalyCovOmFunction	Vector	Evaluation of omission error covariance function of distance

EWDeflectionCovOm	2D Array	Omission error variance-covariance matrix
EWDeflectionCovOmFunction	Vector	Evaluation of omission error covariance function of distance
NSDeflectionCovOm	2D Array	Omission error variance-covariance matrix
NSDeflectionCovOmFunction	Vector	Evaluation of omission error covariance function of distance
InputPotentialCovCom	2D Array	Error covariance matrix for SH coefficients of the gravity potential. This matrix is not part of the IDS. Instead, the path to the matrix in ICGEM format is stored by Covar Object attribute InputPotentialCovComPath. This data structure will not be included in the first release of GUT Default: EGM_GVC_2
GeoidHeightCovCom	2D Array	Commission error variance-covariance matrix. This will not be included in the first release of GUT
GravityAnomalyCovCom	2D Array	Commission error variance-covariance matrix. This will not be included in the first release of GUT
EWDeflectionCovCom	2D Array	Commission error variance-covariance matrix. This will not be included in the first release of GUT
NSDeflectionCovCom	2D Array	Commission error variance-covariance matrix. This will not be included in the first release of GUT

Filter Object

Attributes

Name	Type	Possible values	Default value	Description
`Type`	Flag	Python constants: `SPATIAL_JEKELI`, `SPATIAL_GAUSSIAN`, `SPATIAL_CAP`, `SPATIAL_HANNING`, `SPATIAL_HAMMING`, `SPECTRAL_JEKELI` or `SPECTRAL_PELLINEN`	Python constant: `SPATIAL_JEKELI`	Filter type
`Scale`	Integer	?	?	Length scale
`Annotation`	String	ASCII chars, including	"Default "	Description of the

		'\n'		object

Data

Name	Type	Description
Matrix	2D Array	Filter matrix

4.7 Logical Model of System Functionality

This section discusses the units of computational work and data flows involved in setting up the internal data store and executing the scientific work-flows. Python syntax is used to specify names of logical data structures according to the Object they belong to in the user environment. The calculation of commission error variances and covariances is not mentioned explicitly in any of the discussions concerning scientific work-flows. This is because the ability to handle the error covariance matrix of the SH coefficients of gravity potential is not going to be in the first release of GUT, as discussed in the Introduction to this report. For the first release of GUT, calculation of covariance error variances will instead be provided by interpolation of the relevant GOCE L2 products, EGM_GER_2. This can be achieved by importing the relevant products into "Input" logical data structures in the Spatial Object. The `Import` methods in GUT deal with adapting data to the current, specified grid and reference system, as described in Section 6 ("Specification of Input and Output").

4.7.1 Processing Units

The Processing Units described in this section are related to the Functional Algorithms described in the WP3000 report. They are units of computational work involved in the pre-processing of data and the execution of the scientific work-flows. They are large computational tasks involving one or more spatial or spectral fields. All Processing Units have the following types of input and output.
Input:
- One or more attributes
- One or more data structures representing spatial and/or spectral fields
- Other large data structures such as a filter matrix

Output:
- One or more data structures representing spatial and/or spectral fields
- Other large data structures such as a filter matrix

The Processing Units represent computationally expensive tasks because they each involve a large amount of data. Therefore, they are not likely to be implemented in Python. Instead, they will probably be implemented in C or Fortran as described in Section 3 ("Toolbox Components"). They all exchange data with the internal data store independently of the parent Python application. In the data-flow diagrams in the next section, the output of one Processing Unit is often shown going directly into another Processing Unit. In most cases the output is also shown being directed to a logical data structure corresponding to a scientifically interesting work-flow product. However, in some cases no logical data structure is specified. All transfers of data take place via the internal data store, but logical data structures are not shown unless they correspond to a scientifically interesting intermediate work-flow product.

Each Processing Unit is numbered according to the Functional Algorithm it relates to. The preference selection tasks are not represented by Processing Units because they simply involve assigning values to attributes. That is why there is no Processing Unit related to Functional Algorithm FA01, because selecting the required degree and order for SH synthesis is not a computationally expensive task. However, Functional Algorithms FA02 and FA03, concerning the reference ellipsoid and tide system, are associated with a Processing Unit because they are a mixture of preference selection and data pre-processing. The preference selection part involves assigning values to the attributes defining the ellipsoid and tide system, and the data pre-processing part involves adapting any existing data in the internal data store to the new parameters. Preference selection methods are discussed in the Section 5.1 ("Command Interface") and in the Appendix ("Using the Command Interface"). Processing Units PU15a, PU15b, PU16 and PU17 do not relate to any of the Functional Algorithms.

The Processing Units are described below, with reference to the logical data structures, the Functional Algorithms from WP3000 and the algorithms corresponding to the GUT Products from WP2000. In the Processing Unit definitions, some inputs and outputs are not explicitly mentioned. Firstly, all Processing Units that deal with spatial field vectors have the following attributes as input:

- Spatial Object grid or point list specification parameters
  - `Spatial.GridType`
  - `Spatial.LatMin, Spatial.LatMax, Spatial.LatCell`
  - `Spatial.Lat`
  - `Spatial.LonMin, Spatial.LonMax, Spatial.LonCell`
  - `Spatial.Lon`

Similarly, all Processing Units that deal with surface SH coefficient arrays in the Spectral Object have input from the `Spectral.MaxDegOrdSurface` attribute, and all Processing Units that deal with SH coefficients of the gravity potential have input from the `Spectral.MaxDegOrdPotential` attribute.

All Processing Units take input from the Main Object reference system parameters, defined by the following attributes:

- o `Main.Ellips_GM`
- o `Main.Ellips_a`
- o `Main.Ellips_gamma_a`
- o `Main.Ellips_f`
- o `Main.Ellips_J2`
- o `Main.Ellips_omega`
- o `Main.TideSys`

Pre-processing Units

PU02: Reference ellipsoid adaptation

Adapts one spatial field vector to a new reference ellipsoid specification, using one of the GUT Product algorithms in the range GUT_104 to GUT_105.
Input:
- One Spatial Object vector

- Reference ellipsoid parameters for the vector
- Current Main Object reference ellipsoid parameters
  - `Main.Ellips_GM`
  - `Main.Ellips_a`
  - `Main.Ellips_gamma_a`
  - `Main.Ellips_f`
  - `Main.Ellips_J2`
  - `Main.Ellips_omega`

Output:
- One Spatial Object vector

PU03a: Tide system adaptation in geographical space

Adapts one spatial field vector to a new tide system, using one of the GUT Product algorithms in the range GUT_106 to GUT_107.

Input:
- One Spatial Object vector
- Tide system specifier for the vector
- Current Main Object tide system specifier
  - `Main.TideSys`

Output:
- One Spatial Object vector

PU03b: Tide system adaptation in spectral space

Adapts one set of gravity field potential SH coefficients to a new tide system, using the algorithms described in the definition of FA03. There is no GUT Product algorithm corresponding to this Processing Unit.

Input:
- One Spectral Object array
  - `Spectral.InputPotential`
- Tide system specifier for those coefficients
- Current Main Object tide system specifier
  - `Main.TideSys`
- Maximum degree and order of SH coefficients
  - `Spectral.MaxDegOrdPotential`

Output:
- One Spectral Object array
  - `Spectral.InputPotential`

PU07: Time series averaging

Calculates the average value of one spatial field for a period of a time-series, using the algorithm described in the definition of FA07. There is no GUT Product algorithm corresponding to this Processing Unit.

Input:
- Time-series of a spatial field, on a grid or point list
- Spatial field identifier
- Grid or point list specification parameters describing time series
- Start and end dates for calculation of averages

Output:
- A spatial field on a grid or point list

PU08: Grid adaptation

Adapts one gridded spatial field to a new grid or point list specification, using the algorithm for GUT Product GUT_110.
Input:

- A spatial field on a grid
- Spatial field identifier
- Grid specification parameters describing spatial field
- Spatial Object grid or point list specification parameters
  - `Spatial.GridType`
  - `Spatial.LatMin, Spatial.LatMax, Spatial.LatCell`
  - `Spatial.Lat`
  - `Spatial.LonMin, Spatial.LonMax, Spatial.LonCell`
  - `Spatial.Lon`

Output:

- One Spatial Object vector

Work-flow Processing Units

PU04a: SH synthesis from SH coefficients gravity potential

Computes one geodetic field on a grid or point list, starting from the SH coefficients of the gravity field potential. The geodetic field is either geoid height, gravity anomaly or deflection from the vertical. This Processing Unit corresponds to FA04 with either geoid height, gravity anomaly or geoid deflection specified as an option. One GUT Product algorithm in the range GUT_005 to GUT_012 is used during the execution of this Processing Unit.
Input:

- Geodetic output field specifier
  - o `Main.GeodeticCalc`
- SH Coefficients of gravity potential
  - o `Spectral.InputPotential`
- Maximum degree and order of potential SH coefficients to be used in the synthesis
  - o `Spatial.MaxDegOrdPotential`
- Calculation type
  - o `Spatial.GridCalc`
- Grid or point list specification
  - o `Spatial.GridType`
  - o `Spatial.LatMin, Spatial.LatMax, Spatial.LatCell`
  - o `Spatial.Lat`
  - o `Spatial.LonMin, Spatial.LonMax, Spatial.LonCell`
  - o `Spatial.Lon`

Output:

- One Spatial Object vector, or pair of vectors in the case of gravity field deflection

PU4b: SH synthesis from surface SH coefficients

Converts one set of surface SH coefficients to geographical space. This Processing Unit corresponds to FA04 with Dynamic Topography specified as an option, but it is not restricted to SH synthesis of Dynamic Topography fields. PU04b can be used for SH synthesis from any set of surface SH coefficients. An algorithm similar to those defined in GUT_005 to GUT_012 could be used to perform the SH synthesis.

Input:
- Spectral Object array containing surface SH Coefficients
- Maximum degree and order of surface SH coefficients to be used in the synthesis
    - `Spatial.MaxDegOrdSurface`

Output:
- One Spatial Object vector

PU05: Commission error determination

The first release of GUT will not include the ability to calculate commission error variances and covariances, as discussed in the Introduction to this report. This Processing Unit is included for completeness, although it is not used in any of the scientific work-flows described in this report. In the first release of GUT, commission error variance data for the geoid height, deflection and gravity anomaly fields will be provided by allowing the user to import the relevant gridded GOCE error products (EGM_GER_2).

This Processing Unit computes commission error variance and covariance for one geodetic output field. GUT Product algorithms in the range GUT_016 to GUT_031 are used for each execution of this Processing Unit.

Input:
- Geodetic output field specifier
    - `Main.GeodeticCalc`
- SH Coefficients of gravity potential
    - `Spectral.InputPotential`
- Error covariance matrix of SH coefficients of gravity potential
    - `Covar.InputPotentialCovCom`
- Maximum degree and order of potential SH coefficients in Spectral Object
    - `Spectral.MaxDegOrdPotential`
- Maximum degree and order for SH sythesis of potential SH coefficients
    - `Spatial.MaxDegOrdPotential`
- Calculation type
    - `Spatial.GridCalc`
- Grid or point list specification
    - `Spatial.GridType`
    - `Spatial.LatMin, Spatial.LatMax, Spatial.LatCell`
    - `Spatial.Lat`
    - `Spatial.LonMin, Spatial.LonMax, Spatial.LonCell`
    - `Spatial.Lon`
- Specification of a pair of points
    - `Covar.Lat`
    - `Covar.Lon`

Output:
- One Spatial Object vector containing commission error variances
- One Covar Object array containing commission error covariances

PU06: Omission error determination

Computes omission error variance and covariance for one geodetic output field. One of the GUT Product algorithms in the range GUT_032 to GUT_039 is used for this Processing Unit.

Input:
- Geodetic output field specifier
    - `Main.GeodeticCalc`
- SH Coefficients of gravity potential
    - `Spectral.InputPotential`
- Maximum degree and order of potential SH coefficients in Spectral Object

- o `Spectral.MaxDegOrdPotential`
- Maximum degree and order for SH sythesis of potential SH coefficients
  - o `Spatial.MaxDegOrdPotential`
- Calculation type
  - o `Spatial.GridCalc`
- Grid or point list specification
  - o `Spatial.GridType`
  - o `Spatial.LatMin, Spatial.LatMax, Spatial.LatCell`
  - o `Spatial.Lat`
  - o `Spatial.LonMin, Spatial.LonMax, Spatial.LonCell`
  - o `Spatial.Lon`
- Specification of a pair of points
  - o `Covar.Lat`
  - o `Covar.Lon`
- Specification of function evaluation parameters

Output:
- One Spatial Object vector containing omission error variances
- One Covar Object array containing omission error covariances
- One Covar Object vector containing evaluation of omission error function

PU09: Linear filter (spatial)

Filters one spatial field, using one of five types of linear filter. Filtering algorithms are discussed in the WP3000 report.

Input:
- One spatial field vector in Spatial Object
- Filter identifier and scale parameter
  - o `Filter.Type`
  - o `Filter.Scale`
- Filter matrix (if this filter has already been used for the current Spatial Object grid)
  - o `Filter.Matrix`

Output:
- One spatial field vector in Spatial Object
- Filter matrix
  - o `Filter.Matrix`

PU10: Linear filter (spectral)

Filters one array of surface SH coefficients, using either of two types of linear filter. Filtering algorithms are discussed in the WP3000 report.
- One array of surface SH coefficients
- Filter identifier and scale parameter
  - o `Filter.Type`
  - o `Filter.Scale`
- Filter matrix (if this filter has already been used for the current value of `Spectral.MaxDegOrdSurface`)
  - o `Filter.Matrix`

Output:
- One array of surface SH coefficients
- Filter matrix
  - o `Filter.Matrix`

PU11a: Fill gaps on continent with another global field

Fills continental gaps in one spatial ocean field using values from global gridded field. This Processing Unit is designed for filling continental gaps in MSSH. The appropriate algorithm for this processing unit is described in the description of FA11 in the WP3000 report.

Input:
- One Spatial Object vector with gaps over continents
- One Spatial Object global gridded field for filling the continental gaps in the ocean field

Output:
- One Spatial Object vector

PU11b: Fill gaps on continent with zeros

Fills continental gaps in one spatial ocean field with zeros. This Processing Unit is designed for filling continental gaps in MDT. The appropriate algorithm for this processing unit is described in the description of FA11 in WP3000.

Input:
- One Spatial Object vector with gaps over continents

Output:
- One Spatial Object vector

PU12: SH analysis

Calculates surface SH coefficients for one global, gridded field, using GUT Product algorithm GUT_103.

Input:
- One Spatial Object vector
- Maximum degree and order for surface SH coefficients
    - Spectral.MaxDegOrdSurface

Output:
- One Spectral Object SH coefficient array

PU13: Surface current determination

Calculates northward and eastward components of surface current for one Spatial Object MDT vector. One of two GUT Product algorithms is used during the execution of this Processing Unit: GUT_013 if the Spatial Object data structures specify a grid, and GUT_014 if the Spatial Object data structures specify a point list.

Input:
- Spatial Object MDT vector

Output:
- Spatial Object vector of Northwards component of velocity
- Spatial Object vector of Eastwards component of velocity

PU14a: Sum of two spatial fields

Adds two spatial fields using GUT Product algorithm GUT_100.

Input:
- Two Spatial Object fields

Output:
- One spatial Object field

PU14b: Difference of two spatial fields

Subtracts two spatial fields using GUT Product algorithm GUT_101.

Input:
- Two Spatial Object fields

Output:
- One spatial Object field

PU15a: Sum of two spectral fields

Adds two spectral fields. No algorithm for this Processing Unit is defined in the reports from WP2000 and WP3000.
Input:
- Two Spectral Object fields

Output:
- One Spectral Object field

PU15b: Difference of two spectral fields

Subtracts two spectral fields. No algorithm for this Processing Unit is defined in the reports from WP2000 and WP3000.
Input:
- Two Spectral Object fields

Output:
- One Spectral Object field

PU16: Calculation of geoid height in spectral space

Calculates surface SH coefficients of the geoid height field. No algorithm for this Processing Unit is specified in the reports from WP2000 and WP3000.
Input:
- SH Coefficients of gravity potential
  - o `Spectral.InputPotential`
- Maximum degree and order for SH synthesis of geoid height
  - o `Spatial.MaxDegOrdPotential`
- Maximum degree and order for SH analysis
  - o `Spectral.MaxDegOrdSurface`

Output:
- One array of Surface SH coefficients in the Spectral Object

PU17: Calculation of mean from reference field and average SLA.

Calculates the mean value of a spatial field on a grid or list of points (M), from the reference mean field (REF) and the average SLA for the time period of interest (MSLA). Algorithm: M = REF + MSLA
Input:
- Reference field vector in Spatial Object
- Mean SLA for time period of interest
  - o `Spatial.InputAverageSLA`

Output
- One Spatial Object vector

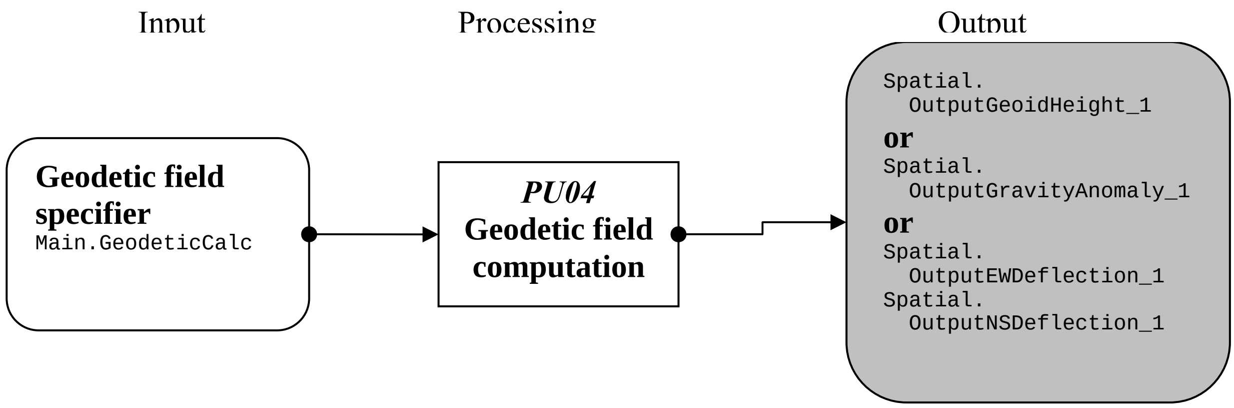
4.7.2 Scientific Data Flows

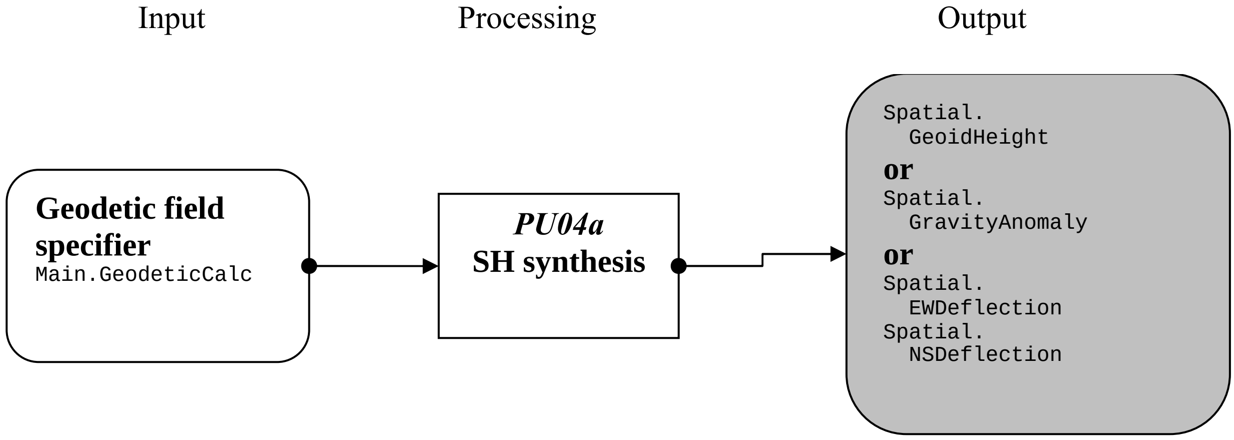
This section relates the work-flows from WP3000 to the Processing Units and the logical data structures defined in Section 7 ("Specification of Logical Data Structures"), i.e. the attributes, vectors and arrays associated with the various types of Object in the user environment. The figures in this section show a series of *data-flow diagrams* to accompany the scientific work-flows from WP3000. There is a data-flow diagram for each scientific work-flow. Python notation is used to describe the data structures, which are labelled according to the following convention.

ObjectName.DataStructureName

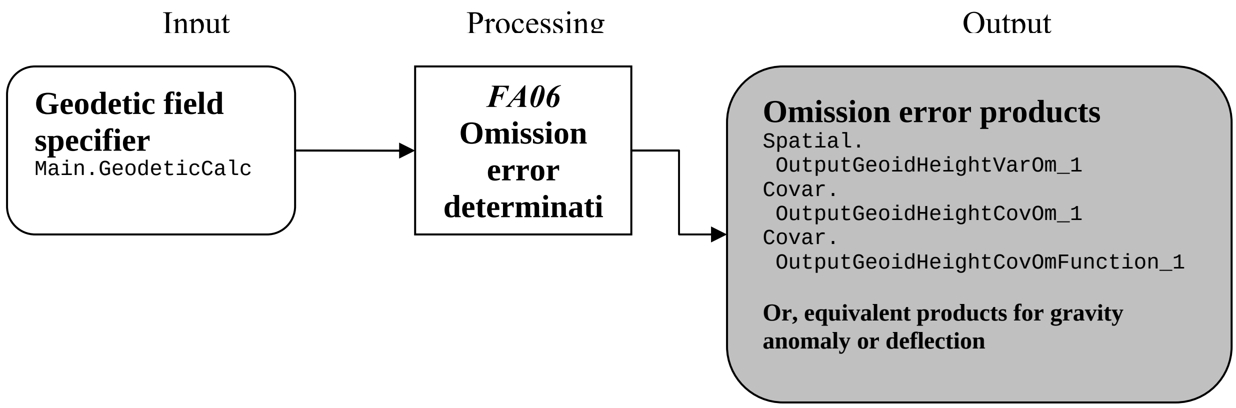
The main difference between the data-flow diagrams and the work-flow diagrams is that preference selection and data pre-processing stages are not shown. The results of preference selection decisions taken by the user are stored in the values of attributes. Similarly, the data pre-processing stages such as grid and reference system adaptation are not shown in the data-flows, because data in the logical data structures has already been through those processes. One advantage of this pre-processing approach is that time consuming processing steps such as grid adaptation do not need to be repeated each time a work-flow calculation is performed. In the data-flow diagrams, all the possible end-points are indicated by shaded boxes. To execute a work-flow up to one of the end points, the `Calc` method corresponding to the desired intermediate or output data structure is used, as described in Section 5.1 ("Command Interface").

All Processing Units take input from, and send output to real data structures in the internal data store. Therefore, all data transfer between processing units shown in the data-flow diagrams takes place via the internal data store. Logical data structures corresponding to these transfers are not shown, except in cases where they correspond to scientifically interesting work-flow products that were included in the work-flow diagrams in the WP3000 report. The scientifically interesting products are directed to intermediate logical data structures, which are accessible to the user. Extra scientifically interesting intermediate data structures can easily be added at a later time. Inputs to, and outputs from Processing Units that come from fixed logical data structures are omitted from the data-flow diagrams. The Process Unit definitions contain details of these fixed inputs and outputs. In cases where the output of one work-flow is to be used as the input to another work-flow, the output from the first work-flow must be exported to an external file, and then imported into the appropriate input logical data structure for the second work-flow.

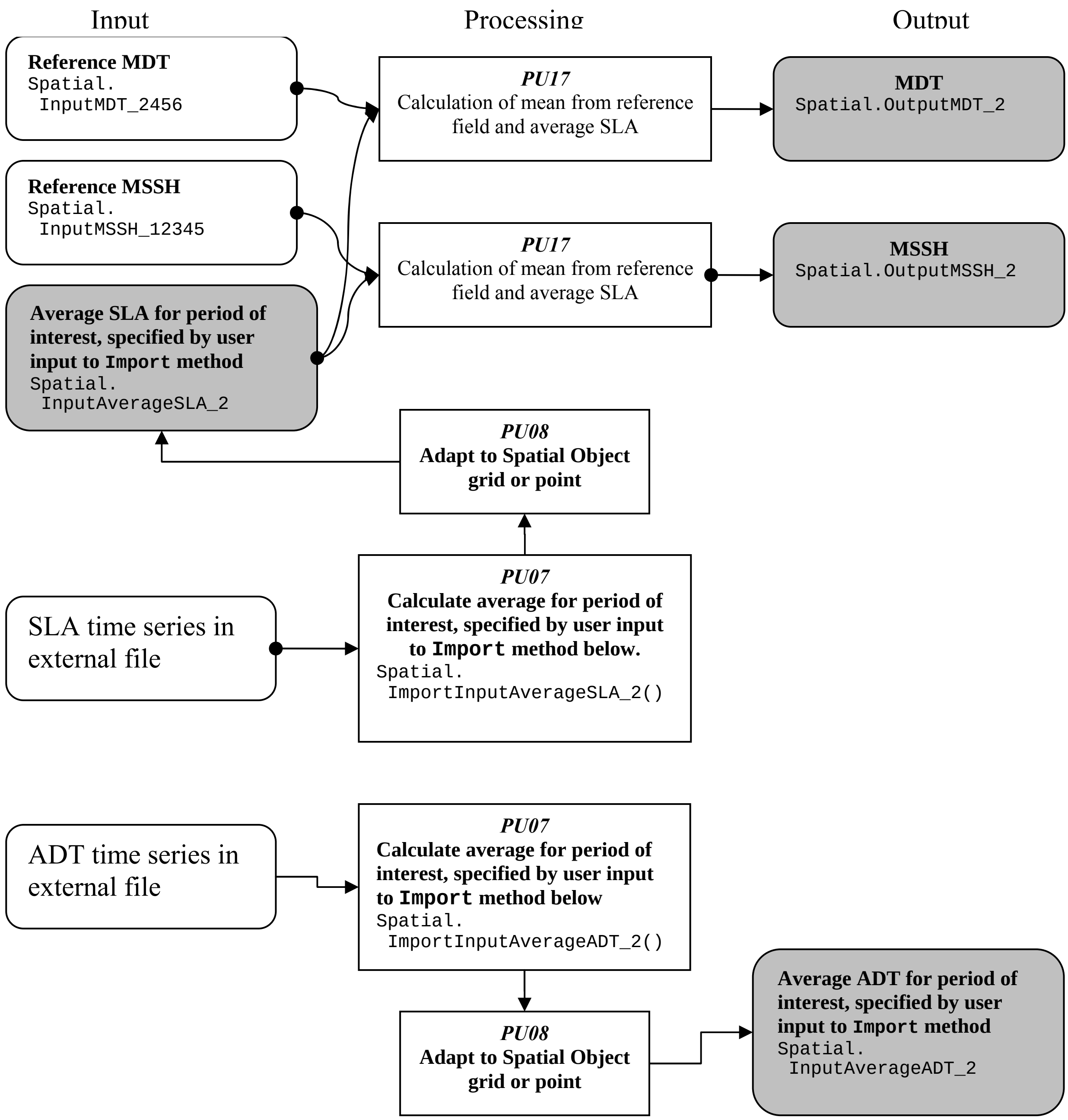




Data-flow 1a: Geoid and gravity field computation.

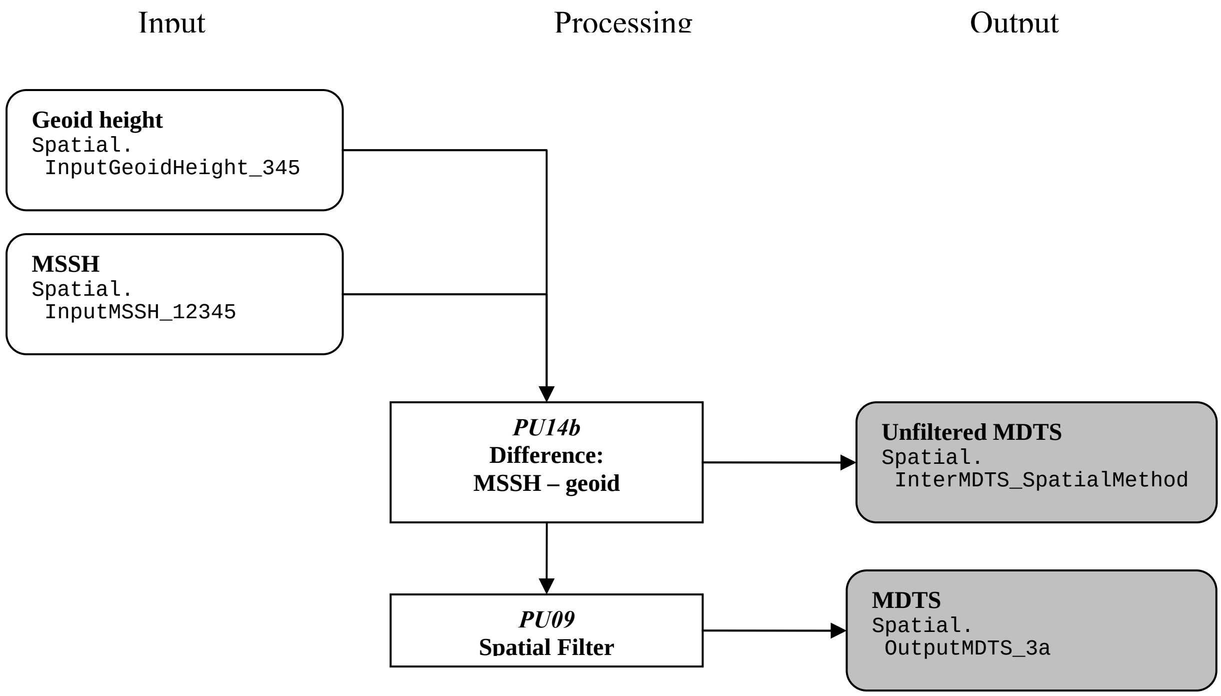


Data-flow 1b. Error computation for geoid and gravity field

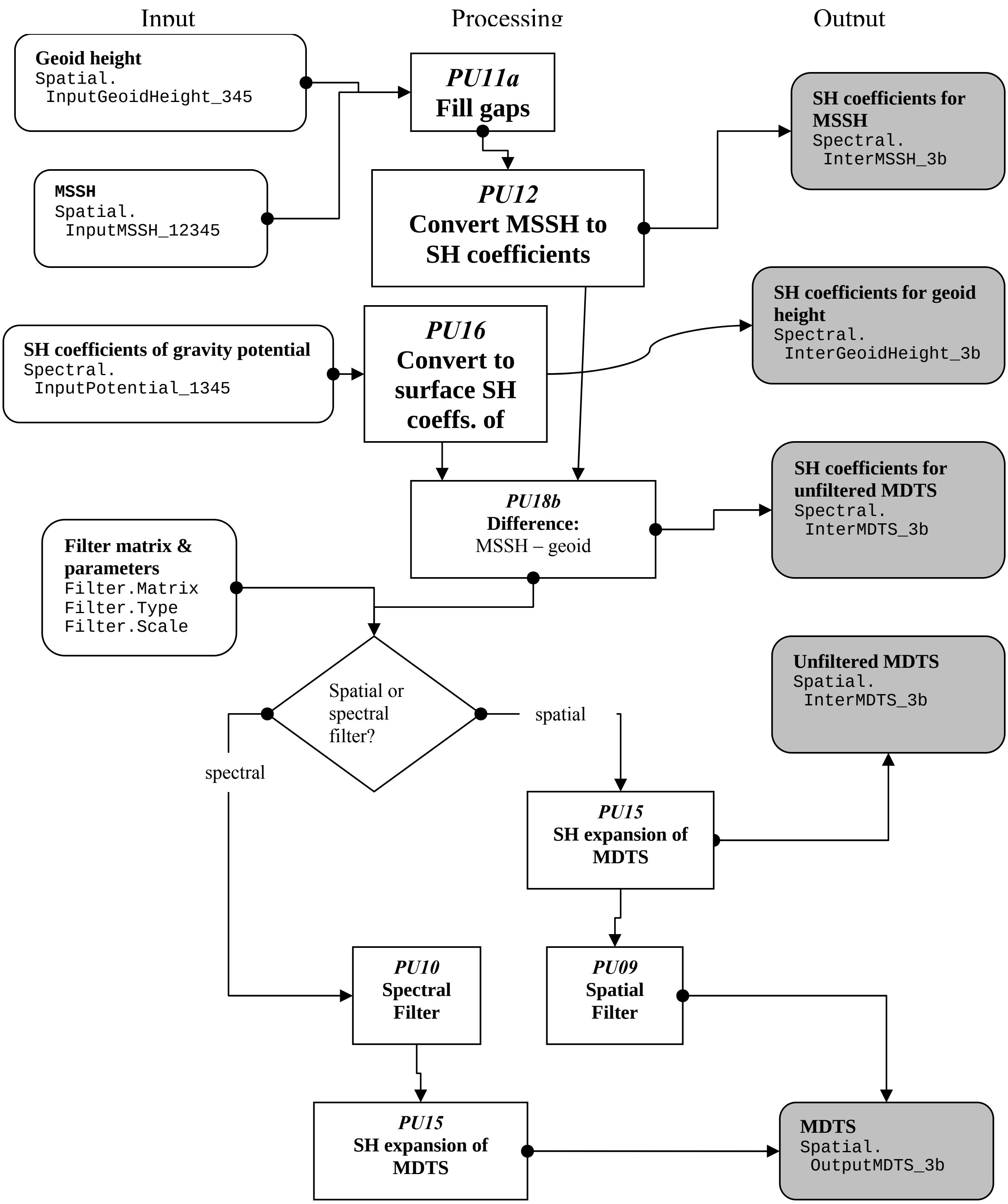


Data-flow 2. Sea surface height and a-priori MDT selection

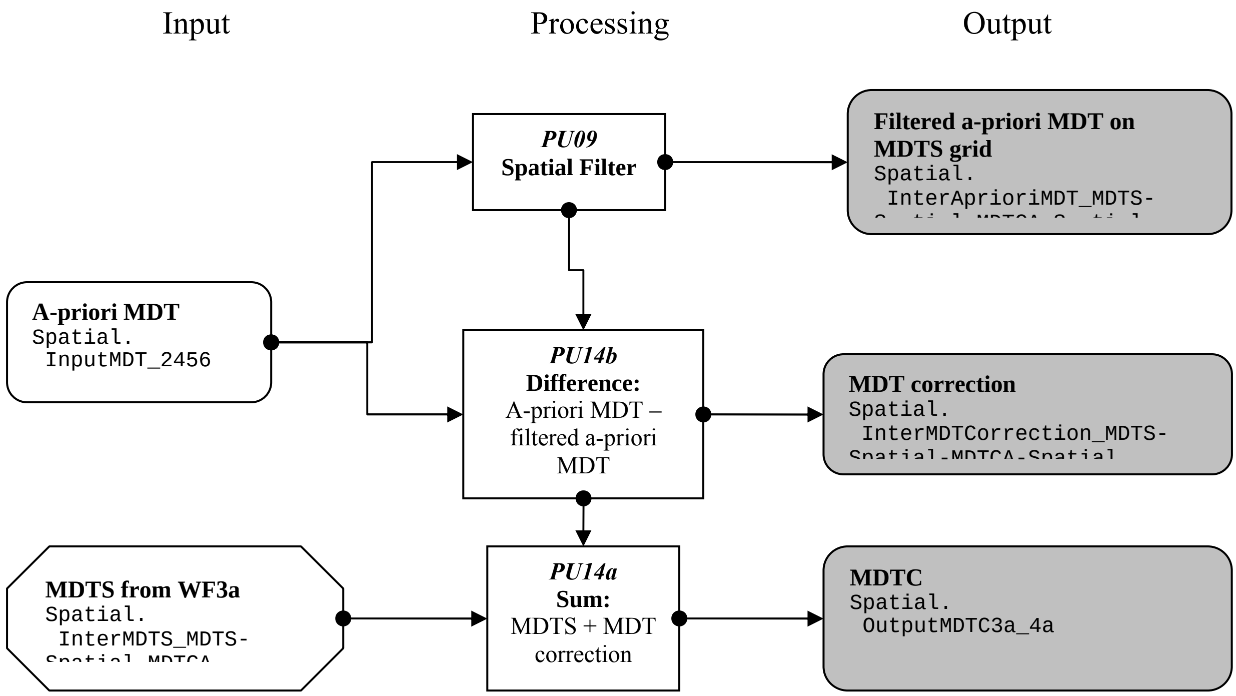
This data-flow diagram corresponds to Work-flow 2, which covers calculating the average for a period of a time series and also calculating a new mean field from the reference mean field and the average SLA in the period of interest. The import methods for average SLA and ADT perform averaging and grid adaptation. These work-flows are different to the others because two of the outputs are labeled as "Input" fields, to allow use of `Import` methods.



Data-flow 3a. Satellite Dynamic Topography computation in geographical space

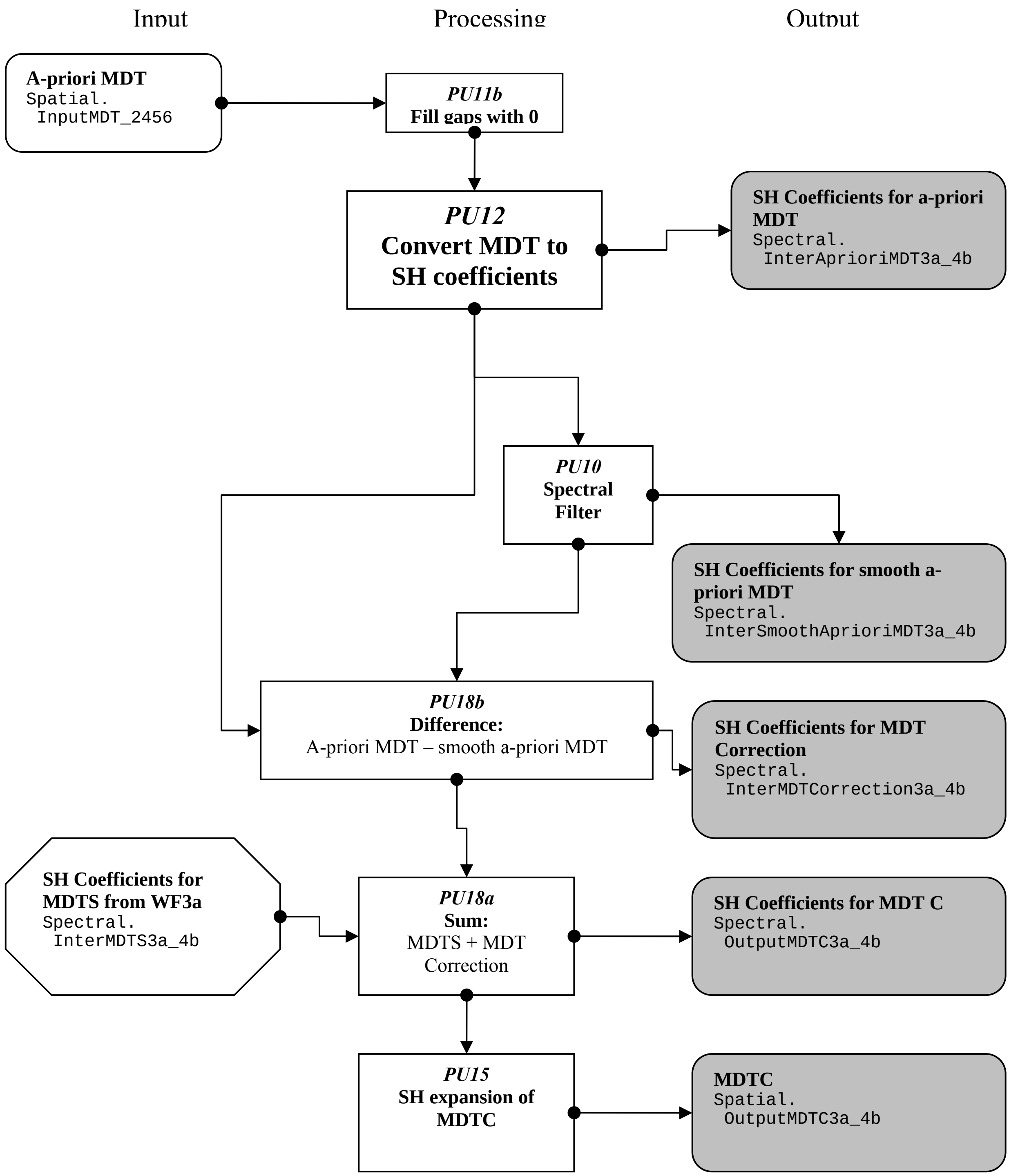


Data-flow 3b. Satellite Dynamic Topography computation in spectral space



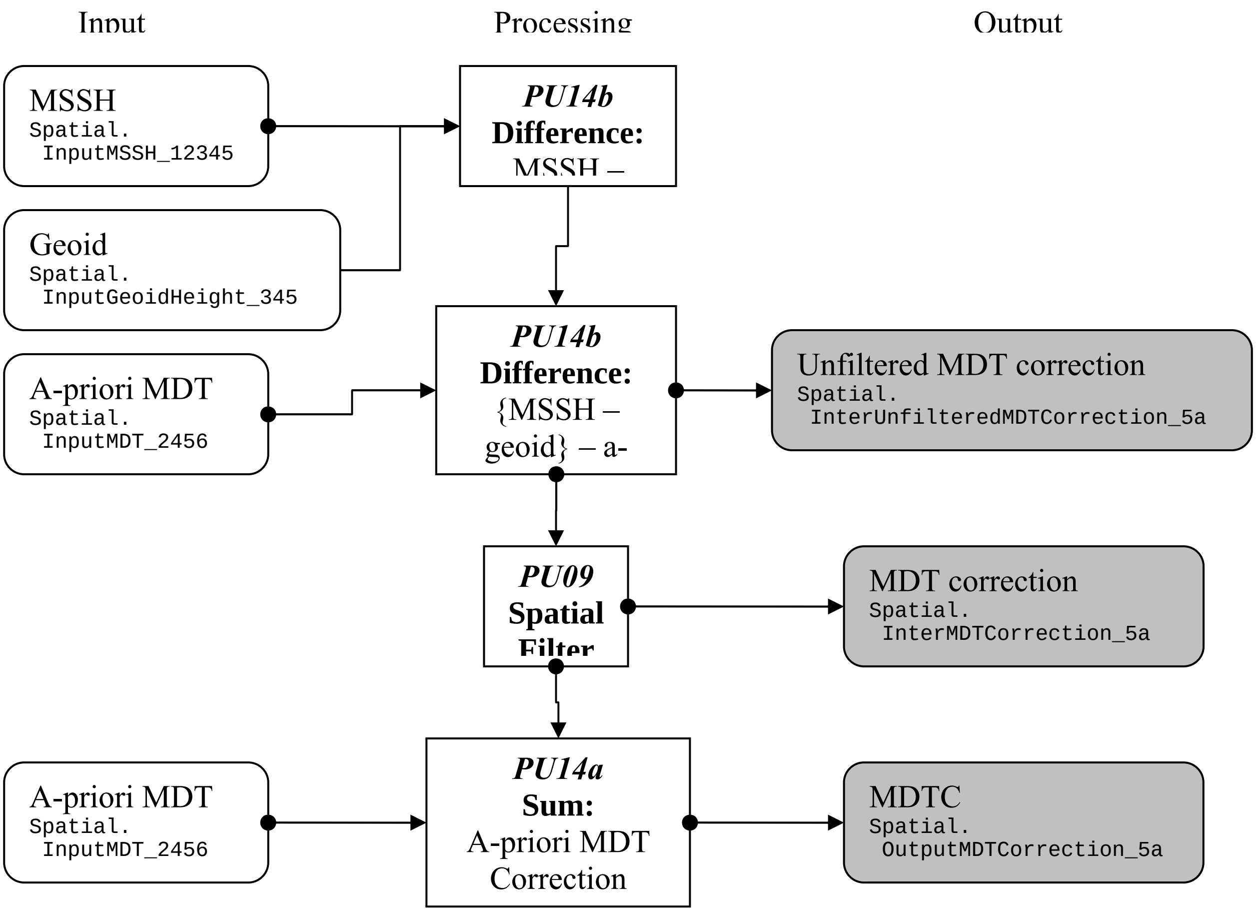
Data-flow 4a: Remove-Restore combined technique A: spatial filtering

The MDTS can be calculated using the method of work-flow 3a or the method of work-flow 3b. This diagram shows the data structures involved when work-flow 3a is used to calculate MDTS for work-flow 4a. If the user wishes to use the MDTS from work-flow 3b instead, the methods and data structures to be used are labeled with "3b_4a" instead of "3a_4a".

