A concept for the estimation of high degree gravity field models in a high performance computing environment

Jan Martin Brockmann · Lutz Roese-Koerner · Wolf-Dieter Schuh

Received: date / Accepted: date

Abstract The determination of global gravity field models of the Earth from the combination of complementary observations within a joint adjustment is a computational expensive task. Global Earth's gravity field models are often parameterized by a finite series of spherical harmonic base functions. The effort depends on the one hand on the maximal resolution of the spherical harmonic expansion and on the other hand on the amount of observations and their stochastic characteristics. Within this contribution a concept for the rigorous estimation of high degree gravity models above degree and order 2 000 is presented. Iterative solvers implemented in a high performance computing environment using several thousands compute cores are used to derive the rigorous least-squares solution from millions of observations for more than 4 000 000 unknown parameters. A flexible design was implemented to process an arbitrary number of observation groups. For each group a variance component can be estimated to derive

J.M. Brockmann
University of Bonn, Institute of Geodesy and Geoinformation, Nussallee 17, 53115 Bonn, Germany
E-mail: brockmann@geod.uni-bonn.de

L Roese-Koerner
E-mail: roese-koerner@geod.uni-bonn.de

W.-D. Schuh
E-mail: schuh@geod.uni-bonn.de

a data adaptive weighting factor. The combined solution is derived in a weighted joint estimation from all observation groups which might be preprocessed normal equations (e.g. from the dedicated gravity field missions like CHAMP, GRACE or GOCE) or observations like data sets of point-wise measured terrestrial gravity field information. The concept of the implemented solver is demonstrated. A small scale closed loop simulation serves as proof of concept.

Keywords high performance computing, iterative solvers, global gravity field recovery, high degree spherical harmonics

1 Introduction

Global Earth's gravity field models are often described as a finite series of spherical harmonic coefficients, (e.g. Heiskanen and Moritz, 1993, p. 59)

$$V\left(r,\theta,\lambda\right)=\frac{GM}{a}\sum_{l=0}^{l_{\max}}\left(\frac{a}{r}\right)^{l+1}\sum_{m=0}^{l}\left(c_{lm}\cos\left(m\lambda\right)+s_{lm}\sin\left(m\lambda\right)\right)P_{lm}\left(cos\theta\right), \qquad (1)$$

where l and m denote the spherical harmonic degree and order, c_{lm} and s_{lm} the coefficients of the spherical harmonic series, a the equatorial radius of the Earth reference ellipsoid, $P_{lm}\left(\cdot\right)$ the fully normalized associated Legendre functions, and GM the gravitational constant of the Earth. $l_{\max}$ is the degree of expansion which defines the spatial resolution. In general, three types of global gravity field models can be distinguished, depending on the observations used and the resolution the models can be resolved to.

The first class consists of the satellite-only gravity field models. Here a consistent gravity field models are derived from the observations from a single dedicated satellite mission like CHAMP (CHAllenging Mini-Satellite Payload, Reigber et al, 2002), GRACE (GRAvity recovery and Climate Experiment, Tapley et al, 2004) or GOCE (Gravity field and steady-state Ocean Circulation Explorer, ESA, 1999). Each of the models reflects the particular strengths of the satellites observation techniques. The observations from a single mission are analyzed, spending a lot of effort in the functional and the stochastic modeling (e.g. Beutler et al, 2010; Mayer-Gürr et al, 2010b; Pail et al, 2011). Nowadays, in addition to the coefficients realistic covariance matrices are provided (see for GRACE and GOCE e.g. Mayer-Gürr et al, 2010a,b; Pail et al, 2011), due to a lot of effort in stochastic modeling. The individual satellites are only sensitive to the long and medium wavelength of the gravity field. Therefore, the models are only resolved to a limited spherical harmonic degree of currently about 250–270 for the GOCE mission (Brockmann et al, 2013a; Bruinsma et al, 2013). For the processing of the satellite-only models, the computational challenge does not result from the number of parameters to be estimated but from the huge number of observations (e.g. several hundreds of millions for GRACE or GOCE over several years) and the complex stochastic properties of the data (see e.g. Schuh et al, 2010; Rummel et al, 2011). Nevertheless, iterative solvers are often used to derive solutions fast (Baur, 2009; Xie, 2005) e.g. for parameter tuning (Brockmann et al, 2010). The final model is then often setup via the assembly and solution of full normal equations (NEQs Pail and Plank, 2003; Bruinsma et al, 2013) to derive in addition the full error covariance matrix.

The second class consists of combined satellite-only gravity field models, which combine the models of the different satellites e.g. on the level of normal equations. As the resolution of those models is limited again by the resolution of the satellite models, the normal equations are quiet small and not bigger than 40 Gigabyte. Thus, for this models, the computational challenges are limited (Pail et al, 2010; Bruinsma et al, 2013; Farahani et al, 2013).

The third class is the class of high resolution combined gravity field models. In addition to satellite observations (in form of their NEQs) also terrestrial measurements like gravity anomalies on land and altimetry over the oceans are taken into account. Using these measurements, which are sensitive to higher degrees, the resolvable resolution considerably increases. Current combined models are available for degree and order (d/o) 360 (GIF48, Ries et al, 2011), 1 949 (EIGEN6C, Förste et al, 2008, 2012) and 2 190 (EGM2008, Pavlis et al, 2012) which corresponds to the number of parameters in the range of 130 000 to 4.8 mio. Whereas the GIF48 model was computed via the assembly and solution of full normal equations (Ries et al, 2011), the higher degree models were computed introducing approximations. EIGEN6C was computed via averaging a full normal equation solution from degree and order 2 to 370 and a blockdiagonal solution for the gridded high resolution data from 2 to degree 1 949 (Förste et al, 2012). EGM2008 was computed from an entire block-diagonal solution. Also the involved satellite NEQs were approximated as a block-diagonal form (Pavlis et al, 2012). Nevertheless, there are some studies, which demonstrate, that the assembly and solution of full normal equations is possible in reasonable time up to d/o 600–720 (Fecher et al, 2011; Brockmann et al, 2013b).

The goal of this study is to explain an implementation of a massive parallel iterative solver which is capable to compute the rigorous least-squares solution for gravity field models of at least d/o 2 000. Standard concepts from high performance and scientific computing are used to derive a portable implementation. Section 2 summerizes the implemented iterative solver with a special focus on the integration of variance component estimation (VCE) to derive data adaptive weights for the observation groups. Section 3 describes the computational tasks and challenges which have to be solved for a massive parallel implementation. A small scale closed loop simulation is performed in Section 4 to verify and to analyze the implementation. A summary and some conclusions are given in Section 5.

2 Implemented Solver

The goal of this section is to review the implemented solver. The algorithms are summarized and adapted for the implementation in a HPC environment. The basic concepts, i.e. the general model, the solver itself and variance component estimation (VCE) to determine the unknown weights of the complementary observation groups within the iterative solver are summarized.

2.1 Basic setup of the model

The basic iterative solver implemented – the preconditioned conjugate gradient solver (Hestenes and Stiefel, 1952) – is well understood and well documented. The solver was already chosen by Schuh (1996) and tailored for the processing of GOCE observations as *Preconditioned Conjugate Gradient Multiple Adjustment* (PCGMA, Schuh, 1996). That implementation was optimized for the requirements of GOCE processing, i.e. the limited spherical harmonic resolution and the huge number of auto-correlated observations. The implementation introduced here should be more flexible and – most important – should be able to derive rigorous solutions for millions of unknown parameters. Nevertheless, the description and summary of the basic algorithm is close to Schuh (1996) and Boxhammer (2006).

2.1.1 Solution of combined system

The goal of the solver is to find the least-squares solution for the unknown spherical harmonic parameters arranged in the vector $\mathbf{x}$. In general, it is assumed that several observation groups should be combined, which might be available as raw observations (OEQs) plus an error description or as preprocessed (band-limited) normal equations (NEQs) which are assumed to be sufficient statistics of the original observations. So on the one hand there is a set of (linearized) observation equations for different groups $o \in \{1, \ldots, O\}$

$$\boldsymbol{\ell}_o + \mathbf{v}_o = \mathbf{A}_o \mathbf{x} \text{ and } \boldsymbol{\Sigma}_{\boldsymbol{\ell}_o \boldsymbol{\ell}_o} = \sigma_o^2 \mathbf{Q}_{\boldsymbol{\ell}_o \boldsymbol{\ell}_o}, \tag{2}$$

where $\mathbf{A}_o$ are the design matrices, $\boldsymbol{\ell}_o$ the vectors of observations, σ_o^2 are the unknown variance components and $\mathbf{Q}_{\boldsymbol{\ell}_o \boldsymbol{\ell}_o}$ the cofactor matrices which are assumed to be known in this context (e.g. derived from empirically estimated covariance functions).

