Earth Sci Inform (2010) 3:149–158
DOI 10.1007/s12145-010-0062-3

RESEARCH ARTICLE

Semi-vectorization: an efficient technique for synthesis and analysis of gravity gradiometry data

Mehdi Eshagh · Makan Abdollahzadeh

Received: 12 January 2010 / Accepted: 23 July 2010 / Published online: 9 August 2010

Abstract The harmonic synthesis and analysis of the elements of gravitational tensor can be done in few minutes if a suitable programming algorithm is used. Vectorization is an efficient technique for such processes, but the size of matrices will increase when the resolution of synthesis or analysis is high; say higher than 0.5°×0.5°. Here, we present a technique to manage the computer memory and computational time by excluding one computational loop from the matrix products and we call this method semi-vectorization. Based on this technique, we synthesize the gravitational tensor using the EGM96 geopotential model and after that we analyze the tensor for recovering the geopotential coefficients. MATLAB codes are provided which are able to analyze 224 millions gradiometric data, corresponding to a global grid of 2.5′×2.5′ on a sphere in 1,093 s by a personal computer with 2 Gb RAM.

Keywords Gradiometry · Spherical harmonics · Computational time · Computer memory

Introduction

The gravity field steady-state and ocean circulation explorer (GOCE) (ESA 1999) is a satellite mission to provide a precise and high resolution Earth's gravity field with accuracies of 1 cm and 1 mGal for the geoid and gravity, respectively. A recovery of the spherical harmonic coefficients of the field to degree and order 250 is expected in this mission. GOCE takes advantage of differential accelerometry at space-borne level to measure second-order derivatives of the geopotential. Gravity field modelling by satellite gradiometry can be carried out in either time or space domain; see Rummel et al. (1993). In the former a time series for each type of gradiometric data is constructed and analyzed to estimate the geopotential coefficients; see e.g. Rummel et al. (1993). The latter method considers the data as functions of position instead of time. The analysis in the space domain can be done in two different ways: one can use either a) least-squares analysis or b) integral formulas. In the least-squares method a linear relation (spherical harmonic series) between the gradiometric data and geopotential coefficients are constructed and a large system of equations is solved to recover the coefficients. Here we consider the space-wise approach and integral formulas for gravity field recovery. In order to obtain the integral formulas, orthogonal base functions with a global support are required. Tensor spherical harmonics are useful tools in this respect, but combinations of gradiometric data are needed to use their orthogonality properties for constructing integral formulas; see e.g. Gelderen and Rummel (2001) and Martinec (2003). The integral formulas, constructed based on the tensor spherical harmonics, were simplified and used by Eshagh (2009a, c) to study the downward continuation of the satellite gradiometric data to sea level. Spherical harmonic analysis by these integrals will take hours to perform unless an efficient algorithm is used.

Kiamehr and Eshagh (2008) and Eshagh (2008) presented two series of MATLAB codes to synthesize the gradiometric data in two different frame using geopotential models. The main problem of their software was the

Communicated by: H. A. Babaie

M. Eshagh (✉) · M. Abdollahzadeh
Division of Geodesy, Royal Institute of Technology, Stockholm, Sweden
e-mail: eshagh@kth.se

M. Abdollahzadeh
e-mail: m_abdollahzadeh@sina.kntu.ac.ir

significant computational delay when a high resolution synthesis in a large extent area was required. The software presented by Eshagh (2009b) used a full-vectorization technique to synthesize and analyze the gradiometric data. His idea was to avoid repeating computations and keep the numerical values in some matrices and the synthesis was summarized in some matrix products. He also showed that full-vectorization of analysis by the integral solutions of the gradiometric boundary value problems (GBVP) was not possible. The disadvantage of this method was to work with large matrices when the resolution of synthesis or analysis was high (say higher than $0.5°\times0.5°$). This limited the work to large matrices which their multiplications required a high memory and long time. In this paper, we present an efficient vectorization technique to perform synthesis and analysis based on Eshagh's (2009a, c) formulas for the gradients and tensor spherical harmonics. We call this method semi-vecorization as one computational loop is excluded from the matrix products to manage the computer memory and speed up computations. Balmino (2009) presented an algorithm to consider the covariance of geopotential coefficients in synthesizing geoid or gravity anomaly by excluding a loop from the vectorized model. Such an idea is very similar to the method we called it semi-vectorization, but we use this method for the synthesis and analysis of the gradiometric data.

Synthesis of gravity gradients in local frame

Spherical harmonic synthesis to generate the geoid, gravity anomalies and deflections of vertical is not a new issue. Many efforts were done by geodesists to find a fast technique to perform such a double summation. Tscherning and Poder (1982) presented efficient algorithms based on the Clenshaw summation method. Holmes and Featherstone (2002a) and Bethencourt et al. (2005) used this method for synthesizing geoid. However, such a process will not be as easy as it is for synthesizing the gravity gradients as they involve first- and/or second-order derivatives of associated Legendre functions in their spherical harmonic expressions. Balmino et al. (1991) compared different software of computing the gravity gradients. They have mentioned that for high resolution and high degree synthesis a computer with large memory is required. However, their studies showed that most of their program codes were made in a FORTRAN platform and the vectorization technique that we present here is totally different with those used by them. Bettadpur et al. (1992) presented a vector technique for synthesizing gravity gradients and their idea was to perform generation of Legendre's functions and summation of the synthesis components simultaneously, while we generate the gradients order by order but in matrix forms. In fact in our approach, order-dependent matrices of gradients are generated so that their summation yields the synthesized data. Ditmar et al. (2003) discussed a fast and accurate computation of the geopotential coefficient from satellite gradiometry. They just considered the diagonal elements of the gravitational tensor in their computations and presented a two-step procedure of synthesis and analysis (Rummel et al. 1993; Sneeuw 1994). They have vectorized the model in such a way that with a simple matrix multiplication the synthesis is done. The inversion method that they used is totally different with that is presented in this paper as they used least-squares solution. Here we continue our discussion by defining the disturbing potential of the Earth's gravity field.