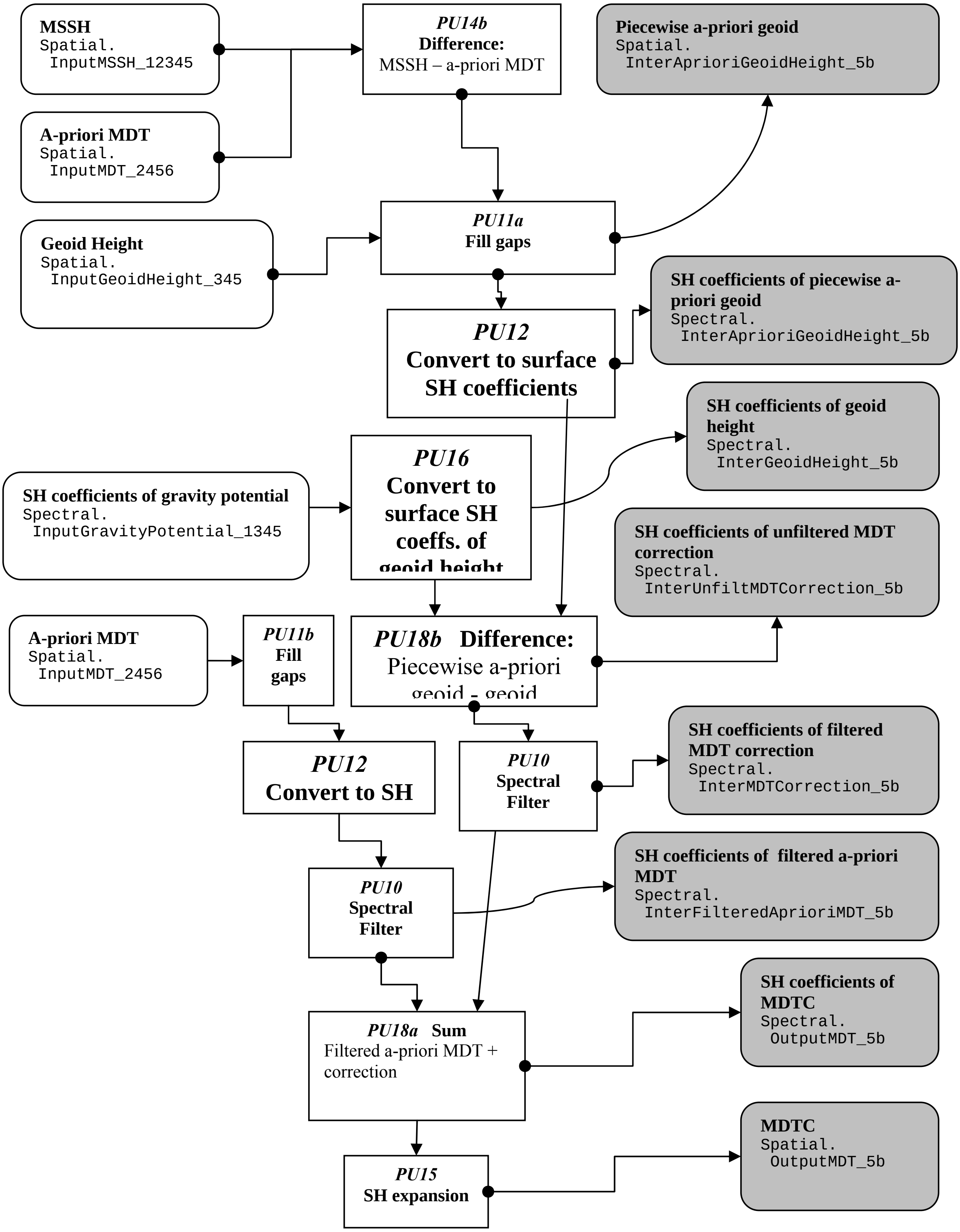


Data-flow 4b: Remove-Restore combined technique A: spectral filtering

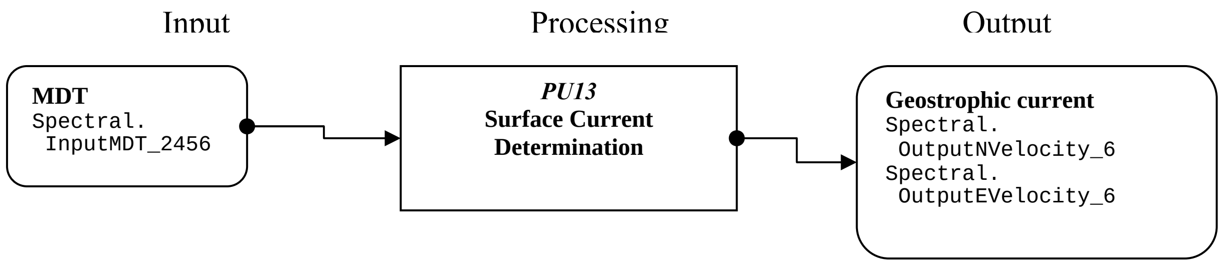
The MDTS can be calculated using the method of work-flow 3a or the method of work-flow 3b. This diagram shows the data structures involved when work-flow 3a is used to calculate MDTS



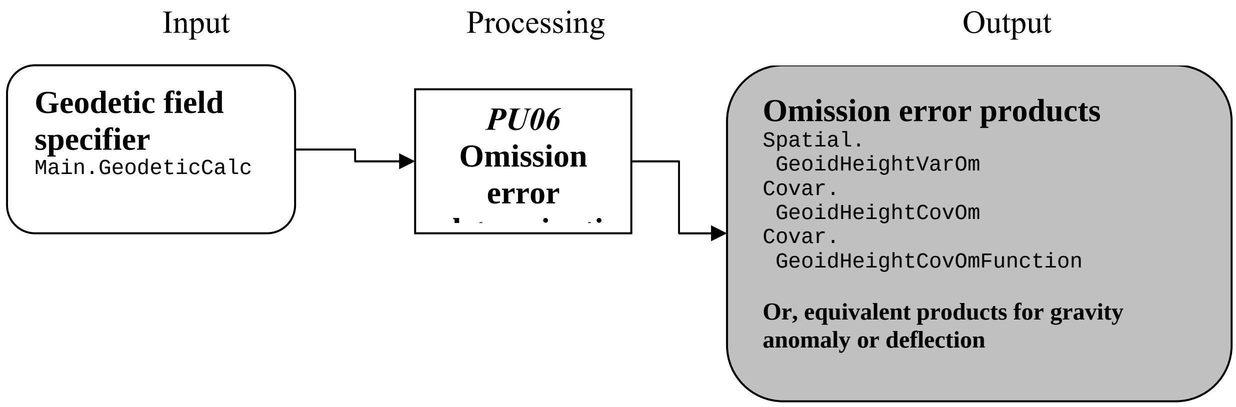
Data-flow 5a: Remove-Restore combined technique B: spatial filtering



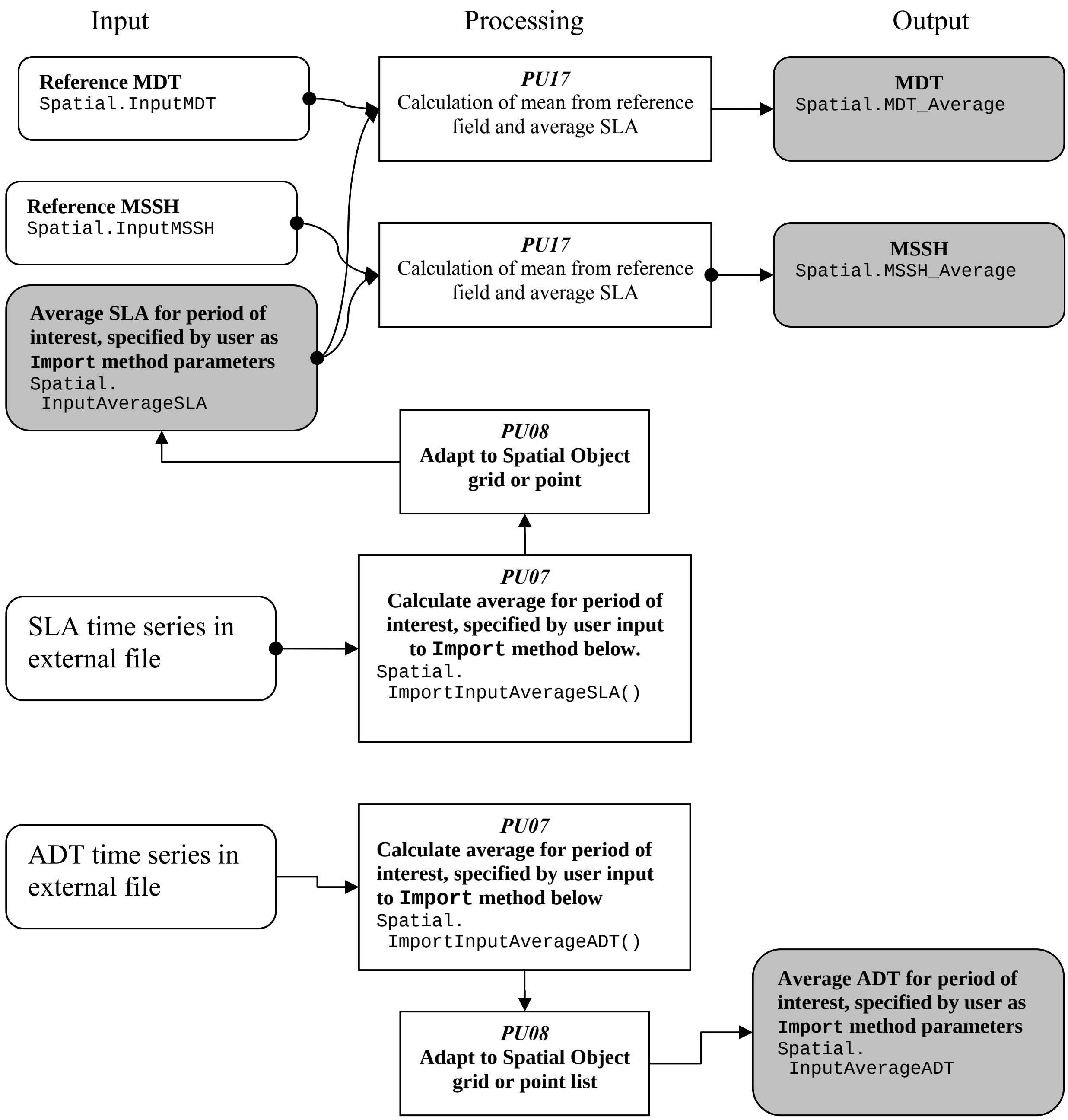
Workflow 5b: Remove-Restore combined technique B: spectral filtering



Data-flow 6: Dynamic Topography-derived quantities

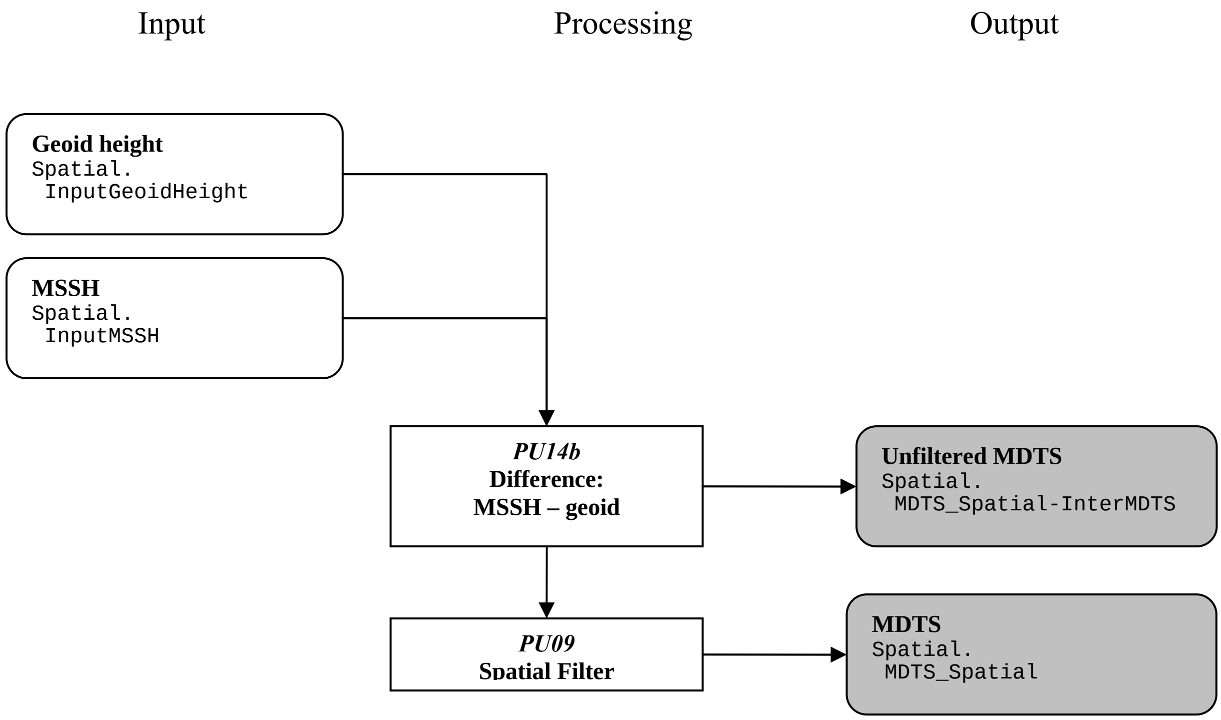


Data-flow 1b. Omission error computation for geoid and gravity field.

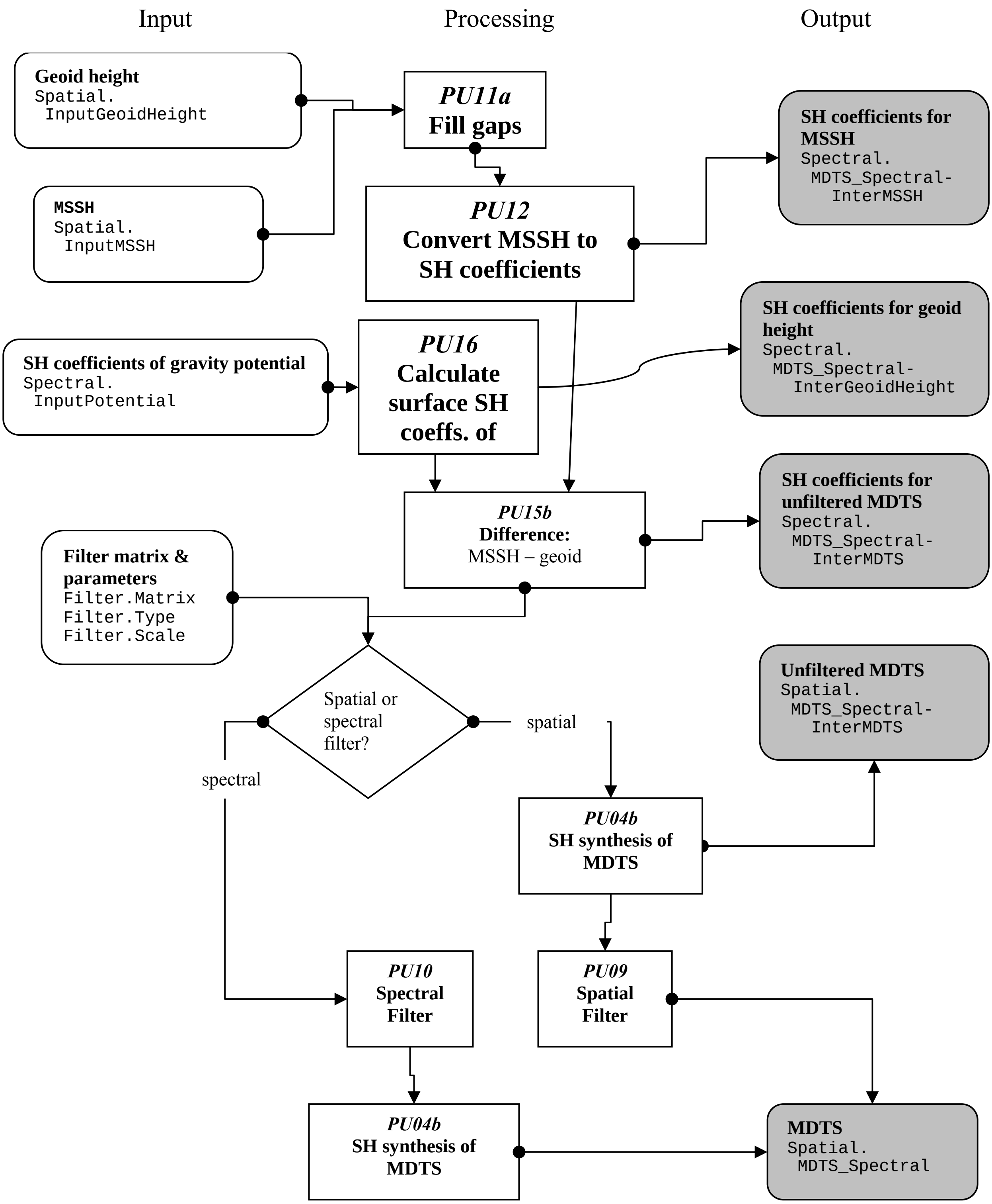


Data-flow 2. Sea surface height and a-priori MDT selection

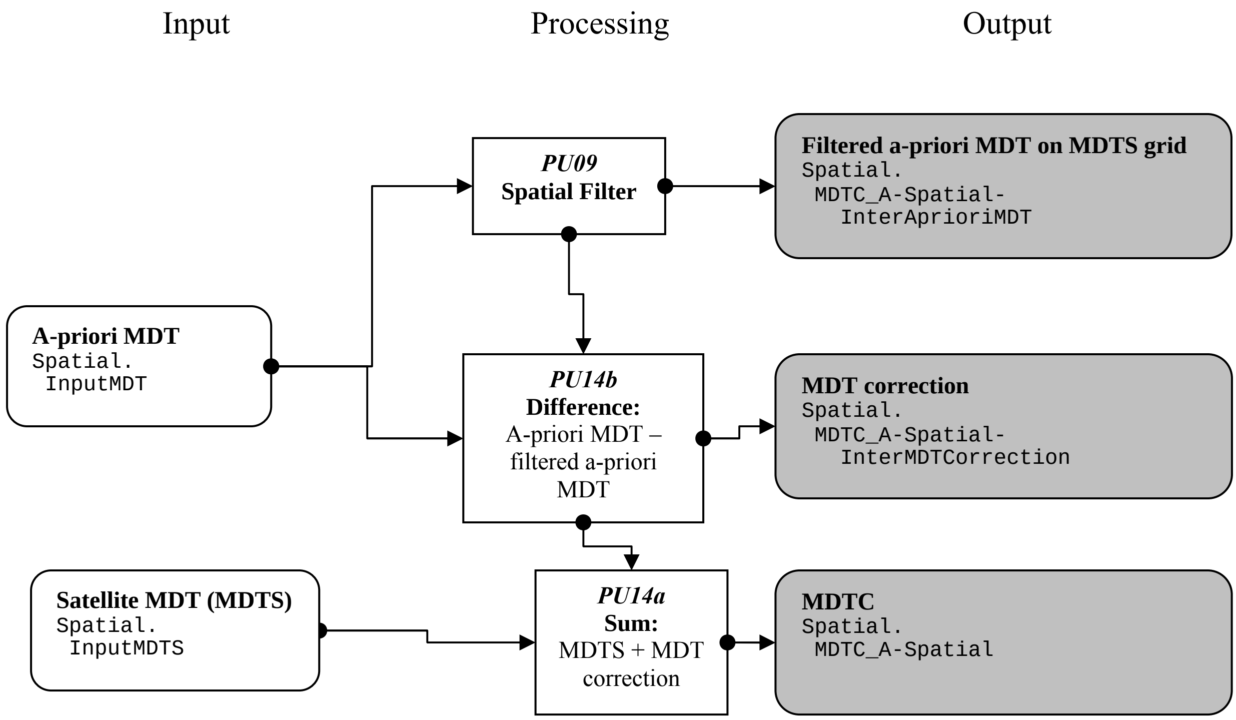
This data-flow diagram corresponds to Work-flow 2, which covers calculating the average for a period of a time series and also calculating a new mean field from the reference mean field and the average SLA in the period of interest. The Import methods for average SLA and ADT perform averaging and grid adaptation



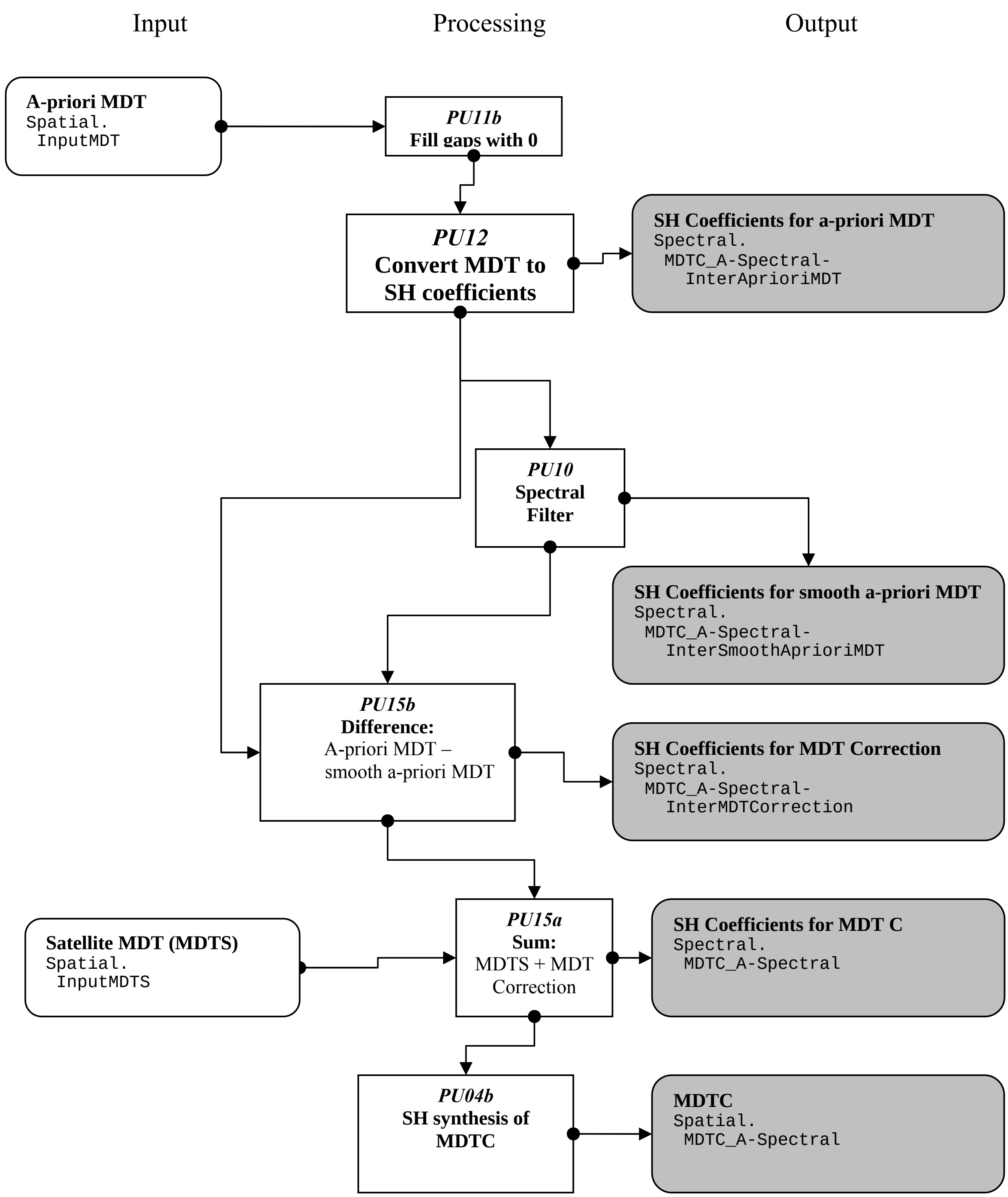
Data-flow 3a. Satellite Dynamic Topography computation in geographical space



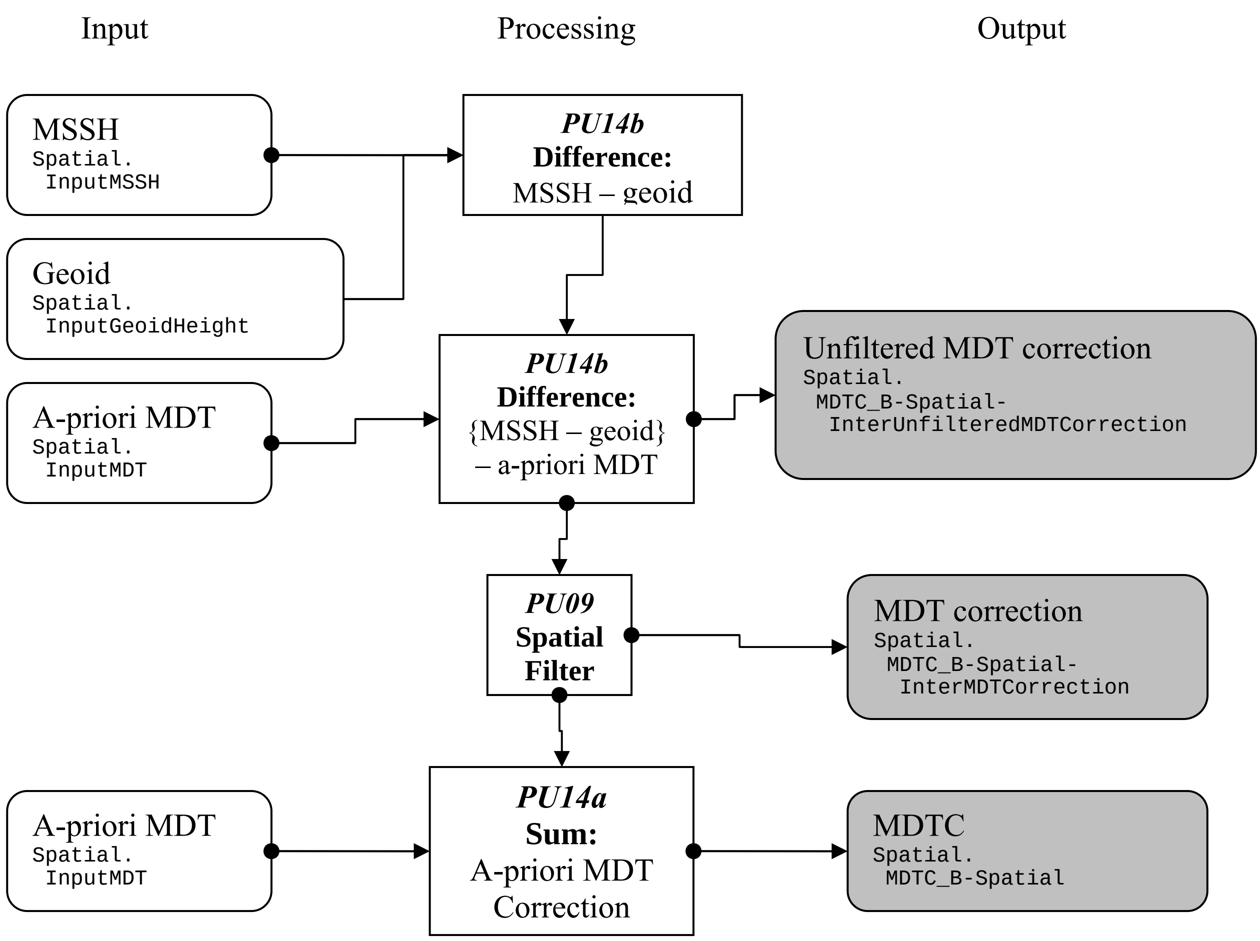
Data-flow 3b. Satellite Dynamic Topography computation in spectral space



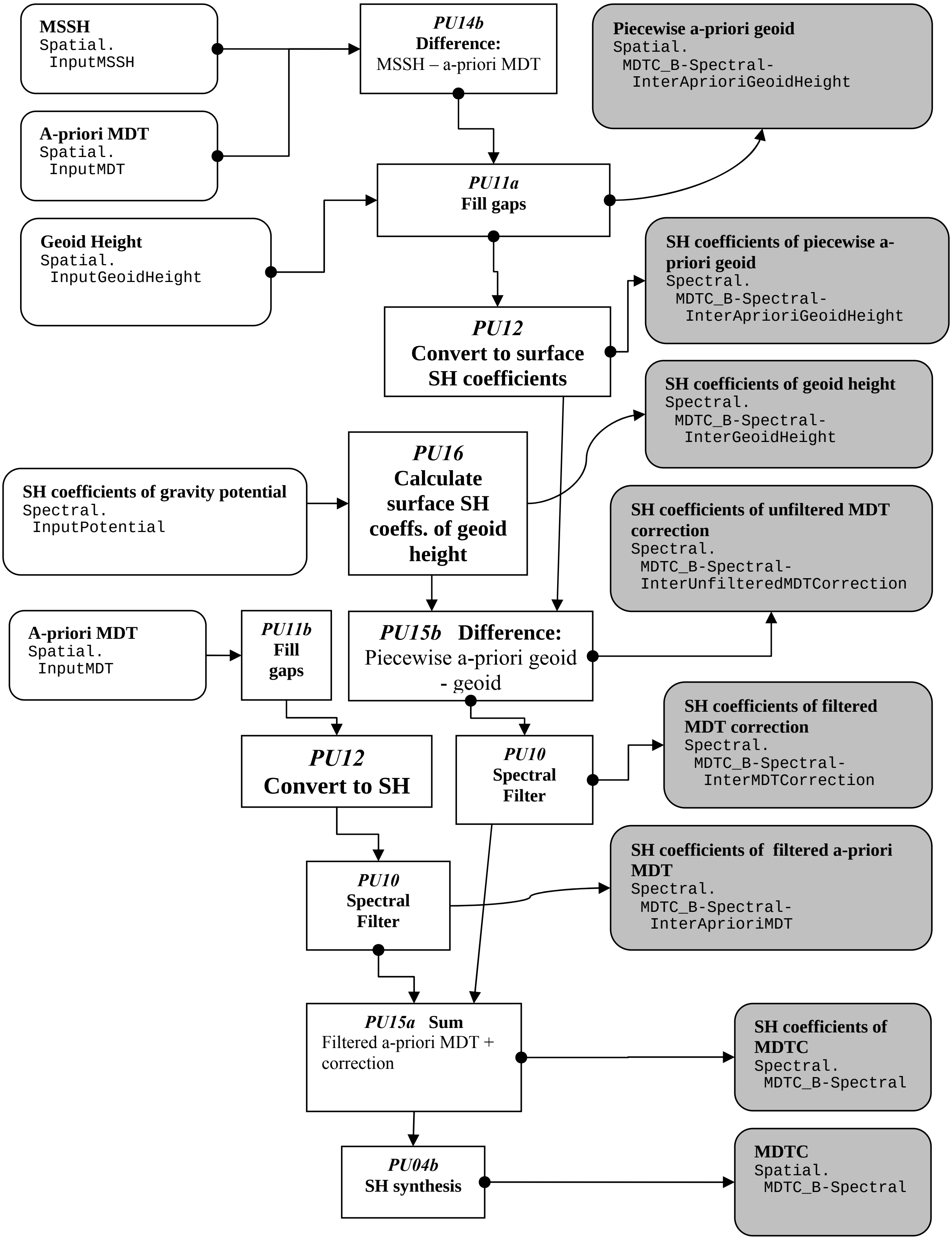
Data-flow 4a: Remove-Restore combined technique A: spatial filtering



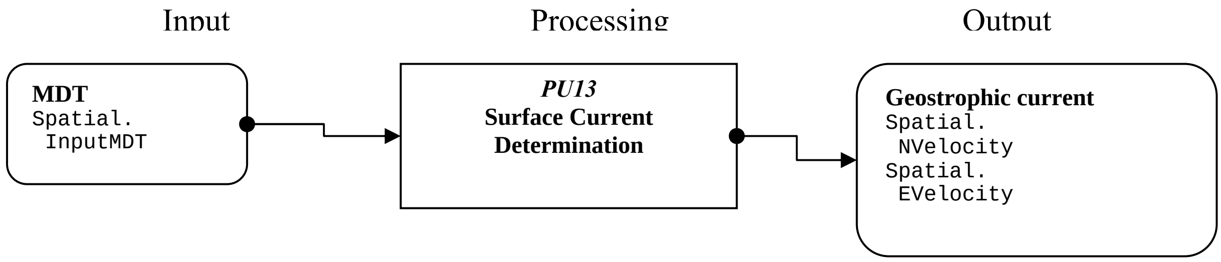
Data-flow 4b: Remove-Restore combined technique A: spectral filtering



Data-flow 4c: Remove-Restore combined technique B: spatial filtering



Workflow 4d: Remove-Restore combined technique B: spectral filtering



Data-flow 5: Dynamic Topography-derived quantities

4.8 List of Abbreviations

API	Application Program Interface
ASCII	American Standard Code for Information Interchange
BEAT	Basic Envisat Atmospheric Toolbox
BRAT	Basic Radar Altimetry Toolbox
CDAT	Climate Data Analysis Tool
DT	Dynamic Topography
ESA	European Space Agency
FTP	File Transfer Protocol
GDR	Geophysical Data Record
GMT	Generalised Mapping Tool
GOCE	Gravity Ocean Circulation Explorer
GUI	Graphical User Interface
GUT	GOCE User Toolbox
GUTS	GOCE User Toolbox Specification
HPF	High level Processing Facility
ICGEM	?
IDE	Integrated Development Environment
IDS	Internal Data Store
MDT	Mean Dynamic Topography
MDTC	Combined MDT
MDTS	Satellite MDT
MSSH	Mean Sea Surface Height
NetCDF	Network Common Data Form
RR	Remove-Restore
SH	Spherical Harmonic
SLA	Sea Level Anomaly
SSH	Sea Surface Height
VTK	Visualisation Toolkit
WF	Work-flow
WP	Work-plan

5 GUT Tutorials

The work realized during the one-year GUT Study has led to the writing and delivery of several reports that have been compiled into the GUTS final report and whose executive summary has been given in the previous introductory section. GUTS deliverables also include a collection of existing software that could, potentially, be used for the GUT construction and that can already act as a first GUT prototype, although they don't cover all GUTS identified functionalities.

The next section is dedicated to the description of the GUTS tutorial whose objective is NOT to give a description of this existing software but rather, in continuity and complement of the WP2000, 3000 and 4000 reports, to provide an overall vision of what the GOCE User Toolbox should look like, that can help later software engineers to construct the toolbox in a user friendly manner.

To achieve this goal, the tutorials have been divided into two main components:

The first part aims at providing general information on the scientific aspects covered by the toolbox, mainly the computation of an ocean mean dynamic topography. This part should help a novice user to better understand why all the toolbox functionalities are needed and why the default workflows have been recommended.

In the second part of the tutorials, concrete use cases have been associated with each function and each workflow identified in WP3000.

For clarity sake, and similarly to what has been done for WP4000 report, the tutorials are written as though GUT already exists.

Although they aim to be an important and useful input for the further construction of GUT, they should not be considered as a restrictive canvas for the writing of the future GUT tutorials.

5.1 Reminder of the GUT objectives

The objectives of the GOCE User Toolbox are mainly twofold:

- The first objective is directed towards a wide group of users from different communities (solid earth, oceanography…). The aim is to facilitate the handling of the GOCE Level-2 products that will be made available from the GOCE HPF (High Processing Facility). This means translating the HPF GOCE Level-2 products, mainly the set of spherical harmonics coefficients and the variance-covariance error matrix, into spatial representations of geophysical quantities (geoid height, gravity anomalies, deflection from the vertical) and their errors.
- The second objective is directed towards the oceanographic community through the computation of ocean mean dynamic topography, which, added to altimetric sea level anomalies, gives access to the ocean absolute dynamic topography and the corresponding ocean geostrophic circulation.

5.1.1 Altimetry and the Mean Dynamic Topography issue

Since the beginning of altimetric missions more than 15 years ago, the lack of an accurate geoid has been hampering the full exploitation of altimetric data for oceanographic studies. Only the variable part of the ocean dynamic topography can be extracted with sufficient accuracy (few centimetres) for oceanographic applications. The estimation of an accurate Mean Dynamic

Topography is mandatory for the correct interpretation of all past, present and future altimetric data and their use for oceanographic analyses. It has also proved to bring significant improvement for their assimilation into operational forecasting systems (Le Provost et al, 1999, Le Traon et al, 2002, results from the EU GOCINA project).

5.1.2 How to compute the ocean mean dynamic topography

The objective of this section is to give a brief outline of the main issues a user has to keep in mind when using GOCE data for oceanographic purposes. Further details on these issues can be found in (Hughes et al, 2006).
The ocean mean dynamic topography (MDT) for a chosen period is the difference between an altimetric mean sea surface (MSSH, computed for the chosen period) and a geoid model N:

$$MDT=MSS-N \qquad \text{Eq.1}$$

This apparently very simple equation is actually quite intricate because:

Altimetric mean sea surfaces and geoid models don't have the same spectral content. Typically, mean sea surfaces are known with a centimetric accuracy at spatial scales down to a few kilometres. On the other hand, the same accuracy on the geoid will be achieved using GOCE data at scales down to around 100 km (GO-ID-HPF-GS-0041). At the present time, this centimetric accuracy is achieved using GRACE data only at scales above around 400 km (Tapley et al, 2003). If a simple difference of the two fields is calculated, the resultant dynamic topography will contain high spatial resolution geoid information, from the altimetric MSSH, that is not included in the geodetic data, giving spurious circulation features. Hence, before subtracting a geoid from a MSSH, the two fields have to be filtered in order to achieve from both of them a similar spectral content. The filtering can be done either in geographical space or in spectral (spherical harmonic) space. In the latter case, the MSSH, which, by construction, is defined only over the oceans, needs to be completed over the continents in order to obtain a global field.

Both altimetric mean sea surface heights and geoid heights are given relative to a reference ellipsoid, which corresponds to a theoretical shape of the Earth. The characteristics of different, currently used, reference ellipsoids are given

Table 1. Before subtracting a geoid from a MSSH, both fields have to be expressed relative to the same reference ellipsoid. If not, the impact on the resulting MDT is large: Figure 1 shows the height differences between the GRIM and Topex ellipsoids on a global grid.

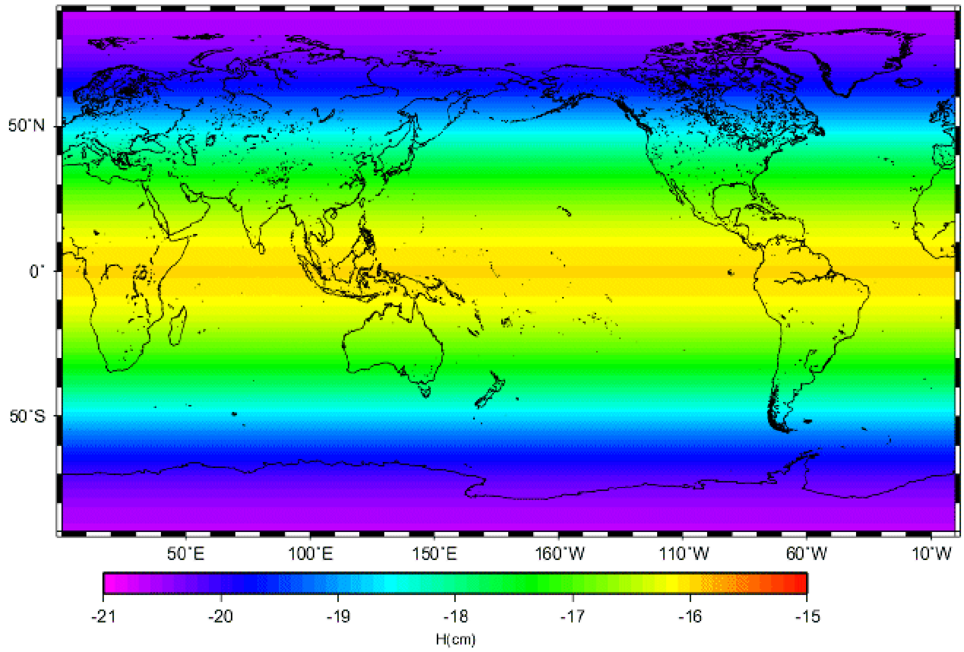


Figure 23: Height difference between the TOPEX and the GRIM ellipsoids.

Ellipsoid name	a (km)	1/f
“GRIM”	6378.13646	298.25765
“TOPEX”	6378.1363	298.257
“GRS80”	6378.137	298.257222101
“WGS84”	6378.137	298.257223563

Table 1: The different reference ellipsoids and their characteristics

Geoid heights (and mean sea surface heights) also differ depending on what tidal system is implemented to deal with the permanent tide effects. In the MEAN TIDE system, the effects of the permanent tides are included in the definition of the geoid. In the ZERO TIDE system, the effects of the permanent tides are removed from the gravity field definition. In the TIDE FREE or NON-TIDAL system, not only the effects of the permanent tides are removed but the response of the Earth to that absence is also taken into account. Altimetric mean sea surfaces are usually expressed in the MEAN TIDE system. The GRACE GGM02 geoids from the CSR are defined relative to the ZERO TIDE system. The GRACE EIGEN geoids from the GFZ are defined relative to the TIDE FREE system. When computing an ocean mean dynamic topography, the MSSH and the geoid first have to be computed in the same system. If not, the impact on the resulting MDT is large: for instance, Figure 24 shows the difference between the TIDE FREE and the MEAN TIDE reference systems.

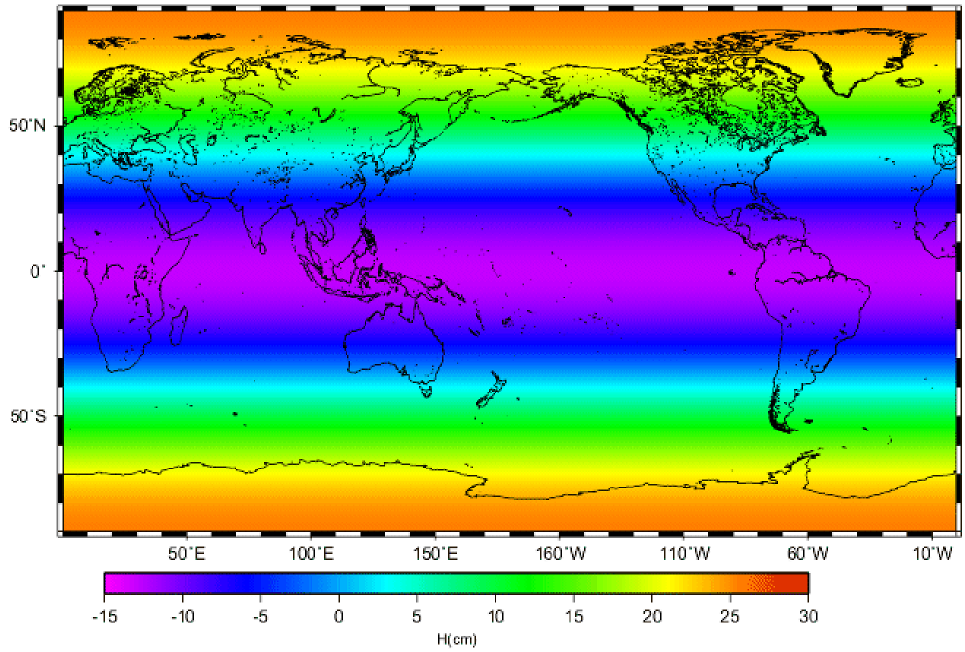


Figure 24: Height difference between the TIDE FREE and the MEAN TIDE reference systems

Once these three points have been taken into account and both the MSSH and the geoid have been adequately processed, the mean dynamic topography can be computed. This MDT product will hereafter been referred to as a “Satellite-only” MDT or MDTS.
By construction, the spectral content of the MDTS is limited by the spectral content of the geoid model. In the case of GOCE, the corresponding MDTS will thus have a centimetric accuracy at a 100 km resolution. In some areas of the world ocean, notably coastal areas, straits, semi-enclosed seas such as the Mediterranean Sea and close to steep bottom topography, the MDT is expected to contain signals at shorter spatial scales.
The GOCE User Toolbox hence provides the user with more sophisticated MDT computation techniques allowing to integrate short-scale information from other MDT sources. These techniques will be further referenced to as Remove-Restore techniques.

Two variants of a remove-restore “combined” technique are included in GUT. The first (method A) utilizes a high-resolution a-priori MDT, eg from hydrodynamic modelling or observations, to restore the small-scale structure in the ‘satellite’ MDTS. The filtered satellite solution here will be the output of the previous MDTS calculation and the filtering can be spatial or spectral – but the filtering of the a-priori MDT must be carried out in the same way.

The second variant (method B) takes the a-priori MDT as the basis and restores the large-scale structure by comparing the spectral equivalents of an a-priori geoid (based on the filtered difference of MSSH and a-priori MDT) and the GOCE geoid. This requires that we use the *unfiltered* version of MDTS (η_G, *i.e.* direct difference of MSSH – Geoid).

These two variants can be used for different purposes. Method A puts higher priority on the MDTS fields and assumes the high resolution features of the a-priori MDT are consistent with MDTS. The second method puts higher priority on the a-priori MDT and would be appropriate (*e.g.*) when using an ocean model for the a-priori to provide an improved model surface suitable for data assimilation fields, that was consistent with the ocean model dynamics and the GOCE geoid.

For both remove-restore variants, as well as for the MDTS calculation, the filtering required can be carried out spatially or spectrally. For better consistency, it is recommended that user remains within the same filtering space.

5.2 Using the toolbox

As recommended in the previous work packages, the GOCE User Toolbox should be designed so that it can be used at different levels, depending on the expertise and the needs of the user. The first level is the use of “workflows” allowing the computation of geoid/gravity field/MDT in one single step, with few inputs required. The toolbox is made of 6 workflows which are described in section 5.2.1

Example1:

User need:
Get a grid of GOCE geoid heights developed at degree/order 120 of spherical harmonic expansion

→ **Workflow 1a of the GOCE User Toolbox**

Example 2:

User need:
Compute a MDT on a ¼° regular grid, at 100 km resolution (SH200) filtering the GUT default geoid and MSSH data in the spectral domain

→ **Workflow 3b of the GOCE User Toolbox**

Each workflow is a succession of processes that can also be called independently by the user.

Example 3:

User need:
Interpolate a grid of GOCE geoid heights along an altimetric track

→ **Use the Grid Interpolation routine of the GOCE User Toolbox**

This is the “single step” approach for which the different available functions are described in section 5.2.2.
Furthermore, many single functions may be called successively, providing an even more complex and flexible processing tool. For instance, when the grid of GOCE geoid heights at 100 km resolution will be available, a GUT user may want to compare it to the latest GRACE geoid model available. He/She will be able to do it through the succession of four GUT functions.

Example 4:

User need:
Compare the default grid of GOCE geoid heights with a user provided grid of GRACE geoid heights

Four steps:
1- Compute the GOCE geoid relative to the same reference ellipsoid as the user provided geoid
2- Compute the GOCE geoid relative to the same tide system as the user provided geoid
3- Compute the GOCE geoid on the same grid as the user provided geoid
4- Compute the difference between the two geoids

This is the "step by step" approach, described in section 5.2.3
In the following sections (5.2.1 to 5.2.3), concrete examples or "use cases" will be given for each of the three approaches. In order to make an explicit link to WP3000, where the workflows and the GUT functionalities have been defined as well as the different input/output, we associate to each use case a "command line" like phrase containing the algorithm/workflow nomenclature followed by a list of options (in red), a list of input files (in yellow) and a list of output files (in green). For the algorithms, the nomenclature defined in WP3000 (GUT_FAXX) is used. Consistently, a GUT_WFXX nomenclature is used for the workflows.

The detailed description of all input and output formats can be found in the WP2000 report (User Toolbox Requirements Document), section "Required functionality, input and output parameters" as well as in the WP3000 report (Toolbox functionality and algorithm specification document), section " Input Output Definition".

The detailed description of all options relative to the different functionalities can be found in the WP3000 report (Toolbox functionality and algorithm specification document), section " Algorithm Specification".

5.2.1 The workflow approach

The main workflow associated with the GOCE User Toolbox is shown Figure 25.

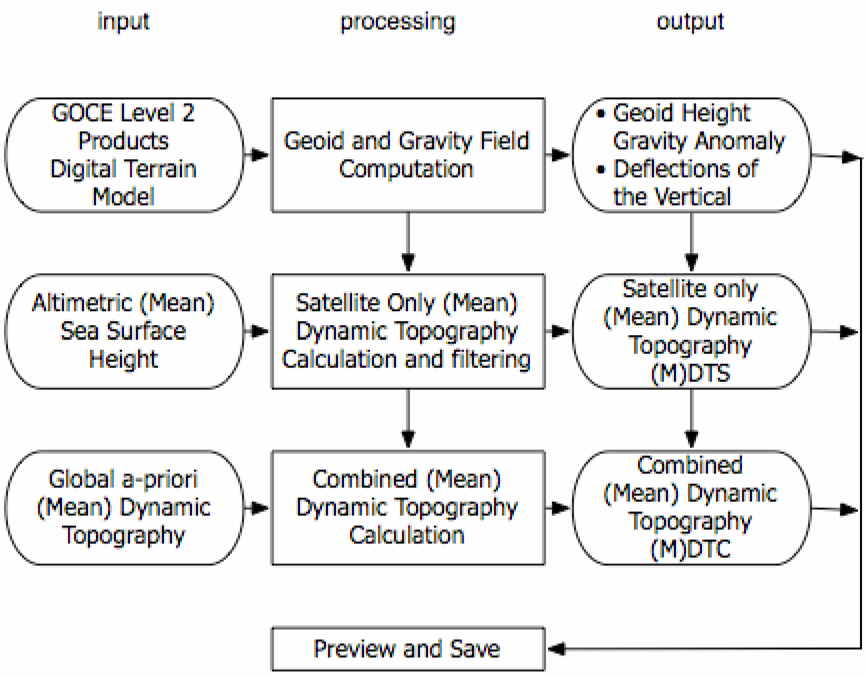


Figure 25: GUT main workflow

Through the use of this workflow, the user has access to the three main outputs of the toolbox, namely geodetic fields (geoid height, gravity anomaly, deflections of the vertical), a satellite-only mean dynamic topography and a combined mean dynamic topography. All products are computed using the default procedures and parameters recommended by the GUTS expert team. For instance, the MDTS is computed in spectral space using a Jekeli filter with a default filter width (that will depend on the GOCE data and is therefore not defined yet – around 100 km). All outputs are gridded fields (1/2° resolution, regular).
When used with the default input fields (MSSH and a-priori MDT) provided with the toolbox, the default MDTS and MDTC are obtained.
This main workflow can be split into 6 sub-workflows that can be called directly by the user for increased flexibility.

Workflow 1

Workflow 1 concerns the computation of different geodetic fields (workflow 1a - Figure 26) and their errors (workflow 1b - Figure 28).
A number of parameters can be defined by the user depending on the desired resolution of the output field (maximum degree and order of spherical harmonic expansion), the reference system he/she would like the output field to be expressed in (reference ellipsoid, tide system), the representation type (area average, single points), the output type (on grid or at user defined points)…

Workflow 1a: The geoid and gravity field computation

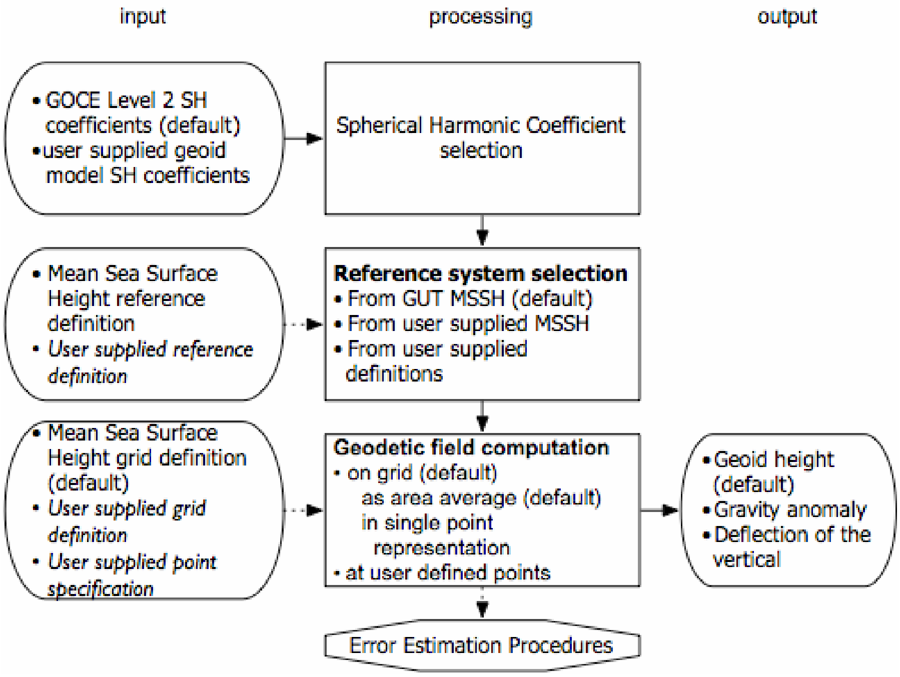


Figure 26: Workflow 1a: The geoid and gravity field computation

Workflow 1a should be used with the following parameters:

Options:
-Fg: geoid height computation (default)
-Fa: gravity anomaly computation
-Fd: geoid deflections from the vertical
-R *e/w/n/s*: Region of interest in the order western, eastern, northern and southern limits in degrees north of the equator and east of the prime meridian (default:79.75/-79.75/0.25/359.75). Only used for regular (latitude-longitude) grid.
-I *dx/dy*: grid increment for regular (latitude-longitude) grid in degrees (default:0.25/0.25)
-Gg *grid_definition_file_x/grid_definition_file_y*: An irregular grid is used. The files contain matrices with latitude/ longitude definitions of the grid. The -I and -Roptions are ignored
-Gu *specific_points_definition_file*: Specific points are defined by a list given in the file. Two columns are expected containing the longitude in the first and the latitude in the second column. The -I and -R options are ignored.
-H *digital_terrain_model*: The geodetic field is computed on the terrain as defined in *digital_terrain_model*. By default the reference ellipsoid is taken. A default terrain model is part of the GUT distribution and is used if no filename is given. -H is ignored if -Fg is selected.
-Ma: The area average for the grid cell is computed. By default the geodetic field is computed on the specified grid points. -Ma is ignored when using -Gu.