A second set of observation groups $n \in \{1, \ldots, N\}$ may be a set of observation groups which were already preprocessed and are available in form of their normal equations, e.g. the normal equations of gravity field mission derived from a single satellite mission like CHAMP, GRACE or GOCE. These data sets are analyzed by experts in the individual missions and are sometimes provided as a solution vector and an unconstrained full covariance matrix. These might be used to directly recover the normal equations. These normal equations (or covariance matrix plus solution vector) are sufficient statistics (e.g. Koch, 1999, p. 150) of the original observations, assuming a realistic functional and stochastic model in their assembly. These datasets consists of NEQs

$$\frac{1}{\sigma_n^2} \mathbf{N}_n \mathbf{x} = \frac{1}{\sigma_n^2} \mathbf{n}_n, \tag{3}$$

$$\Leftrightarrow \frac{1}{\sigma_n^2} \boldsymbol{\Sigma}_n^{-1} \mathbf{x} = \frac{1}{\sigma_n^2} \boldsymbol{\Sigma}_n^{-1} \mathbf{x}_n \text{ and in addition } n_n, \boldsymbol{\ell}_n^T \mathbf{Q}_{\boldsymbol{\ell}_n \boldsymbol{\ell}_n}^{-1} \boldsymbol{\ell}_n, \tag{4}$$

where $\mathbf{N}_n = \boldsymbol{\Sigma}_n^{-1}$ are the preprocessed normal matrices and $\mathbf{n}_n = \boldsymbol{\Sigma}_n^{-1}\mathbf{x}_n$ the preprocessed normal vectors provided. For the use of VCE, to determine the unknown variances σ_n^2, the number of observations n_n used in the assembly of the NEQs and the product $\boldsymbol{\ell}_n^T \mathbf{Q}_{\boldsymbol{\ell}_n\boldsymbol{\ell}_n}^{-1} \boldsymbol{\ell}_n$ have to be known, too.

The joint solution $\mathbf{x}$ determined from all observation groups would yield the combined system of normal equations

$$\left(\sum_{n=1}^{N} \frac{1}{\sigma_n^2} \mathbf{N}_n + \sum_{o=1}^{O} \frac{1}{\sigma_o^2} \mathbf{A}_o^T \mathbf{Q}_{\boldsymbol{\ell}_o\boldsymbol{\ell}_o}^{-1} \mathbf{A}_o \right) \mathbf{x} = \sum_{n=1}^{N} \frac{1}{\sigma_n^2} \mathbf{n}_n + \sum_{o=1}^{O} \frac{1}{\sigma_o^2} \mathbf{A}_o^T \mathbf{Q}_{\boldsymbol{\ell}_o\boldsymbol{\ell}_o}^{-1} \boldsymbol{\ell}_o, \tag{5}$$

$$\mathbf{N}\mathbf{x} = \mathbf{n} \tag{6}$$

where $\mathbf{N}$ and $\mathbf{n}$ are the normal matrices and normal vector of the weighted and combined normal equations. These can be solved to determine the unknown spherical harmonic parameters in $\mathbf{x}$ from all groups. For a limited spherical harmonic resolution the assembly of $\mathbf{N}$ and $\mathbf{n}$ and the solution via Cholesky reduction is directly possible. E.g. Brockmann et al (2013b) demonstrated this for spherical harmonics up to d/o 720 (520 000 unknowns, size of $\mathbf{N}$ is 2 TB) in less then 2 hours on modern supercomputers ($\approx$ 8 000 cores). Nevertheless, increasing the resolution to e.g. d/o 2 000 (4 000 000 unknowns, size of $\mathbf{N}$ is 116 TB), the direct solution technique, although implemented on a supercomputer, becomes unreasonable. Thus, the next section will summarize the iterative solver which is used to estimate $\mathbf{x}$ for problem sizes of spherical harmonic d/o of up to 2 000 without approximations. The iterative solvers avoid the assembly of $\mathbf{N}$ and especially the computation of $\mathbf{N}_o = \mathbf{A}_o^T \mathbf{Q}_{\boldsymbol{\ell}_o\boldsymbol{\ell}_o}^{-1} \mathbf{A}_o$. These are the computationally demanding tasks, as the observation groups o are assumed to contain the complete signal and thus all $l_{\max}^2$ parameters, which are about $4-5$ mio. within the aspired setup.

2.1.2 Estimation of weights as variance components

Combining complementary data types to determine the joint least-squares solution requires to address the question of weighting the individual observation groups. Thus, in addition to the unknown solution $\mathbf{x}$, an individual weight w_i for each group should be estimated from the data. A data adaptive concept within satellite geodesy was developed by Koch and Kusche (2002) who interpreted the weights within a data combination procedure as inverse variance components (Förstner, 1979). Whereas Koch and Kusche (2002) used VCE to determine a regularization parameter, the weight of the prior information, VCE can be used in addition to derive weights for an arbitrary number of complementary observation groups. Thus, the weights $w_i = 1/\sigma_i^2$ for each group $i \in \{n, o\}$ can be determined via VCE. Following again Koch and Kusche (2002) and Koch (1999) they can be iteratively determined via

$$\sigma_i^{(\eta)2} = \frac{\Omega_i^{(\eta)}}{n_i - m_i^{(\eta)}} = \frac{\Omega_i^{(\eta)}}{r_i^{(\eta)}} \tag{7}$$

where n_i is the number of observations in group i, $\Omega_i^{(\eta)}$ the weighted sum of squared residuals after iteration η

$$\Omega_i^{(\eta)} = \mathbf{v}_i^{(\eta)T} \mathbf{Q}_{\boldsymbol{\ell}_i\boldsymbol{\ell}_i}^{-1} \mathbf{v}_i^{(\eta)} = \left(\mathbf{A}_i \mathbf{x}^{(\eta)} - \boldsymbol{\ell}_i\right)^T \mathbf{Q}_{\boldsymbol{\ell}_i\boldsymbol{\ell}_i}^{-1} \left(\mathbf{A}_i \mathbf{x}^{(\eta)} - \boldsymbol{\ell}_i\right) \tag{8}$$

$$= \mathbf{x}^{(\eta)T} \mathbf{N}_i \mathbf{x}^{(\eta)} - 2\mathbf{x}^{(\eta)T} \mathbf{n}_i + \boldsymbol{\ell}_i^T \mathbf{Q}_{\boldsymbol{\ell}_i\boldsymbol{\ell}_i}^{-1} \boldsymbol{\ell}_i \tag{9}$$

and $r_i^{(\eta)}$ the partial redundancy (e.g Koch and Kusche, 2002)

$$r_i^{(\eta)} = n_i - \frac{1}{\sigma_i^{(\eta)2}} \text{trace}\left(\mathbf{N}^{-1}\mathbf{N}_i\right) = n_i - \frac{1}{\sigma_i^{(\eta)2}} \text{trace}\left(\mathbf{N}^{-1}\mathbf{A}_i^T \mathbf{Q}_{\boldsymbol{\ell}_i\boldsymbol{\ell}_i}^{-1} \mathbf{A}_i\right). \tag{10}$$

To handle large systems, Koch and Kusche (2002) proposed a stochastic trace estimator to determine the trace in (10). Within this context, as a preparation for the iterative solver, where $\mathbf{N}$ is unknown, it is required to use the stochastic estimator as well. According to Hutchinson (1990), the estimator

$$\text{trace}\left(\mathbf{B}\right) = E\left\{\boldsymbol{\mathcal{P}}^T \mathbf{B} \boldsymbol{\mathcal{P}}\right\}, \qquad \boldsymbol{\mathcal{P}} \sim \mathcal{U}\left(-1, 1\right), \text{ as discrete uniform distribution} \tag{11}$$

has minimal variance if $\mathbf{B}$ is symmetric. Thus (10) can be rewritten, to obtain symmetric matrices in the trace term. We have to distinguish two cases: the groups as OEQs and the groups available as NEQs.

Partial redundancy for groups of NEQs: Introducing the Cholesky decomposition $\mathbf{N}_n = \mathbf{R}_n^T \mathbf{R}_n$, the partial redundancy can be reformulated as

$$r_n = n_n - \frac{1}{\sigma_n^2} \text{trace}\left(\mathbf{N}^{-1}\mathbf{N}_n\right) = n_n - \frac{1}{\sigma_n^2} \text{trace}\left(\mathbf{N}^{-1}\mathbf{R}_n^T \mathbf{R}_n\right) = n_n - \frac{1}{\sigma_n^2} \text{trace}\left(\mathbf{R}_n \mathbf{N}^{-1} \mathbf{R}_n^T\right), \tag{12}$$

as the trace is invariant under cyclic permutations (e.g. Koch, 1999, p. 40). Due to its quadratic form $\mathbf{R}_n\mathbf{N}^{-1}\mathbf{R}_n^T$ is symmetric. Inserting the stochastic estimation from (11) the partial redundancy is

$$r_n = n_n - \frac{1}{\sigma_n^2} E\left\{\boldsymbol{\mathcal{P}}_n^T \mathbf{R}_n \mathbf{N}^{-1} \mathbf{R}_n^T \boldsymbol{\mathcal{P}}_n\right\}, \text{with } \bar{\boldsymbol{\mathcal{P}}}_n := \mathbf{R}_n^T \boldsymbol{\mathcal{P}}_n \tag{13}$$

$$r_n = n_n - \frac{1}{\sigma_n^2} E\left\{\bar{\boldsymbol{\mathcal{P}}}_n{}^T \mathbf{N}^{-1} \bar{\boldsymbol{\mathcal{P}}}_n\right\}. \tag{14}$$

Inserting k realizations of $\boldsymbol{\mathcal{P}}_n$ instead of the expectation value arranged as columns in the matrix $\mathbf{P}_n$ and applying the same transformation $\bar{\mathbf{P}}_n = \mathbf{R}_n^T \mathbf{P}_n$ the estimate for the partial redundancy as mean value from the k samples is

$$r_n = n_n - \frac{1}{\sigma_n^2} \frac{1}{k} \text{trace}\left(\bar{\mathbf{P}}_n^T \mathbf{N}^{-1} \bar{\mathbf{P}}_n\right). \tag{15}$$

For readability, the iteration indexes $^{(\eta)}$ and $^{(\eta-1)}$ are omitted.