Let the Earth's disturbing potential be expressed by a truncated series of the spherical harmonics (Heiskanen and Moritz 1967):

$$T(P)=\frac{GM}{R}\sum_{n=2}^{N_{\max}}\sum_{m=-n}^{n}\left(\frac{R}{r}\right)^{n+1}t_{nm}Q_m(\lambda)\overline{P}_{n|m|}(\cos\theta), \tag{1a}$$

where

$$Q_m(\lambda)=\begin{cases}\cos m\lambda & m\le 0\\ \sin m\lambda & m>0\end{cases}, \tag{1b}$$

GM is the geocentric gravitational constant, R semi-major axis of a reference ellipsoid, r, θ and λ are geocentric radius, co-latitude, and longitude, respectively. $\overline{P}_{n|m|}=\overline{P}_{n|m|}(\cos\theta)$ is the fully-normalized associated Legendre function of the first kind of degree n and order m, $N_{\max}$ is the maximum degree of truncation and t_{nm} the fully-normalized geopotential coefficients.

The spherical harmonic expressions of gravity gradients are derived by taking second-order partial derivatives of disturbing potential, Eq. 1a. The gradients can be expressed in different frames. The discussion about the expressions of gradients in various frames is not in the scope of this paper and the interested reader is referred to Petrovskaya and Vershkov (2006) or Eshagh (2009a). In this study we consider the local north-oriented frame which is defined as one whose z-axis is pointing upwards in geocentric radial direction, x-axis towards the north and the frame is right-handed, implying that y-axis is directed to the west. The non-singular expressions for the gradients in such a frame are (Eshagh 2009c):

$$T_{zz}(P)=\frac{GM}{R^3}\sum_{n=2}^{N_{\max}}(n+1)(n+2)\left(\frac{R}{r}\right)^{n+3}\sum_{m=-n}^{n}t_{nm}Q_m(\lambda)\overline{P}_{n,|m|} \tag{2a}$$

$$T_{xx}(P)=\frac{GM}{R^3}\sum_{n=2}^{N_{\max}}\left(\frac{R}{r}\right)^{n+3}\sum_{m=-n}^{n}t_{nm}Q_m(\lambda)\times \tag{2b}$$
$$\times\left\{\begin{matrix}a^1_{nm}\overline{P}_{n,|m|-2}+a^2_{nm}\overline{P}_{n|m|}+a^3_{nm}\overline{P}_{n,|m|+2}\\ b^1_{nm}\overline{P}_{n,|m|-2}+b^4_{nm}\overline{P}_{n|m|}+b^3_{nm}\overline{P}_{n,|m|+2}\end{matrix}\right\}$$

$$T_{yy}(P)=\frac{GM}{R^3}\sum_{n=2}^{N_{\max}}\left(\frac{R}{r}\right)^{n+3}\sum_{m=-n}^{n}t_{nm}Q_m(\lambda)\times \times\begin{Bmatrix} b^1_{nm}\overline{P}_{n,|m|-2}+b^2_{nm}\overline{P}_{n|m|}+b^3_{nm}\overline{P}_{n,|m|+2}\\ a^1_{nm}\overline{P}_{n,|m|-2}+a^4_{nm}\overline{P}_{n|m|}+a^3_{nm}\overline{P}_{n,|m|+2}\end{Bmatrix} \tag{2c}$$

$$T_{xy}(P)=\frac{GM}{R^3}\sum_{n=2}^{N_{\max}}\left(\frac{R}{r}\right)^{n+3}\sum_{m=-n}^{n}t_{nm}Q_{-m}(\lambda)\times \times\begin{Bmatrix} c^1_{nm}\overline{P}_{n-1,|m|-2}+c^2_{nm}\overline{P}_{n-1,|m|}+c^3_{nm}\overline{P}_{n-1,|m|+2}\\ d^1_{nm}\overline{P}_{n+1,|m|-2}+d^2_{nm}\overline{P}_{n+1,|m|}+d^3_{nm}\overline{P}_{n+1,|m|+2}\end{Bmatrix} \tag{2d}$$

$$T_{xz}(P)=\frac{GM}{R^3}\sum_{n=2}^{N_{\max}}\left(\frac{R}{r}\right)^{n+3}\sum_{m=-n}^{n}t_{nm}Q_m(\lambda)\left[e^1_{nm}\overline{P}_{n,|m|-1}+e^2_{nm}\overline{P}_{n,|m|+1}\right] \tag{2e}$$

$$T_{yz}(P)=\frac{GM}{R^3}\sum_{n=2}^{N_{\max}}\left(\frac{R}{r}\right)^{n+3}\sum_{m=-n}^{n}t_{nm}Q_{-m}(\lambda) \begin{bmatrix} f^1_{nm}\overline{P}_{n-1,|m|-1}+f^2_{nm}\overline{P}_{n-1,|m|+1}\\ g^1_{nm}\overline{P}_{n+1,|m|-1}+g^2_{nm}\overline{P}_{n+1,|m|+1}\end{bmatrix} \tag{2f}$$

The above expressions are very simple to use as they do not involve the derivatives of the associated Legendre functions and they are non-singular at the poles. The coefficients $a^1_{nm}, a^2_{nm},\ldots,g^1_{nm}$ and g^2_{nm} are constant and can be computed once during computations; see Appendix A for their mathematical formulas. In Eqs. 2b, 2c, 2d and 2f two expressions were presented for each gradient which either of them can be used in practice. The important issue of using such formulas is the existence of two summations on degrees and orders. The use of two loops for performing the summations is suitable only for generating the gradients in few points. If the number of points increases the algorithm will not be efficient. It should be noticed that the presented expressions for T_{xx} and T_{yy} are exactly the same as those derived by Eshagh (2009a, c) but in order to simplify our vectorized models we introduce the new coefficients a^4_{nm} and b^4_{nm} and also made small changes in a^2_{nm} and b^2_{nm}.