Input:
- Data set with spherical harmonic coefficients (GUT default data: EGM_GCF_2)
-SH: Maximum degree & order of the expansion used in the geodetic field computation
-EllipsOut: Reference system selection
-TideOut: tide system selection

Output:
For grids:
- Depending on output parameter:
  -Fg: 2D matrix (grid) of geoid (m)
  -Fa: 2D matrix (grid) of gravity anomaly (mgals)
  -Fd: 2D matrix (grid) of the E-W deflections from the vertical and 2D matrix (grid) of the N-S deflections from the vertical (arcsecs).
- Locations of grid nodes as 2 vector arrays of latitudes and longitudes (latitudelongitude
grid) or as two matrices (irregular grid) in degrees.
- Flag reflecting whether point-wise or area averaged values (-Ma option) have been calculated
For User Specified Points (-Gu):
- Depending on output parameter:
  -Fg: Vector of geoid at each data point (m)
  -Fa: Vector of gravity anomaly at each data point (mgals)
  -Fd: Vector of the E-W deflections from the vertical and vector of the N-S deflections from the vertical at each data point (arcsecs).
- Locations of data points as 2 vector arrays of latitudes and longitudes (degrees).
For All Cases:
- Maximum degree and order and approximate associated spatial scale (in spatial dimensions, ie km or degrees lat) of the calculation.
- Reference ellipsoid definition of the calculated field
-For gravity anomaly or deflections from the vertical the actual height above the reference ellipsoid for the calculation.

Example 1a:
An example of how to use workflow 1a is given below. The spherical harmonics coefficients of the EIGEN-GL04S1 GRACE model computed by the GFZ (http://www.gfz-potsdam.de/pb1/op/grace/results/) are used to compute a ½° resolution grid of geoid heights above the TOPEX reference ellipsoid and in the mean tide system at a 500 km resolution (SH expansion= degree and order 40).

GUT_WF1a

–Fg –R 89.75/-89.75/0.25/359.75 –I 0.5/0.5
–EllipsOut=TP –TideOut=MEAN –SH=40
EIGENGL4S_SH150_coef.fic
EIGENGL4S_SH40_Etp_MT_grid.fic

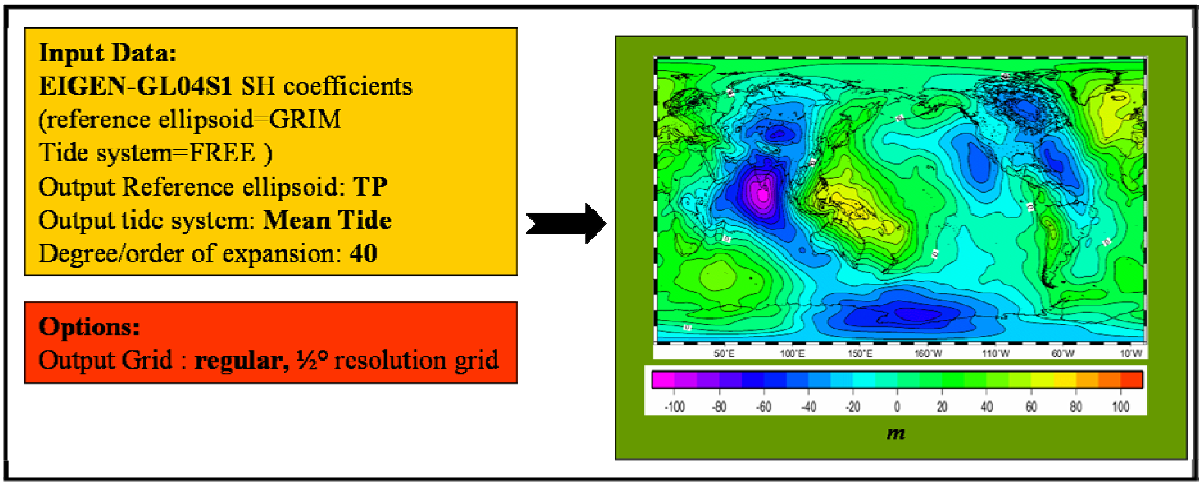


Figure 27: Example 1a

Workflow 1b: Error computation for geoid and gravity field

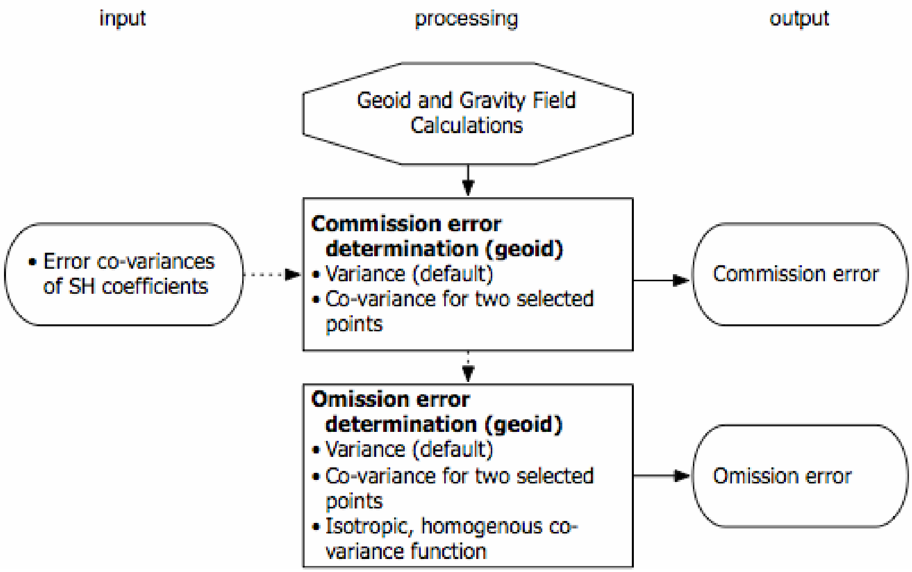


Figure 28: Workflow 1b

Following the recommendations from the previous work packages, this workflow is limited in the first GUT version to functionalities not requiring the storage and handling of the SH coefficients variance/covariance matrix. This means that concerning the commission error, only the gridded commission error variance as provided by HPF as Level-2 product and further interpolated to the required grid or data points, will be available for display. As for the omission error, it is estimated using a model of the expected power spectrum for all spherical harmonics higher than the maximum degree and order included in the spherical harmonic synthesis.

However, the full description of the workflow is detailed below.

Workflow 1b should be used with the following parameters:

Options:
-Om : Omission error computation
-Com : Commission error computation
-Fg: geoid height error computation (default)
-Fa: gravity anomaly error computation
-Fd: geoid deflections from the vertical error computation
-R e/w/n/s: Region of interest in the order western, eastern, northern and southern limits in degrees north of the equator and east of the prime meridian (default:79.75/-79.75/0.25/359.75). Only used for regular (latitude-longitude) grid.
-I dx/dy: grid increment for regular (latitude-longitude) grid in degrees (default: 0.25/0.25).
-Gg grid_definition_file_x/grid_definition_file_y: An irregular grid is used. The files contain matrices with latitude/ longitude definitions of the grid. The -I and -R options are ignored.
-Gu specific_points_definition_file: Specific points are defined by a list given in the file. Two columns are expected containing the longitude in the first and the latitude in the second column. The -I and -R options are ignored.
-H digital_terrain_model: The error field is computed on the terrain as defined in digital_terrain_model. By default the reference ellipsoid is taken. A default terrain model is part of the GUT distribution and is used if no filename is given. -H is ignored if -Fg is selected.
-Ma: The area average for the grid cell is computed. By default the geodetic field is computed on the specified grid points. -Ma is ignored when using -Gu.
-C lon1/lat1/lon2/lat2: The error co-variance is computed for the two points defined by lon1/lat1 and lon2/lat2 (longitude/ latitude in degrees). By default only the variances are computed (see options -R, -I, -Gg, -Gu).
-F d1/d2/dx: Applicable only if the -Om option has been put.
The omission error is provided as an isotropic, homogenous function of distance. The function is evaluated in the intervall [d1,d2] using dx as increment. By default only the variances are computed (see options -R, -I, -Gg, - Gu).

Input:
- Data set with error-covariance for spherical harmonic coefficients used in geodetic field computation (GUT default data: EGM_GVC_2)
-**SH**: Maximum degree & order of the expansion used in the geodetic field computation
-**EllipsOut**: Reference system selection
-**TideOut**: tide system selection

Outputs
For grids:
- Depending on geodetic parameter:
  - **-Fg:** 2D matrix (grid) of geoid error variance (m2)
  - **-Fa:** 2D matrix (grid) of gravity anomaly error variance (mgal2)
  - **-Fd:** 2D matrix (grid) of the error variance in the E-W deflections from the vertical and 2D matric (grid) of the error variance in the N-S deflections from the vertical (arcsec2).
- Locations of grid nodes as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.
- Flag reflecting whether point-wise or area averaged values (-Ma option) have been calculated

For User Specified Points (-Gu):
- Depending on output parameter:
  - **-Fg:** Vector of geoid error variance at each data point (m2)
  - **-Fa:** Vector of gravity anomaly error variance at each data point (mgal2)
  - **-Fd:** Vector of the error variance in the E-W deflections from the vertical and vector of the error variance in the N-S deflections from the vertical at each data point (arcsec2).
- Locations of data points as 2 vector arrays of latitudes and longitudes (degrees).

If co-variances are computed (-C):
- 2x2-matrix containing variances and co-variance of the two specified points
- Locations of the two data points.

If an isotropic homogenous co-variance function is provided for the omission error computation (-Om -F):
- Vector of error co-variances
- Vector of distances corresponding to the error co-variances.

For All Cases:
- Maximum degree and order and approximate associated spatial scale (in spatial dimensions, ie km or degrees lat) of the calculation.
- Reference ellipsoid definition of the calculated field
- For gravity anomaly or deflections from the vertical - the actual height above the reference ellipsoid for the calculation.

Example 1b:
In this example, the cumulative error variance of the EIGEN-GL04S1 geoid is computed for degree 50 of spherical harmonic expansion. The obtained error field ranges between zero (in blue on the colour scale) to three millimetres (in pink on the colour scale).

GUT_WF1b

–Fg –R 89.75/-89.75/0.25/359.75 –I 0.5/0.5
–EllipsOut=TP –TideOut=MEAN –SH=40
EIGENGL4S_SH150_cov_mat.fic
EIGENGL4S_SH50_Etp_MT_com_error_grid.fic

Input Data:
EIGEN-**GL04S1** error covariance matrix of SH coefficients
(reference ellipsoid=GRIM
Tide system=FREE)
Output Reference ellipsoid: **TP**
Output tide system: **Mean Tide**
Degree/order of expansion: **50**

Options:
Output Grid : **regular, ½° resolution grid**

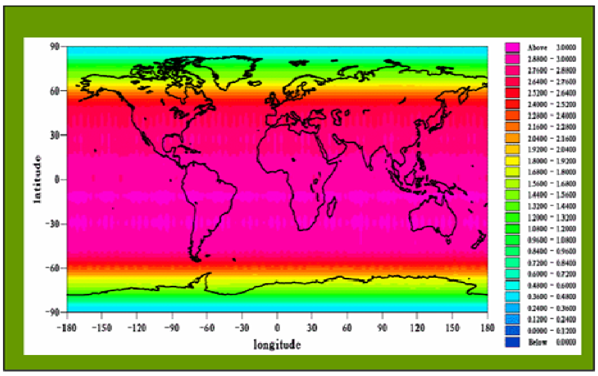


Figure 29: Example 1b

Workflow 2: Sea surface height and a-priori MDT selection

Workflow 2 concerns the selection of the sea surface heights and the a-priori MDT (Figure 30). It is composed of an averaging routine, (in order to compute a mean dynamic topography from a series of time varying dynamic topography fields or to compute a MSSH from a series of time varying sea surface heights) and an adaptation routine in order to interpolate the mean field onto a user specified grid or list of points. It is worth noticing that this functionality could be devolved to BRAT as long as the implemented version of GUT can input the BRAT output data

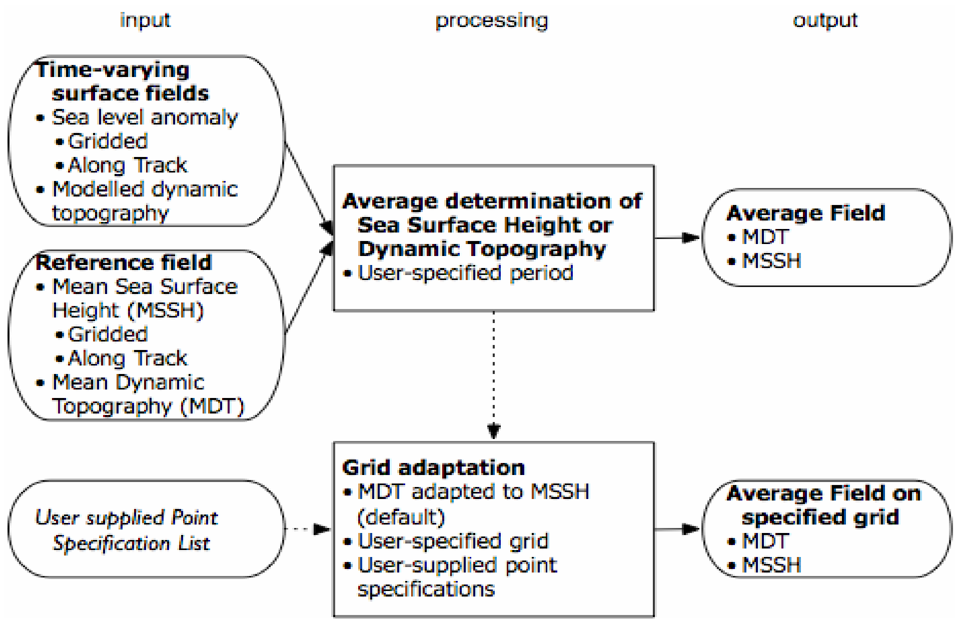


Figure 30: Workflow 2

This workflow should be used with the following parameters:

Options:
-R ***e/w/n/s***: Region of interest in the order western, eastern, northern and southern limits in degrees north of the equator and east of the prime meridian (default: 79.75/-79.75/0.25/359.75). Only used for regular (latitude-longitude) grid.
-I ***dx/dy***: grid increment for regular (latitude-longitude) grid in degrees (default: 0.25/0.25)
-Gg ***grid_definition_file_x/grid_definition_file_y***: An irregular grid is used. The files contain matrices with latitude/ longitude definitions of the grid. The -I and -R options are ignored
-Gu ***specific_points_definition_file***: Specific points are defined by a list given in the file. Two columns are expected containing the longitude in the first and the latitude in the second column. The -I and -R options are ignored.
-Tbeg: First day to consider for the averaging time period
-Tend: Last day to consider for the averaging time period

Input:
Time series of 2D matrices or vectors of parameters to be calculated

Output:
Returned mean value for the specified period at each location specified in cption by the user. Default: the averaged field is given at the same locations than the input fields and the average is done on the entire input serie.

Example 2:
In the example below (Figure 31), a mean dynamic topography of the Mediterranean Sea is computed from 7 years (1993-1999) of dynamic topography outputs from the MFSTEP model.

GUT_WF2 (GUT_FA07)
ADT_MFSTEP_time_serie_1993_1999.list
MDT_MFSTEP_1993_1999.fic

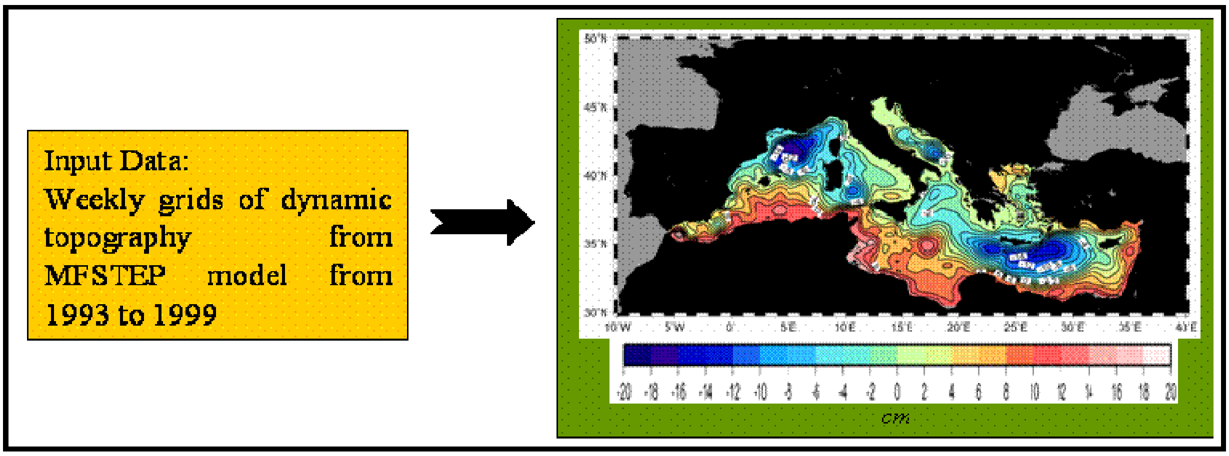


Figure 31: Example 2

Workflow 3: Satellite Dynamic Topography Computation

Workflow 3 is dedicated to the computation of satellite-only mean dynamic topographies either in the space domain (Workflow 3a - Figure 32) or in the spectral domain (Workflow 3b -

Figure 34). A number of parameters can be defined by the user, depending on the selection of filter type and filter width. The different filter types available are described in WP3000 report. The computation can be done with user supplied geoid heights and MSSH heights or using the GUT default files. The consistency between the reference frames of the two surfaces (reference ellipsoid and tide system) is checked automatically inside the workflow and the reference frames are homogenized if necessary.
Following the conclusions of the trade-off study realized in WP3000, the use of the gaussian filter is recommended for filtering in the space domain. Also, although two workflows are available, it is highly recommended to use Workflow 3b (filtering in the spectral domain) when computing MDTS for global grids (See also Bingham et al, 2007).

Workflow 3.a: MDTS computation in the space domain

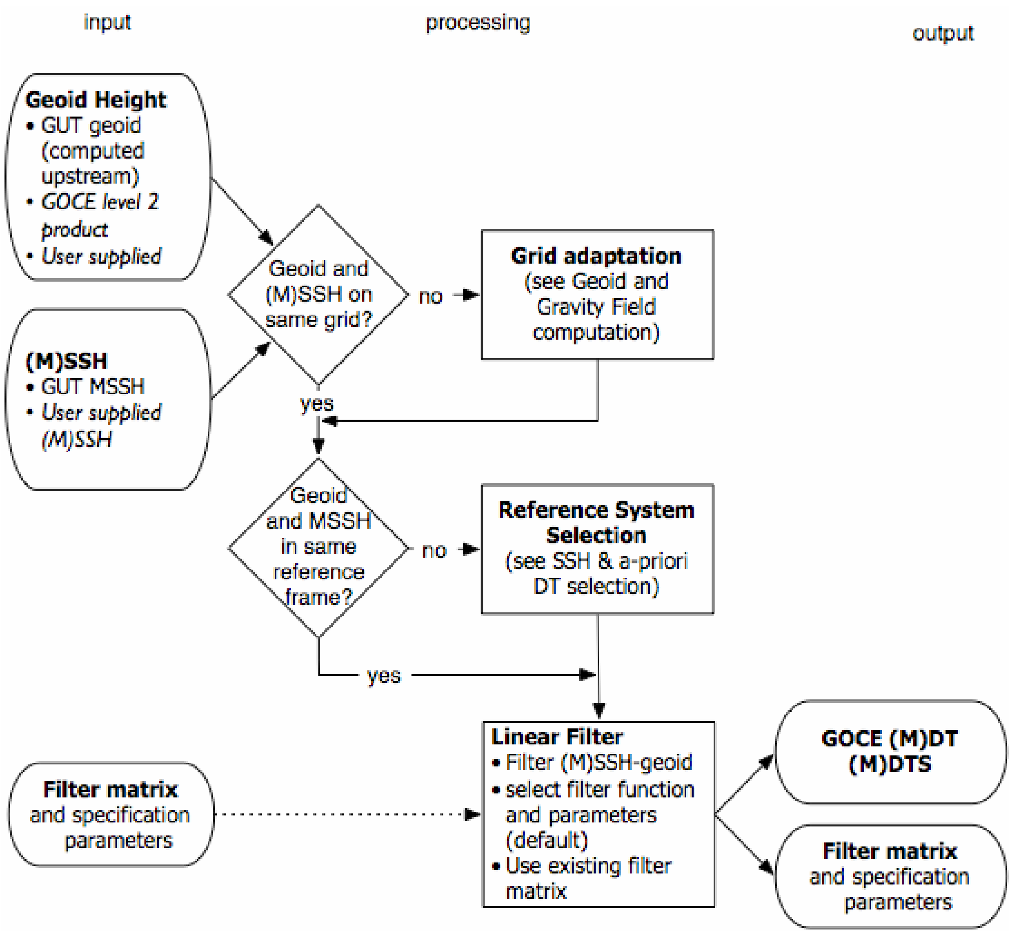


Figure 32: Workflow 3a

Workflow 3a should be used with the following parameters:

```
Options:
-Fj scale: Quasi-gaussian (Jekeli) filter with full filter width scale (default option)
-Fg scale: Gaussian filter with full filter width scale
-Fc scale: Spherical cap with diameter scale
-Fhm scale: Hamming window of width scale
-Fhn scale: Hanning window of width scale
-F filter_matrix_file: The filter matrix is read in from a file.
-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a
limitation for large grids depending on available resources. The limit grid size is
defined in .GUTdefaults.
```

```
Input:
- Grid of geoid height (GOCE level-2 product or user provided)
- Grid of Mean Sea Surface Height (GUT MSSH or user provided)
```

```
Output:
- GOCE MDTS
- Filter matrix
```

Example 3a:
In the example below, the MSSH CLS01 and the EIGEN-GL04S1 geoid model are used as input to workflow 3a to compute the satellite-only mean dynamic topography at a 400 km resolution using a Gaussian filter in geographical space. This filter is quite satisfying in the open ocean, where a quite realistic 400 km resolution MDT is obtained. Along the continental coasts however, strong, unrealistic gradients are created (Indonesian through flow, Western coasts of South America…).
A file name for storing the filter matrix is specified (option –O) so that this filter may be used afterwards for other similar computations.

```
GUT_WF3a
        –Fg400 –O my_filter_matrix.fic
        MSSCLS01_grid.fic EIGENGL04S_SH150_grid.fic
        MSSCLS01_EIGENGL4S_fg400_grid.fic
```

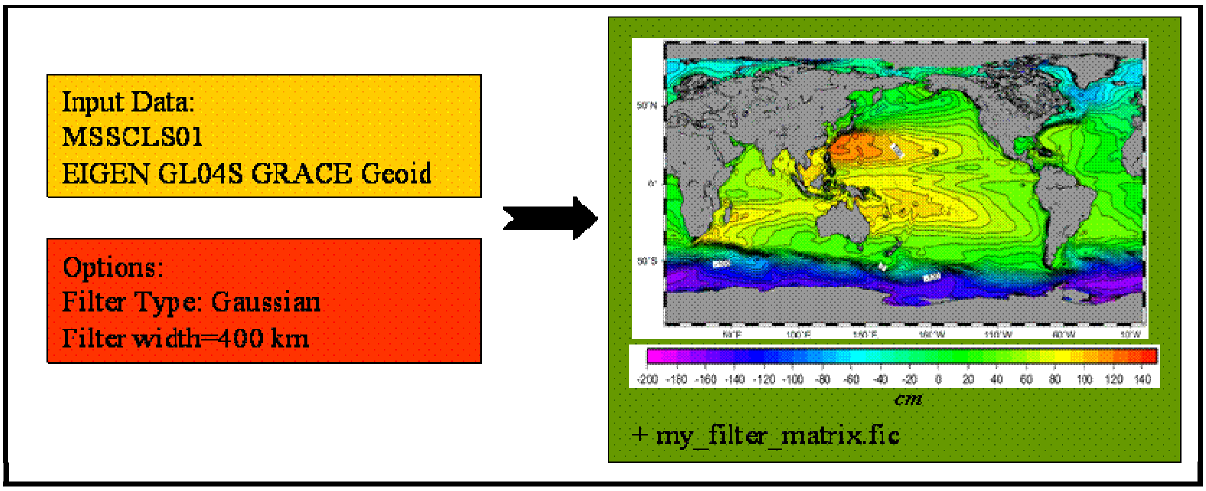


Figure 33: Example 3a

Workflow 3b: MDTS computation in the spectral domain

Using this workflow allows computation of the MDTS in spectral space. This means that the (default or GUT user supplied) MSSH is first completed over the continental gaps, using a geoid field, before being further expanded into spherical harmonics coefficients. The difference between the MSSH and the geoid is then done in the spectral domain (the coefficients are subtracted and MDT SH coefficients are obtained). Then, the user can choose to go back into geographical space for further filtering or to directly filter the MDT SH coefficients (recommended).

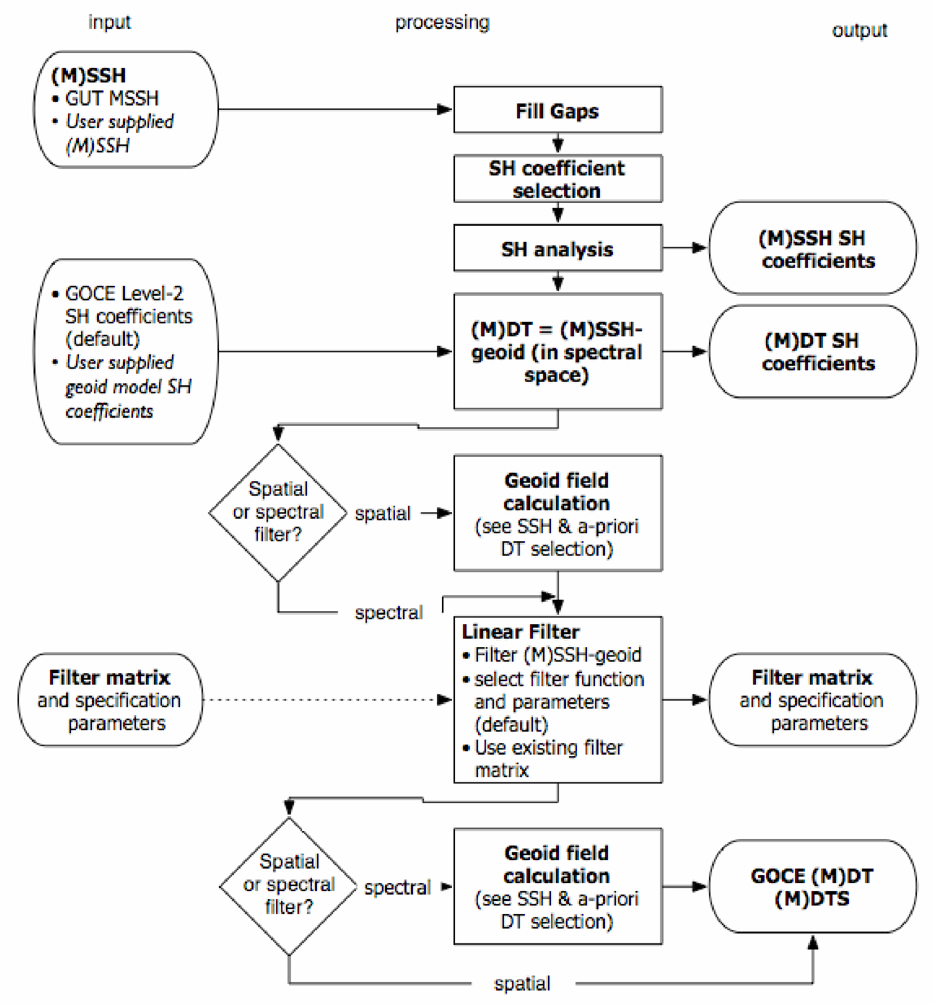


Figure 34: Workflow 3b

Workflow 3b should be used with the following parameters:

Options:
-Spatial: Filtering is done in physical space
-Spectral: Filtering is done in spectral space
If the -Spatial option is chosen:
-Fj scale: Quasi-gaussian (Jekeli) filter with full filter width scale (*default option*)
-Fg scale: Gaussian filter with full filter width scale
-Fc scale: Spherical cap with diameter scale
-Fhm scale: Hamming window of width scale
-Fhn scale: Hanning window of width scale
If the -Spectral option is chosen:
-Fj scale: Quasi-gaussian (Jekeli) filter with full filter width scale (*default option*)
-Fc scale: Pellinen filter with full width scale
-F filter_matrix_file: The filter matrix is read in from a file.
-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a limitation for large grids depending on available resources. The limit grid size is defined in .GUTdefaults.

Input:
- MSSH Grid (GUT or user supplied MSSH)
- GOCE Level-2 or user supplied spherical harmonic coefficients

Output:
-MSSH SH coefficients
-MDT SH coefficients
-Filter matrix
-GOCE MDTS

Example 3b:
In the example below, the MSSH CLS01 and the GGM02S geoid from CSR (Tapley et al, 2005) are used to compute the mean dynamic topography at a 400 km resolution using a Jekeli filter in spectral space. Compared to the MDTS obtained using workflow 3a (Figure 33) the unrealistic gradients near the coasts have been significantly reduced.

GUT_WF3b

–Spectral -Fj400 –O my_filter_matrix.fic
MSSCLS01_filled_with_GGM02S.fic GGM02S_SH160_coef.fic
MSSCLS01_EIGENGL4S_fj400_grid.fic MSSCLS01_coef.fic

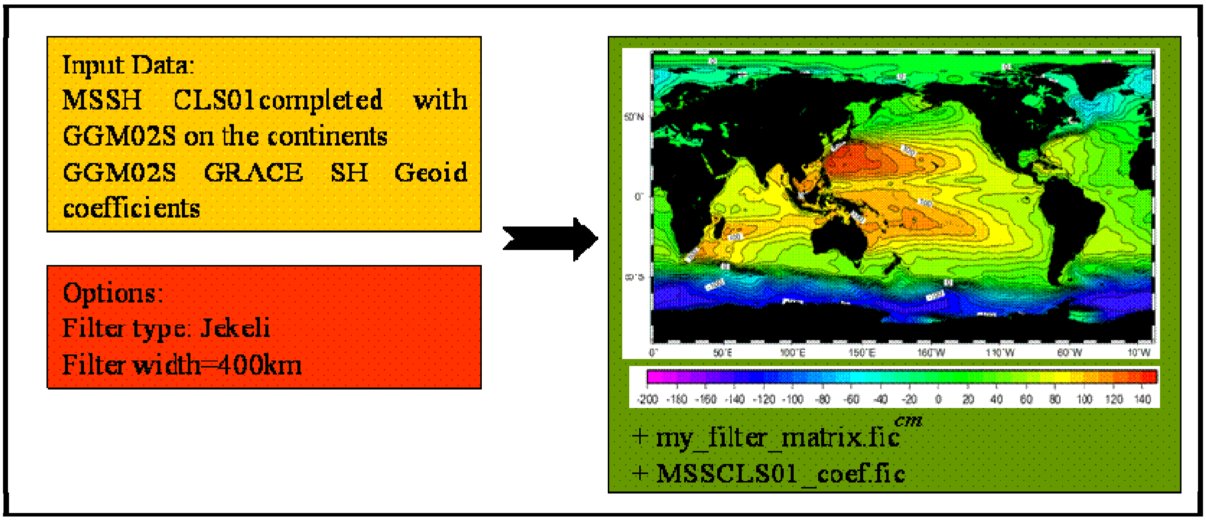


Figure 35: Example3b

Workflow 4 Remove-restore combined technique A

Workflow 4 is dedicated to the computation of combined mean dynamic topographies using the remove-restore technique A as described in section 5.1.2. The computation can be done in geographical space (Workflow 4a - Figure 36) or in spectral space (Workflow 4b - Figure 38). Computation in spectral space is recommended.

Workflow 4a spatial filtering

If spatial filtering is chosen, the a-priori MDT is processed in order to extract its short spatial scales, that are then added to the GUT MDTS as a corrective field. It is essential, for consistency sake, that the filter applied on the a-prior MDT for the short scales extraction is the same as the filter used for computing the MDTS (through workflow 3a or 3b). This can be done simply done by providing, as input, the filter matrix produced during the MDTS computation (output from workflow 3a or 3b).

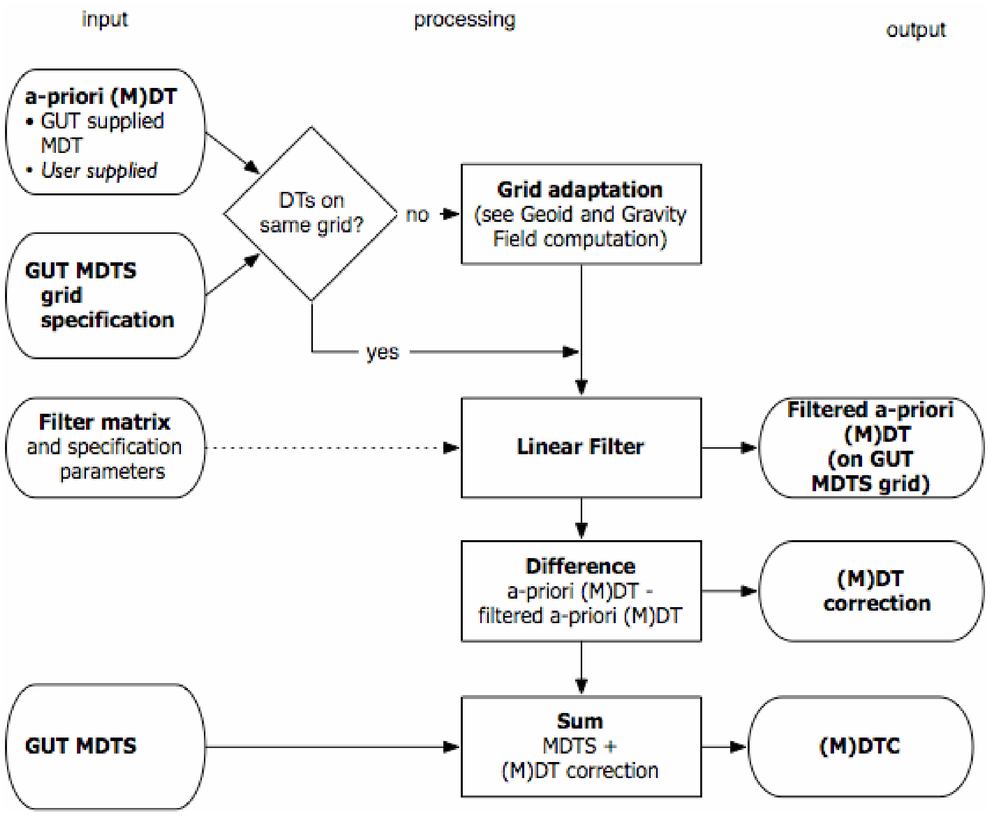


Figure 36: Workflow 4a

Workflow 4a should be used with the following parameters:

Options:
-Fj scale: Quasi-gaussian (Jekeli) filter with full filter width scale (*default option*)
-Fg scale: Gaussian filter with full filter width scale
-Fc scale: Spherical cap with diameter scale
-Fhm scale: Hamming window of width scale
-Fhn scale: Hanning window of width scale
-F filter_matrix_file: The filter matrix is read in from a file.
-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a limitation for large grids depending on available resources. The limit grid size is defined in .GUTdefaults.

Input:
- A-priori MDT (GUT or user-supplied)
- GUT MDTS
- Grid specification of GUT MDTS:
  -R ***e/w/n/s***: Region of interest in the order western, eastern, northern and southern limits in degrees north of the equator and east of the prime meridian.
  -I ***dx/dy***: grid increment for regular (latitude-longitude) grid in degrees

Output:
- Filtered A-priori MDT
- MDT correction
- MDTC

Example 4a:
In the example below, a combined MDT is computed using as a-priori MDT the (Niiler et al, 2003) field provided on a regional grid (Gulf Stream area) and the MDTS computed in workflow 3a. In the obtained field, the large scale structures come from the MDTS, while the shorter scales (signature of the mean of Gulf Stream eddies north and south of the main jet, signature of the Mann eddy) come from the (Niiler et al, 2003) a-priori field.

GUT_WF4a
Niiler_MDT.fic MSSCLS01_EIGENGL04S_fg400_grid.fic Grid.fic
MDTC_niiler_grid.fic

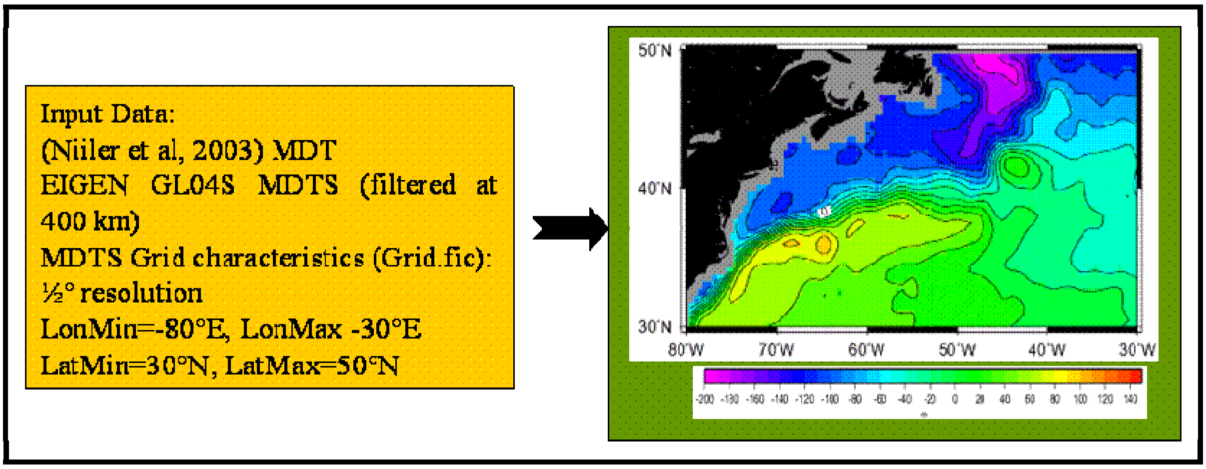


Figure 37: Example 4a

Workflow 4 b spectral filtering
If spectral filtering is chosen, the a-priori MDT is first expanded into Spherical Harmonic coefficients (after the continental gaps have been filled) and the extraction of the a-priori MDT short scales in done in the SH domain. Here again, the filter used must be consistent with the provided MDTS. The filter matrix obtained as output of Workflow 3b in the computation of the MDTS should therefore be used here as input information. After the computation is done in spectral space, the MDTC is further transformed in geographical space and the user can choose the grid specification of the output field.

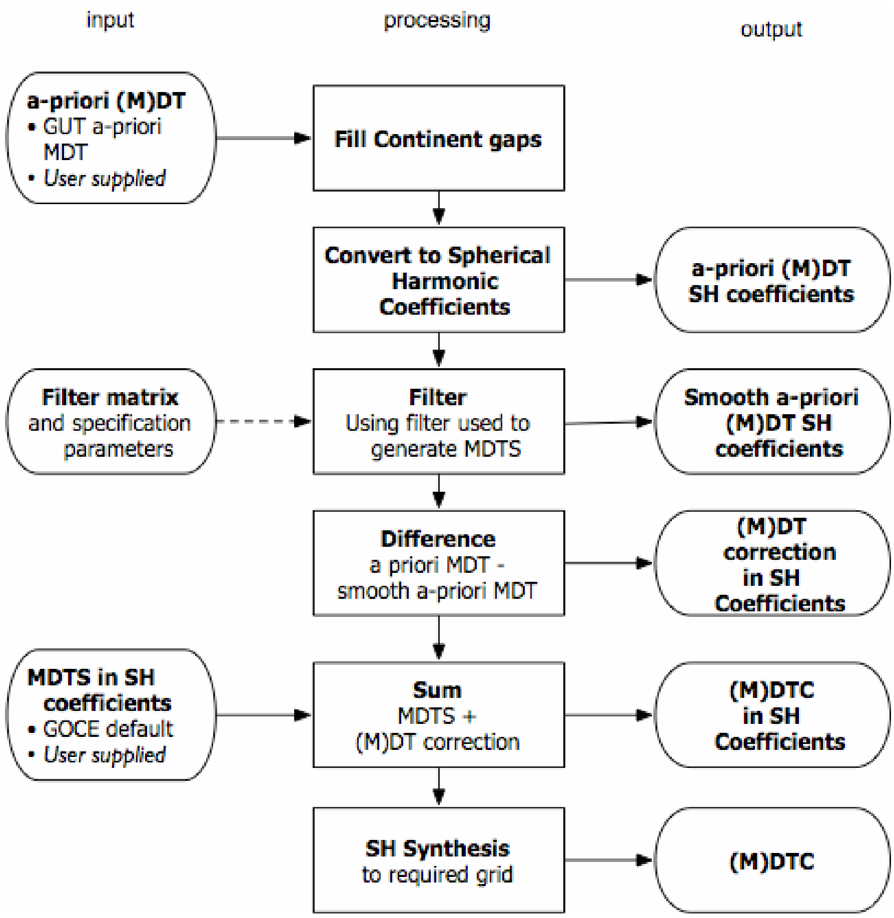


Figure 38: Workflow 4b

Workflow 4b should be used with the following parameters:

Options:
-R *e/w/n/s*: Region of interest in the order western, eastern, northern and southern limits in degrees north of the equator and east of the prime meridian (default: 79.75/-79.75/0.25/359.75). Only used for regular (latitude-longitude) grid.
-I *dx/dy*: grid increment for regular (latitude-longitude) grid in degrees (default: 0.25/0.25)
-Fj scale: Quasi-gaussian (Jekeli) filter with full filter width scale (*default*)
-Fc scale: Pellinen filter with full filter width scale
-F filter_matrix_file: The filter matrix is read in from a file.
-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a limitation for large grids depending on available resources. The limit grid size is defined in .GJTdefaults.

Input:
- SH coefficients of GOCE or user-supplied geoid
- GUT or user-supplied (M)SSH
- A-priori MDT

Output:
- A-priori MDT SH coefficients
- Smooth a-priori MDT SH coefficients
- SH coefficients of MDT correction
- SH coefficients of MDTC
- MDTC on user specified grid

Workflow 5 Remove-restore combined technique B

Workflow 5 is dedicated to the computation of combined mean dynamic topographies using the remove-restore technique B as described in section 5.1.2. The computation can be done in geographical space (Workflow 5a -Figure 39) or in spectral space (Workflow 5b -Figure 41). Here again, computation in spectral space is recommended.

Workflow 5a: spatial filtering

Compared to workflow 4a, the inputs to this workflow are independent of any previously computed MDTS and any type of filter and filter characteristics can be chosen.
Three grids are needed as input, a MSSH, a geoid and an a-priori MDT that are automatically adapted to a consistent grid (the MSSH grid is used as the default). The MDTC output grid characteristics are therefore, by default, identical to the MSSH input grid characteristics.

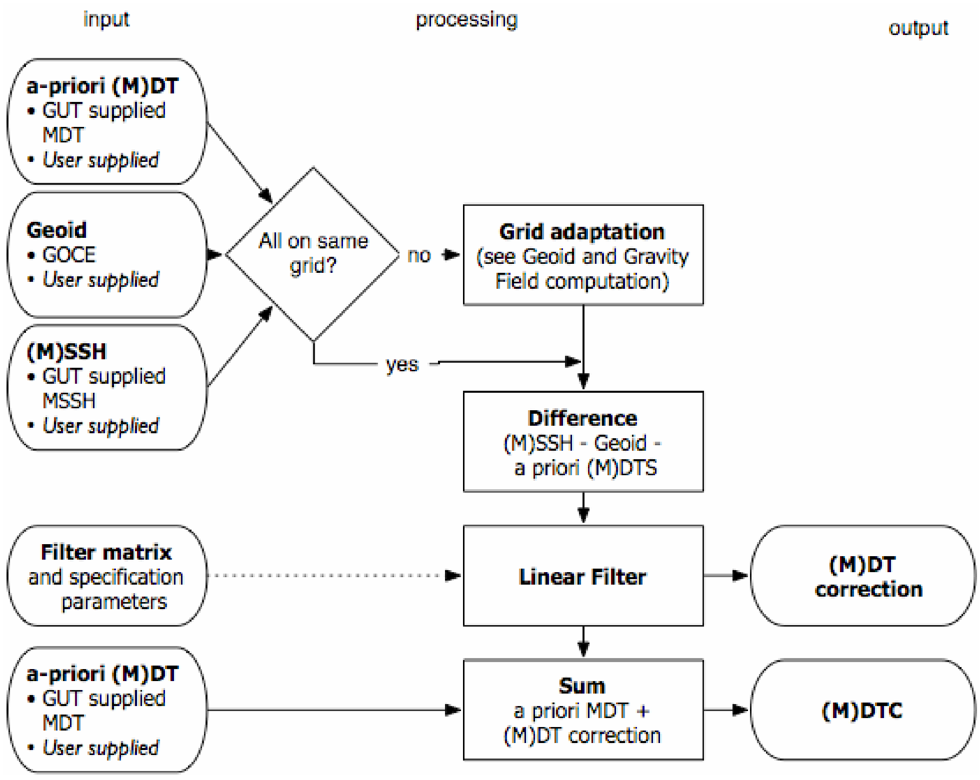


Figure 39: Workflow 5a

Workflow 5a should be used with the following parameters:

```
Options:
-Fj scale: Quasi-gaussian (Jekeli) filter with full filter width scale (default option)
-Fg scale: Gaussian filter with full filter width scale
-Fc scale: Spherical cap with diameter scale
-Fhm scale: Hamming window of width scale
-Fhn scale: Hanning window of width scale
-F filter_matrix_file: The filter matrix is read in from a file.
-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a limitation for large grids depending on available resources. The limit grid size is defined in .GUTdefaults.
```

```
Input:
- GOCE or user-supplied geoid
- GUT or user-supplied (M)SSH
- A-priori MDT
```

```
Output:
- MDT correction
- MDTC
```

Example 5a:

In the example below, the RIO05 MDT (Rio et al, 2005), provided on a regular grid of the North Atlantic, is used as a first guess for the computation of a remove-restore MDTC using Workflow 5a. The EIGEN-GL04S1 geoid model is provided as input as well as the CLS01 MSS. The filter chosen for the processing is Gaussian type with a 400 km width. A correction is computed (right plot on Figure 40) that, added to the a-priori MDT, provides the output MDTC (left plot on Figure 40).

```
GUT_WF5a
        –Fg400
        RIO05_MDT.fic MSSCLS01_grid.fic EIGENGL04S_grid.fic
        MDTC_rio_natl_grid.fic MDTC_rio_natl_correction_grid.fic
```

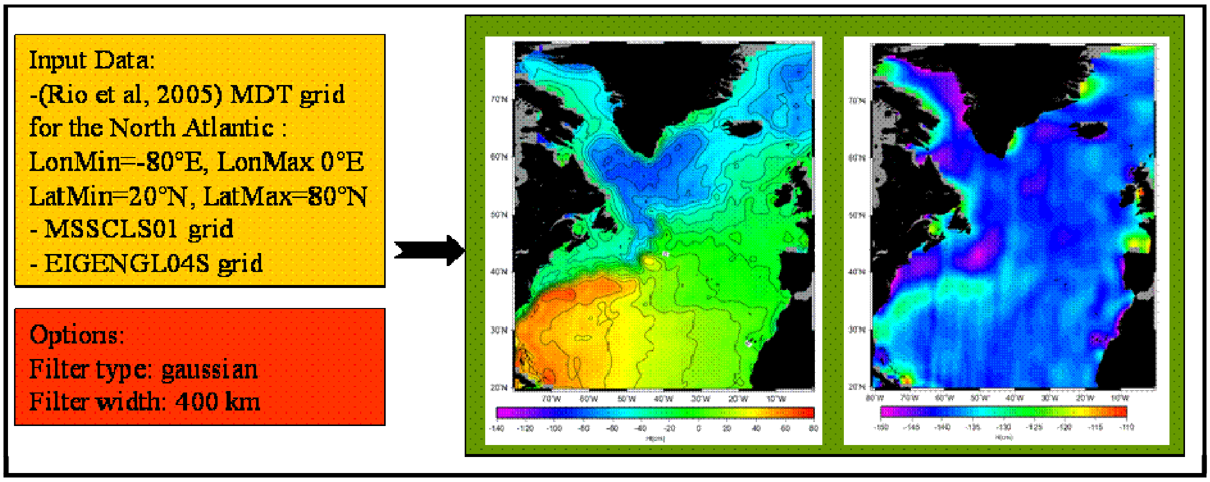


Figure 40: Example 5a

Workflow 5b spectral filtering

In Workflow 5b, the same approach as that of workflow 5a is applied but in spectral space. The user can provide grid specifications for the output grid since the MDTC is first produced in SH coefficients and then developed back to geographical space on the user-required grid.