Partial redundancy for groups of OEQs: Introducing the Cholesky decomposition $\mathbf{Q}^{-1}_{\boldsymbol{\ell}_o \boldsymbol{\ell}_o} = \mathbf{G}_o^T \mathbf{G}_o$, the partial redundancy from (10) can be rewritten as

$$r_o = n_o - \frac{1}{\sigma_o^2} \text{trace}\left(\mathbf{N}^{-1} \mathbf{A}_o^T \mathbf{G}_o^T \mathbf{G}_o \mathbf{A}\right) = n_o - \frac{1}{\sigma_o^2} \text{trace}\left(\mathbf{G}_o \mathbf{A} \mathbf{N}^{-1} \mathbf{A}_o^T \mathbf{G}_o^T\right), \tag{16}$$

with $\mathbf{G}_o \mathbf{A} \mathbf{N}^{-1} \mathbf{A}_o^T \mathbf{G}_o^T$ being symmetric. Thus, again inserting the stochastic trace estimator yields

$$r_o = n_o - \frac{1}{\sigma_o^2} E\left\{\boldsymbol{\mathcal{P}}_o^T \mathbf{G}_o \mathbf{A} \mathbf{N}^{-1} \mathbf{A}_o^T \mathbf{G}_o^T \boldsymbol{\mathcal{P}}_o\right\}, \text{with } \bar{\boldsymbol{\mathcal{P}}}_o := \mathbf{A}_o^T \mathbf{G}_o^T \boldsymbol{\mathcal{P}}_o \tag{17}$$

$$r_o = n_o - \frac{1}{\sigma_o^2} E\left\{\bar{\boldsymbol{\mathcal{P}}}_o{}^T \mathbf{N}^{-1} \bar{\boldsymbol{\mathcal{P}}}_o\right\}. \tag{18}$$

Replacing again the expectation value by k realizations arranged in the matrix $\mathbf{P}_o$ and applying the same transformation $\bar{\mathbf{P}}_o = \mathbf{A}_o^T \mathbf{G}_o^T \mathbf{P}_o$ the estimate for the partial redundancy is

$$r_o = n_o - \frac{1}{\sigma_o^2} \frac{1}{k} \text{trace}\left(\bar{\mathbf{P}}_o^T \mathbf{N}^{-1} \bar{\mathbf{P}}_o\right), \tag{19}$$

already introducing the mean value of all realizations. These representation of VCE can be integrated in the iterative solver as demonstrated in Sect. 2.2.2.

2.2 PCGMA including variance component estimation

For the combination of normal and observation equations the *Preconditioned Conjugate Gradient Multiple Adjustment* (PCGMA) algorithm was proposed by Schuh (1995, 1996) as a connection of the conjugate gradient algorithm proposed by Hestenes and Stiefel (1952) and the extension for adjustment problems by Schwarz (1970). This section is used to summarize the basic (serial) PCGMA algorithm and extend it by VCE following the line of thought of Alkhatib (2007).

2.2.1 Basic PCGMA algorithm

The basic PCGMA algorithm is summarized in Alg. 1. The expensive computations of $\mathbf{N}$, especially $\mathbf{N}_o = \mathbf{A}_o^T \mathbf{Q}_{\boldsymbol{\ell}_o\boldsymbol{\ell}_o}^{-1} \mathbf{A}_o$, are avoided, instead only a few matrix vector multiplications have to be performed in each iteration (especially in the computations of $\mathbf{h}^{(\nu)}$). The result of the algorithm is a least-squares estimate of the unknown parameters $\mathbf{x}$ after ν iterations. In contrast to direct solution methods, a covariance matrix of the parameters can only be computed applying Monte Carlo techniques (cf. Alkhatib and Schuh, 2007). As the expected convergence rate heavily depends on the condition of $\mathbf{N}$ and gravimetric problems tend to be ill-conditioned (e.g. Schwintzer et al, 1997; Ilk et al, 2002), one essential step for the convergence of the PCGMA algorithm is the design of a preconditioner. A preconditioner $\mathbf{N}_\oplus$, a sparse approximations of $\mathbf{N}$, is used to accelerate the convergence. This preconditioner should have the following properties: i) it should reflect the main characteristics of $\mathbf{N}$, ii) it should be sparse and easily computable and iii) the decomposition (e.g. Cholesky for the preconditioning step) should be sparse as well. Whereas very special preconditioners might be designed for very individual problems and setups (e.g. Boxhammer and Schuh, 2006), already quiet simple designs can considerably improve the convergence rate of general spherical harmonic analyses. As shown in Boxhammer (2006, p. 68) a simple block-diagonal preconditioner can be a very efficient general approximation and significantly accelerates the PCGMA convergence.

Algorithm 1: PCGMA algorithm following Schuh (1996).

Data:

$\mathbf{N}_{\oplus}$	sparse preconditioning matrix	$\nu_{\max}$	maximal number of PCGMA iterations
$\mathbf{A}_o$	design matrix $o \in \{1...O\}$	$\boldsymbol{\ell}_o$	vector of observations $o \in \{1...O\}$
$\mathbf{Q}_{\boldsymbol{\ell}_o\boldsymbol{\ell}_o}$	cofactor matrix of observations $o \in \{1...O\}$	w_o	weight for observation group $o \in \{1...O\}$
$\mathbf{N}_n$	normal matrix for group $n \in \{1...N\}$	$\mathbf{n}_n$	right hand side of NEQ for $n \in \{1...N\}$
w_n	weight for observation group $n \in \{1...N\}$	$\mathbf{x}^{(0)}$	initial values for parameters

```
$\mathbf{r}^{(0)} = \sum_n^N w_n \left(\mathbf{N}_n \mathbf{x}^{(0)} - \mathbf{n}_n\right) + \sum_o^O w_o \mathbf{A}_o^T \left(\mathbf{Q}_{\mathbf{l}_o\mathbf{l}_o}^{-1} \mathbf{A}_o \mathbf{x}^{(0)} - \boldsymbol{\ell}_o\right)$ // compute initial joint residuals
$\boldsymbol{\rho}^{(0)} = \text{solve}\left(\mathbf{N}_{\oplus}, \mathbf{r}^{(0)}\right)$ // apply preconditioner to residuals
$\boldsymbol{\pi}^{(0)} = -\boldsymbol{\rho}^{(0)}$ // initial search direction
// PCGMA - iterations
for $\nu = 0$ to $\nu_{\max} - 1$ do
    if $\nu > 0$ then
        // update step for search direction
        $e_\nu = \frac{\mathbf{r}^{(\nu)T} \boldsymbol{\rho}^{(\nu)}}{\mathbf{r}^{(\nu-1)T} \boldsymbol{\rho}^{(\nu-1)}}$
        $\boldsymbol{\pi}^{(\nu)} = -\boldsymbol{\rho}^{(\nu)} + e_\nu \boldsymbol{\pi}^{(\nu-1)}$
    end
    $\mathbf{h}^{(\nu)} = \sum_n^N w_n \mathbf{N}_n \boldsymbol{\pi}^{(\nu)} + \sum_o^O w_o \mathbf{A}_o^T \mathbf{Q}_{\mathbf{l}_o\mathbf{l}_o}^{-1} \mathbf{A}_o \boldsymbol{\pi}^{(\nu)}$
    // update step for parameters
    $q^{(\nu)} = \frac{\mathbf{r}^{(\nu)T} \boldsymbol{\rho}^{(\nu)}}{\boldsymbol{\pi}^{(\nu)T} \mathbf{h}^{(\nu)}}$
    $\mathbf{x}^{(\nu+1)} = \mathbf{x}^{(\nu)} + q^{(\nu)} \boldsymbol{\pi}^{(\nu)}$
    // update step for residuals
    $\mathbf{r}^{(\nu+1)} = \mathbf{r}^{(\nu)} + q^{(\nu)} \mathbf{h}^{(\nu)}$
    // apply preconditioner to residuals
    $\boldsymbol{\rho}^{(\nu+1)} = \text{solve}\left(\mathbf{N}_{\oplus}, \mathbf{r}^{(\nu+1)}\right)$
end
return $\mathbf{x}^{(\nu_{\max})}$
```

2.2.2 VCE in iterative solvers

The basic PCGMA algorithm from Alg. 1 may be extended for a VCE for an arbitrary number of of variance components. As within iterative solvers $\mathbf{N}$ is not computed, but needed in VCE standard form (10), the Monte Carlo based method is used as introduced in (15) and (19) (cf. Alkhatib, 2007; Brockmann and Schuh, 2010).

Whereas Ω_i can be directly computed as for the direct solver, the computation of the partial redundancies, especially the trace term in (15) and (19), has to be adapted and integrated into the iterative procedure. Especially, the computation of the trace term $\bar{\mathbf{P}}_i^T \mathbf{N}^{-1} \bar{\mathbf{P}}_i$ for both groups n and o . Defining

$$\mathbf{Z}_i := \mathbf{N}^{-1} \bar{\mathbf{P}}_i, \qquad \Leftrightarrow \mathbf{N}\mathbf{Z}_i = \bar{\mathbf{P}}_i, \tag{20}$$

$\mathbf{Z}_i$ can be directly determined via solving the system of normal equations with the PCGMA for additional right hand sides $\bar{\mathbf{P}}_i$. The partial redundancies are then determined via

$$r_i = n_i - \frac{1}{\sigma_i^2}\frac{1}{k}\text{trace}\left(\bar{\mathbf{P}}_i^T\mathbf{Z}_i\right). \tag{21}$$

Including this steps into Alg. 1, yields the complete PCGMA algorithm with VCE shown in Alg. 2.