Tensor spherical harmonics and their application in spectral solution of gradiometric boundary value problems

The gravitational tensor can be expressed in terms of tensor spherical harmonics. These harmonics are orthogonal and in fact they are base functions for analyzing the gradiometry data. Three combinations of the gradients are used and called vertical-vertical, vertical-horizontal and horizontal-horizontal GBVP (VVGBVP, VHGBVP and HHGBVP). Based on these boundary values three sets of solutions for the geopotential coefficients are obtained (Martinec 2003 and Eshagh 2009c):

$$t^1_{nm}=\frac{R^3/4\pi GM}{(n+1)(n+2)}\left(\frac{r}{R}\right)^{n+3}\iint_{\sigma}T_{zz}(Q)Y_{nm}(Q)d\sigma \tag{3a}$$

$$t^2_{nm}=\frac{R^3/4\pi GM}{n(n+1)(n+2)}\left(\frac{r}{R}\right)^{n+3}\iint_{\sigma}\left[T_{xz}(Q)E_{nm}(Q)+T_{yz}(Q)F_{nm}(Q)\right]d\sigma \tag{3b}$$

$$t^3_{nm}=\frac{R^3/4\pi GM}{(n-1)n(n+1)(n+2)}\left(\frac{r}{R}\right)^{n+3}\iint_{\sigma}\left\{\left[T_{xx}(Q)-T_{yy}(Q)\right]G_{nm}(Q)-2T_{xy}(Q)H_{nm}(Q)\right\}d\sigma \tag{3c}$$

where

$$E_{nm}(Q)=Q_m(\lambda)\left\{e'^1_{nm}\overline{P}_{n,|m|-1}+e'^2_{nm}\overline{P}_{n,|m|+1}\right\} \tag{4a}$$

$$F_{nm}(Q)=Q_{-m}(\lambda)\begin{Bmatrix} f'^1_{nm}\overline{P}_{n-1,|m|-1}+f'^2_{nm}\overline{P}_{n-1,|m|+1}\\ g'^1_{nm}\overline{P}_{n+1,|m|-1}+g'^2_{nm}\overline{P}_{n+1,|m|+1}\end{Bmatrix} \tag{4b}$$

$$G_{nm}(Q)=Q_m(\lambda)\left\{h^1_{nm}\overline{P}_{n,|m|-2}+h^2_{nm}\overline{P}_{n,|m|}+h^3_{nm}\overline{P}_{n,|m|+2}\right\} \tag{4c}$$

$$H_{nm}(Q)=Q_{-m}(\lambda)\begin{Bmatrix} k^1_{nm}\overline{P}_{n+1,|m|-2}+k^2_{nm}\overline{P}_{n+1,|m|}+k^3_{nm}\overline{P}_{n+1,|m|+2}\\ l^1_{nm}\overline{P}_{n-1,|m|-2}+l^2_{nm}\overline{P}_{n-1,|m|}+l^3_{nm}\overline{P}_{n-1,|m|+2}\end{Bmatrix}. \tag{4d}$$

Equations 4a–4d have been presented by Eshagh (2009c). The constant coefficients $e'^1_{nm}, e'^2_{nm},\ldots,l^1_{nm}$ and l^3_{nm} are given in Appendix B which can be computed once during computations. Using four loops for numerical solution of the integrals (3a)–(3b) with a global coverage of the gravity gradients is very time-consuming and full-vectorization is not possible for such computations (Eshagh 2009b) while we will show that the semi-vectorization technique is greatly efficient.

Semi-vectorization of the mathematical models

We define vectorization as a technique to present the mathematical models in matrix or vector forms (Eshagh

2009b). According to the advances in computer science and technology, it is of importance to use such a technique to make efficient software. MATLAB is one of the powerful platforms for the vectorized models and simple for coding. In the following we explain how to use its capability to handle such models. The important issue is how to verctorize the formulas based on the software capability.

Semi-vectorization in synthesis

Generation of spherical harmonics and their first- and/or second-order derivatives is the most important part of synthesis and analysis but the advantage of Eqs. 2a–2f is to use the associated Legendre functions and not their derivatives. Let us define a special operator which is frequently used in our semi-vector modelling which MATLAB handles it efficiently.

Definition Element-wise product ' # ' is multiplication of each corresponding elements of two matrices, or

$$\begin{bmatrix} a'_{11} & a'_{12} & \cdots \\ a'_{21} & a'_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \# \begin{bmatrix} b'_{11} & b'_{12} & \cdots \\ b'_{21} & b'_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} a'_{11}b'_{11} & a'_{12}b'_{11} & \cdots \\ a'_{21}b'_{21} & a'_{22}b'_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

The element-wise product is specified by ".*" in MATLAB.

We can make a MATLAB code to generate the fully-normalized associated Legendre functions of the first kind and insert them in the following order-dependent matrix:

$$\mathbf{P}_N^m = \begin{bmatrix} P_{2,m}(\theta_0) & P_{3,m}(\theta_0) & \cdots & P_{N_{\max},m}(\theta_0) \\ P_{2,m}(\theta_1) & P_{3,m}(\theta_1) & \cdots & P_{N_{\max},m}(\theta_1) \\ \vdots & \vdots & \ddots & \vdots \\ P_{2,m}(\theta_{N_\theta}) & P_{3,m}(\theta_{N_\theta}) & \cdots & P_{N_{\max},m}(\theta_{N_\theta}) \end{bmatrix} \tag{5}$$

where N stands for a vector containing the values from 2 to $N_{\max}$ and N_θ is number of parallels, which depends on the resolution of synthesis. In Eq. 5 the rows are the associated Legendre functions at the same parallels and orders but with different degrees; and the columns are those with the same degrees and orders but at different parallels (co-latitudes). The size of the matrix changes according to the resolution and maximum degree of synthesis. Equation 5 is order-dependent, i.e. the matrix is updated for each order m.