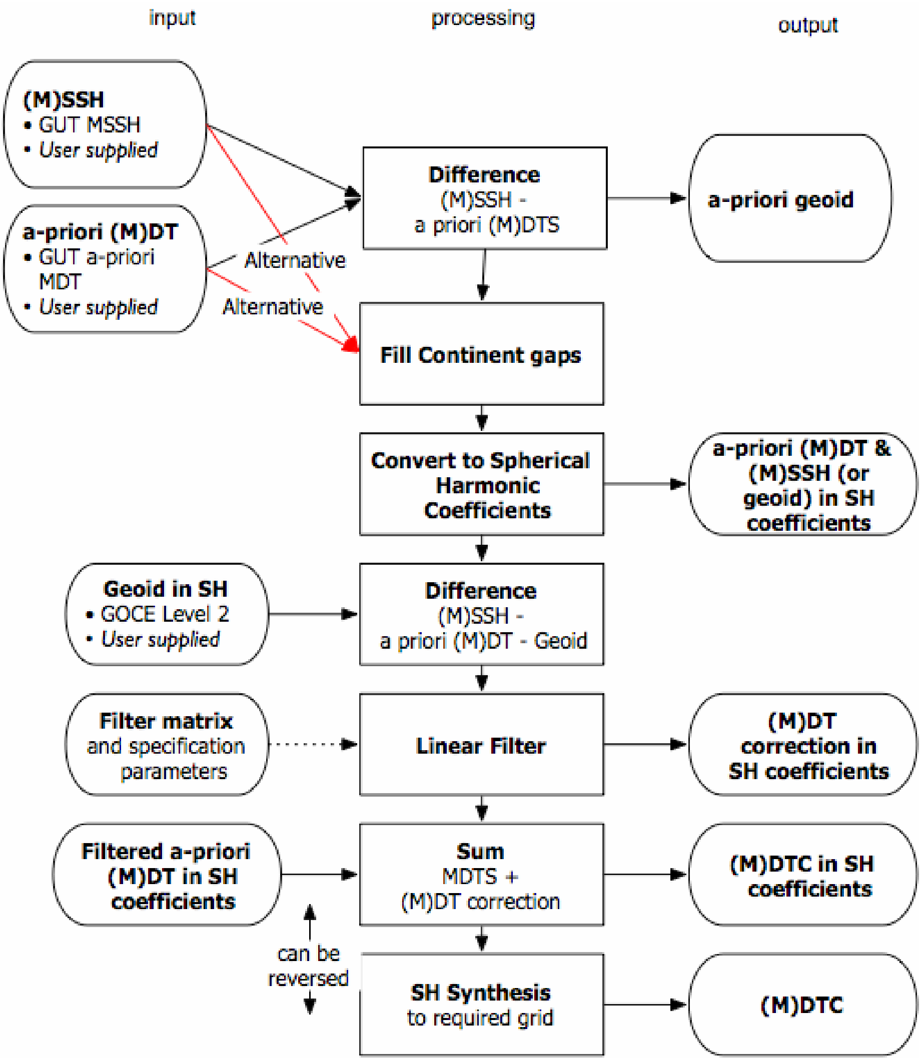


Figure 41: Workflow 5b

This workflow may be used with the following parameters:

Options:
-R *e/w/n/s*: Region of interest in the order western, eastern, northern and southern limits in degrees north of the equator and east of the prime meridian (default: 79.75/-79.75/0.25/359.75). Only used for regular (latitude-longitude) grid.
-I *dx/dy*: grid increment for regular (latitude-longitude) grid in degrees (default: 0.25/0.25)
-Fj scale: Quasi-gaussian (Jekeli) filter with full filter width scale (*default*)
-Fc scale: Pellinen filter with full filter width scale
-F filter_matrix_file: The filter matrix is read in from a file.
-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a limitation for large grids depending on available resources. The limit grid size is defined in .GJTdefaults.

Input:
- SH coefficients of GOCE or user-supplied geoid
- GUT or user-supplied (M)SSH
- A-priori MDT

Output:
- A-priori geoid
- SH coefficients of a-priori MDT
- SH coefficients of MSSH
- SH coefficients of MDT correction
- SH coefficients of MDTC
- MDTC grid

Workflow 6: dynamic topography derived quantities

Workflow 6 (Figure 42) is dedicated to the computation of surface geostrophic currents, from a list of dynamic topographies (mean or time-dependent) distributed on a grid or along transects.

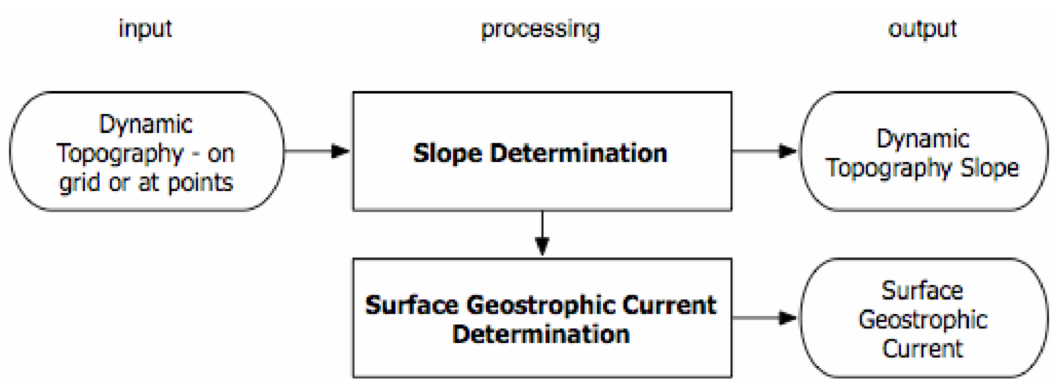


Figure 42: Workflow 6

This workflow may be used with the following parameters:

Options:
-T: Dynamic Topography is provided on a transect (default: topography on 2D grid)
-G: Currents are required on the same grid as the input topography. This involves interpolation on larger scales than the default four point calculation (see algorithm description below)

Input:
For 2D Dynamic Topography:
- 2D matrix of gridded Dynamic Topography (m)
- Locations of grid nodes as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.

For transects:
- Vector containing (mean) dynamic topography (m) at points values
- 2 vectors containing the latitude and longitude coordinates of the points

Output:
For 2D Dynamic Topography:
- 2 2D matrices of East and North velocities respectively
- Locations of grid nodes at which the velocity was calculated, given as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.

For transects:
- 2 vectors containing East and North components respectively of velocity across the transect
- 2 vectors containing the latitude and longitude coordinates of the points at which the velocity was calculated.

By default, for gridded data, the velocity values will be calculated for a grid offset by $\delta\varphi/2$ in latitude and $\delta\lambda/2$ in longitude for the dynamic topography grid (*i.e.* for a grid offset by ½ grid cell dimension from the input grid. For global grid, the longitude will be wrapped.
By default, for transect data, the velocities will be calculated at the midpoints between each location

Example 6:
In the example below (Figure 43), the absolute geostrophic circulation is computed in the Gulf Stream area on June 7th, 2006 from a grid of absolute dynamic topographies (ADT). The ADT map was computed by adding the SLA map to the MDTC computed using workflow 4a.

GUT_WF6 (GUT_FA13)
-G
ADT_June_7th_2006.fic
AbsGeosVel_June_7th_2006.fic

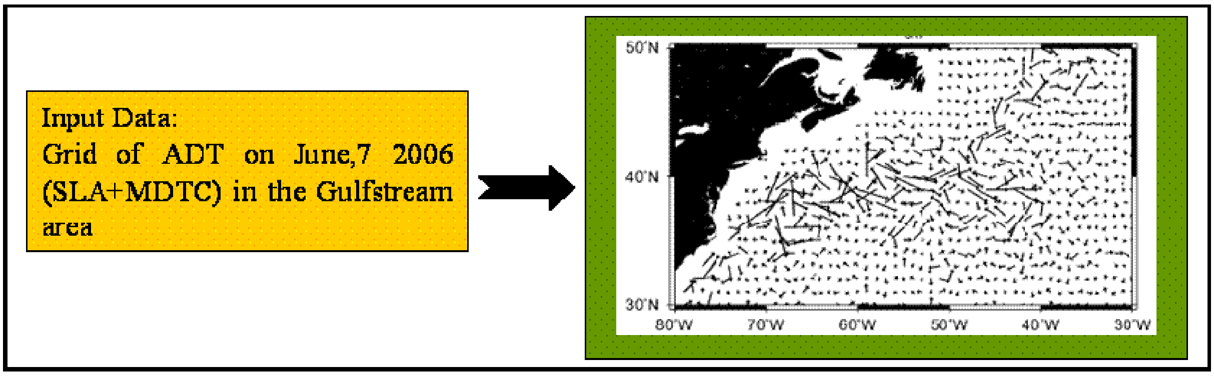


Figure 43: Example 6

5.2.2 The "single step" approach

The different workflows described in the previous section are composed of a number of functionalities that have been defined and described in the GUTS WP2000 and WP3000. Each of these functions can be called independently by the user for specific applications.

GUT_FA08: Grid adaptation

The GUT_FA08 function provides the user with the possibility of interpolating a given grid to another user specified grid or to a list of points.

This function should be used with the following input/output specifications:

```
Input:
- Required new regular grid specification for output fields (default is GUT MSS grid)
- Regular grid specification of existing input field
- Input data field
```

```
Output:
- Input data field interpolated to grid locations of the required new grid
- Latitude and Longitude values of new grid nodes
```

Example

In the example below (Figure 44), the MDTS computed through workflow 3a (Figure 33) is interpolated along a transect through the Drake Passage. This transect could, for example, correspond to an altimeter satellite track along which temperature and salinity CTD profiles have been measured during a dedicated sea campaign. The altimetric SLA, added to the interpolated MDTS, gives the absolute values of the ocean dynamic topography along the altimeter track, which can be compared to the steric dynamic heights deduced from the in-situ T/S measurements. The difference between the two data types gives an estimate of the barotropic component of the flow in this area (since it is contained in the altimetric data but not in the in-situ dynamic heights data).

GUT_FA08

MSSCLS01_EIGENGL4S_fg400.fic Grid.fic Points.fic
MSSCLS01_EIGENGL4S_fg400_points.fic

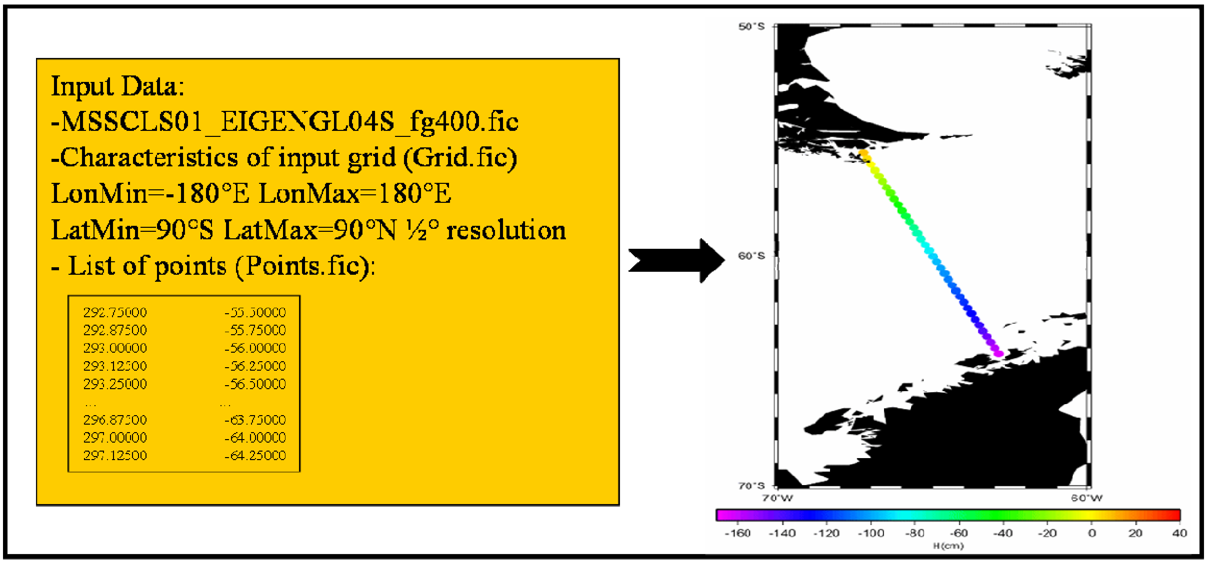


Figure 44: Grid adaptation

GUT_FA09: Filtering in geographical space

The GUT_FA09 function allows the user to filter a gridded data field in geographical space. The user can choose between different types of filters (Quasi-gaussian, Gaussian, Spherical cap, Hamming window, Hanning window) or can provide a filter matrix that was created, for instance, during a previous use of this function.

This function should be called using the following parameters:

```
Options:
-Fj scale: Quasi-gaussian (Jekeli) filter with full filter width scale (default option)
-Fg scale: Gaussian filter with full filter width scale
-Fc scale: Spherical cap with diameter scale
-Fhm scale: Hamming window of width scale
-Fhn scale: Hanning window of width scale
-F filter_matrix_file: The filter matrix is read in from a file.
-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a limitation for large grids depending on available resources. The limit grid size is defined in .GUTdefaults.
```

```
Input:
- Grid specification (default is the default GUT MSS grid specification)
- Input field
```

```
Output:
- Filtered version of input field
```

Example:
In the example below (Figure 45), the difference field (obtained using the later described GUT_FA14 function) between the MSSH CLS01 and the EIGENGL04C geoid model (http://www.gfz-potsdam.de/pb1/op/grace/index_GRACE.html) is filtered using a Gaussian filter of 200 km width. The obtained MDTS contains numerous erroneous short scale, structures due to the poor accuracy of the geoid model at that resolution.

GUT_FA09

–Fg200 –O my_filter_matrix.fic
MSSCLS01_EIGENGL4C.fic Grid.fic
MSSCLS01_EIGENGL4C_fg200.fic

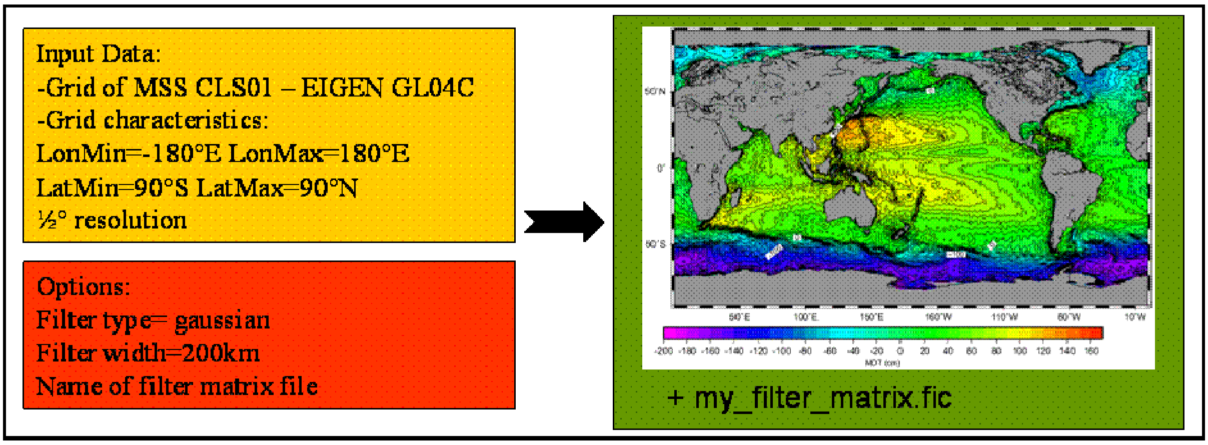


Figure 45:Filtering in geographical space

GUT_FA10: Linear filter (spherical harmonics)

The GUT_FA10 function allows the user to filter a field (defined by a set of spherical harmonic coefficients) in spherical harmonic space.

This function may be called using the following parameters:

Options:
-Fj scale: Quasi-gaussian (Jekeli) filter with full filter width scale (*default*)
-Fc scale: Pellinen filter with full filter width scale
-F filter_matrix_file: The filter matrix is read in from a file.
-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a limitation for large grids depending on available resources. The limit grid size is defined in .GUTdefaults.
-A: The parameter is given on a regular grid and a transformation to spherical harmonic coefficients is needed (default). -A- indicates that the parameter is given in spectral space and no analysis is necessary.
-S: Transformation into physical space is invoked after filtering (default). -S- indicates that the output is required in spectral space.
-D: The parameter filtered is a dynamic topography. No reference surface (as specified in .GUTdefaults) is substracted when transforming to physical space (-S option). Default: Subtraction of reference.

Input:
If parameter provided on regular grid (-A option):
- Grid specification (default is the default GUT MSS grid specification)
- Input field
If parameter is provided in spectral space (-A- option):
- Data set with spherical harmonic coefficients

Output:
- Filtered field on grid (-A option) or as spherical harmonic coefficients (-A- option).

Example:
This function can be used to filter the difference field between the MSSH CLS01 and the GGM02S geoid (expressed in spherical harmonic coefficients) using a Jekeli filter with a 400 km width.

GUT_FA10

MSSCLS01_GGM02S_SH360_coef.fic Grid.fic

MSSCLS01_GGM02S_fj400_grid.fic

In this case the output field is identical to the one displayed on Figure 35 (Output example of workflow 3b).

GUT_FA11: Filling gaps

The GUT_FA11 function allows the user to replace the default value of a grid by values from another grid. In oceanography, this is needed to complete an oceanic field (for example an altimetric mean sea surface or a mean dynamic topography) on the continents in order to obtain a globally defined grid. Globally defined grids are needed so that expansion into spherical harmonic coefficients can be performed.

This function can be called with the following input/output specifications:

```
Input:
- geodetic field
- global field to fill gaps in geodetic field (default GUT gridded geoid)
```

```
Output:
- patched geodetic field
```

Example:
In the example below (Figure 46), the Mean Sea Surface CLS01 is completed over the continents using the geoid heights from the GGM02S model. This field can now be used as input to the next function: GUT_FA12.

GUT_FA11

MSSCLS01.fic GGM02S.fic

MSSCLS01_filled_with_GGM02S.fic

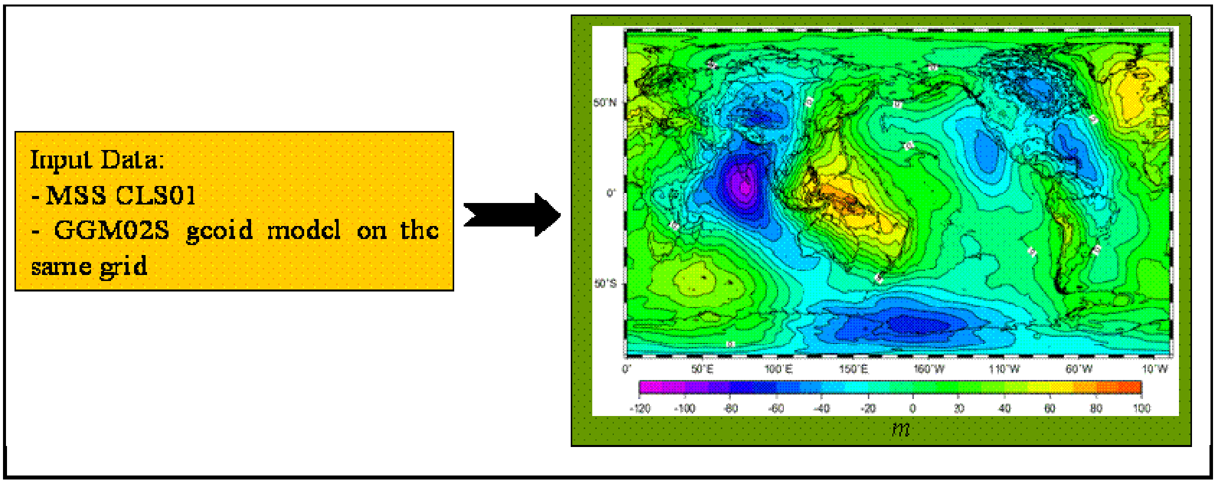


Figure 46: Filling gaps

GUT_FA12: SH analysis

The GUT_FA12 function allows the user to expand a gridded field, defined globally, into spherical harmonic coefficients.

This function may be used with the following input/output specifications:

```
Input:
- Global gridded field
- Regular grid specification of input field
- Maximum degree and order of spherical harmonic expansion to calculate (read from .GUTdefaults)
```

```
Output:
- Spherical Harmonic expansion of coefficents of global field
- Maximum degree and order included in expansion
```

GUT_FA14: Difference/Sum in geographical space

The GUT_FA14 function allows calculation of the sum or difference of two fields defined on the same grid.

This function may be used with the following parameters:

Options:
-Diff: compute the difference
-Sum: compute the sum

Input:
- Input field 1
- Input field 2

Output:
- Difference or Sum of the two grids

Example:

Difference

The example below (Figure 47) computes the differences between two MSSH solutions computed in the North Atlantic area during the GOCINA project. The first one was computed by CLS while the second one was computed by DNSC. computed in the GOCINA area by CLS and the EIGEN-GL4C geoid model. Prior to subtract the two solutions, the user must be careful that both fields have been previously computed relative to the same reference ellipsoid and tide system. This can be done using the further described GUT_FA15 and GUT_FA16 functions. In this particular example, the DNSC MSSH field has been first computed relative to the TP reference ellipsoid and in the mean tide system to be consistent with the CLS MSSH reference frame.

GUT_FA14

-Diff

MSS_CLS_grid_Etp_MT.fic MSS_DNSC_grid_Etp_MT.fic

Diff_MSS_CLS_DNSC_Gocina.fic

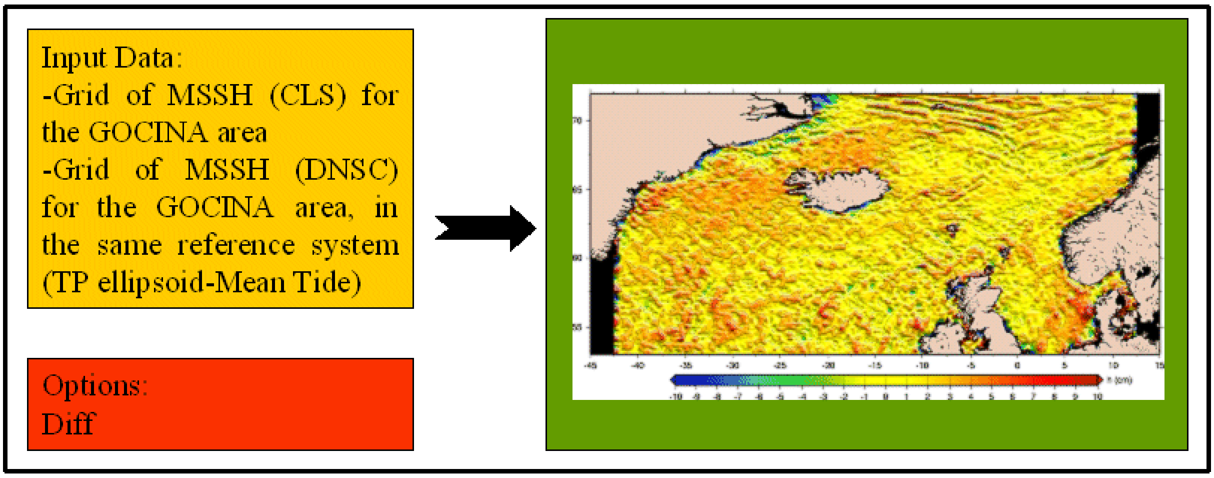


Figure 47: Grid difference

Sum:

The example below (Figure 48) computes the sum of the MDTC obtained using the Workflow 4a (Figure 37) and the sea level anomalies measured on June, 7th 2006 in the Gulf Stream area. The resulting field is the absolute dynamic topography, from which geostrophic velocities may be derived using the GUT_FA13 function (see section 0).

```
GUT_FA14
        -Sum
        MDTC_Niiler_GL4S_fg400_RRA.fic SLA_June_7th_2006.fic
        ADT_June_7th_2006.fic
```

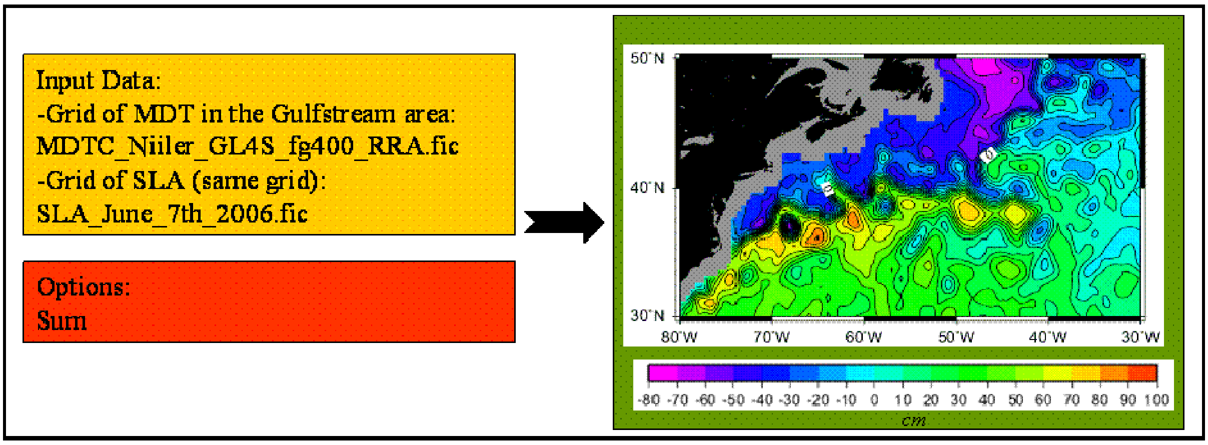


Figure 48: Grid Sum

GUT_FA15: Change tide system

The GUT_FA15 function allows the user to convert a height field from a given tide system to another.

This function may be called using the following parameters:

```
Options:
-TideIn: Tide system of the input field
-TideOut: Tide system required for the output field
```

```
Input:
- Gridded input field
```

```
Output:
- Input field relative to the new tide system
```

Example:

In the example below (Figure 49), the EIGEN-GL4C geoid heights are converted from the TIDE FREE system into the MEAN TIDE system.

GUT_FA15

EIGENGL4C_SH360_grid_Etp.fic –TideIn=FREE –TideOut=MEAN
EIGENGL4C_SH360_grid_Etp_MT.fic

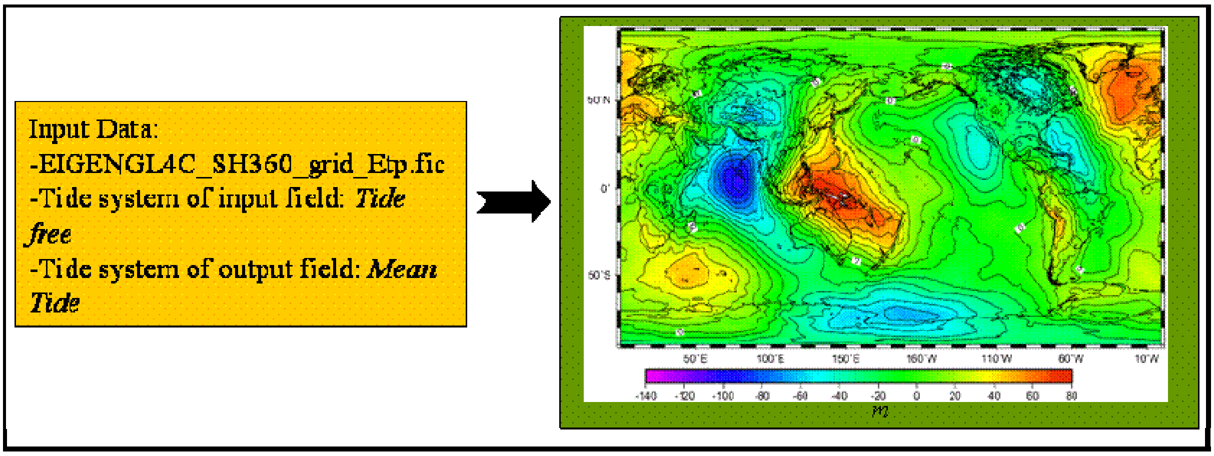


Figure 49: change the tide system

GUT_FA16: Change reference ellipsoid

The GUT_FA16 function allows the user to convert a height field from a given reference ellipsoid to another.

This function may be called using the following parameters:

```
Options:
-EllipsIn: Reference ellipsoid of the input field
-EllipsOut: Reference ellipsoid required for the output field
```

```
Input:
- Gridded input field
```

```
Output:
- Input field relative to the new ellipsoid
```

Example:
In the example below, the EIGEN-GL4C geoid heights are converted from the GRIM reference ellipsoid to the TOPEX reference ellipsoid.

GUT_FA16

EIGENGL4C_SH360_grid.fic –EllipsIn=GRIM –EllipsOut=TP
EIGENGL4C_SH360_grid_Etp.fic

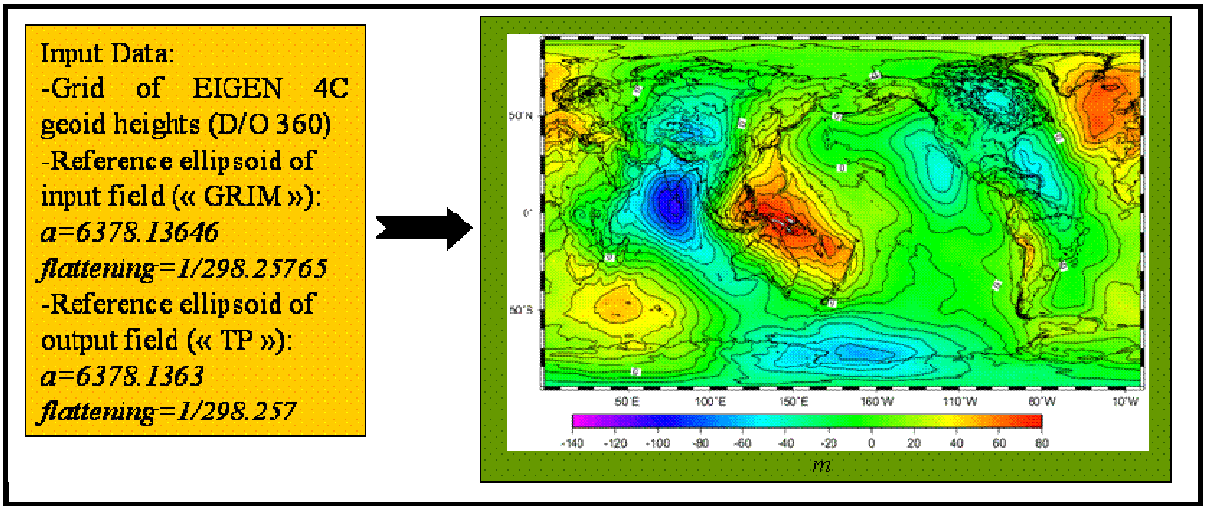


Figure 50: Change the reference ellipsoid

GUT_FA18: SH synthesis

The GUT_FA18 function allows to develop a set of spherical harmonic coefficients into a gridded height field.

This function may be called with the following input/output specifications:

```
Input:
- Spherical Harmonic expansion of coefficents of global field
- Regular grid specification of output field
- Maximum degree and order of spherical harmonic expansion to calculate
```

```
Output:
- Global gridded field
```

Example:

In the example below (Figure 51), the Spherical Harmonic coefficients of the EIGENGL4C geoid model are developed up to degree 360 on a 1/2° regular grid.

GUT_FA18

EIGENGL4C_SH360_coef.fic -SH360 Grid.fic

EIGENGL4C_SH360_grid.fic

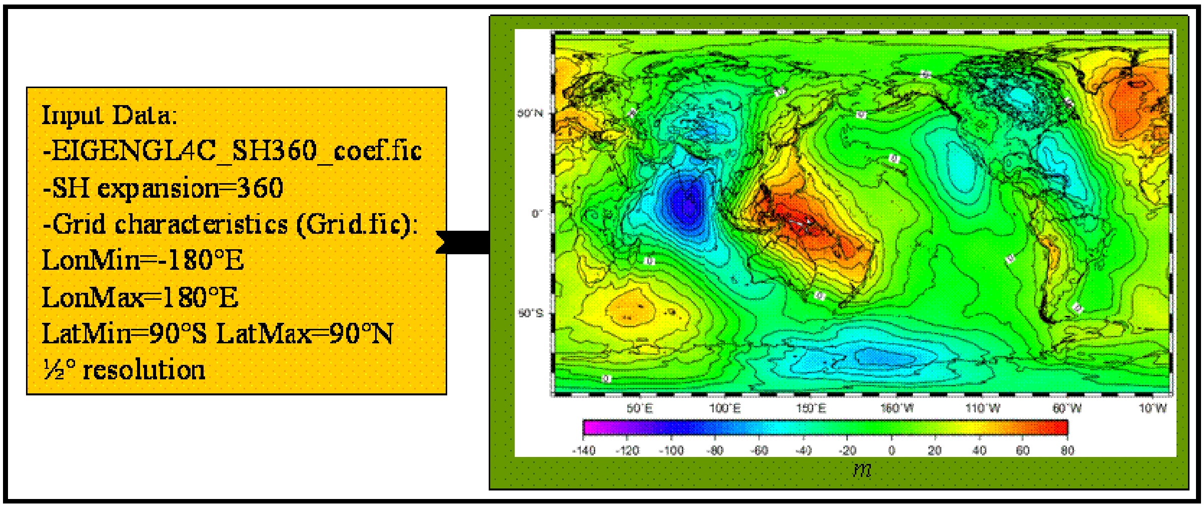


Figure 51: SH synthesis

5.2.3 The step by step approach

The "single step" functions described above can be used individually and successively. This provides the user with the possibility to build his/her own workflows for specific research purposes.
We give below some examples of what can be done combining the different "single step" functions:

Example1: Compare two geoid models

In this example (Figure 52), two different geoid models based on 110 days of GRACE data are compared: the GGM02S model from CSR (Tapley at al, 2005), and the EIGEN-GRACE2S model from GFZ (Reigber et al, 2005). Both models have first been developed at their maximum SH degree (160 for GGM02S and 150 for EIGEN-GRACE2S).
The GGM02S model is provided by CSR relative to the ellipsoid and in the zero tide system while the EIGEN-GRACE02S model is provided by GFZ relative to the GRIM ellipsoid and in the tide free system.
Both models are therefore first processed in order to be expressed relative to the same reference ellipsoid (the TP ellipsoid is chosen) and the same tide system (the mean tide system is chosen) and are then subtracted.

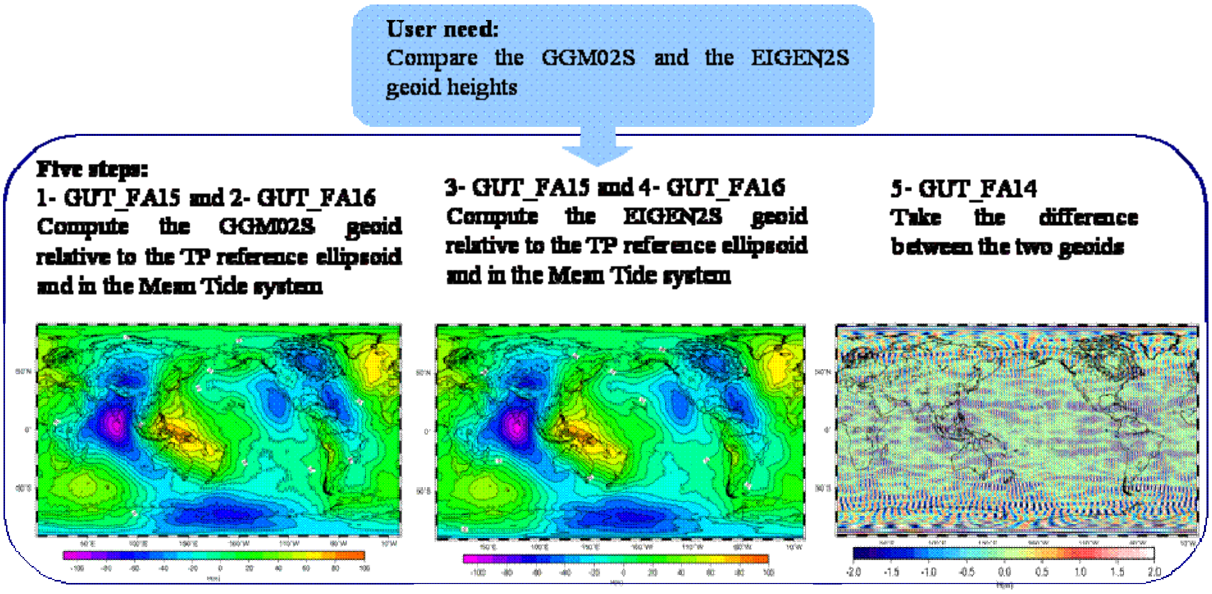


Figure 52: Example1

Example 2: Change the average period of a Mean Dynamic Topography field to another.

In this example (Figure 53), the Mean Dynamic Topography from the ECCO model, corresponding to the 1992-2002 time period, is computed relative to the 1993-1999 time period. This processing is often needed in oceanography when combining SLA and MDT to compute absolute values of the dynamic topography: SLAs are obtained subtracting from the altimetric Sea Surface Heights an altimetric mean profile (computed for a given period). In order to compute absolute values of the dynamic topography, the SLA needs to be added to an estimate of the MDT corresponding to the same time-averaging period. For instance, the SLA distributed by the AVISO center are computed relative to the 1993-1999 period. If the user wants to use the ECCO MDT to compute absolute dynamic topographies from these SLA, he/she first needs to express the ECCO MDT relative to the same 1993-1999 time period. This is done through the following equation:

$$MDT_{1993-1999} = MDT_{1992-2002} - \langle SLA_{1993-1999} \rangle_{1992-2002}$$

Explanation:

$$\left(\begin{array}{l} MDT_{1992-2002} = \langle h \rangle_{1992-2002} = \langle MDT_{1993-1999} + SLA_{1993-1999} \rangle_{1992-2002} \\ = MDT_{1993-1999} + \langle SLA_{1993-1999} \rangle_{1992-2002} \\ \\ \Leftrightarrow MDT_{1993-1999} = MDT_{1992-2002} - \langle SLA_{1993-1999} \rangle_{1992-2002} \end{array} \right)$$

where h stands for dynamic topography and $SLA_{1993\text{-}1999}$ refer to altimetric Sea Level Anomalies computed relative to a 1993-1999 mean profile

Maps of SLA are therefore first averaged over the 1992-2002 time period and the mean is then subtracted from the ECCO MDT.

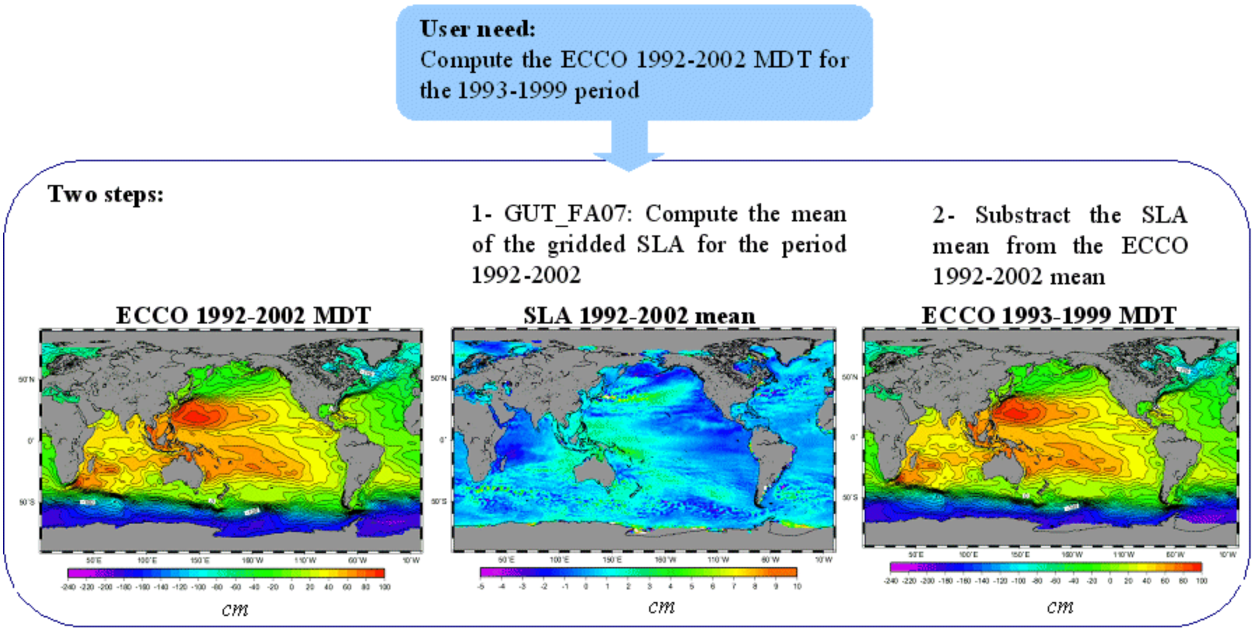


Figure 53: Example 2

Example 3: Compute the MDTC as with workflow 4a, using the step by step approach

The objective of this example (Figure 54) is to decompose the example from Workflow 4a given in the previous section and show the outputs from each single processing step needed when computing a MDTC.
A MDTC can therefore be obtained either using directly Workflow 4a, or using a step by step approach which allow visualizing, modifying, saving, and analyzing each step of the computation.

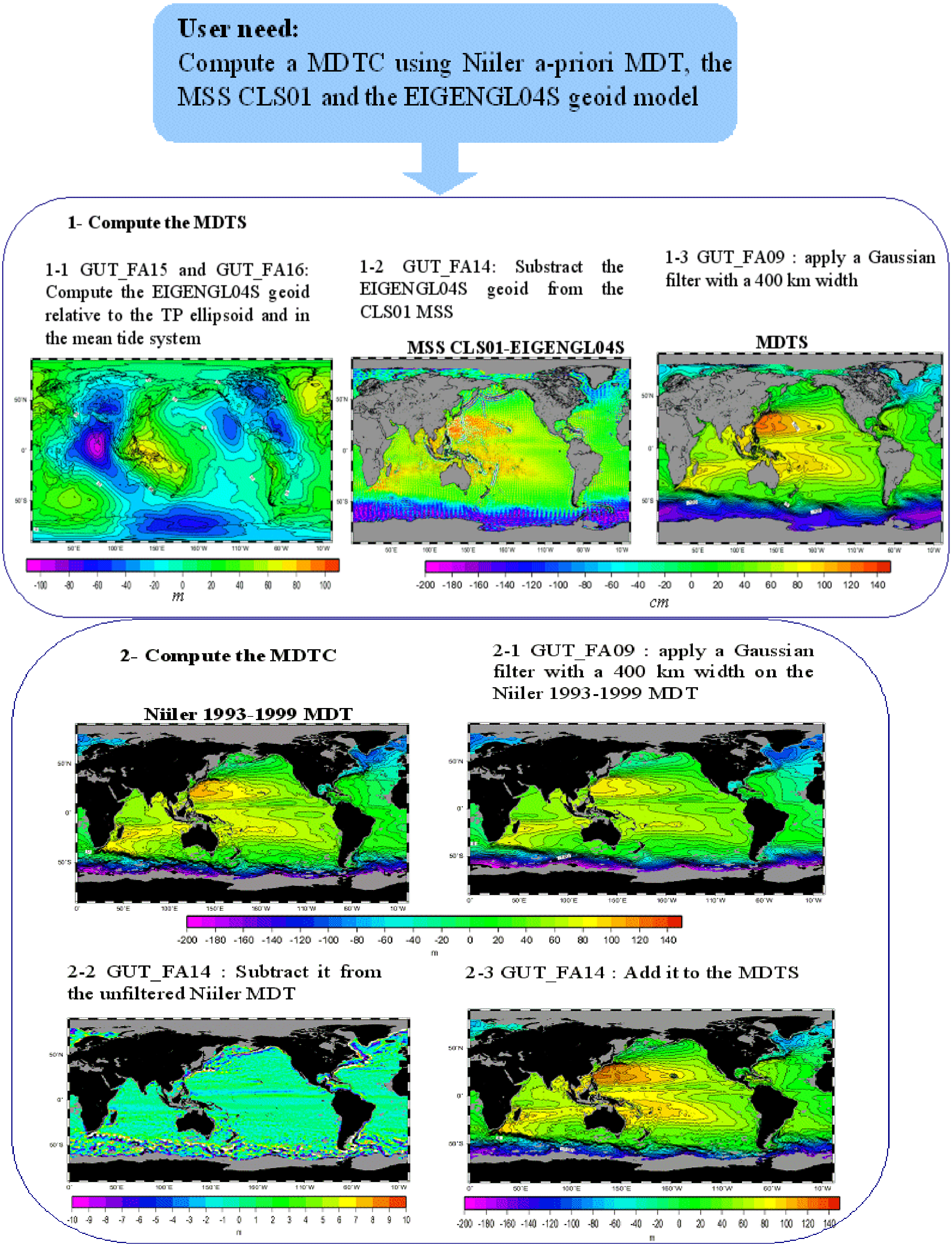


Figure 54: Example 3

5.3 Recommendation for future GUT tutorials

The objectives of the GUTS tutorial were to complete the previous work packages of the GUTS project in order to provide the software engineers who will be involved in the subsequent toolbox building with a clear view of what the GOCE User Toolbox should look like. Although they may serve as a basis for their writing, they differ in content and objective from the future GUT tutorials. It is highly recommended that these should include:

- A detailed and didactic course on the use of geoid data for oceanographic applications (mainly) and their combination with altimetry.
- A “default cases” section explaining and describing the different inputs and outputs that will come out from the toolbox when using the default parameters.
- A number of reproducible use cases: the toolbox should be delivered with a number of input and output fields so that the user, when running the toolbox with the input fields and the parameters described in the tutorial, is able to obtain the provided output fields.

6 Conclusion

The objective of the GOCE User Toolbox Specifications study was to develop – in close collaboration with ESA's HPF effort – algorithms and input and output specification for the subsequent generation of a user toolbox that is required by the general science community for the exploitation of GOCE level 2 and ERS-ENVISAT altimetry. The purpose of the study was accordingly to:

- Consolidate the User Toolbox requirements.
- Carry out a scientific trade off study to select the toolbox processing and viewing functions.
- Produce a Toolbox output specification document.
- Produce an algorithm specification document which details the necessary level for coding.
- Produce a Toolbox architectural design document mapping the required functionality and interfaces such as auxiliary data.

As part of the toolbox requirements consolidation a variety of geophysical studies associated with the Earth's gravity were found to use information in form of geoid heights, gravity anomalies or deflections of the vertical. Those quantities may be represented in the nodes along a profile, in a grid or in discretely located points. Associated with such quantities error covariance information may be needed.

The primary oceanography variable of interest to be provided by a toolbox is the dynamic topography resulting from the difference between altimetric measurements and the geoid model. Altimetric MSSH fields would be auxiliary input data set fields from which a consistently filtered mean dynamic topography need to be computed by the toolbox.

The commonly used quantity in solid earth studies is gravity anomalies and associated error information. Hence, the main required functionalities are covered by general functionalities described above. However, on land the quantities may be needed on the surface of the earth.

In conclusion, this first phase, the GOCE User Toolbox Specification study, has successfully set up the basis for the further implementation of phase 2, the GOCE User Toolbox construction. In addition, it was strongly recommended that two activities should be led in parallel to the toolbox construction as part of the phase 2 project. First, further research is needed into exploiting the full error variance-covariance matrix. Second, it is highly recommended that the GOCE User Toolbox is made available to the users with an updated set of auxiliary data. More particularly, a specific study should be done in order to provide a new altimetric Mean Sea Surface taking advantage of all improved processing of altimetric data, mainly in coastal areas. This new Mean Sea Surface should be provided with an error estimate.

6.1 References

[1] GOCE User Toolbox : User Toolbox Requirement Document.

[2] Haines, K., R. Hipkin, C. Beggan, R. Bingley, F. Hernandez, J. Holt, T. Baker, and R.J. Bingham, Combined use of altimetry and in situ gravity data for coastal dynamics studies. *Space Science Reviews*, 2003. 108(1-2): 205-216.

[3] Losch, M., trade-off study dynamic topography: truth and difference to prior, in *matlab*, wp3300-2.pdf, Editor. 2006.

[4] Losch, M., B.M. Sloyan, J. Schröter, and N. Sneeuw, Box inverse models, altimetry and the geoid: Problems with the omission error. *Journal of Geophysical Research-Oceans*, 2002. 107(C7): -.

[5] Lemoine, F.G., D.E. Smith, L. Kunz, R. Smith, E.C. Pavlis, S.M. Klosko, D.S. Chinn, M.H. Torrence, R.C. Williamson, C.M. Cox, K.E. Rachlin, Y.M. Wang, S.C. Kenyon, R. Salman, R. Trimmer, R.H. Rapp, and R.S. Nerem. The development of the NASA GSFC and DMA joint geopotential model. in *International Symposium on Gravity, Geoid and Marine Geodesy (GraGeoMar96). 1996*. University of Tokyo, Tokyo, Japan.

[6] Fox, A. and K. Haines, Interpretation of water transformations diagnosed from data assimilation. *Journal of Physical Oceanography*, 2003. 33: 485–498.

[7] Haines, K. and R.J. Bingham. Mean dynamic topographies (MDTs) in the GOCINA region. in *GOCINA Workshop. 2006*. Luxembourg.

[8] Wunsch, C. and P. Heimbach, Decadal changes in the North Atlantic meridional overturning and heat flux. *J. Phys. Oceanogr.*, 2006. (in press).