The intention of Alg. 2 is to given a quiet simple overview about the implemented algorithm. Thus, so far, a major point was not addressed and not taken into account in the algorithm. Not all normal equations are assembled for the same set of parameters, instead individual NEQs $\mathbf{N}_n$ are just assembled for a (small) subset of the parameters to be estimated. Thus the NEQs differ in size and thus in parameter ordering. This point will be addressed in the next section, in which the implementation on high performance computers is addressed and a concept for a massive parallel implementation is presented.

3 Computational aspects

Within this section, the massive parallel implementation of the PCGMA algorithm on a high-performance compute cluster is summarized. Standard concepts of scientific computing are used to implement the solver presented in Sect. 2. The goal is a flexible implementation, which is capable to operate on thousands of compute cores. Assume a cluster with $N = R \cdot C$ compute cores, which can be virtually arranged as a $R \times C$ Cartesian grid. These cores, belonging to a set of compute nodes, are connected via a network device. Furthermore it is assumed, that each of the compute cores has only a direct access to its own local main memory. Thus, if data stored in the memory of a specific core is needed by one of the other cores, the data has to be shared via data communication over the network.

Within high-performance computing (HPC), the Message Passing Interface (MPI, MPI-Forum, 2009) standard exists. Different implementations of this standard (e.g. Gabriel et al, 2004; Balaji et al, 2013) allow to start one process of the application on each core and provide dedicated send and receive operations to communicate data along compute cores/processes. In combination of a MPI library and extensions like the Basic Linear Algebra Communication Subprograms library (BLACS, Dongarra and

Whaley, 1997) besides the communication (i.e. message passing routines), the set up of Cartesian grids for the cores is provided. Coordinates are assigned to each individual core (or the process of the application running on that core) such that each of the N cores can be addressed via Cartesian coordinates (r, c) with $r \in \{0, \ldots, R-1\}$ and $c \in \{0, \ldots, C-1\}$.

3.1 General design

The set up and use of MPI and BLACS is a prerequisite for the use of block-cyclic distributed matrices. The concept of distributing large matrices block-cyclically along a Cartesian grid of compute cores is a (quasi-) standard in HPC (Sidani and Harrod, 1996; Blackford et al, 1997, Chap. 4). As this concept is used for all matrices and vectors in the implementation of the PCGMA algorithm, the concept is briefly described. For details, as they are very technical, the reader should refer to Blackford et al (1997).

Given a general dense matrix $\mathbf{A}^g$, the whole matrix (often called global matrix) is divided into blocks of an arbitrary block size $b_r \times b_c$,

$$\mathbf{A}^g = \begin{bmatrix} \mathbf{A}^g_{0,0} & \mathbf{A}^g_{0,1} & \cdots & \mathbf{A}^g_{0,J-1} \\ \mathbf{A}^g_{1,0} & \mathbf{A}^g_{1,1} & \cdots & \mathbf{A}^g_{1,J-1} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{A}^g_{I-1,0} & \mathbf{A}^g_{I-1,1} & \cdots & \mathbf{A}^g_{I-1,J-1} \end{bmatrix}. \tag{22}$$

Except the matrix blocks $\mathbf{A}^g_{I-1,*}$ and $\mathbf{A}^g_{*,J-1}$, all blocks are of dimension $b_r \times b_c$. $\mathbf{A}^g_{I-1,*}$ and $\mathbf{A}^g_{*,J-1}$ might be of a smaller dimension. These blocks are now cyclically distributed along the Cartesian core grid of dimension $R \times C$. The blocks of the first row $(i = 0)$ are distributed alternating to the processors of the first row of the core grid $(0, c)$. Block $\mathbf{A}^g_{0,j}$ is stored on processor with coordinates $(0, \mathrm{mod}\,(j, C))$. The second row of blocks $\mathbf{A}^g_{1,j}$ is distributed alternating along the second row of the core grid. The general rule can be written as $\mathbf{A}^g_{i,j} \mapsto (\mathrm{mod}\,(i, R), \mathrm{mod}\,(j, C))$. If $I > R$, the Ith row will be distributed again over the first row of the processors grid. All matrix blocks mapped to a core (r, c) are arranged there in a local, serial stored, matrix

$$\mathbf{A}^{l}_{r,c} = \begin{bmatrix} \mathbf{A}^{g}_{i,j} & \mathbf{A}^{g}_{i,j+C} & \cdots \\ \mathbf{A}^{g}_{i+R,j} & \mathbf{A}^{g}_{i+R,j+C} & \cdots \\ \vdots & \vdots & \cdots \end{bmatrix} . \tag{23}$$

Thus, instead of operating on a global matrix, local operations have to be performed on $\mathbf{A}^{l}_{r,c}$, containing elements of the global matrix but not necessarily neighboring ones. It is important to realize, that neighboring elements from $\mathbf{A}^{g}$ occur only as neighboring elements in the local matrices $\mathbf{A}^{l}_{r,c}$ within the small sub blocks of size $b_r \times c_c$.

For matrices (and vectors) stored in this scheme, two libraries exist for basic matrix and linear algebra operations (Parallel Basic Linear Algebra Subprograms, PBLAS; Scalable Linear Algebra Package, ScaLAPACK Blackford et al, 1997) containing basic matrix computations. The whole algorithm presented in Sect. 2 is implemented on this block-cyclic distributed matrices, such that huge matrices can be used and operated on. The next subsection will give some details on the implementation using this distributed matrices.

3.2 Implementation of PCGMA in the HPC environment

The PCGMA algorithm summarized in Alg. 2 is directly implemented using the concept of block-cyclic distributed matrices for all matrices (and vectors) involved. The standard operations for matrices like additions, multiplications or solutions can be directly performed using PBLAS and ScaLAPACK libraries. Therfore, within this section, we focus on the implementation of $\mathbf{H}^{(\nu)}$ (see Alg. 2, l. 38), as it is the computationally most demanding step and some individual implementations are needed.

Numbering scheme for the PCGMA algorithm: As the observation equations have to be newly set up in each PCGMA iteration ν, the fast set up of the distributed matrix $\mathbf{A}_o$ is essential within the implementation of the iterative solver. Thus, using block-cyclically distributed matrices, the local matrices $\mathbf{A}^{l}_{o}$ should be locally computable without communication with other processes. In addition, redundant computations of several cores should be avoided, or at least minimized. Redundant computations might occur during the expensive recursive computations of the Legendre functions, which are performed by orders (e.g. Holmes and Featherstone, 2002). Thus, it is important, that all columns of $\mathbf{A}_o$ corresponding to coefficients of the same order, are within a single local matrix $\mathbf{A}_o^l$, or spread over as few local matrices as possible. Therefore, a numbering scheme depending on the parameters of the block-cyclic distribution (b_r and b_c) and the size of the compute grid (R and C) is determined at runtime.

The numbering scheme of the global matrix $\mathbf{A}_o$ is driven by the fact that the local matrices along a row of the processor grid $\hat{\mathbf{A}}_o = \left[\mathbf{A}_{r,0}^l \; \mathbf{A}_{r,1}^l \; \mathbf{A}_{r,2}^l \cdots \; \mathbf{A}_{r,C}^l\right]$ follow an order-wise numbering scheme. Thus, the global numbering scheme might look arbitrary at first view, but results from the fact, that parameters of the same order are collected as good as possible in the same local matrices. This minimizes redundant computation in the recursive computations of Legendre functions.

Reordering of block-cyclic distributed matrices: Choosing a configuration dependent numbering scheme allows that the matrix $\mathbf{A}_o$ can be efficiently computed, but results in a numbering scheme which differs to the fixed numbering scheme of the preconditioner ($\mathbf{N}_\oplus$ is only blockdiagonal in the order-wise numbering). Before applying the preconditioner, $\mathbf{R}^{(\nu)}$ has to be modified via a sequence of row interchanges to arrive – in view of the global matrix – in the order-wise ordering of the preconditioner. As the numbering scheme (parameter sequence) of $\mathbf{R}^{(\nu)}$ is configuration dependent, a reordering between two arbitrary numbering schemes needs to be performed.

A flexible method via a comparison and sorting of the symbolic parameters is used to derive an index vector from two general symbolic numbering schemes. Transforming the index vector to a serial sequence of permutations, a ScaLAPACK helping function for pivoting can be used to reorder rows and columns of a block-cyclic distributed matrix (`pdlapiv`).

Computation of the update vector for the OEQ groups: For the groups which are processed as OEQs, the fast and efficient set up of the observation equations is essential. As described above, using the special numbering scheme, Legendre functions can be computed recursively with only a few redundant computations on different cores. Thus, via the local (i.e. serial) computation of $\mathbf{A}^{l}_{r,l}$ the global matrix $\mathbf{A}_o$ is set up and the computation of $\mathbf{H}^{(\nu)}_n := w_o \mathbf{A}_o^T \mathbf{Q}^{-1}_{\mathbf{l}_o \mathbf{l}_o} \mathbf{A}_o \boldsymbol{\Pi}^{(\nu)}$ and the update step can be directly performed using PBLAS and ScaLAPACK routines. As $\mathbf{A}_o$ is set up directly in the proper numbering scheme, the update step $\mathbf{H}^{(\nu)} = \mathbf{H}^{(\nu)} + \mathbf{H}^{(\nu)}_o$ can be directly performed. Note that it is more efficient to set up $\mathbf{A}_o^T$ than $\mathbf{A}_o$, if column major ordering is used for serial matrix storage.