There is a degree-dependent parameter in Eqs. 2a–2f which we name it **dmp** (damping factor) and vectorize it by:

$$\mathbf{dmp} = [1 \quad 1 \quad \cdots \quad 1]^T_{1\times N_\theta} \otimes [a_2 \quad a_3 \quad \cdots \quad a_{N_{\max}}]_{1\times(N_{\max}-1)} \tag{6a}$$

$$a_i = \frac{GM}{R^3}\left(\frac{R}{r}\right)^{i+3} \times 10^9 \tag{6b}$$

where $\otimes$ stands for Kronecker's product. **dmp** is a matrix with the same dimension of $\mathbf{P}_N^m$ (Eq. 5). The Kronecker product implies that **dmp** has the same rows. $Q_m(\lambda)$ can also be vectrozied by:

$$\mathbf{Q}^m = \begin{cases} \sin m[\lambda_1\lambda_2\cdots\lambda_{N_\lambda}] & m > 0 \\ \cos m[\lambda_1\lambda_2\cdots\lambda_{N_\lambda}] & m \le 0 \end{cases} \tag{7}$$

which is dependent on the sign of m. $\mathbf{Q}^m$ is a $1 \times N_\lambda$ vector of sine or cosine of m times of longitudes. The fully-normalized geopotential coefficients are inserted into two separate matrices $\mathbf{T}^m$ which its rows show the degrees and the columns are the orders:

$$\mathbf{T}^m = \begin{cases} S & m > 0 \\ C & m \le 0 \end{cases}. \tag{8}$$

In order to simplify the vectorized forms of formulas, we put the constant coefficients of Eqs. 2b–2f into separate matrices of ε^j so that the rows are the values of coefficients at the same degrees but at different orders and the columns are vice versa. ε^j, j=1, 2,…, 11 corresponds to the coefficients a^1_{nm}, a^2_{nm}, a^3_{nm}, a^4_{nm}, d^1_{nm}, d^2_{nm}, d^3_{nm}, e^1_{nm}, e^2_{nm}, g^1_{nm} and g^2_{nm}, respectively. These matrices have the same dimensions as those of $\mathbf{T}^m$. The element-wise product of $\mathbf{T}^m$ and ε^j will be:

$$\widehat{\mathbf{T}}^{m,j} = \mathbf{T}^m \# \boldsymbol{\varepsilon}^j. \tag{9}$$

According to the above definitions, Eq. 2b–2f are vectorized by:

$$\mathbf{T}_{xx} = \sum_{m=-N_{\max}}^{N_{\max}} \left[\left(\mathbf{P}_N^{|m|-2} \# \mathbf{dmp}\right) \widehat{\mathbf{T}}^{m,1}_{,|m|+1} + \left(\mathbf{P}_N^{|m|} \# \mathbf{dmp}\right) \widehat{\mathbf{T}}^{m,2}_{,|m|+1} + \left(\mathbf{P}_N^{|m|+2} \# \mathbf{dmp}\right) \widehat{\mathbf{T}}^{m,3}_{,|m|+1} \right] \mathbf{Q}^m \tag{10a}$$

$$\mathbf{T}_{yy} = \sum_{m=-N_{\max}}^{N_{\max}} \left[\left(\mathbf{P}_N^{|m|-2} \# \mathbf{dmp}\right) \widehat{\mathbf{T}}^{m,1}_{,|m|+1} + \left(\mathbf{P}_N^{|m|} \# \mathbf{dmp}\right) \widehat{\mathbf{T}}^{m,4}_{,|m|+1} + \left(\mathbf{P}_N^{|m|+2} \# \mathbf{dmp}\right) \widehat{\mathbf{T}}^{m,3}_{,|m|+1} \right] \mathbf{Q}^m \tag{10b}$$

$$\mathbf{T}_{xy} = \sum_{m=-N_{\max}}^{N_{\max}} \left[\left(\mathbf{P}_{N+1}^{|m|-2} \# \mathbf{dmp}\right) \widehat{\mathbf{T}}^{m,5}_{,|m|+1} + \left(\mathbf{P}_{N+1}^{|m|} \# \mathbf{dmp}\right) \widehat{\mathbf{T}}^{m,6}_{,|m|+1} + \left(\mathbf{P}_{N+1}^{|m|+2} \# \mathbf{dmp}\right) \widehat{\mathbf{T}}^{m,7}_{,|m|+1} \right] \mathbf{Q}^m \tag{10c}$$

$$\mathbf{T}_{xz} = \sum_{m=-N_{\max}}^{N_{\max}} \left[\left(\mathbf{P}_N^{|m|-1} \#\mathbf{dmp} \right) \widehat{\mathbf{T}}_{,|m|+1}^{m,8} + \left(\mathbf{P}_N^{|m|+1} \#\mathbf{dmp} \right) \widehat{\mathbf{T}}_{,|m|+1}^{m,9} \right] \mathbf{Q}^m \tag{10d}$$

$$\mathbf{T}_{yz} = \sum_{m=-N_{\max}}^{N_{\max}} \left[\left(\mathbf{P}_{N+1}^{|m|-1} \#\mathbf{dmp} \right) \widehat{\mathbf{T}}_{,|m|+1}^{m,10} + \left(\mathbf{P}_{N+1}^{|m|+1} \#\mathbf{dmp} \right) \widehat{\mathbf{T}}_{,|m|+1}^{m,11} \right] \mathbf{Q}^m. \tag{10e}$$

In the above equations $\widehat{\mathbf{T}}_{,|m|+1}^{m,j}$ are column vectors and the subscript $,|m|+1$ refers to column $|m|+1$ of the matrix $\widehat{\mathbf{T}}^{m,j}$. Equations 10a–10e generate $N_\theta \times N_\lambda$ matrices of gradients for each order m, i.e. the synthesis process globally performs order by order. The result of the synthesis will be summation of all the matrices in different orders.