[9] Gerlach, C., L. Foldvary, D. Svehla, T. Gruber, M. Wermuth, N. Sneeuw, B. Frommknecht, H. Oberndorfer, T. Peters, M. Rothacher, R. Rummel, and P. Steigenberger, A CHAMP-only gravity field model from kinematic orbits using the energy integral. *Geophysical Research Letters*, 2003. 30(20): -.

[10] Knudsen, P. and C.C.Tscherning. Error characteristics of dynamic Topography models derived from altimetry and GOCE gravimetry. in *IAG Scientific Assembly. 2005*. Cairns.

[11] Jekeli, C., Spherical harmonic analysis, aliasing, and filtering. *Journal of Geodesy*, 1996. 70(4): 214-223.

[12] McCarthy, D.D. and G. Petit (eds.) *IERS Conventions (2003)*. IERS Technical Note. No. 32. 2004, Verlag des Bundesamtes für Kartographie und Geodäsie: Frankfurt am Main. Available from: http://www.iers.org/iers/publications/tn/tn32/tn32.pdf.

[13] Tscherning, C.C., R. Forsberg, and P.Knudsen. The GRAVSOFT package for geoid determination. in *Continental Workshop on the Geoid in Europe. 1992*. Prague: Research Institute of Geodesy, Topography and Cartography, Prague.

[14] Tscherning, C.C., P. Knudsen, and R. Forsberg:, Description of the GRAVSOFT package, 4. ed. 1994, Geophysical Institute, University of Copenhagen. Available from: http://cct.gfy.ku.dk.

[15] Abramowitz, M. and I.A. Stegun (eds.) *Handbook of Mathematical Functions with Formulas, Graphs, and Mathamatical Tables*. Natioan lBureau of Standards Applied Mathematics Series. No. 55. 1964, United States Department of Commerce. Available from: http://www.convertit.com/Go/ConvertIt/Reference/AMS55.ASP.

[16] *Mean Sea Surface*. Available from: http://www.cls.fr/mss/.

[17] GO-ID-HPF-GS-0041: Product Specification for Level-2 Products and Auxiliary Data Products. Available from: http://esamultimedia.esa.int/docs/GO-ID-HPF-GS-0041_2.0_Product_Specification_Level2.pdf.

[18] Flechtner, F. and J. Ries. *GRACE 327-732 (GR-GFZ-FD-001) Gravity Recovery and Climate Experiment. GRACE Gravity Field Solution Data Formats*. 2003; Available from: ftp://podaac.jpl.nasa.gov/pub/grace/doc/GR-GFZ-FD-001_v1.1.pdf.
[19] Tscherning, C.C. and P.Knudsen. Determination of bias parameters for satellite altimetry by least-squares collocation. in *Hotine - Marussi Symposium. 1986*. Rome: Politecnico di Milano.
[20] Tscherning, C.C., Covariance expressions for second and lower order derivatives of the anomalous potential, in *Reports of the Department of Geodetic science*. 1976, The Ohio State University: Columbus.
[21] Tscherning, C.C., Computation of covariances of derivatives of the anomalous gravity potential in a rotated reference frame. *Manuscripta Geodaetica*, 1993. 18(3): 115-123.
[22] Sanso, F. and W.-D. Schuh, Finite covariance functions. *Bulletin Geodesique*, 1987. 61: 331-347.
[23] Tscherning, C.C., Computation of spherical harmonic coefficients and their error estimates using least-squares collocation. *Journal of geodesy*, 2001. 75(1): 12-18.
[24] Tscherning, C.C., Determination of datum-shift parameters using least squares collocation. *Boll. geodesia sc. Aff.*,, 1976. XXXV(2).
[25] Sanso, F. and C.C.Tscherning, Fast Spherical Collocation - Theory and Examples. *Journal of Geodesy*, 2003. 77(1-2): 101_112, DOI 10.1007/s00190_002_0310_.
[26] Siegismund, F. et al, *GUTS Toolbox Functionality and Algorithm Specification Report*
[27] Bos, A., Cadot, S., Neimeijer, S., *Envisat Atmospheric Toolbox – Software Specification Document Part B: VISAN*, Science [&] Technology Corporation, March 2002
[28] Python.org - http://www.python.org/
[29] Python.org - http://docs.python.org/ext/intro.html
[30] SciPy.org - http://www.scipy.org/
[31] Science [&] Technology bv - http://www.science-and-technology.nl/beat/documentation/visan.html
[32] The MathWorks - http://www.mathworks.com/
[33] Program for Climate Model Diagnosis and Intercomparison (PCMDI) - http://www-pcmdi.llnl.gov/software-portal/cdat/
[34] Unidata - http://ftp.unidata.ucar.edu/software/netcdf/
[35] wxWindows.org - http://www.wxwindows.org/
[36] CLS - http://www.cls.fr/html/oceano/general/applications/ccn_heracles_en.html
[37] VTK.org - http://www.vtk.org/
[38] School of Ocean and Earth Science Technology (SOEST), University of Hawaii - http://www.soest.hawaii.edu/gmt/
[39] Eclipse.org - http://www.eclipse.org/
[40] NetBeans.org - http://www.netbeans.org/
[41] ITT Visual Information Solutions - http://www.ittvis.com/idl/
[42] GOCE Level 2 Product Handbook
[43] Andersen, O. B., Vest, A. L., and Knudsen, P., 2005: The KMS04 Multi-Mission Mean Sea Surface. In Cahiers du Centre Européen de Géodynamique et de Séismologie volume 25. Proceedings of the Workshop: "GOCINA: Improving modelling of ocean transport and climate prediction in the North Atlantic region using GOCE gravimetry". Edited by P. Knudsen, J. Johannessen, T. Gruber, S. Stammer, and T. van Dam. Distributed by: ECGS.
[44] Hernandez, F. and Schaeffer, P., 2001. Surface Moyenne Oceanique: Support Scientifique à la mission altimetrique Jason-1, et à une mission micro-satellite

altimétrique. Contrat SSALTO 2945 - Lot2 - A.1. Rapport final n° CLS/DOS/NT/00.341, CLS, Ramonville St Agne.

[45] Bingham, R., Haines, K., Hughes, C, 2007: Calculating the Ocean's Mean Dynamic Topography from a Mean Sea Surface and a geoid. Submitted to Journal of Geophysical Research.

[46] Hernandez, F. and Schaeffer, P., 2001. Surface Moyenne Oceanique: Support Scientifique à la mission altimetrique Jason-1, et à une mission micro-satellite altimétrique. Contrat SSALTO 2945 - Lot2 - A.1. Rapport final n° CLS/DOS/NT/00.341, CLS, Ramonville St Agne.

[47] Hughes, C. W., Bingham, R. J., 2006: An oceanographer's guide to GOCE and the geoid. Ocean Sci. Discuss., 3, 1543-1568

[48] Le Provost, C., et al., 1999. Impact of GOCE for ocean circulation studies. Final Report n° 13175/98/NL/GD, édité par ESA Study Contract 13175/98/NL/GD.
[49] Le Traon, P.-Y., Hernandez, F., Rio, M.-H., Davidson, F., 2002. How operational oceanography can benefit from dynamic topography estimates as derived from altimetry and improved geoid. In: von Steiger, R. (Ed.), ISSI Workshop. Earth Gravity Field from Space — from Sensors to Earth Sciences. Space and Science Reviews. Kluwer Academic Publisher, Bern, Switzerland, pp. 239–249.
[50] Niiler, P., Maximenko, N.A. and McWilliams, J.C., 2003. Dynamically balanced absolute sea level of the global ocean derived from near-surface velocity observations. Geophysical Research Letters, 30(22): 2164-2167.
[51] Rio, M.-H., Schaeffer, P., Hernandez, F., Lemoine, J.-M. , 2005. The estimation of the ocean Mean Dynamic Topography through the combination of altimetric data, in-situ measurements and GRACE geoid: From global to regional studies. Proceedings of the GOCINA Workshop in Luxembourg (April, 13-15 2005)
[52] Christoph Reigber, Roland Schmidt, Frank Flechtner, Rolf König, Ulrich Meyer, Karl-Hans Neumayer, Peter Schwintzer, Sheng Yuan Zhu (2005): An Earth gravity field model complete to degree and order 150 from GRACE: EIGEN-GRACE02S, Journal of Geodynamics 39(1),1-10
[53] Tapley, B.D., Chambers, D.P., Bettadpur, S. and Ries, J.C., 2003. Large scale ocean circulation from the GRACE GGM01 Geoid. Geophysical Research Letters, 30(22): 2163-2166.
[54] Tapley, B., Ries, J., Bettadpur, S., Chambers, D., Cheng, M., Condi, F., Gunter, B., Kang, Z., Nagel, P., Pastor, R., Pekker, T., Poole, S., Wang, F., J., 2005: GGM02 - An improved Earth gravity model from GRACE. Geodesy, 79:467-478, doi:10.1007/s00190-005-0480-z.
[55] Proceedings of the Second International GOCE User Workshop, March 2004, ESA-ESRIN, ESA Special publication SP-569, June 2004.
[56] Proceedings of the GOCINA workshop, April 2005, Luxembourg, Cahiers du Centre Européen de Géodynamique et de Séismologie, Volume 25, 2006.

7 Appendix A. Products, Input and Output Parameters

EGM_GOC_2: Final GOCE gravity field model

This product contains several subproducts:

- EGM_GCF_2: Spherical harmonic series in ICGEM format
- EGM_GEO_2: Grid with geoid heights in Grid format
- EGM_GAN_2: Grid with gravity anomalies in Grid format
- EGM_GVE_2: Grid with east-west vertical deflections in Grid format
- EGM_GVN_2: Grid with north-south vertical deflections in Grid format
- EGM_GER_2: Grid with geoid height errors in Grid format

All of these will be included in the GUT and the application of these will be described later. The two products: EGM_GER_2 and EGM_GCF_2 will be especially useful and will therefore be described in greater detail.

EGM_GER_2: Grid with geoid height errors in Grid format

This contains the geoid errors of the delivered gridded geoid heights, EGM_GEO_2. These will be an easy way of getting a priori errors for some of the products that GUT will deliver.

This product will, as mentioned, be delivered from the HPF in Grid format. Both a header and a data section will be included. The information contained in the header is:

- Data_set_name
- Northern_latitude. The northern border of the grid in geographical coordinates given in sdd.mmss. Latitudes on the southern hemisphere are given as e.g. -70.
- Southern_latitude. Southern border of the grid
- Western_latitude. Western border of the grid.
- Eastern_latitude. Eastern border of the grid.
- Latitude_cell_size. Grid cell size in latitude direction, given in dd.mmss.
- Longitude_cell_size. Grid cell size in longitude direction.
- Number_of_cells_latitude_dir.
- Number_of_cells_longitude_dir.’
- Mean_(0)_or_point_(1)_values. 0=mean-values, the most North West corner of one grid identifies this grid. 1=point values
- Geocentric(0)_geodetic(0)_lat. 0=geocentric latitudes and 1=geodetic latitudes.
- Reference_ellipsoid
- Format_of_data. The data format given in FORTRAN notation.
- gap_value. The value used for specifying an unknown value in the grid. This number must correspond to the format_of_data format.
- Description_of_data. Description of data records.
- Unit
- Any_comments

- End_of_header

The data section is cut into two sections: The Row Leader Record and the Data Record. The Row Leader contains the following:

- Latitude. The latitude of the current row
- Number_of_data_values.

The Data Record consists of the values written from west to east, with rows going north to south.

EGM_GCF_2: Spherical harmonic series in ICGEM format

This is given in the form of a spherical harmonic expansion of the full gravitational potential, *V*. The full expression for the expansion in fully normalized spherical harmonics is

$$V(r,\theta,\lambda) = \frac{GM}{r}\sum_{n=0}^{N_{\max}}\left(\frac{a}{r}\right)^n \sum_{m=0}^{n}(\overline{C}_{nm}\cos m\lambda + \overline{S}_{nm}\sin m\lambda)\overline{P}_{nm}(\cos\theta)$$

$$\overline{C}_{n0} = \frac{1}{\sqrt{2n+1}}C_{n0}, \quad \overline{P}_{n0}(\cos\theta) = \sqrt{2n+1}P_{n0}(\cos\theta)$$

$$\begin{Bmatrix}\overline{C}_{nm}\\ \overline{S}_{nm}\end{Bmatrix} = \sqrt{\frac{(n+m)!}{2(2n+1)(n-m)!}}\begin{Bmatrix}C_{nm}\\ S_{nm}\end{Bmatrix}, \quad \overline{P}_{nm}(\cos\theta) = \sqrt{\frac{2(2n+1)(n-m)!}{(n+m)!}}P_{nm}(\cos\theta)$$

where θ is the co-latitude, λ is the longitude, *a* is the semimajor axis of the ellipsoid used in the determination of the coefficients, *r* is the height at which this is calculated and the C_{nm} and S_{nm} are the spherical harmonic coefficients and the barred versions are the normalized coefficients. What will be provided from this expansion are the coefficients up to degree and order 250. This expansion forms the basis for the other level-2 products that will be used as input. The provided files will contain a header and a data section. The header will contain the information that does not depend on degree and order. The information given in this is:

- Product type, i.e. “gravity_field”.
- Name of the model.
- The value used for *GM*, which in GOCE is $3.986004415 \cdot 10^{14}$ m^3/s^2
- The radius, *a*. The value used in GOCE is 6378136.3 m
- The maximum degree of the expansion, i.e. 250.
- The errors, this will be either “no”, “calibrated”, “formal”, or “calibrated_and_formal”.
- The normalization of the expansion. Either “fully normalized” or “unnormalized”, as default this is “fully normalized”.
- The tide system used. Either “zero_tide”, “tide_free”, or “unknown”.
- Finally an end_of_head to indicate the end of the header.

Depending on the kind of error specified in the header, the data section will contain different data. What kind of data is given is specified by a parameter at the start of each line. This is followed by the data. The parameters are

- gfc: This gives the degree, order, C_{nm}, S_{nm}, and possibly the errors of these.
- gfct: The same as the above and in addition the time is provided.
- dot: This gives the degree, order, the time derivates of the coefficients, and possibly the errors on these.

- For more in depth information on these parameters look at the *Product Specification for Level-2 products and Auxiliary Data Products* from the GOCE High Level Processing Facility. In general these are just a way of specifying which information is included.

The rest of the products described herein, except for the variance-covariance matrix for the coefficients and the gridded geoid height errors, derive from these coefficients.

EGM_GVC_2: Variance-covariance matrix file of the spherical harmonics coefficients

This product contains the variance-covariance matrix of the coefficients of the spherical harmonic expansion of the gravity potential. The variance-covariance matrix is the result of a least squares estimation of the spherical harmonic coefficients.

As the expansion goes to degree and order 250 this matrix is somewhat large. This means that the matrix is split over several files with the matrix entries of each order *m* in a separate file. This set of files containing the matrix entries are accompanied by a meta-data file describing the data. The content of this meta-data file is:

- The product type which is a variance-covariance matrix
- The model name
- The value of *GM* used.
- The value of *a* used.
- The maximum degree of the spherical harmonic expansion.
- Whether the errors are "formal", "calibrated", or "unknown".
- Whether the covariance is a full system or a block-diagonal system.
- The number of coefficients in the matrix, and next the sequence of the coefficients, with each coefficient written on its own line.
- The number of files used to store the variance-covariance matrix, followed by the names. These are ordered by harmonic order.

The data files contains

- The name of the meta-data file.
- The harmonic order of the file.
- The number of entries in the file.
- A begin_data and an end_data

GUT_001: Grid with geoid height, sph. approx.

This product will take the geoid height grid delivered as a level 2 product from the HPF and by use of linear interpolation (GUT_109) generate a user-specified subgrid. This means that the degree and order of the gridded geoid height are predetermined by the HPF.

Input:

- **EGM_GEO_2**.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in GRAVSOFT grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_002: Grid with gravity anomalies, sph. approx.

This product will take the gravity anomaly grid delivered as a level 2 product from the HPF and by use of linear interpolation (GUT_109) generate a user-specified subgrid. This means that the degree and order of the gridded gravity anomalies are predetermined by the HPF.

Input:

- **EGM_GAN_2.**
- Wanted grid specifications:
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_003: Grid with east-west vertical deflections, sph. approx.

This product will take the E-W vertical deflections grid delivered as a level 2 product from the HPF and by use of interpolation (GUT_109) generate a user-specified subgrid. This means that the degree and order of the gridded deflections are predetermined by the HPF.

Input:

- **EGM_GVE_2**.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_004: Grid with north-south vertical deflections, sph. approx.

This product will take the N-S vertical deflections grid delivered as a level 2 product from the HPF and by use of interpolation (GUT_109) generate a user-specified subgrid. This means that the degree and order of the gridded deflections are predetermined by the HPF.

Input:

- **EGM_GVN_2**.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_005: Height anomaly in grid format, full

The height anomaly – which is usable over land as well as over ocean – is found through the following formula

$$\zeta = \frac{(W-U)_P}{\gamma}$$

$$W = V + \frac{\omega^2}{2}(r\cos\theta)^2$$

$$\gamma = |\nabla U| = \left(U_x^2 + U_y^2 + U_z^2\right)^{1/2}$$

$$U_x = \frac{1}{r\cos\theta}\frac{\partial U}{\partial \lambda}, U_y = \frac{1}{r}\frac{\partial U}{\partial \theta}, U_z = \frac{\partial U}{\partial r}$$

γ is the normal gravity at the point Q on the equipotential surface that has $U_Q=W_P$ at all points. The potentials will be found from the given coefficients and parameters and then subtracted. This is done in a grid.

The "full" means that this product does not make the spherical approximations that are made in the grids coming from the HPF.

Input:

- A set of spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2

- Starting longitude, λ_1
- Ending longitude, λ_2
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_006: Height anomaly in points, full

The height anomaly – which is usable over land as well as over ocean – is found through the following formula

$$\zeta = \frac{(W-U)_P}{\gamma}$$

$$W = V + \frac{\omega^2}{2}(r\cos\theta)^2$$

$$\gamma = |\nabla U| = \left(U_x^2 + U_y^2 + U_z^2\right)^{1/2}$$

$$U_x = \frac{1}{r\cos\theta}\frac{\partial U}{\partial\lambda}, U_y = \frac{1}{r}\frac{\partial U}{\partial\theta}, U_z = \frac{\partial U}{\partial r}$$

γ is the normal gravity at the point Q on the equipotential surface that has U_Q=W_P at all points. The potentials will be found from the given coefficients and parameters and then subtracted. This is done in a set of points.

The "full" means that this product does not make the spherical approximations that are made in the grids coming from the HPF.

Input:

- A set of spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
    - GM, a, J_2, ω
    - GM, a, f, ω
    - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Name of input point file
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_007: Gravity anomaly in grid format, full

The gravity anomalies are found from the norm of the gravity vector and the normal gravity vector derived from the reference ellipsoid. It will be done in the following way to avoid using spherical approximations. The *W* here is the potential *V* plus the potential stemming from the rotation of the Earth, all in all this gives

$$W = V + \frac{\omega^2}{2}(r\cos\theta)^2$$

$$g = |\nabla W| = \left(W_x^2 + W_y^2 + W_z^2\right)^{1/2}$$

$$W_x = \frac{1}{r\cos\theta}\frac{\partial W}{\partial\lambda}, W_y = \frac{1}{r}\frac{\partial W}{\partial\theta}, W_z = \frac{\partial W}{\partial r}$$

$$\gamma = |\nabla U| = \left(U_x^2 + U_y^2 + U_z^2\right)^{1/2}$$

$$U_x = \frac{1}{r\cos\theta}\frac{\partial U}{\partial\lambda}, U_y = \frac{1}{r}\frac{\partial U}{\partial\theta}, U_z = \frac{\partial U}{\partial r}$$

$$\Delta g = g - \gamma$$

Note here that the potential *W* needs to be evaluated at the surface of the Earth while the reference potential *U* needs to be evaluated at the point *Q* determined by $U_Q = W_P$. Depending on which height is given, the ellipsoidal height of the point *P*, *h*, or the normal orthometric height of the point *Q*, *H*, the height anomaly needs to be taken into account. For a given ellipsoidal height, *h*, of the point *P* the height of *Q* is found as $H=h-\zeta$ where *U* then needs to be evaluated to find the gravity anomaly. If the orthometric height, *H*, is given the height of *P* will be $h=H+\zeta$ and *W* then has to be evaluated at this height to get the anomaly.

A way of circumventing this, is using the formula

$$\Delta g = g - \gamma - \frac{2}{r\gamma}\frac{\partial\gamma}{\partial n}(W - U)$$

The last term is a correction term where the differentiation is along the normal. Now this can be evaluated at both points *P* and *Q* but it will give a small error in the gravity anomaly.

The “full” means that this product does not make the spherical approximations that are made in the grids coming from the HPF.

Input:

- A set of spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
    - GM, a, J_2, ω
    - GM, a, f, ω
    - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}

- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_008: Gravity anomaly in points, full

The gravity anomalies are found from the norm of the gravity vector and the normal gravity vector derived from the reference ellipsoid. It will be done in the following way to avoid using spherical approximations. The W here is the potential V plus the potential stemming from the rotation of the Earth, all in all this gives

$$W = V + \frac{\omega^2}{2}(r\cos\theta)^2$$

$$g = |\nabla W| = \left(W_x^2 + W_y^2 + W_z^2\right)^{1/2}$$

$$W_x = \frac{1}{r\cos\theta}\frac{\partial W}{\partial\lambda}, W_y = \frac{1}{r}\frac{\partial W}{\partial\theta}, W_z = \frac{\partial W}{\partial r}$$

$$\gamma = |\nabla U| = \left(U_x^2 + U_y^2 + U_z^2\right)^{1/2}$$

$$U_x = \frac{1}{r\cos\theta}\frac{\partial U}{\partial\lambda}, U_y = \frac{1}{r}\frac{\partial U}{\partial\theta}, U_z = \frac{\partial U}{\partial r}$$

$$\Delta g = g - \gamma$$

Note here that the potential W needs to be evaluated at the surface of the Earth while the reference potential U needs to be evaluated at the point Q determined by $U_Q=W_P$. Depending on which height is given, the ellipsoidal height of the point P, h, or the normal orthometric height of the point Q, H, the height anomaly needs to be taken into account. For a given ellipsoidal height, h, of the point P the height of Q is found as $H=h-\zeta$ where U then needs to be evaluated to find the gravity anomaly. If the orthometric height, H, is given the height of P will be $h=H+\zeta$ and W then has to be evaluated at this height to get the anomaly.

A way of circumventing this, is using the formula

$$\Delta g = g - \gamma - \frac{2}{r\gamma}\frac{\partial\gamma}{\partial n}\left(W - U\right)$$

The last term is a correction term where the differentiation is along the normal. Now this can be evaluated at both points P and Q but it will give a small error in the gravity anomaly.

The “full” means that this product does not make the spherical approximations that are made in the grids coming from the HPF.

Input:

- A set of spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}

- Name of input point file, containing heights of the points
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_009: E-W deflections in grid format, full

The vertical deflections are the differences in direction of the true gravity vector and the normal gravity vector. The formula for the east-west deflections without spherical approximation is

$$\eta = \frac{-W_x}{g}$$

This expression needs to be evaluated above the terrain.

The "full" means that this product does not make the spherical approximations that are made in the grids coming from the HPF.

Input:

- A set of spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, Δφ
  - Grid spacing in longitudinal direction, Δλ

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, Δφ
  - Grid spacing in longitudinal direction, Δλ

GUT_010: E-W deflections in points, full

The vertical deflections are the differences in direction of the true gravity vector and the normal gravity vector. The formula for the east-west deflections without spherical approximation is

$$\eta = \frac{-W_x}{g}$$

This expression needs to be evaluated at the terrain.

The “full” means that this product does not make the spherical approximations that are made in the grids coming from the HPF.

Input:

- A set of spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
    - GM, a, J_2, ω
    - GM, a, f, ω
    - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Name of input point file
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_011: N-S deflections in grid format, full

The north-south deflections are given by

$$\xi = \frac{(U_y - W_y)\cos(\theta - \varphi) + (U_z - W_z)\sin(\theta - \varphi)}{g}$$

where φ is the ellipsoidal latitude. This expression needs to be evaluated above the terrain.

The "full" means that this product does not make the spherical approximations that are made in the grids coming from the HPF.

Input:

- A set of spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
    - GM, a, J_2, ω
    - GM, a, f, ω
    - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_012: N-S deflections in points, full

The north-south deflections are given by

$$\xi = \frac{(U_y - W_y)\cos(\theta - \varphi) + (U_z - W_z)\sin(\theta - \varphi)}{g}$$

where φ is the ellipsoidal latitude. This expression needs to be evaluated at the terrain.

The "full" means that this product does not make the spherical approximations that are made in the grids coming from the HPF.

Input:

- A set of spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
    - GM, a, J_2, ω
    - GM, a, f, ω
    - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Name of input point file
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_013: Geostrophic velocities in grid

The geostrophic velocities are calculated from an ocean topography. The algorithms to do this is

$$u = \frac{-\gamma}{fR}\frac{\partial\zeta}{\partial\theta}$$

$$v = \frac{\gamma}{fR\cos\theta}\frac{\partial\zeta}{\partial\lambda}$$

$$f = 2\omega\sin\theta$$

where ζ is the topography input.

Input:

- The ocean topography from which the velocities are to be calculated. This must be in the form of a grid with the same specifications as the wanted grid.
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- A grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_014: Geostrophic velocities in points

The geostrophic velocities are calculated from an ocean topography. The algorithms to do this is

$$u = \frac{-\gamma}{fR}\frac{\partial\zeta}{\partial\theta}$$

$$v = \frac{\gamma}{fR\cos\theta}\frac{\partial\zeta}{\partial\lambda}$$

$$f = 2\omega\sin\theta$$

where ζ is the topography input.

Input:

- Name of input point file containing the values of the topography at the wanted points
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_015: Grid with geoid height errors, Pre-computed.

The pre-computed geoid height error variances have been found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (see GOCE Level 2 product EGM_GVC_2), i.e.

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{a}\,.$$

where E_x is the variance –covariance matrix containing and the vector a contain the evaluation functional for each coefficient associated with geoid heights in the point P where the geoid height error covariance is computed.

$$\mathbf{a}_j = \left\{ \frac{GM}{r\gamma_Q} \left(\frac{a}{r} \right)^n \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j = \begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The geoid errors are calculated through use of this and given in a grid file.

Input:

- **EGM_GER_2**.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_016: Geoid height errors in grid format.

The geoid height error variances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2) where the vector **a** contain the evaluation functional for each coefficient associated with geoid heights in the point P where the geoid height error covariance is computed, i.e.

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{a} .$$

where E_x is the variance –covariance matrix containing and a is

$$\mathbf{a}_j = \left\{ \frac{GM}{r\gamma_Q} \left(\frac{a}{r} \right)^n \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The geoid errors are calculated through use of this and given in a grid file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, Δφ
  - Grid spacing in longitudinal direction, Δλ

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1

- Ending longitude, λ_2
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_017: Geoid height errors in points.

The geoid height error variances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2) where the vector **a** contain the evaluation functional for each coefficient associated with geoid heights in the point P where the geoid height error covariance is computed, i.e.

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{a} .$$

where E_x is the variance –covariance matrix containing and a is

$$\mathbf{a}_j = \left\{ \frac{GM}{r\gamma_Q} \left(\frac{a}{r} \right)^n \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The geoid errors are calculated through use of this and given in a grid file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Name of input point file
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_018: Gravity anomaly errors in grid format.

The error variances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2) where the vector **a** contain the evaluation functional for each coefficient associated with gravity anomalies in the point P where the geoid height error covariance is computed, i.e.

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{a}\,.$$

where E_x is the variance –covariance matrix containing and a is

$$\mathbf{a}_j = \left\{ \frac{GM}{r} \left(\frac{a}{r} \right)^n \frac{(n-1)}{r} \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The geoid errors are calculated through use of this and given in a grid file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2

- Starting longitude, λ_1
- Ending longitude, λ_2
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_019: Gravity anomaly errors in points.

The error variances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2) where the vector **a** contain the evaluation functional for each coefficient associated with gravity anomalies in the point P where the geoid height error covariance is computed, i.e.

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{a} .$$

where E_x is the variance –covariance matrix containing and a is

$$\mathbf{a}_j = \left\{ \frac{GM}{r} \left(\frac{a}{r} \right)^n \frac{(n-1)}{r} \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The geoid errors are calculated through use of this and given in a grid file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Name of input point file
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_020: E-W deflection errors in grid format.

The error variances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2) where the vector **a** contain the evaluation functional for each coefficient associated with E-W deflections in the point P where the geoid height error covariance is computed, i.e.

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{a} .$$

where E_x is the variance –covariance matrix containing and a is

$$\mathbf{a}_j = \left\{ \frac{GM}{r\gamma_Q a \sin(\theta)} \left(\frac{a}{r}\right)^n \begin{pmatrix} -m\sin(m\lambda) \\ m\cos(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The geoid errors are calculated through use of this and given in a grid file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2

- Starting longitude, λ_1
- Ending longitude, λ_2
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_021: E-W deflection errors in points.

The error variances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2) where the vector **a** contain the evaluation functional for each coefficient associated with E-W deflections in the point P where the geoid height error covariance is computed, i.e.

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{a}\,.$$

where E_x is the variance –covariance matrix containing and a is

$$\mathbf{a}_j = \left\{ \frac{GM}{r\gamma_Q a \sin(\theta)} \left(\frac{a}{r}\right)^n \begin{pmatrix} -m\sin(m\lambda) \\ m\cos(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The geoid errors are calculated through use of this and given in a grid file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Name of input point file
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_022: N-S deflection errors in grid format.

The error variances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2) where the vector **a** contain the evaluation functional for each coefficient associated with N-S deflections in the point P where the geoid height error covariance is computed, i.e.

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{a} \, .$$

where E_x is the variance –covariance matrix containing and a is

$$\mathbf{a}_j = \left\{ \frac{GM}{r\gamma_Q a} \left(\frac{a}{r}\right)^n \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} \frac{\partial P_{nm}(\cos\theta)}{\partial\theta} \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The geoid errors are calculated through use of this and given in a grid file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- A sub-grid of the input grid defined by the grid specifications. This grid will be in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
  - Starting latitude, φ_1
  - Ending latitude, φ_2

- Starting longitude, λ_1
- Ending longitude, λ_2
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_023: N-S deflection errors in points.

The error variances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2) where the vector **a** contain the evaluation functional for each coefficient associated with N-S deflections in the point P where the geoid height error covariance is computed, i.e.

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{a}\,.$$

where E_x is the variance –covariance matrix containing and a is

$$\mathbf{a}_j = \left\{ \frac{GM}{r\gamma_Q a} \left(\frac{a}{r}\right)^n \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} \frac{\partial P_{nm}(\cos\theta)}{\partial\theta} \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The geoid errors are calculated through use of this and given in a grid file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Name of input point file
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_024: Geoid height error covariances on a grid.

The error covariances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2)

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{b}$$

where the vectors **a** and **b** contain the evaluation functional for each coefficient associated with geoid heights in the points P and Q respectively where the geoid height error covariance is computed between, i.e.

$$\left\{ \frac{GM}{r\gamma_Q} \left(\frac{a}{r} \right)^n \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the

header from the Gravsoft grid format, but the array will be defined as $F(\varphi_{11},\varphi_{12}, \varphi_{21}, \varphi_{22},\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22})$ with

- Starting latitude axis 1, φ_{11}
- Ending latitude axis 1, φ_{12}
- Starting latitude axis 2, φ_{21}
- Ending latitude axis 2, φ_{22}
- Starting longitude axis 3, λ_{11}
- Ending longitude axis 3, λ_{12}
- Starting longitude axis 3, λ_{21}
- Ending longitude axis 3, λ_{22}
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_025: Geoid height error covariances in points.

The error covariances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2)

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{b}$$

where the vectors **a** and **b** contain the evaluation functional for each coefficient associated with geoid heights in the points P and Q respectively where the geoid height error covariance is computed between, i.e.

$$\left\{ \frac{GM}{r\gamma_Q} \left(\frac{a}{r} \right)^n \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The error covariances are calculated between two points.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_026: Gravity anomaly error covariances on a grid.

The error covariances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2)

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{b}$$

where the vectors **a** and **b** contain the evaluation functional for each coefficient associated with gravity anomalies in the points P and Q respectively where the error covariance is computed between, i.e.

$$\left\{ \frac{GM}{r} \left(\frac{a}{r} \right)^n \frac{(n-1)}{r} \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the

header from the Gravsoft grid format, but the array will be defined as F(φ_{11},φ_{12}, φ_{21}, φ_{22},λ_{11}, λ_{12}, λ_{21}, λ_{22}) with

- Starting latitude axis 1, φ_{11}
- Ending latitude axis 1, φ_{12}
- Starting latitude axis 2, φ_{21}
- Ending latitude axis 2, φ_{22}
- Starting longitude axis 3, λ_{11}
- Ending longitude axis 3, λ_{12}
- Starting longitude axis 3, λ_{21}
- Ending longitude axis 3, λ_{22}
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_027: Gravity anomaly error covariances in points.

The error covariances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2)

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{b}$$

where the vectors **a** and **b** contain the evaluation functional for each coefficient associated with gravity anomalies in the points P and Q respectively where the error covariance is computed between, i.e.

$$\left\{ \frac{GM}{r} \left(\frac{a}{r} \right)^n \frac{(n-1)}{r} \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The error covariances are calculated between two points given in a point file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
    - GM, a, J_2, ω
    - GM, a, f, ω
    - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_028: E-W deflection error covariances in grid.

The error covariances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2)

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{b}$$

where the vectors **a** and **b** contain the evaluation functional for each coefficient associated with E-W deflections in the points P and Q respectively where the error covariance is computed between, i.e.

$$\left\{ \frac{GM}{r\gamma_Q a \sin(\theta)} \left(\frac{a}{r}\right)^n \begin{pmatrix} -m\sin(m\lambda) \\ m\cos(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the

header from the Gravsoft grid format, but the array will be defined as F(φ_{11},φ_{12}, φ_{21}, φ_{22},λ_{11}, λ_{12}, λ_{21}, λ_{22}) with

- Starting latitude axis 1, φ_{11}
- Ending latitude axis 1, φ_{12}
- Starting latitude axis 2, φ_{21}
- Ending latitude axis 2, φ_{22}
- Starting longitude axis 3, λ_{11}
- Ending longitude axis 3, λ_{12}
- Starting longitude axis 4, λ_{21}
- Ending longitude axis 4, λ_{22}
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_029: E-W deflection error covariances in points.

The error covariances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2)

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{b}$$

where the vectors **a** and **b** contain the evaluation functional for each coefficient associated with E-W deflections in the points P and Q respectively where the error covariance is computed between, i.e.

$$\left\{ \frac{GM}{r\gamma_Q a \sin(\theta)} \left(\frac{a}{r}\right)^n \begin{pmatrix} -m\sin(m\lambda) \\ m\cos(m\lambda) \end{pmatrix} P_{nm}(\cos\theta) \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The error covariances are calculated between two points given in point file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_030: N-S deflection error covariances on a grid.

The error covariances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2)

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{b}$$

where the vectors **a** and **b** contain the evaluation functional for each coefficient associated with N-S deflections in the points P and Q respectively where the error covariance is computed between, i.e.

$$\left\{ \frac{GM}{r\gamma_Q a} \left(\frac{a}{r}\right)^n \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} \frac{\partial P_{nm}(\cos\theta)}{\partial\theta} \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the

header from the Gravsoft grid format, but the array will be defined as F(φ_{11},φ_{12}, φ_{21}, φ_{22},λ_{11}, λ_{12}, λ_{21}, λ_{22}) with

- Starting latitude axis 1, φ_{11}
- Ending latitude axis 1, φ_{12}
- Starting latitude axis 2, φ_{21}
- Ending latitude axis 2, φ_{22}
- Starting longitude axis 3, λ_{11}
- Ending longitude axis 3, λ_{12}
- Starting longitude axis 4, λ_{21}
- Ending longitude axis 4, λ_{22}
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_031: N-S deflection error covariances in points.

The error covariances are found using covariance propagation on the coefficient variance-covariance matrix associated with the spherical harmonic coefficients (GOCE Level 2 product EGM_GVC_2)

$$\sigma_x^2 = \mathbf{a}^T \mathbf{E}_{\hat{\mathbf{x}}} \mathbf{b}$$

where the vectors **a** and **b** contain the evaluation functional for each coefficient associated with N-S deflections in the points P and Q respectively where the error covariance is computed between, i.e.

$$\left\{ \frac{GM}{r\gamma_Q a} \left(\frac{a}{r} \right)^n \begin{pmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{pmatrix} \frac{\partial P_{nm}(\cos\theta)}{\partial\theta} \right\}_{j=\begin{pmatrix} n^2+2m \\ n^2+2m+1 \end{pmatrix}}$$

The error covariances are calculated between two points given in a point file.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
    - GM, a, J_2, ω
    - GM, a, f, ω
    - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_032: Geoid height omission error covariances on a grid.

Using spherical harmonic functions, the omission error covariances associated with the gravity field between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$K(\mathrm{P},\mathrm{Q}) = \sum_{i=N_{max}+1}^{\infty} \sigma_i^{TT} P_i(\cos\psi)$$

where σ_i^{TT} are degree variances associated with the anomalous gravity potential field and ψ is the spherical distance between P and Q. Expressions associated with geoid heights and gravity anomalies are obtained by applying the respective functionals on K(P,Q), e.g. $C_{NN}=L_N(L_N(K(\mathrm{P},\mathrm{Q})))$. These functionals are applied to get e.g. from the covariance of the potential to the covariance of the geoid. The functional is the same that generates the transition from the spherical harmonic sum of the potential to the spherical harmonic sum of the geoid.
The degree variances are expressed as

$$\sigma_i^{TT} = \frac{A}{(i-1)(i-2)(i+4)}\left(\frac{R_B^2}{R^2}\right)^{i+1}$$

where A = 1544850 m^4/s^4, $R_B = R - 6.823$ m.

The omission error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- The variance-covariance matrix of the spherical harmonic coefficients.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the header from the Gravsoft grid format, but the array will be defined as $F(\varphi_{11},\varphi_{12}, \varphi_{21}, \varphi_{22},\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22})$ with
    - Starting latitude axis 1, φ_{11}
    - Ending latitude axis 1, φ_{12}
    - Starting latitude axis 2, φ_{21}
    - Ending latitude axis 2, φ_{22}
    - Starting longitude axis 3, λ_{11}
    - Ending longitude axis 3, λ_{12}
    - Starting longitude axis 4, λ_{21}
    - Ending longitude axis 4, λ_{22}
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_033: Geoid height omission error covariances in points.

Using spherical harmonic functions, the omission error covariances associated with the gravity field between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$K(\mathrm{P},\mathrm{Q}) = \sum_{\mathrm{i}=N_{\max}+1}^{\infty} \sigma_{\mathrm{i}}^{\mathrm{TT}}\, \mathrm{P}_{\mathrm{i}}(\cos\psi)$$

where σ_i^{TT} are degree variances associated with the anomalous gravity potential field and ψ is the spherical distance between P and Q. Expressions associated with geoid heights and gravity anomalies are obtained by applying the respective functionals on K(P,Q), e.g. $C_{NN}=\mathrm{L}_N(\mathrm{L}_N(K(\mathrm{P},\mathrm{Q})))$. These functionals are applied to get e.g. from the covariance of the potential to the covariance of the geoid. The functional is the same that generates the transition from the spherical harmonic sum of the potential to the spherical harmonic sum of the geoid. The degree variances are expressed as

$$\sigma_i^{TT} = \frac{A}{(i-1)(i-2)(i+4)}\left(\frac{R_B^2}{R^2}\right)^{i+1}$$

where A = 1544850 m^4/s^4, $R_B = R - 6.823$ m.

The omission error covariances are calculated between two points given in a point file.

Input:

- Parameters A, R_B *and* R.
- Maximum harmonic degree and order, N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_034: Gravity anomaly omission error covariances on a grid.

Using spherical harmonic functions, the omission error covariances associated with the gravity field between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$K(\mathrm{P},\mathrm{Q}) = \sum_{i=N_{max}+1}^{\infty} \sigma_i^{TT} P_i(\cos\psi)$$

where σ_i^{TT} are degree variances associated with the anomalous gravity potential field and ψ is the spherical distance between P and Q. Expressions associated with geoid heights and gravity anomalies are obtained by applying the respective functionals on K(P,Q), e.g. $C_{NN}=L_N(L_N(K(\mathrm{P},\mathrm{Q})))$. These functionals are applied to get e.g. from the covariance of the potential to the covariance of the geoid. The functional is the same that generates the transition from the spherical harmonic sum of the potential to the spherical harmonic sum of the geoid. The degree variances are expressed as

$$\sigma_i^{TT} = \frac{A}{(i-1)(i-2)(i+4)} \left(\frac{R_B^2}{R^2}\right)^{i+1}$$

where A = 1544850 m^4/s^4, $R_B = R - 6.823$ m.

The omission error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- Parameters A, R_B *and* R.
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the header from the Gravsoft grid format, but the array will be defined as F(φ_{11},φ_{12}, φ_{21}, φ_{22},λ_{11}, λ_{12}, λ_{21}, λ_{22}) with

- Starting latitude axis 1, φ_{11}
- Ending latitude axis 1, φ_{12}
- Starting latitude axis 2, φ_{21}
- Ending latitude axis 2, φ_{22}
- Starting longitude axis 3, λ_{11}
- Ending longitude axis 3, λ_{12}
- Starting longitude axis 4, λ_{21}
- Ending longitude axis 4, λ_{22}
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_035: Gravity anomaly omission error covariances in points.

Using spherical harmonic functions, the omission error covariances associated with the gravity field between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$K(\mathrm{P},\mathrm{Q}) = \sum_{i=N_{max}+1}^{\infty} \sigma_i^{TT} P_i(\cos\psi)$$

where σ_i^{TT} are degree variances associated with the anomalous gravity potential field and ψ is the spherical distance between P and Q. Expressions associated with geoid heights and gravity anomalies are obtained by applying the respective functionals on K(P,Q), e.g. C_{NN}=$L_N(L_N(K(P,Q)))$. These functionals are applied to get e.g. from the covariance of the potential to the covariance of the geoid. The functional is the same that generates the transition from the spherical harmonic sum of the potential to the spherical harmonic sum of the geoid. The degree variances are expressed as

$$\sigma_i^{TT} = \frac{A}{(i-1)(i-2)(i+4)} \left(\frac{R_B^2}{R^2}\right)^{i+1}$$

where A = 1544850 m^4/s^4, $R_B = R - 6.823$ m.

The omission error covariances are calculated between two points given in a point file.

Input:

- Parameters A, R_B *and* R.
- Maximum harmonic degree and order, N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_036: E-W deflection omission error covariances on a grid.

Using spherical harmonic functions, the omission error covariances associated with the gravity field between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$K(\mathrm{P},\mathrm{Q}) = \sum_{\mathrm{i}=N_{\max}+1}^{\infty} \sigma_{\mathrm{i}}^{\mathrm{TT}} \mathrm{P_i}(\cos\psi)$$

where σ_i^{TT} are degree variances associated with the anomalous gravity potential field and ψ is the spherical distance between P and Q. Expressions associated with geoid heights and gravity anomalies are obtained by applying the respective functionals on K(P,Q), e.g. $C_{NN}=\mathrm{L}_N(\mathrm{L}_N(K(\mathrm{P},\mathrm{Q})))$. These functionals are applied to get e.g. from the covariance of the potential to the covariance of the geoid. The functional is the same that generates the transition from the spherical harmonic sum of the potential to the spherical harmonic sum of the geoid. The degree variances are expressed as

$$\sigma_i^{TT} = \frac{A}{(i-1)(i-2)(i+4)} \left(\frac{R_B^2}{R^2}\right)^{i+1}$$

where A = 1544850 $\mathrm{m}^4/\mathrm{s}^4$, $R_B = R - 6.823$ m.

The omission error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- Parameters A, R_B *and* R.
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the header from the Gravsoft grid format, but the array will be defined as $F(\varphi_{11},\varphi_{12}, \varphi_{21}, \varphi_{22},\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22})$ with

- Starting latitude axis 1, φ_{11}
- Ending latitude axis 1, φ_{12}
- Starting latitude axis 2, φ_{21}
- Ending latitude axis 2, φ_{22}
- Starting longitude axis 3, λ_{11}
- Ending longitude axis 3, λ_{12}
- Starting longitude axis 4, λ_{21}
- Ending longitude axis 4, λ_{22}
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_037: E-W deflection omission error covariances in points.

Using spherical harmonic functions, the omission error covariances associated with the gravity field between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$K(\mathrm{P},\mathrm{Q}) = \sum_{\mathrm{i}=N_{\max}+1}^{\infty} \sigma_{\mathrm{i}}^{\mathrm{TT}}\, \mathrm{P}_{\mathrm{i}}(\cos\psi)$$

where σ_i^{TT} are degree variances associated with the anomalous gravity potential field and ψ is the spherical distance between P and Q. Expressions associated with geoid heights and gravity anomalies are obtained by applying the respective functionals on K(P,Q), e.g. $C_{NN}=\mathrm{L}_N(\mathrm{L}_N(K(\mathrm{P},\mathrm{Q})))$. These functionals are applied to get e.g. from the covariance of the potential to the covariance of the geoid. The functional is the same that generates the transition from the spherical harmonic sum of the potential to the spherical harmonic sum of the geoid. The degree variances are expressed as

$$\sigma_i^{TT} = \frac{A}{(i-1)(i-2)(i+4)} \left(\frac{R_B^2}{R^2}\right)^{i+1}$$

where A = 1544850 m^4/s^4, $R_B = R - 6.823$ m.

The omission error covariances are calculated between two points given in point file.

Input:

- Parameters A, R_B *and* R.
- Maximum harmonic degree and order, N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_038: N-S deflection omission error covariances on a grid.

Using spherical harmonic functions, the omission error covariances associated with the gravity field between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$K(\mathrm{P},\mathrm{Q}) = \sum_{\mathrm{i}=N_{\max}+1}^{\infty} \sigma_{\mathrm{i}}^{\mathrm{TT}} \, \mathrm{P}_{\mathrm{i}}(\cos\psi)$$

where σ_i^{TT} are degree variances associated with the anomalous gravity potential field and ψ is the spherical distance between P and Q. Expressions associated with geoid heights and gravity anomalies are obtained by applying the respective functionals on K(P,Q), e.g. C_{NN}=$L_N(L_N(K(P,Q)))$. These functionals are applied to get e.g. from the covariance of the potential to the covariance of the geoid. The functional is the same that generates the transition from the spherical harmonic sum of the potential to the spherical harmonic sum of the geoid. The degree variances are expressed as

$$\sigma_i^{TT} = \frac{A}{(i-1)(i-2)(i+4)} \left(\frac{R_B^2}{R^2}\right)^{i+1}$$

where A = 1544850 m^4/s^4, $R_B = R - 6.823$ m.

The omission error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- Parameters A, R_B *and* R.
- Maximum harmonic degree and order, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the header from the Gravsoft grid format, but the array will be defined as F(φ_{11},φ_{12}, φ_{21}, φ_{22},λ_{11}, λ_{12}, λ_{21}, λ_{22}) with

- Starting latitude axis 1, φ_{11}
- Ending latitude axis 1, φ_{12}
- Starting latitude axis 2, φ_{21}
- Ending latitude axis 2, φ_{22}
- Starting longitude axis 3, λ_{11}
- Ending longitude axis 3, λ_{12}
- Starting longitude axis 4, λ_{21}
- Ending longitude axis 4, λ_{22}
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_039: N-S deflection omission error covariances in points.

Using spherical harmonic functions, the omission error covariances associated with the gravity field between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$K(\mathrm{P},\mathrm{Q}) = \sum_{\mathrm{i}=N_{\max}+1}^{\infty} \sigma_{\mathrm{i}}^{\mathrm{TT}}\, \mathrm{P_i}(\cos\psi)$$

where σ_i^{TT} are degree variances associated with the anomalous gravity potential field and ψ is the spherical distance between P and Q. Expressions associated with geoid heights and gravity anomalies are obtained by applying the respective functionals on K(P,Q), e.g. C_{NN}=$L_N(L_N(K(P,Q)))$. These functionals are applied to get e.g. from the covariance of the potential to the covariance of the geoid. The functional is the same that generates the transition from the spherical harmonic sum of the potential to the spherical harmonic sum of the geoid. The degree variances are expressed as

$$\sigma_i^{TT} = \frac{A}{(i-1)(i-2)(i+4)} \left(\frac{R_B^2}{R^2}\right)^{i+1}$$

where A = 1544850 m^4/s^4, $R_B = R - 6.823$ m.

The omission error covariances are calculated between two points given in a point file.

Input:

- Parameters A, R_B *and* R.
- Maximum harmonic degree and order, N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_040: MDT a-priori covariances on a grid.

Using spherical harmonic functions, the a-priori error covariances associated with the MDT between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$C_{\zeta\zeta} = \sum_{i=N_{Min}}^{N_{Max}} \sigma_i^{\zeta\zeta}\, \mathrm{P_i}(\cos\psi)$$

where the degree variances in this expression are associated with the MDT.

$$\sigma_i^{ss} = b \cdot \left(\frac{k_2^3}{k_2^3 + i^3} - \frac{k_1^3}{k_1^3 + i^3} \right) \cdot s^{i+1}$$

where b, k_1, k_2, and s are determined so that the variance and the correlation length agree with empirically derived characteristics, $b = 6.3\ 10^{-4}\ \mathrm{m}^2$, $k_1 = 1$, $k_2 = 90$, $s = ((R\text{-}5000.0)^2/R^2)^2$.