Computation of the update vector for the NEQ groups: The normal equations $\mathbf{N}_n$ are available in a specific parameter ordering and only for a small subset of the parameters. The number of parameters in $\mathbf{N}_n$ is called m_n. Nevertheless, the contribution $w_n \mathbf{N}_n \boldsymbol{\Pi}^{(\nu)}$ to the update vector has to be computed. Thus, the following steps have to be performed:

1. Perform a sequence of row interchanges of $\boldsymbol{\Pi}^{(\nu)}$ so that the parameter order of the first m_n rows correspond to the parameter order of $\mathbf{N}_n$, $\Rightarrow \tilde{\boldsymbol{\Pi}}^{(\nu)}$.
2. Perform the multiplication $\tilde{\mathbf{H}}^{(\nu)}_n := w_n \mathbf{N}_n \tilde{\boldsymbol{\Pi}}^{(\nu)}(1:m_n,:)$, where $\tilde{\boldsymbol{\Pi}}^{(\nu)}(1:m_n,:)$ is the subset of the first m_n rows and all columns.
3. Extend $\tilde{\mathbf{H}}^{(\nu)}_n$ by zeros for parameters not contained in group n. Reorder the rows of the matrix $\begin{bmatrix} \tilde{\mathbf{H}}^{(\nu)}_n & \mathbf{0} \end{bmatrix}^T$ into the numbering scheme of original $\boldsymbol{\Pi}^{(\nu)}$ to obtain $\mathbf{H}^{(\nu)}_n$.
4. Perform the update $\mathbf{H}^{(\nu)} = \mathbf{H}^{(\nu)} + \mathbf{H}^{(\nu)}_n$.

4 Closed loop simulation

To test the concept and to verify the implementation, a small scale closed loop simulation was performed. In this section the simulation is summarized and the runtime behavior of a PCGMA iteration is analyzed.

4.1 Simulated data

The EGM2008 (Pavlis et al, 2012) model was used to generate eighteen synthetic datasets. $N = 3$ datasets were created as normal equations. Real satellite normal equations from GRACE and GOCE

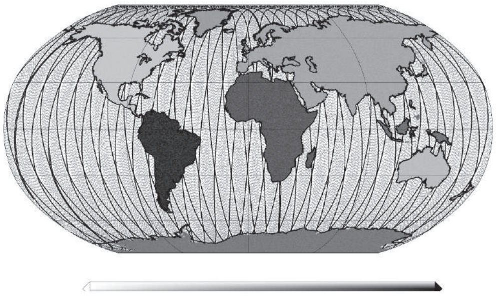


Fig. 1 Geographical distribution of the datasets. Different gray scales reflect the different standard deviations of the observations (cm over the oceans, mGal over land).

were used to generate a synthetic right hand side based on EGM2008 and a noise was added, which follows the normal distribution $\mathcal{N}\left(\mathbf{0}, \sigma_n^2 \mathbf{N}_n^{-1}\right)$. So synthetic datasets with realistic error characteristics and resolutions are available. The datasets used are summarized in Tab. 1.

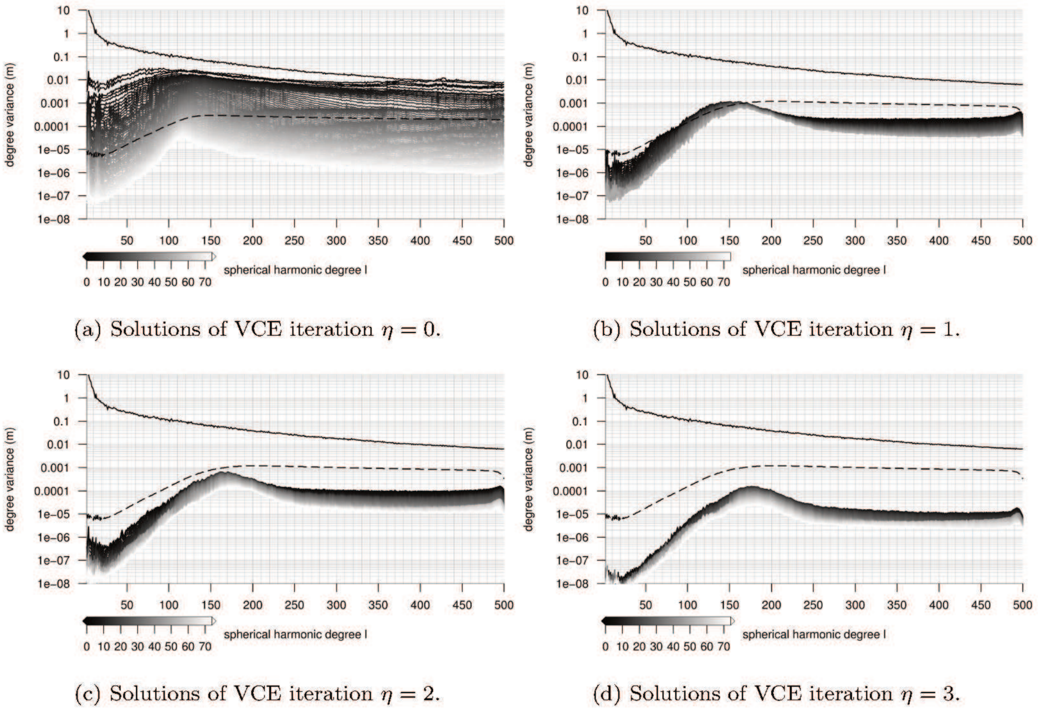
$O = 15$ datasets were created as observation groups. Each observation group stands for a dataset from a geographical region. Over land, gravity anomalies were simulated from EGM2008 for the spherical harmonic degree and order 3-500. As observation error, uncorrelated noise from a diagonal covariance matrix $\boldsymbol{\Sigma}_{\mathrm{l}_o \mathrm{l}_o} = \sigma_o^2 \boldsymbol{Q}_{\mathrm{l}_o \mathrm{l}_o}$ was added. The diagonal entries $\sqrt{\boldsymbol{Q}_{\mathrm{l}_o \mathrm{l}_o}(i,i)}$ of the cofactor matrix were randomly generated for every data point following a uniform distribution between 0.9 and 1.1. This diagonal cofactor matrix is assumed to be known in the algorithm, whereas σ_o is estimated from the data. Over the oceans, altimetry observations along a virtual Cryosat2 orbit were simulated from EGM2008. As a simplification for this first study, the dynamic ocean topography was neglected, thus it is assumed altimetry directly observes the geoid. Four 30 days sub-cycles of different accuracy were introduced as individual observation groups for which a weight should be estimated. The noise is generated as for the gravity anomaly datasets. All necessary information on the datasets is collected in Tab. 2 and illustrated in Fig. 1 (accuracy levels in terms of standard deviations of the involved datasets). The small resolution of d/o 500 was chosen within the simulation to have access to a reference solution via the assembly and solution of the full system of normal equations (Brockmann et al, 2013b).

4.2 Simulation results

The presented massive parallel implementation of the PCGMA algorithm was used to recover the 250 997 spherical harmonic coefficients and the 18 unknown weights (i.e. inverse variance components) from the 18 observation groups. As a preconditioner a blockdiagonal matrix was used. EGM96 (Lemoine et al, 1996) served as a initial solution for the iterative solver and coefficients above degree 360 were set to zero. Initial weights $w_i = 1.0$ were used as starting values for the unknown weights for the iterative VCE. The PCGMA algorithm was iterated $\nu_{\max} = 75$ times within each of $\eta_{\max} = 4$ VCE iterations.

For each of the four VCE iterations, a reference solution was computed via the assembly and solution of full normal equations. The convergence behavior of the PCGMA iterations w.r.t. the reference solutions in terms of degree variances computed from coefficient differences are shown for each VCE iteration in Fig. 2. The black solid lines show the reference solutions and the black dashed lines the degree error variances estimated from the formal coefficient standard deviations of that VCE iteration. The different PCGMA iterations – as difference to the reference solutions – are shown as grey lines. After 50 PCGMA iterations, the coefficients are determined within the formal coefficients accuracy.

In addition to the convergence of the solution, the convergence of the estimated variance components is shown as relative value $\delta\sigma_{n,o}^{(\eta)} = \frac{\sigma_{n,o}^{(\eta)} - \sigma_{n,o}}{\sigma_{n,o}}$ in Fig. 3. $\sigma_{n,o}$ names the true value used in the generation of the synthetic data. For most groups, the standard deviations converged to 10^{-2} – 10^{-3} after iteration 2. Only the groups containing only a small number of observations ($o = 6, 7, 8$) show a slightly worse behavior. Nevertheless, they are recovered better then 10^{-1} after the second VCE iteration. For the Monte Carlo based trace estimation five Monte Carlo samples were used per observation group (although a single sample should be sufficient, Koch and Kusche, 2002). Hence, the system of equations was solved for 91 right hand sides with the PCGMA algorithm.



(a) Solutions of VCE iteration $\eta = 0$.

(b) Solutions of VCE iteration $\eta = 1$.

(c) Solutions of VCE iteration $\eta = 2$.

(d) Solutions of VCE iteration $\eta = 3$.

Fig. 2 Convergence behavior of PCGMA algorithm within the 4 VCE iterations. PCGMA iterations are shown as gray lines. The reference solution is always the solution obtained via assembly and solution of full normal equations of the same VCE iteration.

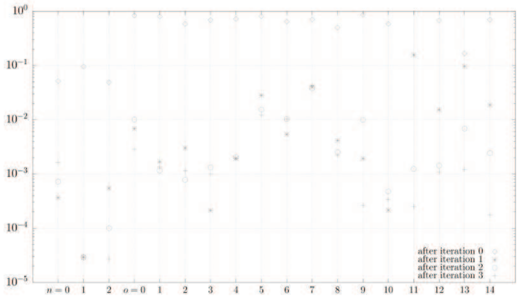


Fig. 3 Relative convergence of square root of variance component $\delta\sigma_{n,o}^{(\eta)} = \frac{\sigma_{n,o}^{(\eta)}-\sigma_{n,o}}{\sigma_{n,o}}$ for all groups n and o.