In a similar manner, we can vectorize Eq. 2a. First we consider the degree-dependent term:

$$a_i = \frac{GM}{R^3}(i+1)(i+2)\left(\frac{R}{r}\right)^{i+3} \times 10^9. \tag{11}$$

According to Eq. 2a and the strategy presented for semi-vectorization, it will not be difficult to convert the equation to the following semi-vectorized model:

$$\mathbf{T}_{zz} = \sum_{m=-N_{\max}}^{N_{\max}} \left(\mathbf{P}_N^{|m|} \#\mathbf{dmp} \right) \mathbf{T}^m \mathbf{Q}^m. \tag{12}$$

Figure 1 presents a flowchart for semi-vectorization of synthesis. The inputs are the vectors θ and λ with dimensions of $1 \times N_\theta$ and $1 \times N_\lambda$, respectively. The argument $\mathbf{N}$ is an $(N_{\max}-1)\times 1$ vector containing the degrees from 2 to $N_{\max}$. The matrices $\mathbf{T}^m$ and ε^j are constructed with dimensions of $(N_{\max}-1)\times(N_{\max}+1)$. The **dmp** matrix is computed using Eqs. 6a and 6b before starting the loop. After that the computation starts using Eqs. 10a–10e and Eq. 12 and follows by a loop on order m. The matrices of the loop are generated order by order and the result will be sum of the matrices. In Fig. 1, $\mathbf{T}_k$, $k=1, 2, \ldots, 6$ corresponds to $\mathbf{T}_{xx}$, $\mathbf{T}_{yy}$, $\mathbf{T}_{zz}$, $\mathbf{T}_{xy}$, $\mathbf{T}_{xz}$, and $\mathbf{T}_{yz}$, respectively.

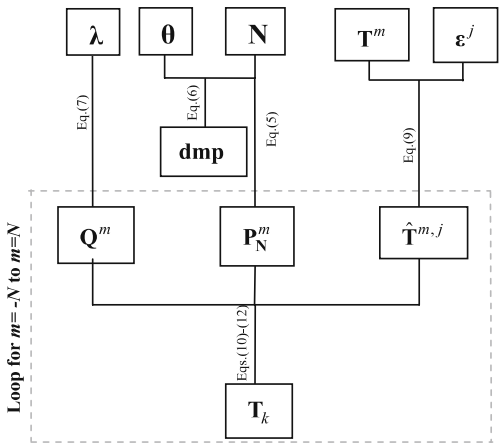

Fig. 1 Flowchart of semi-vectorization in synthesis

Table 1 Elements of $\mathbf{DF}_k$

k	$\mathbf{DF}_k$
1	$R^3/4\pi GM(i+1)(i+2)(r/R)^{i+3}$
2	$R^3/4\pi GMi(i+1)(i+2)(r/R)^{i+3}$
3	$R^3/4\pi GM(i-1)i(i+1)(i+2)(r/R)^{i+3}$

The models presented until now are not fully-vectorized. We have to use a loop to sum up the generated order-dependent matrices. Eshagh (2009b) used the fully-vectorized approach for synthesis without any loop. The problem of this method is to work with very large matrices especially when the resolution of data is high. However, in the method presented here we let one loop to be outside the matrix-vector products for managing the computer memory better than the fully-vectorized model. The synthesis technique we presented here is suited for regular grids on a sphere. However, synthesis on irregular surfaces requires modifications in the algorithm to consider the irregularities. If one does not use semi-vectorization in this case, the synthesis process will be slow but this issue is left for further investigations in the future.

Semi-vectorization in analysis

Equations 3a–3c are three integral solutions for analysis of gradiometric data, which are semi-vectorized here. First let us introduce the following parameter:

$$\mathbf{\Omega} = \Delta\theta\Delta\lambda \left[1 \quad 1 \cdots 1 \right]^T_{1\times(N_{\max}-1)} \otimes \left[\sin\theta_1 \sin\theta_2 \cdots \sin\theta_N \right]_{1\times N_\theta} \tag{13}$$

where $\mathbf{\Omega}$ is a matrix whose columns are surface area of the each cell within parallel and they are repeated in rows $(N_{\max}-1)$ times. In Eqs. 3a–3c the degree-dependent

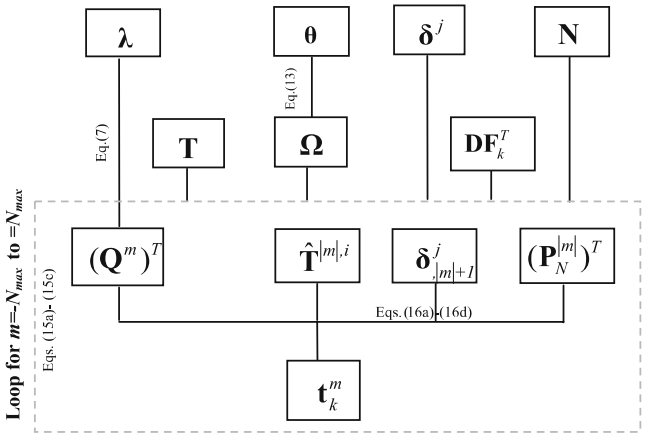

Fig. 2 Flowchart of semi-vectorization in analysis

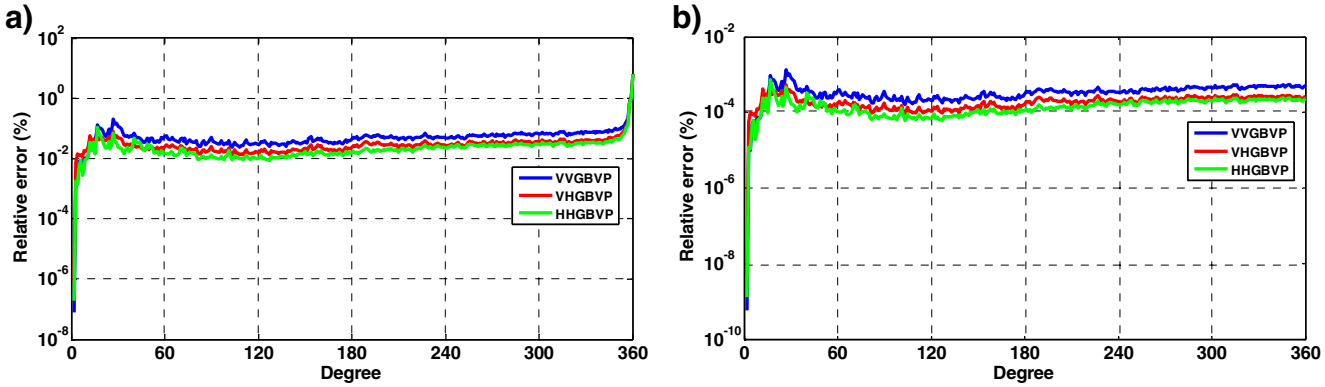

Fig. 3 Percentage of relative error degree variances of recovered coefficients with a resolution of **a** 30′×30′ and **b** 2.5′×2.5′