The omission error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- Parameters b, k_1, k_2 and s.
- Minimum and maximum harmonic degree and order, N_{min} and N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the header from the Gravsoft grid format, but the array will be defined as $F(\varphi_{11},\varphi_{12}, \varphi_{21}, \varphi_{22},\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22})$ with
    - Starting latitude axis 1, φ_{11}
    - Ending latitude axis 1, φ_{12}
    - Starting latitude axis 2, φ_{21}
    - Ending latitude axis 2, φ_{22}
    - Starting longitude axis 3, λ_{11}

- Ending longitude axis 3, λ_{12}
- Starting longitude axis 4, λ_{21}
- Ending longitude axis 4, λ_{22}
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_041: MDT a-priori covariances in points.

Using spherical harmonic functions, the a-priori error covariances associated with the MDT between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$C_{\zeta\zeta} = \sum_{i=N_{Min}}^{N_{Max}} \sigma_i^{\zeta\zeta}\, \mathrm{P_i}(\cos\psi)$$

where the degree variances in this expression are associated with the MDT.

$$\sigma_i^{ss} = b \cdot \left(\frac{k_2^3}{k_2^3 + i^3} - \frac{k_1^3}{k_1^3 + i^3} \right) \cdot s^{i+1}$$

where b, k_1, k_2, and s are determined so that the variance and the correlation length agree with empirically derived characteristics, $b = 6.3\ 10^{-4}\ \mathrm{m}^2$, $k_1 = 1$, $k_2 = 90$, $s = ((R\text{-}5000.0)^2/R^2)^2$.

The omission error covariances are calculated between two points given in a point file.

Input:

- Parameters b, k_1, k_2 and s.
- Minimum and maximum harmonic degree and order, N_{min} and N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_042: MDT a-posteriori covariances on a grid.

Using spherical harmonic functions, the a-posteriori error covariances associated with the MDT between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$C_{\zeta\zeta} = \sum_{i=N_{Min}}^{N_{Max}} \sigma_i \, \mathrm{P_i}(\cos\psi)$$

where the a-posteriori degree variances in this expression are associated with the MDT have been estimated externally.

The omission error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- A-posteriori degree variances σ_i , as an array[N_{min}:N_{max}]
- Minimum and maximum harmonic degree and order, N_{min} and N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the header from the Gravsoft grid format, but the array will be defined as $F(\varphi_{11},\varphi_{12}, \varphi_{21}, \varphi_{22},\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22})$ with
  - Starting latitude axis 1, φ_{11}
  - Ending latitude axis 1, φ_{12}
  - Starting latitude axis 2, φ_{21}
  - Ending latitude axis 2, φ_{22}
  - Starting longitude axis 3, λ_{11}
  - Ending longitude axis 3, λ_{12}
  - Starting longitude axis 4, λ_{21}
  - Ending longitude axis 4, λ_{22}
  - Grid spacing in latitude direction, $\Delta\varphi$

- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_043: MDT a-posteriori covariances in points.

Using spherical harmonic functions, the a-posteriori error covariances associated with the MDT between points P and Q may be expressed as a sum of Legendre's polynomials multiplied by degree variances as

$$C_{\zeta\zeta} = \sum_{i=N_{Min}}^{N_{Max}} \sigma_i \, \mathrm{P_i}(\cos\psi)$$

where the a-posteriori degree variances in this expression are associated with the MDT have been estimated externally.

The omission error covariances are calculated between two points given in a point file.

Input:

- A-posteriori degree variances σ_i , as an array[N_{min}:N_{max}]
- Minimum and maximum harmonic degree and order, N_{min} and N_{max}
- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_044: Geostrophic velocities covariances on a grid.

The covariance functions associated with the geostrophic velocities are given by the formulas

$$C_{uu} = \frac{\gamma^2}{f_P f_Q}\left(-\cos\alpha_{PQ}\cos\alpha_{QP}C_{ll} - \sin\alpha_{PQ}\sin\alpha_{QP}C_{qq}\right)$$

$$C_{vv} = \frac{\gamma^2}{f_P f_Q}\left(-\sin\alpha_{PQ}\sin\alpha_{QP}C_{ll} - \cos\alpha_{PQ}\cos\alpha_{QP}C_{qq}\right)$$

$$C_{u\zeta} = \frac{-\gamma}{f_P}\cos\alpha_{PQ}C_{l\zeta}$$

$$C_{u\zeta} = \frac{\gamma}{f_P}\sin\alpha_{PQ}C_{l\zeta}$$

$$C_{ll} = \frac{1}{R^2}\left(\cos\psi C'_{\zeta\zeta} - \sin^2\psi C''_{\zeta\zeta}\right)$$

$$C_{qq} = \frac{1}{R^2}C'_{\zeta\zeta}$$

$$C_{l\zeta} = \frac{-1}{R}\sin\psi C'_{\zeta\zeta}$$

Where the α_{PQ} is the azimuth between the two points and ψ is the spherical distance between them.

The error covariances are calculated between points given in a grid file. This gives a 4-D output array and should not be attempted with big arrays as the calculation time would be enormous.

Input:

- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
  - Starting latitude, φ_1
  - Ending latitude, φ_2
  - Starting longitude, λ_1
  - Ending longitude, λ_2
  - Grid spacing in latitude direction, $\Delta\varphi$
  - Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The output will be in the form of a 4-D array with two of the axes defined by the latitude limits in the input grid specification and two of the axes defined by the longitude limits in the input grid specification. Thus the header will be identical to the header from the Gravsoft grid format, but the array will be defined as F(φ_{11},φ_{12}, φ_{21}, φ_{22},λ_{11}, λ_{12}, λ_{21}, λ_{22}) with

- Starting latitude axis 1, φ_{11}
- Ending latitude axis 1, φ_{12}
- Starting latitude axis 2, φ_{21}
- Ending latitude axis 2, φ_{22}
- Starting longitude axis 3, λ_{11}
- Ending longitude axis 3, λ_{12}
- Starting longitude axis 4, λ_{21}
- Ending longitude axis 4, λ_{22}
- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_045: Geostrophic velocities covariances in points

The covariance functions associated with the geostrophic velocities are given by the formulas

$$C_{uu} = \frac{\gamma^2}{f_P f_Q}\left(-\cos\alpha_{PQ}\cos\alpha_{QP}C_{ll} - \sin\alpha_{PQ}\sin\alpha_{QP}C_{qq}\right)$$

$$C_{vv} = \frac{\gamma^2}{f_P f_Q}\left(-\sin\alpha_{PQ}\sin\alpha_{QP}C_{ll} - \cos\alpha_{PQ}\cos\alpha_{QP}C_{qq}\right)$$

$$C_{u\zeta} = \frac{-\gamma}{f_P}\cos\alpha_{PQ}C_{l\zeta}$$

$$C_{u\zeta} = \frac{\gamma}{f_P}\sin\alpha_{PQ}C_{l\zeta}$$

$$C_{ll} = \frac{1}{R^2}\left(\cos\psi C'_{\zeta\zeta} - \sin^2\psi C''_{\zeta\zeta}\right)$$

$$C_{qq} = \frac{1}{R^2}C'_{\zeta\zeta}$$

$$C_{l\zeta} = \frac{-1}{R}\sin\psi C'_{\zeta\zeta}$$

Where the α_{PQ} is the azimuth between the two points and ψ is the spherical distance between them. The error covariances are given between two specified points.

Input:

- Specification of the two points for which the covariance is to be calculated.

Output:

- The covariance of the two input points.

GUT_100: Grid addition.

Form sums of the values in two grids:

Grid3 = grid1 + grid2

Input:

- Name of input file of grid1 in gravsoft format
- Name of input file of grid2 in gravsoft format
- Name of output file

Output:

- Values of grid3 in points in the output file in gravsoft grid format.

GUT_101: Grid subtraction.

Form differences between values in two grids:

Grid3 = grid1 – grid2

Input:

- Name of input file of grid1 in gravsoft format
- Name of input file of grid2 in gravsoft format
- Name of output file

Output:

- Values of grid3 in points in the output file in gravsoft grid format.

GUT_102: Grid filtering.

The toolbox will include two filters, a standard box-car filter and a Gaussian filter.

The box-car filter takes an average of the values of the field inside a user defined square box. This means that the user specifies two variables, $\Delta\theta$ and $\Delta\lambda$, which defines the box size. If the point of calculation is called (θ, λ) this leads to the new value at this point given by

$$f(\theta,\lambda) = \frac{1}{n}\sum_{\theta_i,\lambda_i} f(\theta_i,\lambda_j) \quad \text{with}$$

$$\theta - \Delta\theta/2 < \theta_i < \theta + \Delta\theta/2\,, \quad \lambda - \Delta\lambda/2 < \lambda_i < \lambda + \Delta\lambda/2$$

where n is the number of points inside the box.

The Gaussian filter provides a weighted average where the user defines a radius, R, for the filter. Again the user defines a box size with $\Delta\theta$ and $\Delta\lambda$ and the point of calculation is (θ,λ). This gives the equation

$$f(\theta,\lambda) = \frac{\sum_{\theta_i,\lambda_i} e^{r^2/2R^2} f(\theta_i,\lambda_j)}{\sum_{\theta_i,\lambda_i} e^{r/2R^2}} \quad \text{with}$$

$$\theta - \Delta\theta/2 < \theta_i < \theta + \Delta\theta/2\,, \quad \lambda - \Delta\lambda/2 < \lambda_i < \lambda + \Delta\lambda/2$$

$$r^2 = (\lambda - \lambda_j)^2 + (\theta - \theta_i)^2$$

Input:

- Name of input file of grid1 in gravsoft format
- Specification of boxcar or Gaussian filter.
- Filter parameters, $\Delta\theta$, $\Delta\lambda$ and R for the Gaussian
- Name of output file

Output:

- Values of grid2 in points in the output file in gravsoft grid format.

GUT_103: Going from geographical space to spectral.

This functionality will allow the user to get the coefficients of a spherical harmonic expansion from a grid in geographical space. The formulas for this are built on the concept that the spherical harmonics form a complete set. What this means is that if the gridded field is given by $f(\theta,\lambda)$ then the formula to get the coefficient of degree n and order m is

$$\overline{a}_{nm} = \frac{1}{4\pi}\iint_{\sigma} f(\theta,\lambda)\overline{P}_{nm}(\cos\theta)\cos(m\lambda)d\sigma$$

$$\overline{b}_{nm} = \frac{1}{4\pi}\iint_{\sigma} f(\theta,\lambda)\overline{P}_{nm}(\cos\theta)\sin(m\lambda)d\sigma$$

This is the general solution that will not give any factors like those that are usually associated with, e.g. the expansion of the geoid.

For the above formulas to be useful, numerical integration needs to be used. If the field is defined on a grid (θ_i,λ_j) with grid spacings $\Delta\theta$ and $\Delta\lambda$ then the above formulas become

$$\overline{a}_{nm} = \frac{1}{4\pi}\sum_{i,j} f(\theta_i,\lambda_j)\overline{P}_{nm}(\cos\theta_i)\cos(m\lambda_j)\Delta\theta\Delta\lambda$$

$$\overline{b}_{nm} = \frac{1}{4\pi}\sum_{i,j} f(\theta_i,\lambda_j)\overline{P}_{nm}(\cos\theta_i)\sin(m\lambda_j)\Delta\theta\Delta\lambda$$

Input:

- Name of input file containing grid in gravsoft format
- The maximum degree and order for the coefficients

Output:

- A file containing the calculated coefficients in ICGEM format.

GUT_104: Change of reference ellipsoid in grid

To change the reference ellipsoid the following formula will be used.

$$\begin{pmatrix} a\delta\varphi \\ a\cos(\varphi)\delta\lambda \\ \delta h \end{pmatrix} = -\overline{\mathbf{A}}^{-1}\delta\mathbf{r}_0 + \mathbf{C}\begin{pmatrix} \delta\varepsilon_{\overline{X}} \\ \delta\varepsilon_{\overline{Y}} \\ \delta\varepsilon_{\overline{Z}} \end{pmatrix} + \mathbf{F}\begin{pmatrix} \delta a \\ a\delta f \end{pmatrix}$$

$$\overline{\mathbf{A}}^{-1} = \begin{pmatrix} -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ -\sin\lambda & \cos\lambda & 0 \\ \cos\varphi\cos\lambda & \cos\varphi\sin\lambda & \sin\varphi \end{pmatrix}$$

$$\mathbf{C} = a\begin{pmatrix} \sin\lambda & -\cos\lambda & 0 \\ -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{F} = \begin{pmatrix} 0 & \sin 2\varphi \\ 0 & 0 \\ -1 & \sin^2\varphi \end{pmatrix}$$

Here $\delta\mathbf{r}_0$ is the translation vector to the new reference frame and the $\delta\varepsilon$ are the three rotation angles. This gives a correction needed to get from the coordinates of the original reference frame to the new one. The first two parts here are only necessary for a datum shift that is, moving the Earth center and rotating the axis. Thus, usually only the third term is needed.

This will be done on every point of a grid.

Input:

- Grid in gravsoft format
- Transformation parameters, $\delta\mathbf{r}_0$, $\delta\boldsymbol{\varepsilon}$, a and f

Output:

- Transformed grid in gravsoft format

GUT_105: Change of reference ellipsoid in points

To change the reference ellipsoid the following formula will be used.

$$\begin{pmatrix} a\delta\varphi \\ a\cos(\varphi)\delta\lambda \\ \delta h \end{pmatrix} = -\overline{\mathbf{A}}^{-1}\delta\mathbf{r}_0 + \mathbf{C}\begin{pmatrix} \delta\varepsilon_{\overline{X}} \\ \delta\varepsilon_{\overline{Y}} \\ \delta\varepsilon_{\overline{Z}} \end{pmatrix} + \mathbf{F}\begin{pmatrix} \delta a \\ a\delta f \end{pmatrix}$$

$$\overline{\mathbf{A}}^{-1} = \begin{pmatrix} -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ -\sin\lambda & \cos\lambda & 0 \\ \cos\varphi\cos\lambda & \cos\varphi\sin\lambda & \sin\varphi \end{pmatrix}$$

$$\mathbf{C} = a\begin{pmatrix} \sin\lambda & -\cos\lambda & 0 \\ -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{F} = \begin{pmatrix} 0 & \sin 2\varphi \\ 0 & 0 \\ -1 & \sin^2\varphi \end{pmatrix}$$

Here $\delta\mathbf{r}_0$ is the translation vector to the new reference frame and the $\delta\varepsilon$ are the three rotation angles. This gives a correction needed to get from the coordinates of the original reference frame to the new one. The first two parts here are only necessary for a datum shift, that is, moving the Earth center and rotating the axis. Thus, usually only the third term is needed.

This will be done on each point in the list.

Input:

- Point list in gravsoft format
- Transformation parameters, $\delta\mathbf{r}_0$, $\delta\boldsymbol{\varepsilon}$, a and f

Output:

- Transformed points in gravsoft format

GUT_106: Change of tide system in a grid

The three tidal models used are mean tide, zero tide, and non-tide. Conversion of geoid height between these three models uses the following formulas.

The first for conversion from zero tide geoid to mean tide geoid:

$$N_m - N_z = \frac{GMr^2}{4gd}\left(\frac{3}{2}\sin^2\varepsilon - 1\right)\left(3\sin^2\varphi - 1\right)$$

Next for non-tidal geoid to zero geoid:

$$N_z - N_n = k\left(\frac{GMr^2}{4gd}\left(\frac{3}{2}\sin^2\varepsilon - 1\right)\left(3\sin^2\varphi - 1\right)\right)$$

The third is for non-tidal geoid to mean geoid:

$$N_m - N_n = (1-k)\left(\frac{GMr^2}{4gd}\left(\frac{3}{2}\sin^2\varepsilon - 1\right)\left(3\sin^2\varphi - 1\right)\right)$$

Here d is the geocentric distance to the moon, ε is the inclination of the ecliptic to the equator, g is the gravity, and k is a Love's number.

This will be calculated at every point of the grid

Input:

- Grid in gravsoft format
- The parameter k
- The current tidal model and the wanted tidal model

Output:

- Transformed grid in gravsoft format

GUT_107: Change of tide system at points

The three tidal models used are mean tide, zero tide, and non-tide. Conversion of geoid height between these three models uses the following formulas.

The first for conversion from zero tide geoid to mean tide geoid:

$$N_m - N_z = \frac{GMr^2}{4gd}\left(\frac{3}{2}\sin^2\varepsilon - 1\right)\left(3\sin^2\varphi - 1\right)$$

Next for non-tidal geoid to zero geoid:

$$N_z - N_n = k\left(\frac{GMr^2}{4gd}\left(\frac{3}{2}\sin^2\varepsilon - 1\right)\left(3\sin^2\varphi - 1\right)\right)$$

The third is for non-tidal geoid to mean geoid:

$$N_m - N_n = (1-k)\left(\frac{GMr^2}{4gd}\left(\frac{3}{2}\sin^2\varepsilon - 1\right)\left(3\sin^2\varphi - 1\right)\right)$$

Here d is the geocentric distance to the moon, ε is the inclination of the ecliptic to the equator, g is the gravity, and k is a Love's number.

This will be calculated at every point of the grid

Input:

- Point list in gravsoft format
- The parameter k
- The current tidal model and the wanted tidal model

Output:

- Transformed point list in gravsoft format

GUT_108: Calculation of reference potential, grid

A necessary part of the toolbox is being able to calculate a reference potential from a set of variables with values set by the user. There are different sets of variables that can be used namely

- GM, a, J_2, ω
- GM, a, f, ω
- γ_a, a, f, ω

The potential is then calculated through use of

$$U = \frac{GM}{r}\left(1 - \sum_{n=1}^{\infty} J_{2n}\left(\frac{a}{r}\right)^{2n} P_{2n}(\cos\theta)\right)$$

$$J_{2n} = (-1)^{2n}\frac{3e^{2n}}{(2n+1)(2n+3)}\left(1 - n + 5n\frac{C-A}{ME^2}\right), \quad e = \frac{E}{a}$$

where C and A are moments of inertia, C with respect to the rotation axis and A with respect to any axis in the equatorial plane, E is the eccentricity, and e is the first eccentricity. Dependent on the variables given different ways to find the J_{2n}'s must be used. However they all use the formulas

$$G(C-A) = \frac{1}{3}GME^2\left(1 - \frac{2}{15}\frac{me'}{q_0}\right), \; m = \frac{\omega^2 a^2 b}{GM}, \; e' = \frac{E}{b}$$

$$q_0 = \frac{1}{2}\left[\left(1 + 3\frac{b^2}{E^2}\right)\tan^{-1}\frac{E}{b} - 3\frac{b}{E}\right]$$

Input:

- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
    - GM, a, J_2, ω
    - GM, a, f, ω
    - γ_a, a, f, ω
- Maximum harmonic degree, N_{max}
- Altitude of the grid points. This altitude should be above the terrain.
- Wanted grid specifications:
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2

- Grid spacing in latitude direction, $\Delta\varphi$
- Grid spacing in longitudinal direction, $\Delta\lambda$

Output:

- The specified grid in gravsoft grid format. This format contains a line at the start of the file containing in order and in decimal degree notation
    - Starting latitude, φ_1
    - Ending latitude, φ_2
    - Starting longitude, λ_1
    - Ending longitude, λ_2
    - Grid spacing in latitude direction, $\Delta\varphi$
    - Grid spacing in longitudinal direction, $\Delta\lambda$

GUT_109: Calculation of reference potential, point

A necessary part of the toolbox is being able to calculate a reference potential from a set of variables with values set by the user. There are different sets of variables that can be used namely

- GM, a, J_2, ω
- GM, a, f, ω
- γ_a, a, f, ω

The potential is then calculated through use of

$$U = \frac{GM}{r}\left(1 - \sum_{n=1}^{\infty} J_{2n}\left(\frac{a}{r}\right)^{2n} P_{2n}(\cos\theta)\right)$$

$$J_{2n} = (-1)^{2n} \frac{3e^{2n}}{(2n+1)(2n+3)}\left(1 - n + 5n\frac{C-A}{ME^2}\right), \quad e = \frac{E}{a}$$

where C and A are moments of inertia, C with respect to the rotation axis and A with respect to any axis in the equatorial plane, E is the eccentricity, and e is the first eccentricity. Dependent on the variables given different ways to find the J_{2n}'s must be used. However they all use the formulas

$$G(C-A) = \frac{1}{3}GME^2\left(1 - \frac{2}{15}\frac{me'}{q_0}\right), m = \frac{\omega^2 a^2 b}{GM}, e' = \frac{E}{b}$$

$$q_0 = \frac{1}{2}\left[\left(1 + 3\frac{b^2}{E^2}\right)\tan^{-1}\frac{E}{b} - 3\frac{b}{E}\right]$$

Input:

- EGM_GCF_2.
- The specifications for the reference ellipsoid, this input can be in several forms. The choices are
  - GM, a, J_2, ω
  - GM, a, f, ω
  - γ_a, a, f, ω
- Maximum harmonic degree and order, N_{max}
- Name of input point file, containing heights of the points
- Name of output file

Output:

- Values in points in the output file in gravsoft point list format.

GUT_110: Interpolation

Interpolation of some sort will be needed in the GUT to change grids and make subgrids of already existing ones.

The algorithm given here is just an example and is only of use on regular grids. The idea is to be able to find the value of a function F(x,y) at some point (x,y) that is not located on the grid. It will however be contained in a rectangle where the four corners are grid points. These will be called – in the order lower left corner, lower right corner, upper right corner, and upper left corner – (x_1,y_1), (x_2,y_1), (x_2,y_2), and (x_1,y_2) with the values $F(x_1,y_1)$, $F(x_2,y_1)$, $F(x_2,y_2)$, and $F(x_1,y_2)$. Define

$$t = \frac{(x - x_1)}{(x_2 - x_1)}$$

$$u = \frac{(y - y_1)}{(y_2 - y_1)}$$

The value of the function F at the point (x,y) will then be given by

$$F(x,y) = (1-t)(1-u)F(x_1,y_1) + t(1-u)F(x_2,y_1) + tuF(x_2,y_2) + (1-t)uF(x_1,y_2)$$

Input:

- Grid to be interpolated from
- Points or grid wanted in output
    - If grid then the grid specifications need to be given, that is
        - Starting latitude, φ_1
        - Ending latitude, φ_2
        - Starting longitude, λ_1
        - Ending longitude, λ_2
        - Grid spacing in latitude direction, $\Delta\varphi$
        - Grid spacing in longitudinal direction, $\Delta\lambda$
    - If points then specification of latitude and longitude for the points

Output:

- Grid or points specified as input.

7.1 Calculating sums of spherical harmonic functions

The starting points for finding the sums of the spherical harmonic expansions are the recurrence relations for the associated Legendre functions

$$
\begin{aligned}
&\overline{P}_{nm}(\theta) = a_{nm} t \overline{P}_{n-1,m}(\theta) - b_{nm} \overline{P}_{n-2,m}(\theta),\ n > m \\
&\overline{P}_{mm}(\theta) = u\sqrt{\frac{2m+1}{2m}}\overline{P}_{m-1,m-1}(\theta) = u^m \sqrt{3}\prod_{i=2}^{m}\sqrt{\frac{2i+1}{2i}},\ m > 1 \\
&a_{nm} = \sqrt{\frac{(2n-1)(2n+1)}{(n-m)(n+m)}} \quad , \quad b_{nm} = \sqrt{\frac{(2n+1)(n+m-1)(n-m-1)}{(n-m)(n+m)(2n-3)}} \\
&t = \cos\theta \quad , \quad u = \sin\theta
\end{aligned}
$$

These relations are stable up to about degree 2700. The full sum on a grid is then found through

$$
\begin{aligned}
&S = c\sum_{m=0}^{N_{\max}}\Omega_m \\
&\Omega_m = \sum_{\alpha=1}^{2}\begin{cases} X_{m\alpha}\cos m\lambda & \text{for } \alpha = 1 \\ X_{m\alpha}\sin m\lambda & \text{for } \alpha = 2 \end{cases} \\
&X_{m\alpha} = \sum_{n=\mu}^{N_{\max}} \overline{E}_{nm\alpha}\overline{P}_{nm}(\theta)
\end{aligned}
$$

Here μ is a number that varies with m, and $E_{nm\alpha}$ incorporates the coefficients of the expansion. As an example look at the Earth's gravity potential as given in the beginning of this document. Here one would get

$$
\begin{aligned}
&\overline{E}_{nm\alpha} = \begin{cases} \left(\dfrac{a}{r}\right)^n \overline{C}_{nm} & \text{for } \alpha = 1 \\ \left(\dfrac{a}{r}\right)^n \overline{S}_{nm} & \text{for } \alpha = 2 \end{cases} \\
&X_{m\alpha} = \sum_{n=\mu}^{N_{\max}} \left(\frac{a}{r}\right)^n \overline{E}_{nm\alpha}\overline{P}_{nm}(\theta)
\end{aligned}
$$

The reason the sums are done this way is because of computing time. In this way only $X_{m\alpha}$ depend on the latitude and much computation time can be saved. The sum can be found in several ways. One way is to just calculate the Legendre functions to the needed degree and order and then manually doing the sum. Another way is to use Clenshaw summation. Clenshaw summation is useful because it greatly reduces the amount of calculations necessary. Clenshaw summation is tailored to easily do sums in the form of

$$S = \sum_{n=m}^{N} y_n P_n(x) = y^T p$$

where the y_n's are coefficients of some series and P_n is a series of polynomials for which a recurrence relation of the following form exists

$$P_n(x) + a_n(x) P_{n-1}(x) + b_n(x) P_{n-2}(x) = 0$$

The sum is then given by the equations

$$\begin{aligned}
s_n &= -a_{n+1} s_{n+1} - b_{n+2} s_{n+2} + y_n \\
S &= s_m P_m(x) \\
s_{N+1} &= s_{N+2} = 0
\end{aligned}$$

This is a recursive relation which when the termination condition is reached will only depend on the coefficients and the polynomial P_m.

For finding the sum of the derivative polynomial another set of formulas are needed together with the assumption that $a_n(x)$ is linear in x.

$$\begin{aligned}
a_n(x) &= a'_n x + a''_n \\
s'_n &= -a_{n+1} s'_{n+1} - b_{n+2} s'_{n+2} - a'_{n+1} s_{n+1} \\
dS &= s'_m P_m + s_m dP_m \\
s'_{N+1} &= s'_{N+2} = 0
\end{aligned}$$

This will only depend on the coefficients and the polynomial and its first derivative.

It can be seen from the above that Clenshaw summation can be used to find expressions for the $X_{m\alpha}$ only from knowledge of the sectorial associated Legendre functions.

8 Appendix B. Algorithm specification

The algorithms as defined in this section are based on the fundamental algorithms and formulae defined in the User Requirements Document (URD) [1], composited to be consistent with the toolbox functionality as defined above. Each algorithm corresponds to a "processing" block in the workflow definitions. The basic algorithms to be implemented within each defined function are given. Where existing software exists that provides the required functionality, the software is also defined.

For each function, the input is divided into *options*, *input data* and *output data*. Some of the required input data are to be provided with the toolbox as default data sets, such as the GOCE level 2 data sets and a global a-priori MDT. Those data sets are indicated as *GUT default data*. By choosing the default options the functionality as defined in the main workflow given in (Figure 12) can be fully accessed by the user without providing additional data. When using a different option setting, the user may be required to provide additional data. However, each data set provided with GUT can be replaced by the user's own data using a format supported by the toolbox.

The potential sources of these parameters and the final output file format specifications are given in the Input/Output Definitions.

The algorithms are grouped by type. The first section: "Algorithms for Preference Selection" are required to provide parameters to other toolbox functions but do not include basic algorithms defined in the URD. The second section, "Computational Algorithms", generate the primary toolbox products, using the functionality of the toolbox as defined in section 3.3.

8.1.1 Conventions and Definitions:

Throughout the algorithm definition document, unless otherwise explicitly defined, the following standard variable name definitions and conventions will be used

Standard variable name definitions:

θ is geocentric co-latitude

φ is geodetic latitude

λ is longitude

C_{nm} and S_{nm} are the spherical harmonic coefficients a degree *n* and order *m*. An overbar denotes the normalised version.

$P_{n,m}$ are the associated Legendre functions of degree *n* and order *m*. An overbar denotes the normalised version.

N_{max} is the maximum degree and order of the expansion.

W is the geopotential

a is the semimajor axis of the ellipsoid used in the determination of the coefficients (The GOCE product standard has not yet been selected)

f is the inverse flattening coefficient of the reference ellipsoid

GM is the product of *G,* the gravitational constant, and *M*, the combined mass of the earth and atmosphere, which is known to much better accuracy than either individually (for GOCE products, $GM = 3.986004415 \cdot 10^{14}$ m^3/s^2.

r is the height (from the centre of the earth) at which a value is calculated

ω is the angular rotation of the earth reference

V is the full gravitational potential

U is a reference gravitational potential

γ is the normal gravity at a point Q on the equipotential surface that has $U_Q = W_P$ at all points.

ζ is the geoid height anomaly

Conventions

Latitude values are always given in degrees N, *i.e.* 90˚S is given as -90˚.

Longitude values are always given as degrees E, *i.e.* 180˚W is given as -180˚.

All heights are given in m.

Gravity Anomalies are given in mgals.

Deflections from the Vertical are given in arcsec.

Regular grid definition requires:

Minimum latitude
Maximum latitude
Minimum longitude
Maximum longitude
Latitude grid cell dimension
Longitude grid cell dimension

Irregular grid definition requires:

Vector of latitude values of each grid node
Vector of corresponding longitude values of each grid node

Data Point definition requires:

Vector of latitude values of each data point
Vector of corresponding longitude values of each data point

Reference ellipsoid description coefficients will be in one of the recognised forms:

- GM, a, J_2, ω
- GM, a, f, ω
- γa, a, f, ω

8.1.2 Algorithms for preference selection

selection of required degree and order of expansion

Algorithm is required for all cases where spherical harmonic synthesis is to be used in calculation of a spatial field.

Options:

-DO *max_DO*: *max_DO* is the maximum degree and order of expansion to use

-Ddeg *res_deg*: *res_deg* is the required resolution (half wave length) in degrees latitude

-Dkm *res_km*: *res_km* is the required resolution (half wave length) in km.

Input:

Data set with spherical harmonic coefficients

GUT default data: EGM_GCF_2

Output:

Maximum degree and order of input spherical harmonic coefficients to be used in further calculations

Units: dimensionless

Algorithm:

Case:

-: default (no options provided):

output: degree and order to use = the maximum degree and order of the spherical harmonic data (provided as input)

-DO: User provided degree and order of coefficients to use:

where *max_DO* > maximum degree (order) of input data:
output = maximum degree (order) of input data
where *max_DO* <= maximum degree (order) of input data:
output = requested degree (order)

-Dkm: User provided preferred highest resolution of calculated field

Convert resolution in km (if given) to degrees latitude using:
res_deg = *res_km* * 90 / 1000
goto –Ddeg

-Ddeg: User provided preferred highest resolution in degrees latitude

Convert highest requested resolution in degree latitude to an approximate corresponding degree and order using:
Max_DO = 360 / (2**res_deg)* – value to be rounded up to next integer value
Revert to -DO above to return maximum degree and order to be used

Note – in all cases the returned values must be reported. User should be informed of maximum degree and order in the input data. If a lower degree and order than requested has been returned, user should be explicitly informed of this.

URD Algorithm functionality defined by this algorithm

None

Existing software providing this algorithm

None

Reference system selection:

Reference system selection is required in all cases where data, referenced to an earth ellipsoid, are to be combined or compared. It is also required for spherical harmonic synthesis calculations.

Where height data (geoid, mean sea surface etc.) are calculated or reported relative to a reference ellipsoid, the definition of the ellipsoid used must be known and consistent with other fields for combination or comparison, *e.g.* differencing of two geoid fields or generation of a dynamic topography by combination of geoid and sea surface height information. For harmonic synthesis, the values of full reference ellipsoid coefficients used in the geoid model will be required. For conversion of gridded fields, it is assumed that GM and ω will remain constant and only the ellipsoid axis values a and f, will be varied.

The default reference ellipsoid should be that used in the Mean Sea Surface Height included within the GOCE User Toolbox (see the I/O Definitions section 8.2).

The output of this function will be required for:

1) spherical harmonic synthesis calculations
2) conversions of existing spatial fields between reference systems using GUT_104: *Change of Reference Ellipsoid in Grid* and GUT_105: *Change of Reference Ellipsoid in points*

Options:

-

Input:

Preferred reference ellipsoid. (Default: Reference ellipsoid used for GUT *a priori* Mean Sea Surface Height field.)

Output:

Reference ellipsoid description coefficients in one of the recognised forms

Algorithm:

Select default values

Or

Select data definition coefficients

URD Algorithm functionality defined by this algorithm

None:

Existing software providing this algorithm

None

tide system selection

The tide system used for a gravity field model will be one of three types:
Tide-free (or nontidal) - This geoid would exist for a tide-free Earth with all (direct and indirect) effects of the Sun and Moon removed.
Zero-tide - This geoid would exist if the permanent direct effects of the Sun and Moon are removed, but the indirect effect (mean component) related to the elastic deformation of the Earth is retained.
Mean-tide – The normal system used in altimetry where the permanent effects of the tides are retained.
It is expected that either zero or tide-free will be used for the GOCE level 2 products [12]. The tide system used must be known for comparison of gravity field data, *e.g.* differencing of geoid fields.
Only the degree 2 zonal component ($C_{2,0}$ component) is affected.
In order to transfer a gravity field spherical harmonic series between tide-free and zero-tide tide systems the following equation has to be applied to the $C_{2,0}$ coefficient:

$$C_{2,0}^{tide-free} = C_{2,0}^{zero-tide} + 4.201 \times 10^{-9}$$

In order to transfer a spherical harmonic series between mean-tide and zero-tide tide systems the following equation has to be applied to the $C_{2,0}$ coefficient:

$$C_{2,0}^{zero-tide} = C_{2,0}^{mean-tide} + 1.39 \times 10^{-8}$$

The output of this function will be required for:

spherical harmonic synthesis calculations

conversions of existing spatial fields between reference systems using GUT_106: Change of tide system in a grid or GUT_107: Change of tide system at points

Options:

-Tmean-tide

-Tzero-tide

-Ttide-free

Default: The tide system used in the GOCE level 2 geoid harmonic analysis (default tide system)

Input:

None

Output:

Tide system to use: 'zero-tide', 'tide-free' or 'mean-tide'.

Algorithm:

Selection of default or preferred tide system: 'zero-tide', 'tide-free' or 'mean-tide'

Existing software providing this algorithm

None

8.1.3 Computational Algorithms

Geodetic field computation

Required for determination of geoid height, gravity anomaly and deflections of the vertical (E-W and N-S) by the toolbox.

Options:

-Fg: geoid height computation (default)

-Fa: gravity anomaly computation

-Fd: geoid deflections from the vertical

-R *e/w/n/s*: Region of interest in the order western, eastern, northern and southern limits in degrees north of the equator and east of the prime meridian (default: 79.75/-79.75/0.25/359.75). Only used for regular (latitude-longitude) grid.

-I *dx/dy*: grid increment for regular (latitude-longitude) grid in degrees (default: 0.25/0.25)

-Gg *grid_definition_file_x/grid_definition_file_y*: An irregular grid is used. The files contain matrices with latitude/ longitude definitions of the grid. The –I and –R options are ignored

-Gu *specific_points_definition_file*: Specific points are defined by a list given in the file. Two columns are expected containing the longitude in the first and the latitude in the second column. The –I and –R options are ignored.

-H *digital_terrain_model*: The geodetic field is computed on the terrain as defined in *digital_terrain_model*. By default the reference ellipsoid is taken. A default terrain model is part of the GUT distribution and is used if no filename is given. –H is ignored if –Fg is selected.

-Ma: The area average for the grid cell is computed. By default the geodetic field is computed on the specified grid points. –Ma is ignored when using –Gu.

Input

- Data set with spherical harmonic coefficients (*GUT default data:* EGM_GCF_2)
- maximum degree and order of the expansion to use in the computation (from .GUT_FA01).
- Reference system selection (from .GUT_FA02)
- Preferred tide system: (from .GUT_FA03)

Output:

For grids:

- Depending on output parameter:

-Fg: 2D matrix (grid) of geoid (m)

-Fa: 2D matrix (grid) of gravity anomaly (mgals)

-Fd: 2D matrix (grid) of the E-W deflections from the vertical and 2D matrix (grid) of the N-S deflections from the vertical (arcsecs).

- Locations of grid nodes as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.
- Flag reflecting whether point-wise or area averaged values (-Ma option) have been calculated

For User Specified Points (-Gu):

- Depending on output parameter:

-Fg: Vector of geoid at each data point (m)

-Fa: Vector of gravity anomaly at each data point (mgals)

-Fd: Vector of the E-W deflections from the vertical and vector of the N-S deflections from the vertical at each data point (arcsecs).

- Locations of data points as 2 vector arrays of latitudes and longitudes (degrees).

For All Cases:

- Maximum degree and order and approximate associated spatial scale (in spatial dimensions, *i.e.* km or degrees lat) of the calculation.
- Reference ellipsoid definition of the calculated field
- For gravity anomaly or deflections from the vertical – the actual height above the reference ellipsoid for the calculation.

URD Algorithm functionality defined by this algorithm

This functional algorithm includes the application of the GUTS algorithms:

GUT_005: *Height anomaly in grid format, full*

GUT_006: *Height anomaly in points, full*

GUT_007: *Gravity anomaly in grid format, full*

GUT_008: *Gravity anomaly in points, full*

GUT_009: *E-W deflections in grid format, full*

GUT_010: *E-W deflections in points, full*

GUT_011: *N-S deflections in grid format, full*

GUT_012: *N-S deflections in points, full*

The spherical approximation versions of the geodetic field calculations (GUT_001: *Grid with geoid height, sph. approx.*, GUT_002; *Grid with gravity anomalies, sph. approx.* , GUT_003: *Grid with east-west vertical deflections, sph. approx.* and GUT_004: *Grid with* *north-south vertical deflections, sph. approx.*), used in the GOCS level 2 gridded products, are not recommended for inclusion in the toolbox.

It is also not considered a priority to provide the calculation of reference potential (GUT_108: *Calculation of reference potential, grid* and GUT_109: *Calculation of reference potential, point*) in the current implementation of the toolbox

Existing software providing this algorithm

1) Routines within the GRAVSOFT fortran package [13, 14]. In particular the GEOCOL and HARMEX programs. These programs currently expect data in external files, but could be re-coded as functions to allow use of previously input data.

2) Harmonic_synth package of Simon A. Holmes and Nikolaos K. Pavlis.

Fortran programme initiated by a series of editable bourne shell scripts.

Current implementation required generation of a specific fortran source code file for each activation. The source code file contains internal parameters that set the required input for the algorithm. This source code is then compiled and executed.

The current implementation of this code uses the spherical harmonic approximations of the required parameters (GUT_001, GUT_002, GUT_003, GUT_004) and would require code changes to implement the 'full' calculation as recommended for the toolbox.

Existing package could be recoded rather easily to allow input of required parameters from external script using values defined by required or optional input as specified above.

Alternative software solutions are found in the GSHBUNDLE matlab software of Nico Sneeuw. These functions could, potentially, be recoded in fortran to provide the required functionality.

Commission error determination (geoid)

- Ideally, the toolbox would allow the calculation of the full variances and error covariances using the spherical harmonic error covariance data from the level 2 product: EGM_GVC_2. However, it is recommended that, in order to create a toolbox with realistic computational requirements, the error calculations are restricted to interpolations if the error fields generated from the level 2 gridded products.

The calculation of full error covariances for a grid is considered beyond the scope of the GOCE user toolbox due to computational resource constraints.
Even where it might be considered a possible option, as the full error covariance information available for a high degree and order spherical harmonic expansion of a geoid field is of the order of several Gigabytes in size, it is not feasible to separate the read method from the input data file. The algorithms will need to extract parts of the error covariance matrix data file.

Options:

-Fg: geoid height commission error computation (default)

-Fa: gravity anomaly commission error computation

-Fd: geoid deflections from the vertical commission error computation

-R *e/w/n/s*: Region of interest in the order western, eastern, northern and southern limits in degrees north of the equator and east of the prime meridian (default: 79.75/-79.75/0.25/359.75). Only used for regular (latitude-longitude) grid.

-I *dx/dy*: grid increment for regular (latitude-longitude) grid in degrees (default: 0.25/0.25).

-Gg *grid_definition_file_x/grid_definition_file_y*: An irregular grid is used. The files contain matrices with latitude/ longitude definitions of the grid. The –I and –R options are ignored.

-Gu *specific_points_definition_file*: Specific points are defined by a list given in the file. Two columns are expected containing the longitude in the first and the latitude in the second column. The –I and –R options are ignored.

-H *digital_terrain_model*: The error field is computed on the terrain as defined in *digital_terrain_model*. By default the reference ellipsoid is taken. A default terrain model is part of the GUT distribution and is used if no filename is given. –H is ignored if –Fg is selected.

-Ma: The area average for the grid cell is computed. By default the geodetic field is computed on the specified grid points. –Ma is ignored when using –Gu.

-Clon1/lat1/lon2/lat2: The commission error co-variance is computed for the two points defined by lon1/lat1 and lon2/lat2 (longitude/ latitude in degrees). By default only the variances are computed (see options –R, -I, -Gg, -Gu).

Input:

- Data set with error-covariance for spherical harmonic coefficients used in geodetic field computation (*GUT default data:* EGM_GVC_2)

- maximum degree and order of the expansion used in the geodetic field computation (from .GUT_FA01).
- Reference system selection (from .GUT_FA02)
- Preferred tide system: (from .GUT_FA03)

Output:

For grids:

- Depending on geodetic parameter:

-Fg: 2D matrix (grid) of geoid error variance (m^2)

-Fa: 2D matrix (grid) of gravity anomaly error variance ($mgal^2$)

-Fd: 2D matrix (grid) of the error variance in the E-W deflections from the vertical and 2D matrix (grid) of the error variance in the N-S deflections from the vertical ($arcsec^2$).

- Locations of grid nodes as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.
- Flag reflecting whether point-wise or area averaged values (-Ma option) have been calculated

For User Specified Points (-Gu):

- Depending on output parameter:

-Fg: Vector of geoid error variance at each data point (m^2)

-Fa: Vector of gravity anomaly error variance at each data point ($mgal^2$)

-Fd: Vector of the error variance in the E-W deflections from the vertical and vector of the error variance in the N-S deflections from the vertical at each data point ($arcsec^2$).

- Locations of data points as 2 vector arrays of latitudes and longitudes (degrees).

If co-variances are computed (-C):

- 2x2-matrix containing variances and co-variance of the two specified points
- Locations of the two data points.

For All Cases:

- Maximum degree and order and approximate associated spatial scale (in spatial dimensions, *i.e.* km or degrees lat) of the calculation.
- Reference ellipsoid definition of the calculated field
- For gravity anomaly or deflections from the vertical – the actual height above the reference ellipsoid for the calculation.

URD Algorithm functionality defined by this algorithm

GUT_016: *Geoid height errors in grid format*

GUT_018: *Gravity anomaly errors in grid format*

GUT_020: *E-W deflection errors in grid format*

GUT_022: *N-S deflection errors in grid format*

GUT_017: *Geoid height errors in points*

GUT_019: *Gravity anomaly errors in points*

GUT_021: *E-W deflection errors in points*

GUT_023: *N-S deflection errors in points*

GUT_025: *Geoid height error covariances in points*

GUT_027: *Gravity anomaly error covariances in points*

GUT_029: *E-W deflection error covariances in points*

GUT_031: *N-S deflection error covariances in points*

The following URD algorithms *will not* be implemented, as they require computing resources in excess of those anticipated necessary to run the toolbox.

GUT_024: *Geoid height error covariances on a grid*

GUT_026: *Gravity anomaly error covariances on a grid*

GUT_028: *E-W deflection error covariances on a grid*

GUT_030: *N-S deflection error covariances on a grid*

Existing software providing this algorithm

Covar.m:

Matlab function to determine errors on a grid based on the covariance matrix

Omission error determination (geoid)

is estimated using a model of the expected power spectrum for all spherical harmonics higher than the maximum degree and order included in the spherical harmonic synthesis.

The calculation of the omission errors will be limited to calculation of:

- the variances for grids of points
- full co-variances for discrete pairs of points
- an isotropic, homogenous covariance function.

The calculation of full error co-variances for a grid is considered beyond the scope of the GOCE user toolbox.
As the full error covariance information available for a high degree and order spherical harmonic expansion of a geoid field is of the order of several Gigabytes in size, it is not feasible to separate the read method from the input data file. The algorithms will need to extract parts of the error covariance matrix data file.

Options:

-Fg: geoid height omission error computation (default)

-Fa: gravity anomaly omission error computation

-Fd: geoid deflections from the vertical omission error computation

-R *e/w/n/s*: Region of interest in the order western, eastern, northern and southern limits in degrees north of the equator and east of the prime meridian (default: 79.75/-79.75/0.25/359.75). Only used for regular (latitude-longitude) grid.

-I *dx/dy*: grid increment for regular (latitude-longitude) grid in degrees (default: 0.25/0.25).

-Gg *grid_definition_file_x/grid_definition_file_y*: An irregular grid is used. The files contain matrices with latitude/ longitude definitions of the grid. The –I and –R options are ignored.

-Gu *specific_points_definition_file*: Specific points are defined by a list given in the file. Two columns are expected containing the longitude in the first and the latitude in the second column. The –I and –R options are ignored.

-H *digital_terrain_model*: The error field is computed on the terrain as defined in *digital_terrain_model*. By default the reference ellipsoid is taken. A default terrain model is part of the GUT distribution and is used if no filename is given. –H is ignored if –Fg is selected.

-Ma: The area average for the grid cell is computed. By default the geodetic field is computed on the specified grid points. –Ma is ignored when using –Gu.

-C *lon1/lat1/lon2/lat2*: The omission error co-variance is computed for the two points defined by *lon1/lat1* and *lon2/lat2* (longitude/ latitude in degrees). By default only the variances are computed (see options –R, -I, -Gg, -Gu).

-F *d1/d2/dx*: The omission error is provided as an isotropic, homogenous function of distance. The function is evaluated in the interval [*d1,d2*] using *dx* as increment. By default only the variances are computed (see options –R, -I, -Gg, -Gu).

Input:

- Data set with spherical harmonic coefficients (*GUT default data:* EGM_GCF_2)
- maximum degree and order of the expansion used in the geodetic field computation (from .GUT_FA01).
- Reference system selection (from .GUT_FA02)
- Preferred tide system: (from .GUT_FA03)

Output:

For grids:

- Depending on geodetic parameter:
  -Fg: 2D matrix (grid) of geoid error variance (m^2)
  -Fa: 2D matrix (grid) of gravity anomaly error variance ($mgal^2$)
  -Fd: 2D matrix (grid) of the error variance in the E-W deflections from the vertical and 2D matrix (grid) of the error variance in the N-S deflections from the vertical ($arcsec^2$).
- Locations of grid nodes as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.
- Flag reflecting whether point-wise or area averaged values (-Ma option) have been calculated

For User Specified Points (-Gu):

- Depending on output parameter:

-Fg: Vector of geoid error variance at each data point (m^2)

-Fa: Vector of gravity anomaly error variance at each data point ($mgal^2$)

-Fd: Vector of the error variance in the E-W deflections from the vertical and vector of the error variance in the N-S deflections from the vertical at each data point ($arcsec^2$).

- Locations of data points as 2 vector arrays of latitudes and longitudes (degrees).

If co-variances are computed (-C):

- 2x2-matrix containing variances and co-variance of the two specified points
- Locations of the two data points.

If an isotropic homogenous co-variance function is provided (-F):

- Vector of error co-variances
- Vector of distances corresponding to the error co-variances.

For All Cases:

- Maximum degree and order and approximate associated spatial scale (in spatial dimensions, *i.e.* km or degrees lat) of the calculation.
- Reference ellipsoid definition of the calculated field
- For gravity anomaly or deflections from the vertical – the actual height above the reference ellipsoid for the calculation.

URD Algorithm functionality defined by this algorithm

GUT_032: *Geoid height omission error covariances on a grid*

GUT_033: *Geoid height omission error covariances in points*

GUT_034: *Gravity anomaly omission error covariances on a grid*

GUT_035: *Gravity anomaly omission error covariances in points*

GUT_036: *E-W deflection omission error covariances on a grid*

GUT_037: *E-W deflection omission error covariances in points*

GUT_038: *N-S deflection omission error covariances on a grid*

GUT_039: *N-S deflection omission error covariances in points*

Existing software providing this algorithm

Some elements of the required software exist in the GRAVSOFT GEOCOL program.

Average determination of SSH or DT

This functionality is provided in a simple form to allow the calculation of a mean sea surface height (SSH) or dynamic topography (DT) from existing gridded or collocated locations using a simple pointwise averaging technique. The Basic Radar Altimeter Toolbox (BRAT) provides more advances algorithms for carrying more complex averaging and gridding and could be used to pre-condition data for input to GUT. However, it is recommended that this simple functionality is retained in GUT for tutorial purposes and simple re-calculation of averaged properties.

No interpolation will be carried out to account for missing data values, and these should be accounted for in the input fields.

Options:

none

Input:

Time series of 2D matrices or vectors of parameters to be calculated

Output:

Returned mean value at each input location: matrix identical in size to each input field.

Algorithm

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

where x is the property to average, overbar denotes the mean, i is the location of each measurement of property x and n is the total number of unique locations.