4.3 Analysis of the algorithm

In addition to the verification of the implemented algorithm, the simulation scenario is used to analyze the performance of the algorithm as well as the massive parallel implementation on a supercomputer. This covers on the one hand runtime for the different steps within one PCGMA iteration and on the other hand the scalability of the implementation. That means the decrease of runtime compared to the increase of the number of compute cores is measured. Note that all absolute (and relative) numbers and all conclusions given are only valid for the small scenario analyzed. Within this context, the word scenario means the whole set up: I) the problem to be solved itself (number of observations, number of datasets, number of unknown parameters, ...), thus the input to the algorithm and II) the technical configuration which is mainly the hardware and configuration (e.g. arrangement of compute grid, number of cores involved, parameters for block cyclic distribution, ...).

Runtime of a single PCGMA iteration: This section is used to briefly analyze the runtime of a single PCGMA iteration step for the datasets introduced above. The given numbers were derived on the JUROPA supercomputer at FZ Jülich on a 24×24 core grid (576 cores) using the block size of distributed matrices $b_r = b_c = 80$. On average one PCGMA iteration in the set up summarized above took $222\,s$. Not surprisingly, the major part is consumed by the computation of the update vector $\mathbf{H}^{(\nu)}$ which took $204\,s$ (92 %). The next time-intensive step from Alg. 2 is the preconditioning step. However it can be performed in less then $8\,s$. As the computation of $\mathbf{H}^{(\nu)}$ consists of two major parts, i.e. i) the processing of NEQs and ii) the handling of OEQs, these parts can be looked at separately. The processing of the normal equations took $48\,s$ (22%), where most of the time is spend for reading the normal equations from disk. This is a part of the implementation which can be easily be accelerated. It is possible to read all NEQ matrices once within a VCE iteration and perform the weighted combination only once. The consequence would be that the combined NEQ has to be held in-core during all PCGMA iterations (which is possible as the NEQ groups are limited in size). But still the major runtime is consumed during the ii) step, computation of the update of $\mathbf{H}^{(\nu)}$ for the groups of observation equations. That is again not surprising, as the observation equations have to be set up for all observations, and for all parameters as it is assumed that the observation groups contain full signal content. The runtime for all O groups is $156\,s$, which corresponds to 70% of the whole runtime per iteration. Within this step the following three operations are performed: I) Set up/build of $\mathbf{A}_o$: $39\,s \hat{=} 25\,\%$ (18 % of overall time),

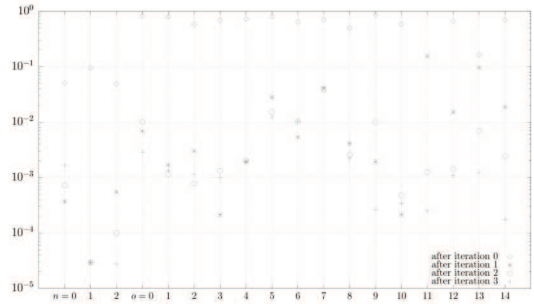


Fig. 4 Scaling behavior of the implementation, normalized to the runtime of 4 × 4 cores. The solid line shows the ideal behavior. Circles show the scaling behavior for the processing of OEQs and NEQs, whereas the stars show the scaling behavior for the processing of OEQs only (excluding the intensive I/O of matrices in each iteration).

and II) Computation of $\mathbf{G}_o^{(\nu)} := \mathbf{A}_o\boldsymbol{\Pi}^{(\nu)}$: $66\,s\hat{=}42\,\%$ ($30\,\%$ of overall time), and III) Computation of $\mathbf{H}_o^{(\nu)} = \mathbf{A}_o^T\mathbf{G}_o^{(\nu)}$: $51\,s\hat{=}33\,\%$ ($22\,\%$ of overall time). Note that the use of a full covariance matrix of the observations would be an additional significant step in the computation of $\mathbf{H}_o^{(\nu)}$ which is neglected here as the covariance matrix in the simulation was just diagonal. Nevertheless, a full covariance matrix is forseen in the implementation.

Scaling behavior: For the scenario above a scaling test was performed. One PCGMA iteration was performed varying the size of the compute core grid $R \times C$. Only quadratic grids were used from 4×4 (16 cores) up to 44×44 (1 936 cores). For the small simulation scenario, a nearly linear scaling behavior was observed from 16 to 576 cores. A still good scaling behavior could be observed up to 1 024 cores (64 times more cores compared to 16 cores, runtime reduction by a factor of 45 compared to the runtime at 16 cores), cf. Fig. 4. Again, note that this simulation scenario is a really small one and not really representative for the scenarios the implementation was designed for.

5 Summary and Conclusions

If global gravity field solutions should be derived as a least-squares solution for very high degrees, a huge number of parameters has to be derived from millions of observations. These computations are computationally demanding, if approximations are avoided. At some point, it becomes unreasonable to derive the least-squares solution from the assembly and solution of full normal equations. Whereas this is a reasonable option (Brockmann et al, 2013b) for resolution up to about d/o 720 (520 000 parameters, 2Terabyte NEQ). For this purpose, well known iterative solvers are used to obtain the rigorous solution for the unknown parameters of very high degrees. However, if the number of parameters is huge and millions of observations should be processed, a tailored massive parallel implementation of the algorithm is needed in an high performance computing environment (supercomputers, compute clusters). Within this study, the PCGMA algorithm (Schuh, 1996) was implemented in a HPC environment making use of standard concepts and libraries from scientific and high performance computing. In addition to the solution, the algorithm is extended to estimate unknown variance components for an arbitrary number of (complementary) observation groups to address the concept of relative weighting. The implementation was designed for the estimation of gravity field models up to degree and order 2 000. To derive the solution within a reasonable time, the implementation is capable to run on thousands of compute cores in parallel. This contribution is used to present the basic idea, trying to avoid too many technical and implementional details. A small scale closed loop simulation study was used to verify the algorithm and implementation. A gravity field model to d/o 500 (250 000 parameters) was estimated from 18 datasets in a joint least-squares adjustment. A rigorous solution is determined without restricting the stochastic modeling of the observations as well as the data distribution. Thus, full covariance matrices of datasets can be handled (although not shown in the simulation) and observations on arbitrary grids, or as shown along satellite orbits, can be directly processed in-situ. This would not be the case for widely used methods working with a block-diagonal approximation, where all data have to be interpolated to a regular grid within a preprocessing step. The algorithm and the implementation was tested in a HPC environment. Successful computations were performed on 16 up to 2 000 cores for the described setup. In addition first successful tests show that computations up to degree and order 2 000 are possible using the described implementation. It is demonstrated, that using HPC concepts, approximative solutions – like the block-diagonal approximation – can be avoided and the rigorous non-approximative solutions can be determined. Nevertheless the study pointed out some potential to accelerate the computations, e.g. via the use more sophisticated preconditioners (the number of iterations needed is much higher then expected) or via some detailed implementational updates (e.g. minimizing file reading operations for normal equations).

Acknowledgements The study and the publication was founded within the DFG project G/O2000+ (SCHU2305/3-1). The computations were performed on the JUROPA supercomputer at FZ Jülich. The computing time was granted by the John von Neumann Institute for Computing (project HBN15). The Open Source HPC computing libraries ATLAS, OpenMPI, BLACS, PBLAS and ScaLAPACK are gratefully acknowledged.

References

Alkhatib H (2007) On Monte Carlo methods with applications to the current satellite gravity missions. PhD thesis, Institute of Geodesy and Geoinformation, University of Bonn, Bonn

Alkhatib H, Schuh WD (2007) Integration of the Monte Carlo covariance estimation strategy into tailored solution procedures for large-scaled least squares problems. J. Geod 70:53–66, DOI 10.1007/s00190-006-0034-z

Balaji P, Bland W, Dinan J, Goodell D, Gropp W, Latham R, Pena A, Thakur R (2013) MPICH User's Guide. Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, 3rd edn, URL http://www.mpich.org

Baur O (2009) Tailored least-squares solvers implementation for high-performance gravity field research. Comput. & Geosci. 35(3):548 – 556, DOI DOI:10.1016/j.cageo.2008.09.004,

Beutler G, Jäggi A, Mervart L, Meyer U (2010) The celestial mechanics approach: application to data of the GRACE mission. Journal of Geodesy 84(11):661–681, DOI 10.1007/s00190-010-0402-6

Blackford LS, Choi J, Cleary A, D'Azevedo E, Demmel J, Dhillon I, Dongarra J, Hammarling S, Henry G, Petitet A, Stanley K, Walker D, Whaley R (1997) ScaLAPACK Users Guide. SIAM, Philadelphia, 2nd edn, URL http://www.netlib.org/scalapack/slug/index.html

Boxhammer C (2006) Effiziente numerische Verfahren zur sphärischen harmonischen Analyse von Satellitendaten. PhD thesis, Institute of Geodesy and Geoinformation, University of Bonn, Bonn

Boxhammer C, Schuh WD (2006) GOCE gravity field modeling: computational aspects - free kite numbering scheme. In: Rummel R et al (eds) Observation of the Earth System from Space, Springer, Berlin - Heidelberg, pp 209–224