downward continuation factor (**DF**) can be written as a $(N_{\max}-1)\times 1$ vector:

$$\mathbf{DF}_k = [\, a_2 \quad a_3 \quad \cdots \quad a_{N_{\max}} \,]^T \tag{14}$$

The elements of $\mathbf{DF}_k$ depend on the solution type k, Eqs. 3a–3c, and they are given in Table 1.

The fully-normalized geopotential coefficients can be computed by the following semi-vectorized models:

$$\mathbf{t}_1^m = \mathbf{DF}_1 \# \left[\left(\left(\mathbf{P}_N^{|m|} \right)^T \# \boldsymbol{\Omega} \right) \mathbf{T}_{zz} (\mathbf{Q}^m)^{\mathrm{T}} \right] \tag{15a}$$

$$\mathbf{t}_2^m = \mathbf{DF}_2 \# \left[\left(\widehat{\mathbf{T}}^{|m|,1} \# \boldsymbol{\Omega} \right) \mathbf{T}_{xz} (\mathbf{Q}^m)^{\mathrm{T}} - \left(\widehat{\mathbf{T}}^{|m|,2} \# \boldsymbol{\Omega} \right) \mathbf{T}_{yz} (\mathbf{Q}^m)^{\mathrm{T}} \right] \tag{15b}$$

$$\mathbf{t}_3^m = \mathbf{DF}_3 \# \left[\left(\widehat{\mathbf{T}}^{|m|,3} \# \boldsymbol{\Omega} \right) (\mathbf{T}_{xx} - \mathbf{T}_{yy}) (\mathbf{Q}^m)^{\mathrm{T}} - 2 \left(\widehat{\mathbf{T}}_m^{|m|,4} \# \boldsymbol{\Omega} \right) \mathbf{T}_{xy} (\mathbf{Q}^m)^{\mathrm{T}} \right] \tag{15c}$$

where $\mathbf{t}_1^{\mathrm{m}}, \mathbf{t}_2^{\mathrm{m}}$ and $\mathbf{t}_3^{\mathrm{m}}$ for $m>0$ are the vectors of sine coefficients for all degrees but in the same orders and for $m<0$ they are cosine coefficients. The matrices $\mathbf{T}_{xx}$, $\mathbf{T}_{yy}$, $\mathbf{T}_{zz}$, $\mathbf{T}_{xy}$, $\mathbf{T}_{xz}$ and $\mathbf{T}_{yz}$ are the gradients with a global coverage and a specific resolution. In above equations the matrices $\widehat{\mathbf{T}}^i$, $i = (|m|,1),(|m|,2),(|m|,3)$ and $(|m|,4)$ are defined by:

$$\widehat{\mathbf{T}}^{|m|,1} = \boldsymbol{\delta}^{\mathbf{1}}_{,|\mathbf{m}|+\mathbf{1}} \otimes \mathbf{I} \# \left(\mathbf{P}_{\mathbf{N}}^{|\mathbf{m}|-\mathbf{1}} \right)^{\mathrm{T}} + \boldsymbol{\delta}^{\mathbf{2}}_{,|\mathbf{m}|+\mathbf{1}} \otimes \mathbf{I} \# \left(\mathbf{P}_{\mathbf{N}}^{|\mathbf{m}|+\mathbf{1}} \right)^{\mathbf{T}} \tag{16a}$$

$$\widehat{\mathbf{T}}^{|m|,2} = \boldsymbol{\delta}^{3}_{,|m|+1} \otimes \mathbf{I} \# \left(\mathbf{P}_{N-1}^{|m|-1} \right)^{T} + \boldsymbol{\delta}^{4}_{,|m|+1} \otimes \mathbf{I} \# \left(\mathbf{P}_{N-1}^{|m|+1} \right)^{T} \tag{16b}$$

$$\widehat{\mathbf{T}}^{|m|,3} = \boldsymbol{\delta}^{5}_{,|m|+1} \otimes \mathbf{I} \# \left(\mathbf{P}_{N}^{|m|-2} \right)^{T} + \boldsymbol{\delta}^{6}_{,|m|+1} \otimes \mathbf{I} \# \left(\mathbf{P}_{N}^{|m|} \right)^{T} + \boldsymbol{\delta}^{7}_{,|m|+1} \otimes \mathbf{I} \# \left(\mathbf{P}_{N}^{|m|+2} \right)^{T} \tag{16c}$$

$$\widehat{\mathbf{T}}^{|m|,4} = \boldsymbol{\delta}^{8}_{,|m|+1} \otimes \mathbf{I} \# \left(\mathbf{P}_{N}^{|m|-2} \right)^{T} + \boldsymbol{\delta}^{9}_{,|m|+1} \otimes \mathbf{I} \# \left(\mathbf{P}_{N}^{|m|} \right)^{T} + \boldsymbol{\delta}^{10}_{,|m|+1} \otimes \mathbf{I} \# \left(\mathbf{P}_{N}^{|m|+2} \right)^{T} \tag{16d}$$

where **I** is an identity row vector:

$$\mathbf{I} = [\, 1 \quad 1 \quad \cdots \quad 1 \,]_{1\times N_\theta} \tag{17}$$