The algorithm will only average over the number of valid data measurements – missing data will not be included.

URD Algorithm functionality defined by this algorithm

None

Existing software providing this algorithm

None

Grid adaptation

Algorithm is required for all cases where gridded fields are supplied and are required to be differenced.

Note: this function is not required where spectral differencing is to be carried out

Options:

None

Input:

- Required new regular grid specification for output fields (default is GUT MSS grid)
- Regular grid specification of existing input field
- Input data field for

Output:

Input data field interpolated to grid locations of the required new grid

Latitude and Longitude values of new grid nodes

URD Algorithm functionality defined by this algorithm

GUT_110: *Interpolation*

Existing software providing this algorithm

geoip from the GRAVSOFT package

Linear filter (spatial)

This function will be used for filtering of the dynamic topography, or of mean sea surface fields, ready for combination with geoid models.

The type of filter(s) to be implemented has been determined from the trade-off study examining the filtering.

Options:

-Fj scale: Quasi-Gaussian (Jekeli) filter with full filter width scale (*default option*)

-Fg scale: Gaussian filter with full filter width scale

-Fc scale: Spherical cap with diameter scale

-Fhm scale: Hamming window of width scale

-Fhn scale: Hanning window of width scale

-F filter_matrix_file: The filter matrix is read in from a file.

-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a limitation for large grids depending on available resources. The limit grid size is defined in .GUTdefaults.

Input:

- Grid specification (default is the default GUT MSS grid specification)
- Input field

Output:

- Filtered version of input field

URD Algorithm functionality defined by this algorithm

GUT_102: *Grid filtering*

Existing software providing this algorithm

None

Linear filter (Spherical Harmonic)

This function will be used for filtering of a geodetic 2D parameter, especially the dynamic topography, in spherical harmonic space. This is the proposed default option for filtering of MSSH- geoid. The function includes expansion of a gridded field into spherical harmonic coefficients, application of the filter kernel and transformation back to physical space. The analysis and synthesis steps to change between physical space and spectral representation of the geodetic parameter can both be switched off, if spherical coefficients are provided as input or required as output.

Options:

-Fj scale: Quasi-Gaussian (Jekeli) filter with full filter width scale (*default*)

-Fc scale: Pellinen filter with full filter width scale

-F filter_matrix_file: The filter matrix is read in from a file.

-O filter_matrix_file: The filter matrix is stored in filter_matrix_file. There is a limitation for large grids depending on available resources. The limit grid size is defined in .GUTdefaults.

-A: The parameter is given on a regular grid and a transformation to spherical harmonic coefficients is needed (default). –A- indicates that the parameter is given in spectral space and no analysis is necessary.

-S: Transformation into physical space is invoked after filtering (default). –S- indicates that the output is required in spectral space.

-D: The parameter filtered is a dynamic topography. No reference surface (as specified in .GUTdefaults) is subtracted when transforming to physical space (-S option). Default: Subtraction of reference.

Input:

If parameter provided on regular grid (-A option):

- Grid specification (default is the default GUT MSS grid specification)
- Input field

If parameter is provided in spectral space(-A- option):

- Data set with spherical harmonic coefficients

Output:

- Filtered field on grid (-A option) or as spherical harmonic coefficients (-A- option).

URD Algorithm functionality defined by this algorithm

None

Existing software providing this algorithm

Spectral filtering is available through the gauss_av function, part of the sh_mdt fortran program by Rory Bingham.

In addition, Matlab routines that were used for the filtering study (section 3.2) exist and are available from Martin Losch. These routines would need re-coding for use in the toolbox.

Fill gaps on continent

To perform expansion into spherical harmonic coefficients a global field is needed. This function fills gaps in a 2D parameter by replacing 'NaN' values by values from a second, globally defined, field, which is defined on the same grid.
This function should correctly address the merging of the fields at the coastlines by applying suitable smoothing. However, this smoothing will have an impact on the results of filters applied. The recommended remove restore methods are relatively robust regarding the smoothing and in an initial toolbox we have not recommended any specific smoothing method. It should be possible for users to include their own smoothing algorithms here.

Input:

- geodetic field
- global field to fill gaps in geodetic field (*default GUT gridded geoid)*

Output:

- patched geodetic field

Existing software providing this algorithm

None

SH analysis

Required for all instances where a global gridded field is to be used for spherical harmonic filtering.

Input:

- Global gridded field
- Regular grid specification of input field
- Maximum degree and order of spherical harmonic expansion to calculate (read from .GUTdefaults)

Output:

- Spherical Harmonic expansion of coefficients of global field
- Maximum degree and order included in expansion

URD Algorithm functionality defined by this algorithm

GUT_103: *Going from geographical space to spectral*

Existing software providing this algorithm

The shpgric fortran program provides this function for gridded data.

gsha.m: Matlab script, part of the GSHBUNDLE software package of Nico Sneeuw.

Surface current determination

Possible for regular gridded data fields or line / transect data.

Options:

- -T: Dynamic Topography is provided on a transect (default: topography on 2D grid)
- -G: Currents are required on the same grid as the input topography. This involves interpolation on larger scales than the default four point calculation (see algorithm description below)

Input:

For 2D Dynamic Topography:

- 2D matrix of gridded Dynamic Topography (m)
- Locations of grid nodes as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.

For transects:

- Vector containing (mean) dynamic topography (m) at points values
- 2 vectors containing the latitude and longitude coordinates of the points

Output:

For 2D Dynamic Topography:

- 2 2D matrices of East and North velocities respectively
- Locations of grid nodes at which the velocity was calculated, given as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.

For transects:

- 2 vectors containing East and North components respectively of velocity across the transect
- 2 vectors containing the latitude and longitude coordinates of the points at which the velocity was calculated.

By default, for gridded data, the velocity values will be calculated for a grid offset by $\delta\varphi/2$ in latitude and $\delta\lambda/2$ in longitude for the dynamic topography grid (*i.e.* for a grid offset by ½ grid cell dimension from the input grid. For global grid, the longitude will be wrapped.

By default, for transect data, the velocities will be calculated at the midpoints between each location

Algorithm:

GUT_013: Geostrophic velocities in grid

GUT_014: Geostrophic velocities in points.

Note for a grid, we use a four point calculation by default, which as per [15] is for a regular grid:

$$u = \frac{-\gamma}{fR}\frac{\partial \varsigma_{i+.5,j+.5}}{\partial \theta}$$

$$\frac{\partial \varsigma_{i+.5,j+.5}}{\partial \theta} = \frac{1}{4\delta\theta}\left(\varsigma_{i,j+1} - \varsigma_{i,j} + \varsigma_{i+1,j+1} - \varsigma_{i+1,j}\right)$$

$$v = \frac{-\gamma}{fR\cos\theta}\frac{\partial \varsigma_{i+.5,j+.5}}{\partial \lambda}$$

$$\frac{\partial \varsigma_{i+.5,j+.5}}{\partial \lambda} = \frac{1}{4\delta\lambda}\left(\varsigma_{i+1,j+1} - \varsigma_{i,j+1} + \varsigma_{i+1,j} - \varsigma_{i,j}\right)$$

where *h* is the dynamic topography defined at points *i,j i+1,j i,j+1 and i+1,j+1* are input grid nodes surrounding the output grid node *i+.5,j+0.5* respectively, where *I* and *j* are the longitude and latitude locations of the input grid nodes separated by δλ and δθ.

If velocity is required on the same grid, 5 point calculations are used:

Existing software providing this algorithm

Combination of GUT_slopes.m, GUT_srfcurrnt.m for 5 point calculation.

Difference or Sum of Gridded Fields

Possible for regular gridded data fields or line / transect data.
Required for generation of a “synthetic geoid” field (difference of (M)DT and (M)SSH) necessary for the remove restore techniques, and also for replacing the residuals when using the remove restore technique.

Options:

-sum: Provide sum of input fields

-diff: Provide difference of 2 fields

Input:

For 2D Fields:

- pair of 2D matrices containing values to sum / difference
- Locations of grid nodes as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.

For transects:

- Pair of vector containing values to sum / difference
- 2 vectors containing the latitude and longitude coordinates of the points

Output:

For 2D Fields:

- A 2D matrix of sum (–sum option) or difference (-diff option) of input fields
- Locations of grid nodes as 2 vector arrays of latitudes and longitudes (latitude-longitude grid) or as two matrices (irregular grid) in degrees.

For transects:

- A vector containing sum (–sum option) or difference (-diff option) of input vectors
- 2 vectors containing the latitude and longitude coordinates of the points

Algorithm:

GUT_100: Grid addition

GUT_101: Grid subtraction.

Existing software providing this algorithm

The fc2 program of the GRAVSOFT package.

8.2 Input Output Definition

Within this section, we will identify the input and output files that are required, or are optionally user supplied, for which the toolbox with require input and output methods. Required input data files should be included within the toolbox package and provide the default for appropriate workflows.
Optional, user supplied, input data files must be in a format that the toolbox is able to input and can be substituted for the supplied input files by the user.
As a minimum requirement, the toolbox documentation must provide the format specifications of the included input data files in order that the user can supply their own data in the same format.
Where the toolbox provides functionality to input a range of input formats, the toolbox documentation must also provide the format specifications (or reference to the appropriate format specification documentation) to enable users to ensure their data are in a compatible format.
[In addition, the toolbox should supply the ability for users to write their own input modules for expansion of the toolbox to cover the widest possible range on input data formats]
All height fields (geoid, mean sea surface, dynamic height, terrain model etc..) must be given relative to a consistent reference ellipsoid. Candidate ellipsoids are:

GRS80 (semi-major axis a = 6378137.0 m, flattening f = 1 / 298.257222101.)

"topex" ellipsoid (semi-major axis a = 6378136.3 m, flattening f = 1 / 298.257)

"Envisat" ellipsoid (semi-major axis a = 6378137, flattening f = 1 / 298.257223563)

Height field requiring reference system conversion.

8.2.1 Required Input Data files: to be included on the toolbox:

Geoid harmonic expansion (SH & error covariance products)

The official HPF GOCE level 2 products:

EGM_GOC_2

EGM_GVC_2

These files are in XML IGCEM format [2].
The read method must input to the toolbox:

Spherical harmonic coefficients of the product

Maximum degree and order of the supplied spherical harmonic expansion

Tidal system (for use by GUT_FA002)

Reference ellipsoid definition: GM, r, a,

Mean Sea Surface Height Field

It is proposed that the included MSS height field is the most recent version of CLS, currently CLS01 [16], or the most recent KMS solution (currently KMS04: http://geodesy.spacecenter.dk/~gocina/publications/4_1_kmss04-lux.pdf). Results of the GOCINA project indicate that the updated, global, KMS solution may be the most appropriate, subject to improved fields being generated by the time of release of the toolkit.
CHARACTERISTICS:

Name CLS01

Reference ellipsoid T/P

Referencing time period 1993-1999 (7 years)

Domain Global (80°S to 82°N) – Oceanwide where altimetric data are available. EGM96 elsewhere and on continents.

Spatial resolution Regular grid with a 1/30° (2 minutes) spacing (*i.e.* ~4 km)

Grid 10800 points in longitudes / 4861 points in latitude

MSS determination technique Local least square collocation method on a 6' grid where altimetric data in a 200-km radius are selected. Estimation on a 2' grid based on SSH-EGM96 values (remove/restore technique to recover the full signal). The inverse method uses local isotropic covariance functions that witness the MSS wavelength content.

Estimation error level YES (in m) – Negative values are flagging coastal areas where the smoothed junction with the continental EGM96 geoid is computed.

Altimetric dataset

T/P 7 years mean profile

ERS-1/2 5 years mean profile

GEOSAT 2 years mean profile

ERS-1 geodetic data

The read method must input to the toolbox:

2-dimensional matrix of mean sea surface height field (m)

2 vectors containing the latitude and longitude of the grid nodes of the matrix.

Reference ellipsoid GM, r etc (for use by GUT_FA_002)

Mean Dynamic Topography and associated error covariance products

In order to address the complex issues of creating dynamic topography fields that are consistent with the length scales included in the GOCE data, and also try and include the higher resolution capability of altimeter derives mean sea surfaces, it is proposed that he toolbox contains 2 mean dynamic topography data sets.
Primary Mean Dynamic Topography: Contains only scales resolved within GOCE data must be compatible with MSSH field in terms of reference averaging period: an example of the output from the simple, default workflow of the toolbox
Secondary Mean Dynamic Topography: high resolution MDT for use as a-priori in remove restore or similar combined MDT calculations. Candidates include OCCAM model at 1/12 degree
The read method must input to the toolbox:

2-dimensional matrix of mean dynamic topography field (m)

2 vectors containing the latitude and longitude of the grid nodes of the matrix.

Reference ellipsoid GM, r etc (for use by GUT_FA_002)

Both products should contain appropriate error covariance data for MDT products:

Sea level Anomaly fields

Sea level Anomaly fields: Include e.g. 12 monthly average fields, relative to MSSH, or 12 annual fields that can be averaged to predefined mssh fields, which can be filtered and added to MDT. Default format files will be the CLS NetCDF format data files.
Alternate MSSH & SLA referenced to different mean periods.

Along Track Altimetric Sea Surface Height fields

Along track SSH and or SLA: **either** use AVISO products as the default **or** using BRAT ingestion interface.

Digital Terrain Model

Digital terrain model: For use in tutorials, the ETOPO2v2 digital elevation model is recommended (http://www.ngdc.noaa.gov/mgg/fliers/06mgg01.html)

8.2.2 Optional User Supplied Input Data Files

It is anticipated that even a basic GOCE User Toolbox should be capable of accepting input fields from user supplied data files as alternatives to the included data files defined in section 8.2.1 above.

As a minimum requirement, the toolkit will include the complete format specification of the included data files to enable the user to transform their preferred data source to the appropriate data format. In this case, the toolbox input method must allow:

The selection of an alternative data file for all data inputs

The toolbox should be capable of reading any gridded data field from a GRAVSOFT format ASCII file, as defined in the URD

GEOID – ICGEM this is commonly used format and common geoids are all downloadable in ICGEM format

Format must be in a format recognized by the toolbox. Candidate formats include:

a) GOCE XML ICGEM [17]
b) GRACE SHM-FORMAT (Earth Gravity Spherical Harmonic Model Format), in use within the GRACE project for Earth gravity model solutions in terms of spherical harmonic coefficients [18].

MSS gridded– in AVISO MSS format

SLA / SSH – can we use BRAT ingestion routines?

8.2.3 Output Data Formats

The toolbox must provide output capability to save all computed fields in standard formats. The standard formats must also include appropriate metadata to describe the saved fields. As the toolbox users will be from a range of scientific disciplines, with a range of discipline specific standards, it is appropriate that at least two output formats are supported for the toolbox products. These formats should be, as far as possible, platform independent and widely recognised by potential post-processing and display software packages. The recommended formats are:

1) ASCII format, following the GRAVSOFT data standards, suitable for all output fields. Additional metadata may be required to describe the product. This will be provided as a separate text file associated with the product.
2) NetCDF format, suitable for all gridded fields. Additional metadata should be included in the file attributes.

8.2.4 File Naming Conventions

It is anticipated that the toolbox users will use the toolbox for data sources beyond those included in the toolbox. As many of these data sources already have appropriate final naming conventions, it is anticipated that the users may wish to use file names that follow these conventions. It is, therefore, inappropriate to enforce any specific file naming conventions on the user supplied input or output data files used by the toolbox.
Default filenames should be suggested for saving files:

GEO

MSSH

MDTS

MDTC

Use GUT products name somehow?
GRAVSOFT format files should use the .dat extension, whilst NetCDF should use the CDF extension.
All file naming should be able to be over-ridden by the user.

9 Appendix C. The GRAVSOFT formats

From Document: GOCE_s1.doc Date: 2001.05.08, Author: C.C.Tscherning,

URL: http://www.gfy.ku.dk/~cct/GOCE_s1.htm.

With slight modifications

GRAVSOFT Standard.

The basis for some of the specifications proposed below are the so-called GRAVSOFT standards, (Tscherning et al., 1994). In brief they are:

All data are in ASCII code with blanks as delimiters.

Spherical harmonic coefficients:

Degree n, order m, C_{nm}, S_{nm}, σ (C_{nm}, S_{nm}). Coefficients are unitless. All coefficients are included (also (0,0)).

Geo-located data:

Geographical coordinates:

<Unique id, integer> <latitude, decimal degrees (number of significant digits to be specified)>, <longitude, decimal degrees from 0 to 360 degrees, positive East > < altitude in m, significant digits to be specified , ellipsoidal height or orthometric height must be specified> <data 1> <data 2> ….. <data n >

UTM coordinates: like geographical coordinates, but Northing instead of latitude and Easting instead of longitude.

Cartesian Coordinates:

Like geographical coordinates, but <unique id. >, <X>, <Y>, <Z>, and then data.

Equal-angular grids of data: Data start with a grid label which describes the extend and spacing of the data grid, see the precise description below. Then the data are organized in "bands" of equal latitude (or UTM Northing) from North to South where the bands start in west and move east. The coordinates refer to the center of the cell.

Grid label: Geographical coordinates: minimum, maximum latitude, θ_1 and θ_2, minimum, maximum longitude, λ_1 and λ_2, spacing in latitude, $\Delta\theta$, spacing in longitude, $\Delta\lambda$. UTM: Northing instead of latitude, Easting instead of longitude, last value is the UTM zone.

This results in a grid that has

$\frac{\lambda_2 - \lambda_1}{\Delta\lambda} + 1.5$ entries per line and

$\frac{\theta_2 - \theta_1}{\Delta\theta} + 1.5$ lines.

Example: Grid with 4 elements.

54.0 56.0 10.0 12.0 2.0 2.0

4.1 5.0

4.5 5.6

Data-specifications which agree with GRAVSOFT:

Auxiliary data needed are:

A-priori set of spherical harmonic coefficients and associated error-estimates.

Global digital mean terrain/depth data in a 5' x 5' grid, globally. Units m.

For all the following data, the horizontal coordinates must be given as geographical coordinates in WGS84/ITRF ??. All associated heights are heights above mean-sea level. (Datum to be specified if not mean value).

5' mean Free-air gravity anomalies with <data 1> = <gravity anomaly, mgal with 1 decimal>, <data 2> = < standard error>.

Height anomalies from GPS and levelling <data 1> = < height anomaly in units of m with 3 decimals> <data 2> = < error estimates > <data 3> = < datum name (ISO)>

Deflections of the vertical: Meridian and Prime-vertical components.

<data 1> = <Meridian component, arcseconds with 2 decimals >, <data 2> = < Prime-vertical component >, <data 3 > , <data 4> = <error-estimates of the components > <datum id.>

Deflection of the vertical, one component only. As above with only two data-items.

Satellite data:

Geo-located observational data on GRAVSOFT format.

SGG data: Geographical coordinates. Altitude is ellipsoidal height in mm. <data 1> … <data n>, <standard deviations of data 1 … data n> <3 attitude angles: tilt, roll, pitch in decimal degrees with 4 decimals>

SST data: Cartesian coordinates

Corrections to geo-located data on same format.

Products:

Spherical harmonic coefficients and their variance-covariance matrix.

Grids of free-air anomalies, deflections of the vertical and height anomalies referring to the mean terrain height.

Corresponding grids of error-estimates. The grid-labels will tell whether the grids are global or local.

Geo-located and corrected satellite data on the same format as above.

Non-GRAVSOFT data-types.

The gravity field statistics.

The physical correlation of gravity field quantities may be expressed through a global or regional covariance functions. Such functions may be specified using the following parameters:

<error-degree-variance modifiers> < depth to Bjerhammar-sphere (m) > < gravity anomaly variance ($mgal^2$) >. The degree-variance modifiers are the following

< weight factor on error degree-variances > < number of error-degree-variances used > < indicator of whether the scale factor is to be applied on all degree-variances or so that it changes linearly from degree zero to the maximal degree >

A file should be established with the current values (they depend on the a-priori spherical harmonic model used). It should in one record hold the boundaries of the areas of validity and the parameters.

Variance-covariance matrices of a-priori given or estimated quantities.

These matrices must be on binary form, REAL*8. Since the matrices are symmetric, the matrix elements should be stored from column 1 to "N" , from element no. 1 to the diagonal element.

Example: The symmetric 2 x 2 matrix

1 3 becomes 1, 3, 4.

3 4

Reference:

Tscherning, C.C., P.Knudsen and R.Forsberg: Description of the GRAVSOFT package. Geophysical Institute, University of Copenhagen, Technical Report, 1991, 2. Ed. 1992, 3. Ed. 1993, 4. ed, 1994.

Appendix D. Existing Software Packages

Use of existing software packages to provide the functionality required for the toolbox.

9.1 HARMONIC_SYNTH (Version 05/01/2006)

This package is a fortran 77 based, open source programme to calculate the range of parameters required by GUT_FA003. However, as stands, the program calculates the spherical approximation of the parameters (as per algorithms GUT_001 – GUT_004). The code would require modification to include the full method, recommended to be used in the GOCE Toolbox.
The present version of HARMONIC_SYNTH can perform harmonic synthesis up to degree and order 2700.
HARMONIC_SYNTH is intended primarily for solid spherical harmonic synthesis using high-degree Earth Gravitational Models (EGMs). Quantities can be computed at scattered locations whose coordinates are read from an input file, or on an equiangular grid on the surface of either a mean-earth reference ellipsoid or a geocentric sphere. Ellipsoidal harmonic synthesis can be executed on the surface of the ellipsoid to which the ellipsoidal spectrum is referenced.

Capabilities

Table 1 summarizes the computational capabilities of the present version of HARMONIC_SYNTH. The following abbreviations are used in Table 1:

- isw: the selection code for the computed quantity
- quantity: the quantity that is computed
- g_mn: option for area-means over geographic rectangles on a regular latitude/longitude grid
- g_pt: option for point values on a regular latitude/longitude grid
- scat: option for output at arbitrarily scattered points
- EGM: requires spherical harmonic gravitational model
- HGHT: requires spherical harmonic elevation model
- CORR: requires spherical harmonic correction model
- EHM: requires ellipsoidal harmonic model units: the units of the computed quantity

isw	quantity	g_mn	g_pt	scat	EGM	HGHT	CORR	EHM	units
00	ζ	x	x	x	x				m
01	Dg01	x	x	x	x				mGal
02	ht02	x	x	x		x			m
03	Dgr	x	x	x	x				mGal/km
04	ζ_r	x	x	x	x				m/km
05	bias	x	x	x			x		mGal
06	Dgrr	x	x	x	x				$mGal/km^2$
07	ξ		x	x	x				arcsec
08	η		x	x	x				arcsec
09	Tr	x	x	x	x				mGal
10	Ty		x	x	x				mGal

11	Tx		x	x	x				mGal
12	Trr	x	x	x	x				E.U.
13	Tyy		x	x	x				E.U.
14	Txx		x	x	x				E.U.
50	Dg50		x	x	x				mGal
80	N80			x	x				m
81	N81			x	x	x			m
82	N82			x	x		x		m
100	Dg100	x	x	x				x	mGal
101	ht101	x	x	x				x	m

Table 1. Summary of the functionality of HARMONIC_SYNTH.

Exact names and formulae for these quantities are as follows. Let (r, φ, λ, h) denote (geocentric distance, geocentric latitude, longitude, geodetic height), respectively. T is the disturbing potential at the computation point and © is the magnitude of normal gravity at that point. Note that, for the radial derivatives $\partial(f)/\partial r$ and $\partial^2(f)/\partial r^2$, the radial partial differential ∂r is in the direction of increasing radial distance (away from the geocentre).

isw=00: "height anomaly" in metres ζ=T/©

isw=01: "spherically approximated gravity anomaly" in mGal $Dg01 = -\partial \mathrm{T}/\partial r - 2^{\mathrm{T}/r}$

isw=02: "elevation" in metres

isw=03: "radial gravity anomaly gradient" in mGal per km $Dg\partial(Dg01)/\partial r$

isw=04: "radial height anomaly gradient" in metres per km $\zeta_r = \partial\zeta/\partial r = (1/\gamma)\times\partial \mathrm{T}/\partial r - \mathrm{T}\,(\partial\gamma/\partial r)/\gamma^2$

isw=05: "gravity anomaly systematic bias" in mGal

isw=06: "gravity anomaly second radial derivative" in mGal per km squared $Dgrr = \partial^2(Dg01)/\partial r^2$

isw=07: "spherically approximated north-south vertical deflection" in arcseconds $\xi = -(\partial \mathrm{T}/\partial\varphi)/\gamma r$

isw=08: "spherically approximated east-west vertical deflection" in arcseconds $\eta = -(\partial \mathrm{T}/\partial\lambda)/(\gamma r\cos\varphi)$

isw=09: "radial component of gravity disturbance" in mGal $Tr = \partial(T)/\partial r$ (Note that the sign is opposite to convention)

isw=10: "north-south component of gravity disturbance" in mGal $Ty = -(\partial(T)/\partial\prod)/r$

isw=11: "east-west component of gravity disturbance" in mGal $Tx = -(\partial(T)/\partial\lambda)/(r*cos(\prod))$

isw=12: "second radial derivative of T" in Eotvos Units $Trr = \partial^2(T)/\partial r^2$

isw=13: "second north-south derivative of T" in Eotvos Units $Tyy = (1/r)*\partial(T)/\partial r + (1/r^2)*\partial^2(T)/\partial\prod^2$

isw=14: "second east-west derivative of T" in Eotvos Units $\mathrm{Txx} = (1/r)*\partial(\mathrm{T})/\partial r + (\tan(\prod)/r^2)*\partial(\mathrm{T})/\partial\prod + (1/(r*\cos(\prod))^2)*\partial^2(\mathrm{T})/\partial\lambda^2$

isw=50: "linearly approximated gravity anomaly" in mGal $Dg50 = \partial(T)/\partial h + ((\partial(©)/\partial h))/©)*T$

isw=80,81,82: "geoid height" in metres (see the section on GEOID HEIGHTS below)

isw=100: "spherically approximated gravity anomaly" in mGal (see the section on ELLIPSOIDAL HARMONICS below)

isw=101: "elevation" in metres (see the section on ELLIPSOIDAL HARMONICS below)

Operation

The user can run HARMONIC_SYNTH by setting a collection of input parameters, in the parameter statements at the beginning of the main program, to their desired values. These input parameters are grouped under the following classifications:

- General input parameters
- Geodetic reference system
- Gridded computations
- Scattered point computations
- Specialized computations

The following sections describe how to operate HARMONIC_SYNTH by setting the values of these parameters.

General Input Parameters

HARMONIC_SYNTH uses the following form for the spherical harmonic expansion of the gravitational potential:ηφρ

V(r,θ,λ) = (GM/r)*SUM_n[(a/r)^n SUM_m [(C_{nm}*cos(m*λ)+S_{nm}*sin(m*λ))*Pnm(cos(θ))]]

for which:

V: gravitational potential
r: geocentric radial distance
θ: northern spherical polar distance
λ: longitude
GM: product of Newton's gravitational constant and the mass of the Earth.
a: scale factor associated with the coefficients
n: spherical harmonic degree
m: spherical harmonic order
C_{nm},S_{nm}: fully-normalized spherical harmonic coefficients of the gravitational potential
Pnm: fully-normalized Associated Legendre Function of the first kind
SUM_n: summation over degree from n=Nmin to Nmax
SUM_m: summation over order from m=0 to n

The following parameters describe the use of HARMONIC_SYNTH for computing gravimetric quantities from a spherical harmonic gravitational model.

iumi: is the integer input FORTRAN unit number for any spherical harmonic gravitational model. This parameter can be set to 1 as a default.

lmin: is the minimum harmonic degree required for synthesis of the gravitational spectrum.

lmax: is the maximum harmonic degree required for synthesis of the gravitational spectrum.

path_mod: is the path to the directory containing any harmonic model to be used by this program.

name_mod: is the name of the ASCII file containing the spectrum of the gravitational model (see FORMATS AND UNITS FOR INPUT FILES).

aegm: is a scale factor used in developing the gravitational model and which scales its coefficients. It usually represents the length in metres of the semi-major axis of some reference ellipsoid.

gmegm: is also a scale factor used in developing the gravitational model and which scales its coefficients. It represents the EGM value for the product of Newton's gravitational constant and the mass of the Earth. It is input in units of $(m^3/s^2)*10^5$.

isub: is an integer parameter flag for removing a reference field from the input spectrum. Set isub=1 to remove a zonal reference field from the input gravitational spectrum to produce a spectrum of the disturbing potential. Set isub=0 if there is no need to modify the even zonal harmonics of the input spectrum. See THE GEODETIC REFERENCE SYSTEM for a description of the reference field applied to the input spectrum.

igrid: is an integer parameter flag that determines whether to generate values at scattered locations read from an input file, or values on a regular equiangular grid. Set igrid=0 for scattered values, and igrid=1 for an equiangular grid.

isw: is an integer parameter flag for selecting the quantity to be computed.

exclude: is the value that flags and replaces any undefined or non-computed quantities in the output. For example, HARMONIC_SYNTH will return the exclude value for vertical deflections at the exact poles. Set this value to 9999.d0 as a default.

path_out: is the path to the directory in which the output file containing the generated values is to be placed.

name_out: is the name of the output file containing the generated values.

Geodetic Reference System

Four parameters are used to define the Geodetic Reference System (GRS) to be used for the synthesis task. Together they uniquely define the mean-earth reference ellipsoid of revolution for the GRS, as well as the associated zonal reference potential field. These four parameters can be either (ae, gm, omega, c20), or (ae, gm, omega, rf) where:

ae: is the length (metres) of the semi-major axis of the reference ellipsoid of revolution used for the GRS.

GM: is the GRS value for the product of Newton's gravitational constant and the mass of the earth. Its units are $(m^3/s^2)*10^5$.

omega: is the angular velocity in radians/second assumed for the GRS.

C_{20}: is the second degree zonal harmonic coefficient of the Somigliana-Pizzetti gravitational potential associated with the chosen reference ellipsoid of revolution.

rf: is the reciprocal flattening of the reference ellipsoid of revolution.

Notes

1) To select which of the two quantities (C_{20} or *rf*) will be used to define the GRS, the defining quantity must be set to its correct value and the NON-defining quantity must be set to zero. The subroutine GRS will return a derived value for the non- defining quantity. The user may observe that choosing to define the GRS with the parameter c20 may yield slightly different values for the derived GRS parameters than those obtained from choosing a value for rf. These differences are due to using double precision (8-byte) FORTRAN rather than extended double

precision (16-byte) FORTRAN in the GRS subroutine and can be ignored for all practical purposes.
2) HARMONIC_SYNTH rescales all input spherical harmonic coefficients of the gravitational potential, so that they refer to the GRS scale factors ae and gm, rather than to aegm and gmegm, respectively. This rescaling is applied regardless of the value of the flag "isub". If isub=1, the spherical harmonic coefficients of the disturbing potential are obtained as the difference between two consistently scaled sets of coefficients: the rescaled gravitational potential (input) coefficients minus the corresponding reference zonal GRS coefficients. HARMONIC_SYNTH then uses the GRS values of ae and gm to scale the gravimetric quantity computed from the harmonic summation of the coefficients of the disturbing potential.
3) HARMONIC_SYNTH computes height anomalies with respect to an IDEAL mean-Earth ellipsoid, whose semi-major axis "a" remains (formally) unspecified. This is done by setting lmin=2 (thereby excluding zero and first degree terms from the harmonic summation), regardless of the possible existence of non-zero such terms in the input gravitational potential file.
If the user requires height anomalies with respect to some specific semi-major axis "a", these can be obtained as discussed by Rapp in (Lemoine et al., 1998, chapter 11). The accuracy of such conversion depends inherently on the accuracy of knowledge of the "ideal" values of "GM" and "W0".

Permanent Tide Considerations

With regard to the Permanent Tide, in its present form the program HARMONIC_SYNTH assumes that:
a) The defining quantity (c20 or rf) of the GRS is given in the ZERO-TIDE system.
b) The input gravitational potential model's second degree zonal coefficient C(2,0) is given in the TIDE-FREE (also known as NON- TIDAL) system.
HARMONIC_SYNTH converts the model's C(2,0) from TIDE-FREE to ZERO- TIDE, within the main program, in the statement:
dc20 = 3.11080d-8*0.3d0/dsqrt(5.d0)
and, within the subroutine "dhcsin", in the statements:
if (nmin.le.2.and.nmax.ge.2) then $C_{nm}(3,1) = C_{nm}(3,1)$ - dc20 endif ! nmin,nmax
If the input gravitational model's C(2,0) is not given in the TIDE- FREE system, the above statements have to be modified accordingly. For any harmonic summation NOT using a spherical harmonic expansion of the gravitational (disturbing) potential, no dc20 correction is applied to the input spectrum. In these cases, the call statement to the "dhcsin" subroutine will assume a zero value for the dc20 correction factor. For details regarding the permanent tide systems, see Rapp et al. (1991), and the discussion by Rapp in (Lemoine et al., 1998, chapter 11).

Gridded Computations

The following parameters are used when computing quantities on a regular equiangular grid (igrid=1). None of these parameters will affect the computation at scattered locations (igrid=0).

deast: is the longitude (decimal degrees: 0.d0 to 360.d0) of the eastern boundary of the computation grid.

dwest: is the longitude (decimal degrees: 0.d0 to 360.d0) of the western boundary of the computation grid. For computations that stride the Greenwich (zero) meridian, use a negative longitude for the western boundary so that dwest is never larger than deast. For global computations, set dwest=0.d0 and deast=360.d0

dnorth: is the latitude (decimal degrees: 90.d0 to -90.d0) of the northern boundary of the computation grid.

dsouth: is the latitude (decimal degrees: 90.d0 to -90.d0) of the southern boundary of the computation grid.

dlon: is the equiangular spacing in longitude (decimal degrees) of the gridded quantities.

dlat: is the equiangular spacing in latitude (decimal degrees) of the gridded quantities.

iell: is an integer flag for selecting the surface upon which the gridded quantities will be situated. Set iell=1 to compute values on the GRS mean earth ellipsoid described above in THE GEODETIC REFERENCE SYSTEM. Set iell=0 to compute gridded values on the surface of a geocentric sphere.

rme: is used when computing values on the surface of a geocentric sphere (iell=0). It specifies the radius in metres of this sphere.

alt: is an auxiliary parameter used to vary the radius set in the rme parameter above. In this case, set rme to some constant mean earth value, and use the alt parameter to specify the radial altitude of the actual computation surface above/below this mean earth value. The gridded quantities will be computed on a geocentric sphere of radius rme+alt. Alternatively, simply set alt=0.d0 and vary the rme parameter to determine the radius of the computation sphere.

iglat: is an integer flag that determines the type of latitude by which the computed values will be equally spaced (dlat) along the meridian. Set iglat=0 for equal spacing in geodetic latitude, iglat=1 for equal spacing in geocentric latitude, and iglat=2 for equal spacing in reduced latitude. Note that, for computation on the surface of a sphere (iell=0), HARMONIC_SYNTH will ONLY accept a latitudinal spacing in terms of geocentric latitude (iglat=1).

iflag: is an integer flag for selecting an output of gridded point values or area means. Set iflag=0 to compute gridded point values. Set iflag=1 to compute area means.

A brief explanation of the geometry of the output grid is useful here. To construct the output grid, HARMONIC_SYNTH will first subdivide the total grid area, delineated by dwest, deast, dnorth and dsouth, into equiangular blocks. These blocks are dlat degrees tall and dlon degrees wide. If the grid boundaries have been chosen such that they do not contain an integer number of blocks, then it is the eastern and southern boundaries of the output grid that will not correspond exactly to the input parameter values (deast & dsouth) for these boundaries. In this case, HARMONIC_SYNTH will either overshoot or undershoot the eastern and southern boundaries, whichever gives the closest match between the dimensions of the input grid boundary parameters and the actual output grid. In all cases, it is the north-western most block that will lie directly against the northern (dnorth) and western (dwest) boundaries selected in the parameter statement.

For iflag=1, HARMONIC_SYNTH will generate a mean value of the desired quantity over each of these equiangular blocks. For iflag=0, HARMONIC_SYNTH will generate point values. Depending on the value of icell (see next) these point values can be situated either at the center of each of these blocks (the "x" position in the sketch below), or at the corners of these blocks (the "+" position in the sketch below).

icell: is an integer flag used when computing gridded point values (iflag=0). Set icell=1 to compute point values at the center of the equiangular blocks. This option yields an output grid of the same number of rows and columns as the iflag=1 option. The only difference between these two options is that for iflag=0 & icell=1, each output value represents the value of the quantity at the center-point of the cell, whereas for iflag=1 these values represent the mean value over the entire cell. Conversely, set icell=0, to compute point values at all corners of the equiangular blocks. This option will yield an extra row and an extra column compared to the icell=1 option.

Scattered Point Computations

The following parameters are used when computing quantities at scattered locations for which the coordinates are read from an input file (igrid=0). None of these parameters will affect the actual values computed for an equiangular grid (igrid=1).

iusc: is the integer FORTRAN input unit number for the ASCII input file holding the coordinates of the scattered points. An explanation of the format of this file is given in the 'FORMATS AND UNITS FOR INPUT FILES' section.

path_pnt: is the path to the directory containing the ASCII input file of coordinate information for the scattered points.

name_pnt: is the name of the ASCII input file containing the coordinate information for the scattered points.

maxpt: is an integer value that sets the maximum number of scattered points for which HARMONIC_SYNTH will generate output in a single execution. If maxpt is less than the number of records in the unit iusc, then HARMONIC_SYNTH will only generate gravimetric quantities for the first maxpt points listed in this file. If maxpt is greater than or equal to the number of records in the unit iusc, then HARMONIC_SYNTH will generate gravimetric quantities for all the points listed in this file. See the section on RAM MANAGEMENT for more on the use of maxpt.

Specialized Computations

This section deals with synthesis tasks that use surface spherical harmonics for input spectra, either instead of, or in addition to, the spherical harmonic expansions of the gravitational potential. Output quantities that require such input spectra are (see Table 1):

isw=2: elevation (m) isw=5: gravity anomaly systematic bias (mGal) isw=80,81,82: geoid height (m)

In addition to the spherical harmonic spectra of the gravitational potential, HARMONIC_SYNTH makes use of two additional classifications of spherical harmonic spectra:

* Surface elevation models
* Correction models

Additional parameters associated with these two model types are as follows.

Elevation model:

iuhi: is the integer FORTRAN input unit number for any spherical harmonic spectrum of elevation. This parameter can be set to 2 as a default.

kmin: is the minimum harmonic degree required for synthesis of the elevation spectrum.

kmax: is the maximum harmonic degree required for synthesis of the elevation spectrum.

name_hgt: is the name of the ASCII file containing the spectrum of the elevation. name_hgt must be in the directory path specified by path_mod.

Notes

1) As shown in Table 1, elevation models are used on their own to compute surface elevations (isw=2) and in conjunction with gravitational potential models to compute geoid heights (isw=81). The values set for the four elevation parameters above will only affect the computation in HARMONIC_SYNTH when isw=2 or isw=81. There is no need to set any valid values for the above four parameters when isw is neither 2 nor 81.

2) The isw=2 option constitutes an unscaled surface spherical harmonic synthesis of the spherical harmonic coefficients in the input unit iuhi. This option can be used for any surface harmonic synthesis task for which the units of the output are completely determined by the units of the input coefficients.

3) To compute gridded (igrid=1) elevations that are equally spaced in terms of geocentric latitude (iglat=1), the user can avoid specifying a GRS by setting iell=0. For iglat=1, setting iell=1 will yield the same result as setting iell=0, except that HARMONIC_SYNTH will require a GRS to be defined. However, to compute gridded elevations that are equally spaced in terms of either geodetic latitude (iglat=0) or reduced latitude (iglat=2), the user must set iell=1 and specify the GRS that will define these latitude measures.

Correction model:

iuci: is the integer FORTRAN input unit number for any spherical harmonic spectrum of correction values. This parameter can be set to 3 as a default.

jmin: is the minimum harmonic degree required for synthesis of the correction spectrum.

jmax: is the maximum harmonic degree required for synthesis of the correction spectrum.

name_cnv: is the name of the ASCII file containing the spectrum of correction values. name_cnv must be in the directory path specified by path_mod.

Notes

1) As shown in Table 1, correction models are used on their own to compute gravity anomaly biases (isw=5) and in conjunction with gravitational potential models to compute geoid heights (isw=82). The values set for the four correction parameters above will only affect the computation in HARMONIC_SYNTH when isw=5 or isw=82. There is no need to set any valid values for the above four parameters when isw is neither 5 nor 82. Note that a correction model used when isw=5 (mGal) will be different from a correction model used when isw=82 (m).

2) The harmonic summation involving elevation and/or correction models may start from as low a degree as zero, while the harmonic summation involving gravitational potential models cannot start from a degree that is less than two.

FORMATS AND UNITS FOR INPUT FILES

HARMONIC COEFFICIENT INPUT FILES

HARMONIC_SYNTH expects ASCII files for the input harmonic models. The first four columns of these files should contain:

COLUMN: 1 2 3 4 degree (n), order (m), C_{nm}, S_{nm}

These data are read using free FORMAT. For example, the first few lines of an ASCII input file containing a gravitational potential spectrum may look something like:

```
2 0 -0.484165371736E-03 0.000000000000E+00
2 1 -0.186987635955E-09 0.119528012031E-08
```

2 2 0.243914352398E-05 -0.140016683654E-05
3 0 0.957254173792E-06 0.000000000000E+00
3 1 0.202998882184E-05 0.248513158716E-06
3 2 0.904627768605E-06 -0.619025944205E-06
3 3 0.721072657057E-06 0.141435626958E-05

Notes

1) The specific ordering of the coefficients within the file is irrelevant. The file may contain the coefficients written such that the "slow" varying index is the degree (as above) or such that the "slow" varying index is the order, or any other ordering whatsoever.
2) HARMONIC_SYNTH expects to read two coefficients even when the record corresponds to a zonal term, *i.e.*, all Sn0 are supposed to be filled with zero values in the input coefficient file. See the S(2,0) and S(3,0) coefficients in the example above.
3) HARMONIC_SYNTH zeroes out all coefficients BEFORE beginning to read values from an input coefficient file.
4) Notes (1) through (3) apply to all types of input coefficient files (gravitational potential, elevation, and correction).

Scattered Point Input File

For gravimetric quantities computed at arbitrarily scattered locations from a spherical harmonic spectrum of the gravitational potential, the position of each computation point can be defined either by geodetic or geocentric coordinates. Geodetic coordinates are preceded by an integer flag of it=1, whereas geocentric coordinates are preceded by an integer flag of it=2. Thus the input record in each case should contain:
(it) (decimal degrees) (decimal degrees) (metres) 1 geodetic latitude longitude geodetic height 2 geocentric latitude longitude geocentric distance
These data are read using free FORMAT. For quantities computed solely from either a surface harmonic spectrum of elevations (isw=2) or of correction values (isw=5), the position of each computation point should be defined by a geodetic latitude & longitude coordinate pair (it=1), or a geocentric latitude and longitude coordinate pair (it=2). In this case the input record should contain:
(it) (decimal degrees) (decimal degrees) 1 geodetic latitude longitude 2 geocentric latitude longitude
Again, these data are read using free FORMAT. In the case of geodetic coordinate entry (it=1) immediately above, HARMONIC_SYNTH will assume the point to be situated on the mean earth ellipsoid (see GEODETIC REFERENCE SYSTEM), and transform the geodetic latitude to geocentric latitude.

Geoid Height Computation

HARMONIC_SYNTH provides three options for computing geoid heights (also known as geoid undulations) at scattered points (igrid=0). These options are selected using isw=80, 81 or 82. All three options compute geoid heights through a correction to the height anomaly (ζ) or the quasi height anomaly (ζ*).
isw=80:
This option computes the geoid undulation according to the following formulae:
N = ζ + (Dg_B/©_overbar)*H_ortho
where,

ζ: height anomaly at the computation point (ζ in Table 1)
Dg01: free-air anomaly at the computation point (Dg01 in Table 1)
Dg_B: Bouguer anomaly at the computation point, computed from Dg01 and H_ortho using the formula: Dg_B = Dg01 - 0.1119*H_ortho
H_ortho: orthometric height at the computation point
©_overbar: mean normal gravity along the plumbline, computed using Heiskanen and Moritz (1967, Eq. 4-42), using geodetic rather than normal heights.
For this computation, the free-air anomaly and the height anomaly are computed from the gravitational potential model (input unit iumi), for a point whose location is specified by the first three coordinates in the records below. The orthometric height of the computation point is supplied with the other three coordinate data in one of the two following formats:
(it) (dec. degrees) (dec. degrees) (metres) (metres) 1 geodetic_lat. longitude geodetic height orth. height 2 geocentric_lat. longitude geoc. distance orth. height
isw=81:
This option computes the geoid undulation in a manner almost identical to that for the isw=80 option. The only difference here is that the orthometric height is not supplied from the coordinate line input, but is instead computed for each point by harmonic synthesis of a spherical harmonic model of the elevation (input unit iuhi), which in this case represents the topographic elevations above mean sea level. The format for the coordinate input file is:
(it) (decimal degrees) (decimal degrees) (metres) 1 geodetic_latitude longitude geodetic height 2 geocentric_latitude longitude geocentric distance
isw=82
This option differs from the previous two options. Here, HARMONIC_SYNTH computes a quasi height anomaly (ζ*) from a gravitational potential model input from unit iumi. ζ* is computed for a point residing on the surface of the ellipsoid (geodetic height = zero), whose location is specified by the two coordinates in the records below. HARMONIC_SYNTH also computes, from a predetermined correction model (input unit iuci), the correction term required to convert the quasi height anomaly ζ* directly to the corresponding geoid undulation. Both ζ* and this correction term are computed for each input point via harmonic synthesis of the respective spherical harmonic model. In this mode no geodetic or orthometric height information is needed for the computation of the geoid undulation. This mode follows exactly the procedure described in Rapp (1997). Here the format for coordinate input is:
(it) (decimal degrees) (decimal degrees) 1 geodetic_latitude longitude 2 geocentric_latitude longitude

Ellipsoidal Harmonic Synthesis

Table 1 shows that HARMONIC_SYNTH provides for two modes of surface ellipsoidal harmonic synthesis:
isw=100: spherically approximated gravity anomaly (mGal)
isw=101: elevation (m)
In both cases, the surface ellipsoidal harmonic synthesis is controlled using the same GENERAL INPUT PARAMETERS as for spherical harmonic synthesis from a gravitational potential model. In these cases we have:
iumi: is the integer input unit for the ellipsoidal harmonic spectrum (set to 1 as a default).
lmin: is the minimum harmonic degree required for synthesis of the ellipsoidal spectrum.
lmax: is the maximum harmonic degree required for synthesis of the ellipsoidal spectrum.

path_mod: as previously specified.
name_mod: is the name of the ASCII file containing the ellipsoidal spectrum. The format for this file is identical to the format for a spherical spectrum.
aegm: is a scale factor applied to the ellipsoidal harmonic spectrum (see below under "isw=100").
gmegm: is a scale factor applied to the ellipsoidal harmonic spectrum (see below under "isw=100").
isub: this parameter should be set to isub=0.
igrid: Set igrid=0 for scattered values on the reference ellipsoid, and igrid=1 for an equiangular grid on the ellipsoid.
exclude: not used for ellipsoidal harmonic computations.
For isw=100 and isw=101, HARMONIC_SYNTH will only generate values on the reference ellipsoid defined by the parameters in the GEODETIC REFERENCE SYSTEM section above. The input ellipsoidal spectrum must be referenced to the same mean-earth ellipsoid defined by these parameters.
For synthesis at scattered points, the input record for each coordinated location should read:
(it) (decimal degrees) (decimal degrees) 1 geodetic_latitude longitude 2 geocentric_latitude longitude
where both forms of coordinate input define points on the surface of the reference ellipsoid.
isw=100:
This option will generate spherically approximated gravity anomalies of the same form as for Dg01 in Table 1.
Dg100 = -∂ (T)/ ∂r - 2*T/r
However, for isw=100, the input coefficients represent the ellipsoidal spectrum of the harmonic quantity:
r * Dg01 = -r * ∂(T)/∂r - 2*T
In addition, these coefficients are assumed to have been scaled by a factor of (aegm*aegm)/((x-1)*gmegm) where: x=n for all ellipsoidal harmonic degrees ‘n' above n=1 (n=2,3,4,ä) and x=2 for n=0 and n=1. The aegm and gmegm values refer to an arbitrary spherical harmonic spectrum of the gravitational potential against which the ellipsoidal spectrum is being compared, such that:
gnm_e = gmegm/(aegm*aegm) * (x-1) * C_{nm}_e
where Cnm_e are the ellipsoidal harmonic coefficients required by HARMONIC_SYNTH for isw=100 and "gnm_e" are the coefficients used by Gleason (1988, Eq. 3.1).
Note
Recall that conversions between ellipsoidal and spherical harmonic coefficients do not preserve the maximum degree. See (Jekeli, 1988, page 112) for details. Consider a set of ELLIPSOIDAL harmonic coefficients Cnm_e up to maximum degree N_e. Consider also the corresponding set of SPHERICAL harmonic coefficients Cnm_s, which extend to a sufficient maximum degree N_s, so as to describe the same gravity field as the Cnm_e. In this case, point gravity anomalies (gridded or scattered) generated from the Cnm_e coefficients (isw=100, lmax=N_e, iflag=0) will match the point gravity anomalies generated from the Cnm_s coefficients (isw=01, lmax=N_s, iflag=0) by HARMONIC_SYNTH. With N_e=2160 and N_s=2190, we have verified that the discrepancies between two such global sets of anomalies (5'x5' grid size) never exceeded 0.7 microGal. However, for area-mean value computations on the surface of the ellipsoid (iell=1, igrid=1, iflag=1), the user may observe some small

discrepancies between the anomalies computed from Cnm_e and those computed from Cnm_s. These discrepancies arise because, for spherical harmonic synthesis of area-mean values on the surface of the ellipsoid, HARMONIC_SYNTH assumes a constant geocentric radius over each geographic block. Using N_e=2160, N_s=2190 and 5'x5' block-size for area-mean gravity anomaly computation, this discrepancy had a global RMS value of +/-2.6 microGal and a maximum absolute value of about 0.08 mGal. This type of discrepancy decreases with decreasing meridional extent of the block. Using N_e=2160, N_s=2190 and 1'x5' block-size, its maximum absolute value was found to be about 4.5 microGal and its global RMS value +/-0.1 microGal.

isw=101:

This option will generate elevations (m) from an ellipsoidal harmonic elevation model. It is essentially the ellipsoidal harmonic equivalent of the isw=2 option for spherical harmonic expansions. That is, the isw=101 option performs an unscaled surface ellipsoidal harmonic synthesis of the coefficients in the input unit iumi. Thus this option can be used for any surface ellipsoidal harmonic synthesis task for which the units of the output are completely determined by the units of the input coefficients.