Brockmann JM, Schuh W (2010) Fast variance component estimation in GOCE data processing. In: Mertikas SP (ed) Gravity, Geoid and Earth Observation, Springer Berlin Heidelberg, IAG Symp, vol 135, pp 185–193, DOI 10.1007/978-3-642-10634-7_25

Brockmann JM, Kargoll B, Krasbutter I, Schuh WD, Wermuth M (2010) GOCE data analysis: From calibrated measurements to the global earth gravity field. In: Flechtner F et al (eds) System Earth via Geodetic-Geophysical Space Techniques, Advanced Technologies in Earth Sciences, Springer Berlin Heidelberg, pp 213–229, DOI 10.1007/978-3-642-10228-8_17

Brockmann JM, Höck E, Krasbutter I, Mayer-Guerr T, Pail R, Schuh WD, Zehentner N (2013a) Performance of the fourth generation GOCE time-wise earth gravity field model. In: Geophysical Research Abstracts, Vienna, Austria, vol 15

Brockmann JM, Roese-Koerner L, Schuh WD (2013b) Use of high performance computing for the rigorous estimation of very high degree spherical harmonic gravity field models. In: (tbc) N.N. (ed) Gravity Geoid and Height Systems, Springer Berlin Heidelberg, IAG Symp, vol 141, accepted

Bruinsma S, Förste C, Abrikosov O, Marty JC, Rio MH, Mulet S, Bonvalot S (2013) The new esa satellite-only gravity field model via the direct approach. Geophysical Research Letters, DOI 10.1002/grl.50716

Dongarra JJ, Whaley RC (1997) A user's guide to the BLACS v1.1. Tech. Rep. 94, LAPACK Working Note, URL `http://www.netlib.org/lapack/lawnspdf/lawn94.pdf`, originally released March 1995

ESA (1999) The four candidate earth explorer core missions - gravity field and steady-state ocean circulation mission. ESA Report SP-1233(1), Granada

Farahani H, Ditmar P, Klees R, Liu X, Zhao Q, Guo J (2013) The static gravity field model DGM-1S from GRACE and GOCE data: computation, validation and an analysis of GOCE missions added value. Journal of Geodesy pp 1–25, DOI 10.1007/s00190-013-0650-3

Fecher T, Pail R, Gruber T (2011) Global gravity field determination by combining GOCE and complementary data. In: Ouwehand L (ed) Proceedings of the 4th International GOCE User Workshop, ESA Publication SP-696, ESA/ESTEC, URL `http://www.spacebooks-online.com/product_info.php?cPath=104&products_id=17254`

Förste C, Schmidt R, Stubenvoll R, Flechtner F, Meyer U, König R, Neumayer H, Biancale R, Lemoine JM, Bruinsma S, Loyer S, Barthelmes F, Esselborn S (2008) The GeoForschungsZentrum Potsdam/Groupe de Recherche de Gèodésie Spatiale satellite-only and combined gravity field models: EIGEN-GL04S1 and EIGEN-GL04C. J. Geod. 82(6):331–346, DOI 10.1007/s00190-007-0183-8

Förste C, Bruinsma S, Flechtner F, Marty J, Lemoine JM, Dahle C, Abrikosov O, Neumayer H, Biancale R, Barthelmes F, Balmino G (2012) A preliminary update of the direct approach GOCE processing and a new release of EIGEN-6C. San Francisco, no. 0923 in AGU Fall Meeting

Förstner W (1979) Ein Verfahren zur Schätzung von Varianz- und Kovarianzkomponenten. Allgemeine Vermessungsnachrichten 11:446–453

Gabriel E, Fagg G, Bosilca G, Angskun T, Dongarra J, Squyres J, Sahay V, Kambadur P, Barrett B, Lumsdaine A, Castain R, Daniel D, Graham R, Woodall T (2004) Open MPI: Goals, concept, and design of a next generation MPI implementation. In: Proceedings, 11th European PVM/MPI Users' Group Meeting, Budapest, Hungary, pp 97–104

Heiskanen W, Moritz H (1993) Physical Geodesy, reprint, Institute of Physical Geodesy, Technical University, Graz

Hestenes M, Stiefel E (1952) Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Stand. 49 (6), Research Paper 2379

Holmes SA, Featherstone WE (2002) A unified approach to the clenshaw summation and the recursive computation of very high degree and order normalized associated Legendre functions. J. Geod. 76(5):279–299, DOI 10.1007/s00190-002-0216-2

Hutchinson M (1990) A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. Commun. Stat. 19(2):433–450, DOI 10.1080/03610919008812866

Ilk KH, Kusche J, Rudolph S (2002) A contribution to data combination in ill-posed downward continuation problems. J. Geodyn. 33:75–99, DOI 10.1016/S0264-3707(01)00056-4

Koch K (1999) Parameter Estimation and Hypothesis Testing in Linear Models, 2nd edn. Springer Berlin/Heidelberg

Koch K, Kusche J (2002) Regularization of geopotential determination from satellite data by variance components. Journal of Geodesy 76(5):259–268, DOI 10.1007/s00190-002-0245-x

Lemoine F, Smith D, Kunz L, Smith R, Pavlis E, Pavlis N, Klosko S, Chinn D, Torrence M, Williamson R, Cox C, Rachlin K, Wang Y, Rapp R, Nerem R (1996) The development of the NASA GSFC and NIMA joint geopotential model. In: Proceedings of the "International Symposium on Gravity, Geoid, and Marine Geodesy", Tokyo, Japan

Mayer-Gürr T, Eicker A, Kurtenbach E, Ilk KH (2010a) Itg-GRACE: Global static and temporal gravity field models from GRACE data. In: Stroink L et al (eds) System Earth via Geodetic-Geophysical Space Techniques, Advanced Technologies in Earth Sciences, Springer Berlin Heidelberg, pp 159–168, DOI 10.1007/978-3-642-10228-8_13

Mayer-Gürr T, Kurtenbach E, Eicker A (2010b) ITG-GRACE2010. URL http://www.igg.uni-bonn.de/apmg/index.php?id=itg-grace2010

MPI-Forum (2009) Mpi: A message-passing interface standard 2.2. online, URL http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf

Pail R, Plank G (2003) Comparison of numerical solution strategies for gravity field recovery from GOCE SGG observations implemented on a parallel platform. Advances in Geosciences 1:39–45

Pail R, Goiginger H, Schuh WD, Höck E, Brockmann JM, Fecher T, Gruber T, Mayer-Gürr T, Kusche J, Jäggi A, Rieser D (2010) Combined satellite gravity field model GOCO01S derived from GOCE and GRACE. Geophy. Res. Lett. 37:L20,314, DOI 10.1029/2010GL044906

Pail R, Bruinsma S, Miggliaccio F, Förste C, Goiginger H, Schuh WD, Höck E, Reguzzoni M, Brockmann J, Abrikosov O, Veicherts M, Fecher T, Mayrhofer R, Krasbutter I, Sansó F, Tscherning CC (2011) First GOCE gravity field models derived by three different approaches. J. Geod. 85(11):819 – 843, DOI 10.1007/s00190-011-0467-x

Pavlis NK, Holmes SA, Kenyon S, Factor JK (2012) The development and evaluation of the earth gravitational model 2008 (EGM2008). J. Geophy. Res. DOI 10.1029/2011JB008916

Reigber C, Lühr H, Schwintzer P (2002) Champ mission status. Advances in Space Research 30(2):129 – 134, DOI DOI:10.1016/S0273-1177(02)00276-4

Ries JC, Bettadpur S, Poole S, Richter T (2011) Mean background gravity fields for GRACE processing. Austin, GRACE Science Team Meeting

Rummel R, Yi W, Stummer C (2011) GOCE gravitational gradiometry. J. Geod 85(11):777–790, DOI 10.1007/s00190-011-0500-0

Schuh W, Brockmann JM, Kargoll B, Krasbutter I, Pail R (2010) Refinement of the stochastic model of GOCE scientific data and its effect on the in-situ gravity field solution. In: ESA Living Planet Symposium, SP-686, Bergen

Schuh WD (1995) SST/SGG tailored numerical solution strategies. Tech. rep., ESA-Project CIGAR III, WP 221, Final-Report, Part 2

Schuh WD (1996) Tailored Numerical Solution Strategies for the Global Determination of the Earth's Gravity Field, Mitteilungen der geodätischen Institute der Technischen Universität Graz, vol 81. TU Graz, Graz

Schwarz H (1970) Die Methode der konjugierten Gradienten in der Ausgleichsrechnung. ZfV 95:130–140

Schwintzer P, Reigber C, Bode A, Kang Z, Zhu SY, Massmann FH, Raimondo, Biancale R, Balmino G, Lemoine JM, Moynot B, Marty JC, Barlier F, Boudon Y (1997) Long-wavelength global gravity eld models: Grim4-s4, grim4-c4. J. Geod. 71(4):189–208, DOI 10.1007/s001900050087

Sidani M, Harrod B (1996) Parallel matrix distributions: Have we been doing it all right? Tech. Rep. 116, LAPACK Working Note, URL http://www.netlib.org/lapack/lawnspdf/lawn116.pdf

Tapley B, Bettadpur S, Ries J, Thompson P, Watkins M (2004) GRACE Measurements of Mass Variability in the Earth System. Sci. 305(5683):503–505, DOI 10.1126/science.1099192

Xie J (2005) Implementation of Parallel Least-Squares Algorithms for Gravity Field Estimation. No. 474 in Reports of the Department of Geodetic Science, Ohio State University (OSU), Ohio

Algorithm 2: PCGMA algorithm extended for VCE.