and the matrices δ^j, j=1, 2, …, 10 correspond to e'^1_{nm}, e'^2_{nm}, g'^1_{nm}, g'^2_{nm}, h^1_{nm}, h^2_{nm}, h^3_{nm}, k^1_{nm}, k^2_{nm} and k^3_{nm} of Eqs. 4a–4d, the subscript $,|m|+1$ refers to $(m+1)$th column of δ^j. A flowchart for analysing the gradiometric data is presented in Fig. 2. The inputs are the $N_\theta \times N_\lambda$ matrices **T** containing the gradients. The constant coefficients matrices δ^j are constructed with dimensions of $(N_{\max}-1)\times(N_{\max}+1)$. Eshagh (2009b) presented a semi-vetorized model to analyze the gradients only by considering vectorization for the double integrals. However, here we presented a technique in which the integrals and degrees are vectorized together and the vectorization is done by one simple loop on the orders. In this case the geopotential coefficients are computed order by order.

Software presentations

We made some MATLAB functions for synthesizing and analyzing the gravity gradients. In order to test the correctness of the functions, the EGM96 geopotential model (Lemoine et al. 1998) to degree and order 360 is used corresponding to a resolution of 0.5°×0.5° for sysnthesis. Then the generated gradients are used in the analysis functions to recompute the geopotential coefficients. The difference between the computed and the original geopotential coefficients of EGM96 will be

Table 2 Computational times of global gradiometric synthesis with a resolution of 30′×30′

Quantity	Non-vec.	Fully-vec.	Semi-vec.
T_{ZZ}	$8^h24^m36^s$	5^m3^s	9^s
$T_{XZ}T_{YZ}$	$18^h51^m00^s$	11^m19^s	26^s
$T_{XX}T_{YY}T_{ZZ}$	$28^h55^m48^s$	18^m45^s	36^s

Table 3 Computational times of global gradiometric analysis with a resolution of 30′×30′

Quantity	Vec. integral	Semi-vec.
VVGBVP	$0^h30^m21^s$	9^s
VHGBVP	$0^h55^m18^s$	30^s
HHGBVP	$1^h05^m39^s$	40^s

a criterion for correct performance of our functions. Figure 3a demonstrates the percentage of relative error degree variances of recovered coefficients. It should be noted that the integral formulas for analysis, Eqs. 3a–3c, suffer by the discretization error. The magnitude of such an error depends on the resolution of analysis, but by increasing the resolution the error should be reduced. To show this, the resolution of data is increased to 2.5′×2.5′ and the relative error degree variances of the recovered coefficients are plotted. As is shown in Fig. 3b the relative errors are noticeably reduced. Figure 3a shows that the errors are increasing when the degree n is approaching 360 while when the resolution is increased such a problem will not occur. It is also observed that the relative error spectra are below 1% but when the resolution is increased they will be below 0.01%.

In order to show the efficiency of the presented vectorized technique, we consider the CPU time of computations. The computations are done on a personal computer with 2.5 GHz dual-core CPU and 2 Gb RAM. The capability of this technique will be adopted when a comparison with non-vectorized and fully-vectocrized methods is made. Tables 2 and 3 clearly show efficiency of the new technique in synthesis and analysis of gradiometric data. It is seen that this method significantly speeds up the computations comparing to non-vectorized models. It is worth mentioning that the main issue related to full-vectorization is to store very large matrices for high resolution of data and high degree of synthesis and analysis. The study of Eshagh (2009b) revealed that a maximum spatial resolution of 30′×30′ for data is achievable by the full-vectorization technique while the new technique is capable of synthesizing of the data up to a spatial resolution of 2.5′×2.5′. In order to draw a comparison for the analysis process, the analysis was carried out based on VVGBVP using non-vectorized models and a computational time of $20^h56^m7^s$ was observed. We restricted our computations just up to degree 200 with the spatial resolution of 30′×30′ for gradiometric data. Such process reduces to 9 s for the semi-vetorized models; see Table 3. Vec. integral in this table means that just the integrals are vectorized and the analysis is performed for each degree and order of harmonic coefficient.

In Fig. 4a and b the CPU times of global synthesis and analysis with different resolutions are plotted. It is observed that the synthesis functions work well when the maximum degree of synthesis is 360 and the resolution is 2.5′×2.5′. In this case the functions generate 224 million gradients. The synthesis of T_{xx}, T_{yy}, and T_{xy} takes around 1,950 s, but their analysis takes less than 550 s based on a resolution of 2.5′×2.5′. The CPU times for synthesizing and analyzing T_{xz} and T_{yz} are around 1,300 and 410 s, respectively. Such processes take 739 and 144 s for T_{zz}. The total CPU time of syntheses of all types of gradient is 1.1 h and 1,093 s for their analyses. The synthesis and analysis times will be reduced to 1,100 and 512 when the resolution is 5′×5′.

The synthesis and analysis functions can also show their efficiency if a high degree geopotential model like EGM08 (Pavlis et al. 2008) is used. They are able to perform the synthesis up to degree and order 2,160 with maximum spatial resolution of 5′×5′. We observed 3^h20^m and 3^h30^m CPU times for synthesis of all gradients and for analysis of the data, respectively.