External Routines

HARMONIC_SYNTH is written so as to be as self-contained (and "portable" across computing platforms) as possible. Apart from standard FORTRAN 77 intrinsic functions, the only machine- dependent call is that used to read the machine clock and thus time the run. For this purpose, HARMONIC_SYNTH calls the function "secife" which was kindly provided by Heiner Denker in 2001. The user will have to modify this routine to read the internal clock on their platform.

In the present version of HARMONIC_SYNTH calls to the external routine "flush" have been de-activated (commented out). This routine is useful only for monitoring the progress of HARMONIC_SYNTH while this is executing.

9.2 GRAVSOFT

The GRAVSOFT suite of software provides similar functionality to the HARMONIC_SYNTH software, but is divided into a number of separate routines.
Here we describe the geocol, geoip and harmexp functions referred to in the algorithm specification section (section 3.4).

The GRAVSOFT software take input in standard formats, defined in the URD [1].

GeoCol

Programmed by : C.C.Tscherning,

University of Copenhagen, Denmark.

2005-04-27, revised 2005-10-10.

The program is copyright by the author first time 1975, and latest 2005. It may be copied and transferred to non-commercial users on the same conditions as other GRAVSOFT programs.

A file http://cct.gfy.ku.dk/geocol17.log contains a record of suspected and corrected errors, updates and planned updates. It also contains test input and output.

The primary function of the program is the computation of an approximation to the anomalous potential of the earth, T, using stepwise least squares collocation, with maximally 3 steps. The computations are done in spherical approximation (logical variable lspher is true) or no approximation is used. In spherical approximation is the radial distance r = RE+h, where h is the ellipsoidal or orthometric height and RE the mean earth radius and the geodetic latitude is put equal to the geocentric latitude.

The program may also be used for the evaluation of a spherical harmonic series or a datum transformation, (logical variables lpot and lncol are .true., see below).

The method requires the specification of (1) one or two sets of observed quantities with known standard deviations and (2) one or two covariance functions, cov1, cov2. Spherical harmonic coefficients may also be used in a remove-restore mode (logical variable lpot is .true.). The sets of observed quantities may consist of data in different files being of different kinds (such as gravity anomalies and height anomalies).

The observations obs(j) are associated to T by a linear functional L_j, to contingent parameters X [19] and to the noise n_j according to the equation

$$Obs(j) = L_j(T) + A_J X + n_j$$

The basic covariance functions used are isotropic and harmonic outside a Bjerhammar sphere with radius R_i, where i=1 or 2. They are specified by a set of empirical anomaly degree-variances σ_k^2 of degree less than an integer variable imax, and by one of three anomaly degree-variance models for the degree- variances of degree greater than imax [20]. The functions depend on the coordinates of two points, P, Q, with radial distances equal to r and r', respectively and spherical distance psi (ψ), so that the covariance of two values of the anomalous potential are

$$\mathrm{cov}_i\left(T(P),T(Q)\right)=\mathrm{cov}_i(t,r,r')=a_i\sum_{k=i\max+1}^{\infty}\sigma_k^2\frac{A_i}{P_{ktype}(k)}\left(\frac{R^2}{rr'}\right)^{k+1}P_k(t)$$

$t=\cos(\psi), P_k$ Legendre Polynomials a_i, A_i constants and
$p_1(k)=(k-1)(k-1), p_2(k)=(k-1)(k-2)(k+ik)\ or\ p_3(k)=(k-1)(k-1)(k+ik)(k+ik1)$

The covariances between two quantities are derived by applying the associated functionals L_j on the covariance function [20].

The observations may be potential coefficients, mean or point gravity anomalies (delta g), gravity disturbances (Tr), height anomalies (zeta), deflections of the vertical (ksi, eta), gravity gradients (Tij) and (quasi-harmonic) density contrasts.

A filtering takes place simultaneously with the determination of the anomalous potential. Therefore all observed must have assigned an error estimate either the same for all records in an file or as a value found in the data file in a position right after the data element(s) in the record. If the program is used for gross-error detection, the data analyzed must also have associated an error- estimate.

The observations may be given in a local (*e.g.* instrumental) frame in which case the attitude angles must be input [21]. They may also have correlated errors, in which case the error-covariance function must be defined, e.g. using the model found in [22].

The determination is made in a number of steps equal to the number of sets of observations. When potential coefficients are used, will the- se form a separate set. Contributions from a terrain potential may not be computed by this version, but may be input and will then be added to or subtracted from the various quantities. Each dataset (each step) will determine a harmonic function, $\tilde{T}_i$, and the anomalous potential will be equal to the sum of these functions.

Potential coefficients will determine a function $\tilde{T}_0$ equal to the coefficients $\overline{C}_{ij}$ multiplied by GM and the corresponding (fully normalized) solid spherical harmonics, V_{ij}.

The up to two sets of data different from potential coefficients will each be used to determine constants b(j), which define the harmonic functions $\tilde{T}_i$. The functions are then equal to the constants multiplied by the covariance between the observations and the value of the anomalous potential in a point, P.

$$\tilde{T}_i(P)=\sum_{j=1}^{N}b_j\,\mathrm{cov}_i\left(T(P),obs(j)\right)$$

The estimated approximation to T is then

$$\begin{aligned}\tilde{T}(P)&=T_0(P)+\tilde{T}_1(P)+\tilde{T}_2(P)\\&=GM\sum_{ij}^{n}\overline{C}_{ij}V_{ij}(P)+\sum_{j=1}^{N_1}\mathrm{cov}_1\left(T(P),obs(j)\right)b(j)+\sum_{j=N_1+1}^{N_2}\mathrm{cov}_2\left(T(P),obs(j)\right)b(j)\end{aligned}$$

The constants b(j) are computed by solving one or two system of normal equations, using the subroutine NES.

$$\{b_j\} = \overline{C}_i^{-1}\{obs(j)\} = \{\text{cov}_i(obs(j), obs(k)) + \sigma_{jk}\}^{-1}\{obs(k)\}$$

In case also parameters X have to be estimated, the equations become slightly more complicated involving the matrix A, formed by the vectors A_j.

$$\tilde{T}_i(P) = \{\text{cov}_i(T(P), obs(j))\}^T \overline{C}_i^{-1}\{obs(k) - A_k\tilde{X}\}$$

$$\tilde{X} = \left(A^T\overline{C}_i^{-1}A\right)^{-1} A^T\overline{C}_i^{-1}\{obs(k)\}$$

The main function of the program is, besides the computation of the constants b(i) and the parameter vector X, the prediction of the quantities zeta, ksi,eta, delta g, Tr, gravity gradients, density contrasts in points Q, and the errors or error-correlations of the predicted quantities.

Corrections to a set of spherical harmonics may also be computed (lspharm is put equal to .true.) [23] and compared to a reference set.

Datum-shift parameters may also be determined by the program in the form of the change in the longitude and latitude components of the deflection of the vertical and of the height-anomaly in a point with given latitude and longitude, cf. [24], or one or more of the parameters of a 7-parameter datum shift. In this case observations of the difference between geocentric and local Geodetic coordinates may be used. Bias and tilt parameters may also be determined. The different types of parameters are defined through parameter codes.

The data used to create one solution may be preserved and used in order to re- establish the solution or as a building stone for a new solution. In the first case a logical variable lwrsol must be true and in the second case must the variable lresol be true. The Cholesky-reduced normal equations may also be preserved and used for error-estimation or as building stones for approximations which use more data (logical variable lsaneq is given the value .true.).

The different kinds of data are identified using kind-codes.

Data-kind codes and units used in this version:

data-kind	**codes:**		**units:**
height-anomaly or geoid undulation (use 11 for satellite altimetry)	1	11	meters
anomalous potential (t)	51		m**2/s**2
gravity disturbance (-dt/dr)= g-gamma	12		mgal
and dt/dr		52	mgal
gravity anomaly	2	13	mgal
	43	53	
vertical gravity anomaly gradient		14	e.u.
vertical gravity disturbance gradient		15	e.u.
deflection of the vertical, meridian com.	3	16	arcsec
	43	56	
deflection of the vertical, prime verti.	4	17	arcsec
	44	57	
gravity anomaly gradient, meridian comp.		18	e.u.
gravity anomaly gradient, prime vert. co.		19	e.u.
gravity disturbance gradient, meridian co.	60	20	e.u.
gravity disturbance gradient, prime vert.	61	21	e.u.
second order derivative in northern direction	62	22	e.u.

2 * mixed second order derivative of t	63	23		e.u.
second order derivative in eastern direction	64	24		e.u.
difference between second order derivatives in prime vertical and meridian planes		25		e.u.
pair of deflections of the vertical	5 45	26 66	96	arcsec
pair of horizontal gravity anomaly grad.	28	68		e.u.
pair of horizontal gravity disturb. grad.	30	70		e.u.
pair of kind (25, 23)	35	75		e.u.
second order derivatives (15, 30, 35), only permitted when lncol is true.	37			e.u.
fully normalized spherical harmonic coeff.	27			
ellipsoidal height difference old minus new datum values	6			meters
latitude and cos(latitude)*longitude difference, new minus old datum values.	7			arcsec
satellite altimetry cross-over difference	9			meters.
anomalous potential	8			m**2/s**2
density contrasts	10			g/cm**3*scale factor

If code 13 is used for gravity, spherical approximation is used, and if code 2 is used, the potential coefficient set approximation is used. Code 13 is recommended in general.

Codes > 40 indicate that a quantity is given in a local reference system, east/north/up. A logical variable lsatp is put true and an integer isat is input equal to 1 if a rotation in the horizontal plane is needed, (then azimuth must be given) and equal to 2, 3 or 4 if a 3D rotation is needed. Then the either the azimuth tilt and roll must be given in decimal degrees or the 3 x 3 rotation matrix (from an East-North-Radial Up frame).

Codes in the interval from 25 to 39 are used to indicate that two quantities (like ksi and eta) are input or predicted simultaneously. In this case error- covariances must not be computed and the data must not have correlated errors. Note, that it is of advantage to use or compute pairs of quantities because covariances or contributions from spherical harmonic expansions may be computed simultaneously for these quantities.

Code=96 indicates use of an ngs format. Codes 42, 45, 70 and 75 were also used in IAG SSG 3.70 and SSG 3.90 cooperation.

Files needed for running the program:

Unit number	used for	temporary	permanent
5, 6	standard input and output files	yes	
8	direct access for normal equations		yes if needed later
14	direct access to store rotation elements	yes	
16	direct access, used to store observation coordinates and the solution	yes	
17	formatted, used for restart file or result output (lwrsol or lpunch true).		yes
18	binary, used for storage of covariance function parameters on binary form.		yes
19	binary, used for storage of solution.		yes
11	formatted used for error in grid		yes

9	input unit for potential coefficients formatted or binary depending on "lbin".		yes
3	temporary storage of potential coefficients if density anomalies are used or computed, and permanent if lbipot is set true		yes
2	temporary storage of contributions from data only associated with parameters.	yes	
4	or different from other unit numbers, input of data if lin4 is true	yes	
12	file with detected gross-errors (lerr, lcomp, lstat must be true).		yes

An updated list is found as a part of the comments in the program.

Brief summary of input specifications.

Details are found in the main program, with reference back to this summary through the numbers (1), (2), ... etc.

(0) input of logical variable, linter, true if interactive input. (It may also be used if the user wants the input instructions to be copied to the output file.)

(1) input of logical variables determining the execution, and contingently names of files holding restart-file, and of normal equation files. most important are: lspher - the computations are made in spherical approximation. ltran - data may have to be transformed from local datum to geocentric system not already defined in Subroutine icosys. lpot - potential coefficients used to compute contribution from a spherical harmonic expansion. lparam- datum or bias parameters to de determined. lncol - no collocation solutions are wanted. liosol- establish or use restart files on character or binary form (units 17, 18 and 19). if liosol is true, input of 5 logical parameters: including the logical lwrsol- write restart file on unit 17.

(1c) if lncol is false, input of number of files used to hold upper symetric part of normal equations, the name of each file, its associated Fortran unit number and its size in number of blocks (note that the block-size may vary between different versions of the program)

(2) input of integer identifying geocentric system. (System definition already given in subroutine icosys).

(3) if ltran is true input of parameters for system in which data contingently are given, and parameters for transformation to geocentric system. The system identification code is 0 for this system.

If linsol is true, jump to (9).

(4) if lpot is true input of specifications for potential coefficients, including a parameter lfm, which is true when the coefficients are in the standard input file (unit 5) and false, if they are input through unit 9. It is possible to input formatted coefficients.

(5) if lfm is true input of coefficients.

If lncol is true, jump to (15)

(6) general specification of covariance function degree- variance model number (1, 2, 3), and the constants ik and/or ik1.

(7) details concerning covariance function: Bjerhammar-sphere radius, Bi, scale factor ai, error-degree-variances, variance of gravity anomalies at zero height (used to calculate scale factor Ai) and contingently variables defining tables used for fast interpolation of values.

(8) if lparm is true, input of logical variable lallp, true if all parameters are of the same kind. If true, followed by parameter codes, otherwise the codes must be given for each data-set. Also the name of a scratch files (unit 2) to be used to hold variance- covariance of parameters. (Only in use, if data independent of T, such as cross-over differences of satellite altimetry, are used).

(9) input of specification (format and sequence of elements) of data set, including value of lin4, true if the observations are input from unit inz (normally equal to 4).

(10) if lin4 is false, then input of observation records from unit 5 else from inz.

When last record is encountered:

(11) input of lstop, true if the data set is the final one contri- buting to the current collocation step, and of lresol, true if the solutions or the reduced normal equation matrix are to be input or re-used, respectively.

If lstop is false, jump to (9)

(12) if lresol is true, input of lsaneq and ifc. lsaneq is true if the ifc first reduced colums of the normal equations are stored.

(13) if ifc is equal to the total number of observations and lresol is true, input of the solutions. (otherwise, the last columns will be established, reduced, and the equations solved).

If the first collocation step now is terminated, input of variables telling whether a second step should be made, otherwise jump to (15).

(14) input of lcref, true if a second step is to be made and of lparm, true if parameters are to be determined in a second step.

If lcref is true, then jump back to (7).

(15) input of lgrid, true if predictions are to be made in a grid, lerno, true if errors of prediction are to be computed or re- produced in output (lncol true), lcomp, true if observed and computed quantities are to be compared (differenced) and lsphar, true if corrections to spherical harmonics are to be determined.

If lerno is true, it is possible to compute error-correlations. In this case 3 file-names must be input. If predictions are done in a grid and errors are to be computed it is possible to defer the computation of error-estimates, in which case a file-name to hold covariances must be input.

If lspharm is true input of various information concerning spherical Harmonic prediction (15C).

If lgrid and lsphar are false, then input of specifications of quantities to be predicted.

(9) and coordinates of prediction points (10), then jump to (18).

(16) input of grid specifications (start point, steps etc.).

(17) if lcomp is true, input of observations in the grid points.

(18) input of lstop. if it is false, jump to (15), otherwise the program will finish.

9.2.1.1.1 Output:

If linter is true, all input specifications will be output. If ltest is true, different types of test-output will be available such as covariances and normal equation matrix.

First 20 values of solution vector b(j) as well as estimated parameters, X, will be output (if lparm is .true.).

Names of input and output files will be output.

Predicted quantities with or without error estimates will be output. Error Correlations will be output to a separate file.

If lncol is false, the Cholesky-reduced normal equations are stored in binary form on one or more files. If the program stops before the completion of the reduction of the normal-equations, the already reduced part may be re-used.

Therefore the column-number of the last reduced column is output during reduction. This number may then be used as described under input (12).

If lcomp and lerno is true contingent output of suspected gross-errors.

References:

In the program text are found references to equations, which are found in the following papers and reports.

Ref(a): Tscherning, C.C.: Covariance expressions for second and lower order derivatives of the anomalous potential. Reports of the Department of Geodetic science no. 225, The Ohio State University, Columbus, 1976.

Ref(b): Tscherning, C.C..: A fortran IV program for the determination of the anomalous potential using stepwise least squares collocation, Department of Geodetic science, the Ohio State University, Report no. 212, 1974.

Ref(c): Heiskanen W.A. and H.Moritz: Physical Geodesy, 1967.

Ref(d): Tscherning, C.C..: Computation of the second-order derivatives of the normal potential based on the representation by a legendre-series. manuscripta geodaetica, vol.1, pp. 71-92, 1976.

Ref(e): Tscherning, C.C..: Determination of datum-shift parameters using least squares collocation, Boll.geodesia sc. Aff., ann. XXXV, no. 2, 1976.

Ref(f): Tscherning, C.C.: Implementation of algol-procedures for covariance computation on the r4000-computer. The Danish Geodetic institute internal report no. 12, 1976.

Ref(g): Sanso, F. and W.-D. Schuh: Finite covariance functions. Bulletin Geodesique, Vol. 61, pp. 331-347, 1987.

Ref(h): Tscherning, C.C.: Prediction of spherical harmonic coefficients using least-squares collocation. JoG, 2001.

Ref(i): Tscherning, C.C.: Local gravity field approximation, Proc. Beijing int. summer school, pp. 277-261, 1984.

Ref(j): Tscherning, C.C. and P.Knudsen: Determination of bias parameters for satellite altimetry by least-squares collocation. Proceedings 1. Hotine - Marussi Symposium, Rome, June 3-6, 1985, pp. 833-852. Politecnico di Milano, 1986.

Ref(k): Tscherning, C.C.: Computation of covariances of derivatives of the anomalous gravity potential in a rotated reference frame. Manuscripta Geodaetica, Vol. 18, no. 3, pp. 115-123, 1993.

Ref(l): Tscherning, C.C.: Geoid determination by least-squares collocation using GRAVSOFT. Lecture Notes "Int. School of the Determination and Use of the Geoid", Milano, Oct., 1994, pp. 135 - 164, published by International Geoid Service, 1994.

Input examples:

The following examples are from Ref(l), see also http://www.gfy.ku.dk/~cct/milano21.htm

Example 1. Computation of contribution from the spherical harmonic coefficients OSU91 and subtraction of the contribution from gravity data in the file nmfa.

```
/disk1/cct/dgravsoft/geocol16<<!
T                                   Input (0)
t  f T f F f t f                    Input (1)
t                                   Logical to confirm correctness of
input
 5                                  Input (2)
OSU91A                              Input (4)
 3.98600500E+14  6378137.0 -484.1655 360 f f T F F
 (2i3,2d19.12)                      Format of coefficients
/disk1/cct/cctf/osu91a1f            Name of file holding coefficients
F F t f                             Input (15)
  1  2  3  3  4  5  0 13 -1      0.00 t F F F F f t f F t      Input (9)
/disk1/cct/cctf/nmfiles/nmfa        Name of file holding data
25                                  Fortran unit number of file
nmfa.osu91                          Output file name
5.0                                 Bin-size of histogram
t                                   Logical to confirm correctness
T F                                 Input (18) – stop
```

Example 2. Computation of residual geoid heights from gravity anomalies, from which the contribution of OSU91 and the topography have been subtracted.

```
../../dgravsoft/geocol16<<!
t                                   Input (0)
t F T T F f F t                     Input (1)
t f f f f f
nmrestart
1                                   Input (1c)
nmneq1
20 20
 F T f f f f

T                                   Confirmation of correctness
 5                                  Input (2)
OSU91A                              Input (4)
 3.98600500E+14 6378137.0 -484.1655 36 f f T F f
 (2i3,2d19.12)
/cct/cctf/osu91a1f
```

```
 2                                 Input (6)
 4
-6.061 196.20 360 F f T f          Input (7)
 -1 4 0.207227
/cct/cctf/osu91.edg
t                                  Confirmation of correctness
 -1 2 3 3 4 7 0 -13 -1 1700.00 f F F t F f t t F t Input (9)
/cct/cctf/nmfiles/nmfa.rd          Name of data file with gravity
25                                 Fortran unit number
5.0                                Bin-size of histogram
32.5 34.5 -107.5 -105.5            Sampling area boundaries
0.2                                Common error-estimate
t                                  Confirmation of correctness
T f                                Input (11)
F F F                              Input (14)
F t f f                            Input (15)
 1 2 3 3 4 0 0 11 -1 1700.0 t f F F F F f T F T    Input (9)
/cct/cctf/nmfiles/nm.h2            Name of input file with positions
27                                 Fortran unit number
nm.geoid                           Name of output file
33.0 34.0 -107.0 -106.0            Data sampling boundaries
t                                  Confirmation of correctness
f                                  Input (18)
F t t f                            Input (15)
 1 2 3 3 4 5 6 -5 -1 1700.0 f F F t F F T T F T    Input (9)
/cct/cctf/nmfiles/nmdfv.rd         Data file name (deflections)
21                                 Fortran unit number
0.2                                Bn-size of histogram
33.0 34.0 -107.0 -106.0            Boundaries
0.4                                Common error estimate
-1.0                               Indicator of no gross-error est.
t                                  Confirmation of correctness
t                                  Input (18)
```

GEOIP

programmer: rene forsberg, danish geodetic institute.
university of calgary, mar 87. modified october 87, honefoss.
minor changes, rf thessaloniki nov 88
revised for swedish map projection and more points, rf june 89
selection added, greenland aug 12, 1992, rf
new modes for ice etc., nov 1992, rf
allowing grid interpolation outside (with 9999's) in linear mode, dec95

Interpolation (linear or splines) from grid(s) to points or grid to grid.
Also implements subtract/add or 3-D "sandwich" grid interpolation.
UTM grids are fully handled, included UTM to UTM interpolation.
Program for interpolating values from a grid using bilinear or spline interpolation. the grid or the prediction points may be in either geographical or utm coordinates. the spline prediction is performed in a window of size 'nsp' x 'nsp' points around the wanted points, with typical value of nsp being 8 for a good interpolation.

The grid file must be in standard format, *i.e.* scanned in e-w bands from n to s, initiated by a label (lat1,lat2,lon1,lon2,dlat,dlon). for utm grid northing and easting replaces lat and lon, and additio- nally ellipsoid number (1: wgs84, 2:ed50, 3:nad27) and utm zone must be given in label. if utm zone 99 is specified, this signals the swedish national projection rt39 (only an approximative transformation is currently implemented, only for geoid use etc.) the program may interpolate from one utm zone to another.

The grid file may be in direct access binary format, as produced by program 'gbin'. the use of direct access format speeds up access time. the program recognizes binary files by a special code (777) written in the first record.

The program attempts to interpolate all points with a distance 'rmin' or more from the margins (for fft applications, e.g., points near the margin are often useless). the program may interpolate in very big grid files, but assumes the prediction points to be reaso- nably close, reading in the smallest necessary subgrid to perform the wanted interpolations.

Special options:

- the program may interpolate in two grids in the same file. this option is especially designed for deflections of the vertical. the two grids must have identical labels. two grid interpolation is signalled by negative mode or mode > 100, see below. two-grid interpolation can only be done for grid files in txt format.

- two grids representing data in different elevations may be used to interpolate values at some elevation between the two grids. in this case mode = mode+100 must be specified

- the program may also be used to convert a list of terrain corrections into rtm-effects through a bouguer reduction to the interpolated reference level. this approximation is only valid for long-wavelength reference grids. stations at sea (negative heights) will have height set to zero.

- in bilinear interpolation mode the grid file may contain 9999-values (signals unknown), interpolated values are assigned to 9999 if any unknown values are encountered in the closest 4 points. For grid interpolation the wanted grid may be to large (9999's will be written at unknown values) *program input:*

```
  gridfile,
  outputfile,
  mode, nsp, rmin, lsel
  (lat1,lat2,lon1,lon2 - for lsel true)
  (pointfile - for mode 1 - 3 and modes > 10)
  (idno - for modes 11-14,19-22 etc.)
  (lint,lat1,lat2,lon1,lon2,dlat,dlon - for mode 4 only)
  (iell,izone - for utm only)
  (h1,h2 - for modes > 100 only)
where
mode ...
   1: prediction point list in geographic coordinates (degrees)
   2: do, with lat and lon given in degrees, minutes, seconds
   3: prediction point list in utm coordinates
   4: predictions wanted in grid (geographic or utm)
   5: individual prediction points in lat, lon (degrees)
   6: do, with lat, lon in degrees, minutes, seconds
   7: individual prediction points in utm
   10: like 1, with a data value in file written after predictions
   11: like 1, predictions  subtracted  from values given in file
       (data value must follow after the height)
   12: do, but predictions added to values in file
   13: like 11, with a second data value for each point not hanged
   14: like 12, doc
   15: 'pointfile' contains a grid, from which the interpolated
       values from 'gridfile' are subtracted i.e. 'outfile' =
       'pointfile' - 'gridfile'
   16: 'pointfile' contains a grid, to which the interpolated values
       from gridfile are added.
   17: 'pointfile' contains a grid which defines the interpolation
       points. the given grid values are only used in two-height
       interpolation mode (117), see below.
   18: 'pointfile' contains a grid with unknown values (9999). the
       unknown values are interpolated from the gridfile, other grid
       values left untouched.
   19: list of terrain corrections converted to rtm anomalies through
       bouguer reduction to reference level 'gfile'
   20: list of free-air anomaly data converted to rtm-reduced data
       using a bouguer plate approximation only additional input:
       density
   21: conversion of free-air data to Bouguer using grid additional
       input: density
   22: migration of ERS-1 data over ice caps. grid is slope grid in
       degrees
   31, 32, ..: like 11, 12, .. for utm coordinates in 'pointfile'
   - if mode is negative, mode = abs mode and two grid interpolation
       is performed (the two grids must be inthe same file, e.g.
       deflections of the vertical fromprogram "geofour")
   - if mode > 100 then point interpolation is done betweentwo grids
       in different heights using mode = mode-100.the levels to which
```

```
      the two grids refer must be input.if mode=117 then 'pointfile'
      is assumedto be a height grid defining the interpolation
      level.
  nsp .... spline window size. nsp=0 means bilinear
      interpolation,nsp=1 is equivalent to a 8x8 spline (nsp=8)
  rkm .... minimum required distance to closest edge (km) for
      interpolation to take place
  lsel  .. select only data wihin a subregion (lat1-lat2, lon1-lon2)
      this option only works with point lists
  idno  .. data number in line (statno,lat,lon,height,d(1),d(2)..)
  lint ... a logical variable specifying that output should be in
      integer grid format (only needed for mode 3 and 4)
```

Files:

```
unit10  gridfile   grid file in standard format
 unit20  pointfile  station list file (no, lat, lon, height, ...)
 unit30  outfile    output file
 unit31,32          scratch files for intermediate storage of all prediction
points in binary format
```

Input example #1: Simple interpolation of grid file ("geoid.gri"), to obtain linearly interpolated values at point locations in file "points.dat". Note that the output file name always must be the 2^{nd} file name in the output stream (not very logical, but necessary due to the many options in GEOIP!)

```
geoip <<!
geoid.gri
output.dat
1 0 0
points.dat
!
```

Input example #2: Interpolation of grid file with two grids, representing a spherical harmonic gravity grid at heights 0 and 3 km, interpolated at the actual station height in the data file "borneo.rd", and then subtracted from an observed gravity file "borneo.faa".

```
geoip <<!
egm96gh.gr
..\data\borneo.rd
113 0 0 f            ! mode 113 is the sandwich mode
..\data\borneo.faa
1    ! datano in file
0 3000               ! heights of reference grids
!
```

HARMEXP

Modification of program from d. arabelos, university of thessaloniki, nov 88, rf. original subroutines from tscherning and goad.

Modified and updated, university of new south wales vax, feb 89

Modified nov 2002,rf (gnu fortran, last binary coeff must be nmax,nmax) (normal gravity field update)

Evaluates geoid, gravity and deflections in grids or points (alternative to GEOCOL)

Program for computing grid of geoid, gravity and deflections from a high degree and order spherical harmonic expansion, using 'gpotdr'.

Output will be in three files (geoid, gravity and ksi/eta) in units of m, mgal, and arcsec, respectively, in standard grid format.

input:

```
  coefficient file name,
  irefsys, mode
irefsys determines a and gm coefficients:
      0: use grs80 constants (i.e., only sum from J2)
      1: egm96 with grs80 normal field
      2: champ eigen-2 with grs80
* additional input, mode = 1 (grid computation):
  geoidfile,
  gravityfile,
  deflectionfile,
  nmax, lbin,
  fi1, fi2, la1, la2, dfi, dla (degrees), h (m)
'lbin' is true for coefficients on binary form.
  file name 'dummy' or '0' ensures no data of that kind is written.
* mode = 2 (point computation):
  pointfile
  outputfile
  kind, omode
where kind =  1: geoid
              2: gravity
              3: deflections
      omode = 0: list computed values
              1: list difference (pointfile - ref.field.)
              2: list sum (pointfile + ref.field.)
```

<u>Input example</u>: Prediction of geoid and gravity effects in a grid.

```
harmexp<<!
0 1
egm96geoid.gri
egm96grav.gri
0
360 t
33 37 22 28 .1 .1 0
!
```

9.3 Sphgric and sh_mdt

Sphgric

Program for the prediction of spherical harmonic coefficients using fast Least_Squares Collocation (LSC) from data gridded equidistantly in longitude. Error estimates are also calculated and error_correlations may be calculated optionally [25].
Programmed 2000_10_31 by cct, update 2006_02_07.
For one given latitude the data must be of the same kind, be in the same altitude and have the same error. However it is possible to have two or more kinds of data associated with the same parallel. Presently 9 kinds of data can be used, identified by an integer code:

Data type	code	units
anomalous potential (T)	10	m**2/s**2
geoid height	11	m
gravity disturbance	12	mgal
gravity anomaly	13	mgal
radial gravity gradient (Trr)	15	E.U.
derivative in nothern direction, Tn		
16 mgal		
2. order derivatives Tnr,	20	E.U.
2.order derivative Tnn,	22	E.U.
2. order derivative Tee	24	E.U.

All functionals are of the kind which when applied on a spherical harmonic expansion leaves the terms dependent on longitude unchanged, (expect for the sign for Tee).
Data may be point or mean values. Means are calculated as the mean of nstep*nstep values. Actual value of nstep is found in a parameter statement and must be an odd number.
The data may be generated using a spherical harmonic expansion, input from a file containing data generated by an earlier run of the program or observed values. This may also be used to test the program, i.e. the input and output coefficients should be close (seen in relation to the error estimate).
Note that data_types may be mixed and in different altitudes. In this case either geodetic or geocentric coordinates may be used.
The program may be run in spherical approximation, i.e. so that the distance from the origin is calculated as the sum of the mean earth_radius and the altitude. (Not recommended in general).
The covariance function used may either be represented by a finite Legendre series or as the sum of a Legendre series and a closed expression. See the subroutines covax, covbx and covcx and the references in these modules.
Various testing possibilities have been implemented. The most important is that a full LSC solution may be calculated when the number of data is small. Furthermore an alternative method for calculating associated Legendre functions has been implemented, and the results for the routine (lecurs) may be compared to the simpler spharm0 routine which is based on standard recursion algorithms. Problems are to be expected for high degree and high latitude.

Input:

Input may be done interactively or by preparing an input file for batch processing. Input has the following structure in the most simple case:

_________________________ INPUT (1) _________________________

use spherical approximation ? (t/f) F

_________________________ INPUT (2) _________________________

Test of program ? (T/F) F

_________________________ INPUT (3) _________________________

Is the grid equidistant in latitude ? (T/F) T

If the answer here is F (=false), the data and error specifications must be given for each parallel, see INPUT (3.2.1) _ (3.2.4).

_________________________ INPUT (3.1.1) _________________________

input gravsoft grid label (latmin,max,lonmin,max,dlat,dlon) using geodetic latitude.
-80.0 80.0 0.0 340.0 20.0 20.0

_________________________ INPUT (3.1.2) _________________________

input altitude of points (m) 300000.0

_________________________ INPUT (3.1.3) _________________________

input functional type (10: anomalous potential, 11: geoid, 13: gravity, 15: Trr 13

_________________________ INPUT (3.1.4) _________________________

input common standard deviation of noise 0.005

_________________________ INPUT (3.1.5) _________________________

are all data mean values (T/F) ? F

_________________________ INPUT (4) _________________________

input degree of spherical harmonic expansion to be used for test or comparison purpose 180

_________________________ INPUT (5) _________________________

input name of file with coefficients used for comparison or data generation pcoeff

_________________________ INPUT (6) _________________________

input name of file to hold calculated coefficients ccoeff

_________________________ INPUT (7) _________________________

read formatted (T/F) ? T

_________________________ INPUT (7.1) _________________________

input format e.g. (2I4,2d19.12) (2I4,2d20.12)

_________________________ INPUT (8) _________________________

Use of closed expressions ? F

If the answer here is T, input of specifications of the closed expression, see subroutine INCOV, input (6) _ (7D).

_________________________ INPUT (8.2.1) _________________________

input minimum and maximal degree of degree_variances 2 180

_________________________ INPUT (8.2.2) _________________________

input name of file with degree_variances (units mgal**2) egm96.edg

_________________________ INPUT (9) _________________________

input maximal degree for prediction 8

_________________________ INPUT (10) _________________________

output of error_estimate for max. degree to file ? T

_________________________ INPUT (10.1) _________________________

input name of file to hold error_estimates eco8.dat

________________________ INPUT (10.2) ________________________
input t if error_correlations are to be computed F
________________________ INPUT (11) ________________________
Input observations from file(s) ? (T/F) F
See Input (11.1.1) _ (11.1.4.1) if the answer is T: IF T, then input of number of files and for each file file_name, number of data_items, and the number in the data list of the data to be used followed (on a new line) t if data are geodetic coordinates and f if they are geocentric. data kind identifier (10, 11, 12, 13, 15, 16, 20, 22, 24) If F, data will be generated by the program from the coefficients.
________________________ INPUT (12) ________________________
Output of observations to file (T/F) F
If the answer is T, input of files names to hold observations.
________________________ INPUT (13) ________________________
Will covariances be input from file (T/F)? F
If T, input of file_names (13.1), otherwise the coefficients are output to one or more files, the name of which must be input. This can be used if the same data points are used again, with a new standard deviation of the error or with new values.
________________________ INPUT (14) ________________________
Output of coefficients and differences to current output? (T/F) T
If the answer is F, only output to the file with name given in Input (6).
For more complex input see the instructions contained in the program file sphgric.f.
Output consist of the input parameters, predicted and observed coefficients, their standard deviation and contingently the error covariances.
A summary of the results are given. If both full LSC and fast LSC are used a comparison of the results is made.
The covariance functions used in fast LSC are stored on a file 'covsph' an overwritten in the next run if not renamed. It may be rather large.

sh_mdt

Author: Rory Bingham, Proudman Oceanographic Laboratory
The program sh_mdt.f90 computes an MDT, filtered if required, by the spectral method. It takes as input a gravity model in the GGM format and a gridded MSS that has been extended globally (as described in Bingham et al). A land mask on the same grid as the MDT is required (Ideally it would use a high resolution land mask that could be interpolated to the required MDT grid) The final output of the program is the spectral MDT (smoothed if spectral smoothing is enabled), and (for spatial filtering) the smoothed MDT, and the residual. The gridded input and gridded output is in a very simple self-descriptive format: number of longitude and latitude points (rows and cols of array), longitudes, latitudes, and data. I have included a result of the program with the parameters as supplied.
The user definable parameters can be found at the beginning of the program. They are:
1. Parameters that define the regular lat/lon grid for the MDT.
2. The desired grace gravity model.
3. The truncation degree.
4. The spectral filter half-weight radius (set to zero if not required), the spatial filter radius (set to zero if not required), and the spatial filter type.

5. Four parameters which determine the reference ellipsoid (note the maths of my program assumes a normal Earth (best fitting) ellipsoid).
The program calls three subroutines:
1. grid2sh - This takes a field on a regular lat/lon grid at from it computes a set of spherical harmonic coefficients.
2. sh_mdt - The takes a set of shperical harmonic coefficients for the gravity model and for the MSS, and the parameters for the reference ellipsoid, and computes an MDT with filtering in the spectral domain if required.
3. filter - This takes a field on a regular lat/lon grid and applies a spatial filter, returning the smoothed field and the residual (the part removed by the filter). The user can choose between a boxcar, a Gaussian, a truncated Gaussian, a Hamming, or a Hanning filter type. The width of the filter specified by the user is the half-weight radius, apart from for the box car where it is the radius.
Below these higher level subroutines are more fundemental subroutines:
1. ref_pot - Calculates a reference ellipsoid potential.
2. norm_grav - Computes the normal gravity on the ellipsoid at a given latitude.
3. legendre - Evaluates the fully normalised Legendre functions at required latitude up to a specified degree and order.
4. gd2gc_clat - Converts from geodetic to geocentric colatitude and calculates the area elements on the now irregular grid.
5. gauss_av - Computes the weights for Gaussian filtering in the spectral domain for a given smoothing radius.
6. convert_tide - Converts between tide systems.

10 Appendix D: Using the Command Interface

The user interacts with GUT by entering commands to execute Python methods. We could also describe this as *calling* Python methods, *running* Python methods or *invoking* Python methods. The idea is for the user to set up a Main Object with one particular work-flow in mind, and then to execute a single method that performs calculations leading to the output of the desired product. To allow the user to experiment with one particular part of a scientific work-flow, methods for performing calculations up to intermediate stages are provided. After finishing with one particular work-flow, the user can then turn his or her attention to another work-flow, making any necessary changes to the attributes and data in the Main Object. Alternatively, it may be easier to start again from scratch with a new Main Object. As an extension to the concept of the Main Object, it would be possible to design a user interface based on a set of Work-flow Objects, one for each of the GUT work-flows. A Work-Flow Object would be the same as a Main Object, except that only the attributes, data and methods relating to one particular work-flow would be present. This would simplify the command interface and enable the names of the Python methods used in commands to be rationalized. This is because many of the method names are derived from the names of the logical data structures they act on; a Work-flow Object dealing with only one particular work-flow would need a much reduced set of logical data structures and corresponding methods, which could, as a consequence have shorter names. For example, a Work-flow Object dealing with Work-flow 3a would only need to have one type of output MDT. In the Main Object there are six types of MDT output, all of which must have different names so that the user has a record of which work-flow was used to produce them. It would be useful for two or more Work-flow Objects to be able to share the same internal data store in order to avoid unnecessary duplication of data on disk. Work-flow Objects are not discussed further in this document, but this approach should be considered for implementation in the first release of GUT.

The command interface includes methods for preference selection, data pre-processing and carrying out work-flow calculations. Preference selection tasks are accomplished by assigning values to attributes. Most data pre-processing occurs when data from external sources is imported into the internal data store using `Import` methods. Work-flow calculations are initiated by executing the `Calc` method associated with the desired work-flow output data structure. The main preference selection tasks involve grid parameters, the maximum degree and order of SH coefficients, the reference ellipsoid and the tide system. These are discussed separately below.

Grid specification

The Spatial Object attributes `LatMin, LatMax, LonMin, LonMax, LatCell` and `LonCell` are used to define a regular grid. If all these values are defined then the Spatial Object represents a regular grid, and the value of Spatial Object attribute `GridType` is REGULAR. If `LatCell` and `LonCell` are undefined (i.e. set to `None`, Python's null value) then the Spatial Object represents an irregular or unstructured grid, and the value of the `GridType` attribute is UNSTRUCTURED. If they are all undefined then the Spatial Object represents a set of positions, and the value of the `GridType` attribute is LIST. There are three ways in which these attributes can be changed.

1. Creating a new Spatial Object from scratch using the Spatial Object `New()` method. The Spatial Object is over-written with the default attributes and data (default MSSH and grid).
2. Importing gridded data into a blank Spatial Object. A blank Spatial Object has all its attributes and data set to None, and can be created using the Spatial Object `Blank()` method. Data can be imported into any field. If the file's metadata contains all seven grid specification parameters then these are used to define the values the grid specification attributes. If one or more parameters are missing from the metadata then the `Import` method prompts the user to enter the value.
3. Their values can be changed using the `DefineGrid()` method, which requires the user to specify all seven values, any of which can be specified as None. If data are already present in the Spatial Object when these attributes are changed, grid adaptation of that data takes place.

Maximum degree and order for SH synthesis

This task is referred to as Functional Algorithm FA01 in the WP3000 report. The maximum degree and order for SH synthesis is governed by the value of Spatial Object attributes `MaxDegOrdPotential` and `MaxDegOrdSurface`. The former refers to the SH coefficients representing the gravity potential, and the latter refers to surface SH coefficients. There are three Spatial Object methods that can be used to change the value of `Spatial.MaxDegOrdPotential`.

a. `MaxDegOrdPotential`, for specifying the maximum degree and order directly
b. `MaxResKMPotential`, for specifying a resolution in kilometres.
c. `MaxResDegreesPotential`, for specifying a resolution in degrees latitude.

Three similar methods are provided for changing the value of `Spatial.MaxDegOrdSurface`.

Maximum degree and order for SH analysis

When converting a global, gridded field from geographical space to SH coefficients, the maximum degree and order of the resulting coefficients is governed by the value of Spectral Object attribute `MaxDegOrdSurface`. There are three Spectral Object methods that can be used to change the value of `Spectral.MaxDegOrdSurface`.

d. `MaxDegOrdSurface`, for specifying the maximum degree and order directly
e. `MaxResKMSurface`, for specifying a resolution in kilometres.
f. `MaxResDegreesSurface`, for specifying a resolution in degrees latitude.

Three similar Spectral Object methods are provided for changing the value of `Spectral.MaxDegOrdPotential`. This parameter would not normally be changed manually because it is a property of the SH coefficients of gravity potential. It is not used for SH analysis.

Reference ellipsoid specification

This task is part of Functional Algorithm FA02 in the WP3000 report. The parameters used to specify the reference ellipsoid are stored by the following Main Object attributes:

`Ellips_GM`, `Ellips_a`, `Ellips_gamma-a`, `Ellips_J2`, `Ellips_omega` and `Ellips_f`. These attributes are located in the Main Object because they apply to both the spatial domain (represented by the Spatial Object) and the spectral domain (represented by the Spectral Object). The attributes do not all have to be defined in order to specify the ellipsoid. Undefined ellipsoid parameters have the value `None`. There are two ways in which the reference ellipsoid can be changed.

1. A single Main Object method for changing all the ellipsoid attributes is provided. This is called `Ellips()`, and takes six parameters, some of which can be given as `None`. Unlike most attributes in GUT, the ellipsoid attributes can not be changed individually using methods with the same names as the attributes. This is because the attributes must be consistent with the data at all times, and changing a single "Ellipse" attribute could result in an invalid ellipsoid specification.
2. When creating a new Spatial Object by importing data from a file, the reference ellipsoid of the Main Object becomes that of the data being imported. The ellipsoid parameters are obtained from the file's metadata if possible, and if not the user is prompted to enter the missing parameters. Subsequent `Import` operations will result, if necessary, in the imported data being adapted to the reference system specified by the Main Object.

Tide System Specification

This task is part of Functional Algorithm FA03 in the WP3000 report. The tide system is specified by Main Object attribute `TideSys`. As with the reference ellipsoid, the `TideSys` attribute can be changed in either of two ways.

1. Using the `TideSys()` method in the Main Object
2. Creating a new Spatial Object by importing spatial data in to a blank Spatial Object. The tide system becomes that of the first field to be imported. Subsequent imports result in data being adapted to the existing tide system.

Many powerful features of the Python language will be available to the GUT user, but it is important to emphasise that it will be easy for novice users to access the GUT work-flows. The best way to illustrate this point is by giving an example sequence of commands. The following three commands, issued at the start of a GUT session, are all that are required in order to calculate geoid heights on the default grid using the default parameters, and to export those results to a file.

```
MyOb = GUT.New()
MyOb.CalcGeoidHeight()
MyOb.Spatial.ExportGeoidHeight("heights.dat", "height_field", GRAVSOFT)
```

Below are some examples of Python command sequences for carrying out some of the work-flows described in the WP3000 report. These examples are intended to give the reader an indication of how the command interface will look, but the exact commands are likely to change during the implementation phase of GUT.

Examples from work-flow 1a: Geoid and gravity field computation.

- Create default Main object with default data

```
MyOb = GUT.New()
```

- Load a saved Main Object from directory old_ob
  MyOb = GUT.Load("old_ob")

- Geoid heights
  MyOb.CalcGeoidHeight()
- Gravity deflections (E-W and N-S deflections are both calculated)
  MyOb.CalcDeflection()

- Create new Spatial Object with default grid and data (overwriting existing object)
  MyOb.Spatial.New()
- Create a new blank Spatial Object (overwriting existing Object) and specify grid parameters manually
  MyOb.Spatial.Blank()
  MyOb.Spatial.DefineGrid(min_lat, max_lat,…)
- Import grid or list of points from a file containing data for one of the input fields. Must start with *blank* Spatial Object, otherwise imported data will be adapted to existing grid and reference system.
  MyOb.Spatial.Blank()
  Myob.Spatial.ImportInputMSSH("mssh.cdf", "mssh_field", NETCDF)

- Create default 2-D plot of geoid heights
  MyOb.Spatial.ViewGeoidHeight()
- Export heights to a NetCDF file
  MyOb.Spatial.ExportGeoidHeight("heights.cdf", "height_field", NETCDF)

Examples from Workflow 1b: Error computation for geoid and gravity field

- Calculation of geoid height omission error variances
  MyOb.CalcGeoidHeightVarOm()
- Define pair of positions in Covar Object
  MyOb.Covar.AddPos(lat1, lon1)
  MyOb.Covar.AddPos(lat2, lon2)
- Calculate geoid height omission error covariances
  MyOb.CalcGeoidHeightCovOm()
- Export geoid height omission error variances
  MyOb.Spatial.ExportGeoidHeightVarOm("heightOmErr.cdf", "err_field", NETCDF)

Examples from Workflow 2: Sea surface height and a-priori MDT selection

- Import average SLA and ADT for specified period from time-series.
  MyOb. Spatial.ImportInputAverageSLA("monthly_sla_timeseries.cdf", start_date, end_date)
  MyOb. Spatial.ImportInputAverageADT("daily_adt_timeseries.cdf", start_date, end_date)
- Calculate MSSH as average SLA + Reference MSSH
  MyOb.CalcMSSH_Average()
- Calculate MDT as average SLA + Reference MDT
  MyOb.CalcMDT_Average()

Examples from Workflow 3a: Satellite Dynamic Topography computation in geographical space

- Calculate MDTS using spatial domain method using default MSSH field and default filter.
```
MyOb.Spatial.ImportInputGeoidHeight("geoid.cdf", "height_field", NETCDF)
MyOb.CalcMDTS_Spatial()
```

Examples from Workflow 4b: Remove-Restore combined technique A: spectral filtering

- Calculate the Satellite MDT using work-flow 3b, then export the output
```
MyOb.CalcMDTS_Spectral()
MyOb.Spatial.ExportMDTS_Spectral("temp_mdts.cdf", "mdt_field", NETCDF)
```
- Import the MDTS calculated above
```
MyOb.Spatial.ImportInputMDTS("temp_mdts.cdf", "mdt_field", NETCDF)
```
- Calculate MDTC by Spectral RR A and spatial filtering, using the output of work-flow 3b as the input MDTS
```
MyOb.CalcMDTC_A-Spectral()
```

After reviewing the example commands above, the reader may observe that many of the commands are rather long, giving the mistaken impression that a lot of typing will be required in order to use GUT. Method names such as `"ImportInputLandHeight"` and `"CalcGeoidHeightVarCom"` may seem excessively long, but there are good reasons for having names of this form. The names will be designed specifically for use with a *command completion* facility in Python, which behaves in the same way as file and directory name completion in Linux and Unix shell environments. This is implemented using the `rlcompleter` package in Python. Command completion ensures that GUT users do not need to type out long commands in their entirety. Furthermore, the command completion facility allows the user to select from a list of appropriate methods available for a particular object. The design of the method names allows the user to find and enter the desired command in the minimum number of steps with the minimum amount of typing. For users who prefer a more traditional console application, the following methods launch a command based wizard that allows the user to select from numbered lists of possible actions.

- `Calc()`
- `Convert()`
- `Filter()`
- `Import()` and `Export()`
- `Delete()`
- `View()`

Not all the types of object have all these methods.

Another advantage of longer method names is that they make Python scripts more readable. Python scripts can be written using a conventional text editor or with an integrated development environment (IDE) such as Eclipse [39] or NetBeans [40], many of which allow the programmer to select an appropriate method or variable name from a list. However, IDEs are sophisticated, powerful programmers' tools that may not appropriate for all GUT users.