Data:

$\mathbf{N}_{\oplus}$	sparse preconditioning matrix	$\nu_{\max}$	maximal number of PCGMA iterations
$\mathbf{A}_o$	design matrix $o \in \{1...O\}$	$\boldsymbol{\ell}_o$	vector of observations $o \in \{1...O\}$
$\mathbf{Q}_{\boldsymbol{\ell}_o\boldsymbol{\ell}_o}$	cofactor matrix of observations $o \in \{1...O\}$	w_o	weight for observation group $o \in \{1...O\}$
$\mathbf{N}_n$	normal matrix for group $n \in \{1...N\}$	$\mathbf{n}_n$	right hand side of NEQ for $n \in \{1...N\}$
w_n	weight for observation group $n \in \{1...N\}$	$\mathbf{x}^{(0,0)}$	initial values for parameters
$\eta_{\max}$	maximal number of VCE iterations	k	number of Monte Carlo samples

```
for $\eta = 1$ to $\eta_{\max}$ do
    $\mathbf{X}^{(\eta,0)} = \left[\mathbf{X}^{(\eta-1,0)}\ \mathbf{0}\ \cdots\ \mathbf{0}\right]$ // compose start solution, 0 for the k*(N+O) MC parameters
    $\mathbf{R}^{(0)} = \mathbf{0}$ // initilize combined residuals with 0, dimension: # parameter $\times (N+O)k+1$
    // add contribution by NEQ group $n$
    for $n = 1$ to $N$ do
        // Generate Monte Carlo Samples for all groups $n$, dimension: #parameter $\times$ $k$
        $\mathbf{P}_n = \pm 1 \sim \mathcal{U}(-1,1)$
        // sample transformation
        $\mathbf{R}_n = \mathrm{chol}(\mathbf{N}_n)$
        $\bar{\mathbf{P}}_n = \mathbf{R}_n^T \mathbf{P}_n$
        // right hand sides for group $n$, first column to insert $\bar{\mathbf{P}}_n$: $1 + (n-1)k$
        $\mathbf{B}_n = \left[\mathbf{n}_n\ \mathbf{0}\ \cdots\ \mathbf{0}\ \bar{\mathbf{P}}_n\ \mathbf{0}\ \cdots\ \mathbf{0}\right]$
        $\mathbf{R}^{(0)} += w_n^{(\eta-1)}\left(\mathbf{N}_n \mathbf{X}^{(\eta,0)} - \mathbf{B}_n\right)$
    end
    // add contribution by OEQ group $o$
    for $o = 1$ to $O$ do
        // Generate Monte Carlo Samples for all groups $n$, dimension: #observations $\times$ $k$
        $\mathbf{P}_o = \pm 1 \sim \mathcal{U}(-1,1)$
        $\mathbf{G}_{\mathbf{o}} = \mathrm{chol}\left(\mathbf{Q}^{-1}_{\boldsymbol{\ell}_o\boldsymbol{\ell}_o}\right)$
        $\bar{\mathbf{P}}_o = \mathbf{A}_o^T \mathbf{G}_{\mathbf{o}}{}^T$
        // right hand sides for group $n$, first column to insert $\mathbf{P}_o$: $1 + Nk + (o-1)k$
        $\mathbf{L}_o = \left[\mathbf{G}_{\mathbf{o}}\boldsymbol{\ell}_o\ \mathbf{0}\ \cdots\ \mathbf{0}\ \mathbf{P}_o\ \mathbf{0}\ \cdots\ \mathbf{0}\right]$
        $\mathbf{R}^{(0)} += w_o^{(\eta-1)} \mathbf{A}_o^T \mathbf{G}_{\mathbf{o}}{}^T \left(\mathbf{G}_{\mathbf{o}} \mathbf{A}_o \mathbf{X}^{(\eta,0)} - \mathbf{L}_o\right)$
    end
    $\boldsymbol{\Gamma}^{(0)} = \mathrm{solve}\left(\mathbf{N}_{\oplus}, \mathbf{R}^{(0)}\right)$ // apply preconditioner to residuals
    $\boldsymbol{\Pi}^{(0)} = -\boldsymbol{\Gamma}^{(0)}$ // initial search direction
    // PCGMA - iterations
    for $\nu = 0$ to $\nu_{\max} - 1$ do
        if $\nu > 0$ then
            // update step for search direction
            $\mathbf{E}_\nu = \frac{\mathbf{R}^{(\nu)T}\boldsymbol{\Gamma}^{(\nu)}}{\mathbf{R}^{(\nu-1)T}\boldsymbol{\Gamma}^{(\nu-1)}}$                element-wise division!
            $\boldsymbol{\Pi}^{(\nu)} = -\boldsymbol{\Gamma}^{(\nu)} + \mathbf{E}_\nu \circ \boldsymbol{\Pi}^{(\nu-1)}$                $\circ :=$ scaling of column $k$ with $\mathbf{E}_\nu(k,k)$!
        end
        $\mathbf{H}^{(\nu)} = \sum_n^N w_n \mathbf{N}_n \boldsymbol{\Pi}^{(\nu)} + \sum_o^O w_o \mathbf{A}_o^T \mathbf{Q}^{-1}_{\mathbf{l}_o\mathbf{l}_o} \mathbf{A}_o \boldsymbol{\Pi}^{(\nu)}$
        // update step for parameters
        $\mathbf{Q}^{(\nu)} = \frac{\mathbf{R}^{(\nu)T}\boldsymbol{\Gamma}^{(\nu)}}{\boldsymbol{\Pi}^{(\nu)T}\mathbf{H}^{(\nu)}}$                element-wise division!
        $\mathbf{X}^{(\eta,\nu+1)} = \mathbf{X}^{(\eta,\nu)} + \mathbf{Q}^{(\nu)} \circ \boldsymbol{\Pi}^{(\nu)}$                $\circ :=$ scaling of column $k$ with $\mathbf{Q}_\nu(k,k)$!
        // update step for residuals
        $\mathbf{R}^{(\nu+1)} = \mathbf{R}^{(\nu)} + \mathbf{Q}^{(\nu)} \circ \mathbf{H}^{(\nu)}$                $\circ :=$ scaling of column $k$ with $\mathbf{Q}_\nu(k,k)$!
        // apply preconditioner to residuals
        $\boldsymbol{\Gamma}^{(\nu+1)} = \mathrm{solve}\left(\mathbf{N}_{\oplus}, \mathbf{R}^{(\nu+1)}\right)$
    end
    // update weights of OEQ groups $i$ ($o$ and $n$) using eq. (7) and (8), (21)
    $w_i^{(\eta+1)} = \mathrm{computeVCE}(\mathbf{X}^{(\eta,\nu_{\max})},\ \bar{\mathbf{P}}_i)$
    $\eta = \eta + 1$
end
return $\mathbf{X}^{(\eta_{\max},\nu_{\max})},\ w_o^{(\eta_{\max})},\ w_n^{(\eta_{\max})}$
```

Table 1 Data sets based on NEQs used in the closed loop simulation.

n	name	NEQ/Σ_{nn}	RHS	d/o	σ_n
1	GRACE	ITG-Grace2010s	EGM08+ $\mathcal{N}\left(\mathbf{0}, \sigma_n^2 \mathbf{N}_n^{-1}\right)$	2-180	1.054
2	GOCE SST	EGM_TIM_RL03	EGM08+ $\mathcal{N}\left(\mathbf{0}, \sigma_n^2 \mathbf{N}_n^{-1}\right)$	2-100	0.954
3	GOCE SGG	EGM_TIM_RL03	EGM08+ $\mathcal{N}\left(\mathbf{0}, \sigma_n^2 \mathbf{N}_n^{-1}\right)$	2-250	0.913

Table 2 Data sets o introduced to the algorithm as observations and used in the simulation. The observations were generated from EGM2008 for d/o 3 – 500.

o	name	functional	n_o	σ_o	$\sqrt{\boldsymbol{Q}_{1_o 1_o}(i,i)}$
1	Africa	anomalies	63 750	6.0 mGal	$\mathcal{U}(0.9, 1.1)$
2	Antarctica	anomalies	155 576	5.0 mGal	$\mathcal{U}(0.9, 1.1)$
3	Australia	anomalies	17 276	2.4 mGal	$\mathcal{U}(0.9, 1.1)$
4	Eurasia	anomalies	163 691	3.2 mGal	$\mathcal{U}(0.9, 1.1)$
5	Greenland	anomalies	17 061	3.6 mGal	$\mathcal{U}(0.9, 1.1)$
6	Indonesia	anomalies	5 095	5.2 mGal	$\mathcal{U}(0.9, 1.1)$
7	Island	anomalies	497	2.8 mGal	$\mathcal{U}(0.9, 1.1)$
8	New Zealand	anomalies	724	3.4 mGal	$\mathcal{U}(0.9, 1.1)$
9	North America	anomalies	75 048	2.0 mGal	$\mathcal{U}(0.9, 1.1)$
10	South America	anomalies	38 504	7.2 mGal	$\mathcal{U}(0.9, 1.1)$
11	North pole Cap	anomalies	23 400	2.4 mGal	$\mathcal{U}(0.9, 1.1)$
12	Cryosat2 sc1	geoid height	577 272	1.0 cm	$\mathcal{U}(0.9, 1.1)$
13	Cryosat2 sc2	geoid height	577 293	3.1 cm	$\mathcal{U}(0.9, 1.1)$
14	Cryosat2 sc3	geoid height	577 479	1.2 cm	$\mathcal{U}(0.9, 1.1)$
15	Cryosat2 sc4	geoid height	577 211	3.3 cm	$\mathcal{U}(0.9, 1.1)$