Summary

Semi-vectorization is an efficient algorithm in spherical harmonic synthesis and analysis of any quantity and it is

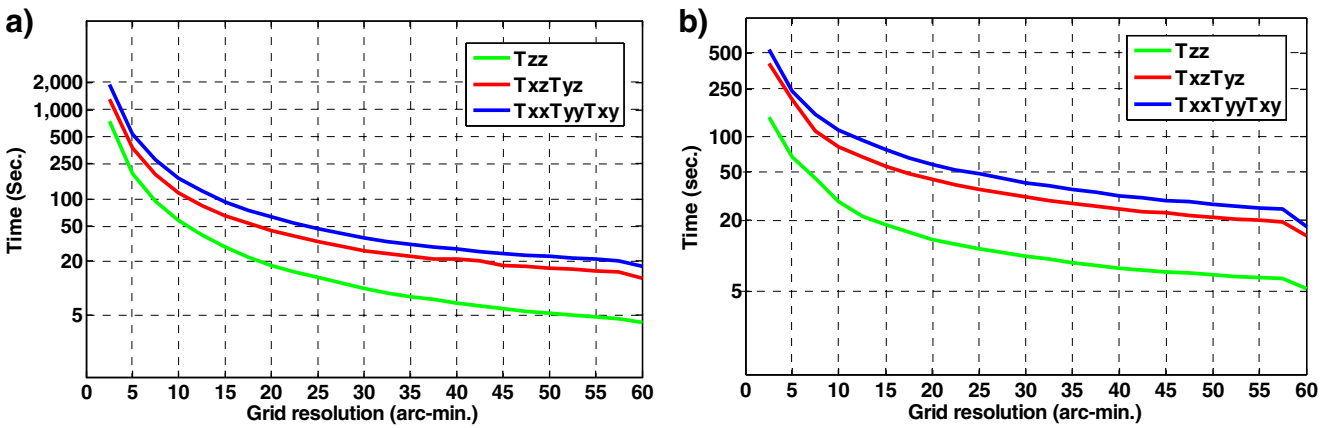

Fig. 4 CPU times for global **a** synthesis and **b** analysis with different resolutions

not restricted to gradiometry. However, gradiometry is one of the difficult examples which were investigated in this paper. The most important issue in this study is related to the way of using Eqs. 2a–2f for synthesis and Eqs. 3a–4d for analysis. In spite of other methods of synthesis and analysis, the method presented here takes the advantage of some new formulas for the gradients and tensor spherical harmonics which do not involve first- and/or second-order derivatives of associated Legendre functions. The analysis functions can be used for analyzing the GOCE data for the validation process and to recover the geopotential coefficients. The functions have the capability of processing 224 millions data using a personal computer with 2 Gb RAM.

Acknowledgment The first author is thankful to the Swedish National Space Board (SNSB) for financial support of the project 63:07:1.

Appendix A

The constant coefficients of Eqs. 2b–2f (Eshagh 2009c):

$$a^1_{nm}=\frac{1}{4}\sqrt{n+|m|}\sqrt{n+|m|-1}\sqrt{n-|m|+1}\times\sqrt{n-|m|+2}\sqrt{\frac{2-\delta_{|m|,0}}{2-\delta_{|m|-2,0}}} \tag{A.1}$$

$$a^2_{nm}=-\frac{1}{4}[(n+|m|)(n-|m|+1)+(n-|m|)(n+|m|+1)]-(n+1) \tag{A.2}$$

$$a^3_{nm}=\frac{1}{4}\sqrt{n+|m|+2}\sqrt{n+|m|+1}\sqrt{n-|m|}\times\sqrt{n-|m|-1}\sqrt{\frac{2-\delta_{|m|,0}}{2-\delta_{|m|+2,0}}} \tag{A.3}$$

$$a^4_{nm}=a^2_{nm}+(n+1)(n+2) \tag{A.4}$$

$$b^1_{nm}=\frac{1}{4}\sqrt{\frac{2-\delta_{|m|,0}}{2-\delta_{|m|-2,0}}}\sqrt{n+|m|}\sqrt{n-|m|+1}\times\sqrt{n-|m|+2}\sqrt{n+|m|-1} \tag{A.5}$$

$$b^2_{nm}=\frac{1}{4}\left[(n+|m|)(n+|m|-1)+\frac{|m|-1}{|m|+1}(n-|m|)(n-|m|-1)+\frac{2(n+|m|+1)(n+|m|+2)}{|m|+1}\right]-(n+1) \tag{A.6}$$

$$b^3_{nm}=\frac{1}{4}\sqrt{\frac{2-\delta_{|m|,0}}{2-\delta_{|m|+2,0}}}\sqrt{n-|m|}\sqrt{n-|m|-1}\times\sqrt{n+|m|+2}\sqrt{n+|m|+1} \tag{A.7}$$

$$b^4_{nm}=b^2_{nm}+(n+1)(n+2) \tag{A.8}$$

$$c^1_{nm}=\frac{m}{4|m|}\sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|-2,0})(2n-1)}}\sqrt{n+|m|}\times\sqrt{n+|m|-1}\sqrt{n+|m|-2}\times\sqrt{n-|m|+1} \tag{A.9}$$

$$c^2_{nm}=\frac{m}{2}\sqrt{n+|m|}\sqrt{n-|m|}\sqrt{\frac{2n+1}{2n-1}} \tag{A.10}$$

$$c^3_{nm}=\frac{m}{4|m|}\sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|+2,0})(2n-1)}}\sqrt{n-|m|}\times\sqrt{n-|m|-1}\sqrt{n-|m|-2}\times\sqrt{n+|m|+1} \tag{A.11}$$

$$d^1_{nm}=\frac{m}{4|m|}\sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|-2,0})(2n-3)}}\sqrt{n+|m|}\times\sqrt{n-|m|+1}\sqrt{n-|m|+2}\times\sqrt{n-|m|+3} \tag{A.12}$$

$$d^2_{nm}=\frac{m}{4|m|}\times[(n+|m|)(n-|m|+1)-(n-|m|)(n-|m|+1)]\times\sqrt{\frac{(2n+1)(n+|m|+1)}{(2n+3)(n-|m|+1)}} \tag{A.13}$$

$$d^3_{nm} = -\frac{m}{4|m|}\sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|+2,0})(2n+3)}}\sqrt{n-|m|} \times \sqrt{n+|m|+1}\sqrt{n+|m|+2} \times \sqrt{n+|m|+3} \quad (A.14)$$

$$e^1_{nm} = \frac{n+2}{2}\sqrt{n+|m|}\sqrt{n-|m|+1} \times \sqrt{\frac{2-\delta_{|m|,0}}{2-\delta_{|m|-1,0}}} \quad (A.15)$$

$$e^2_{nm} = -\frac{n+2}{2}\sqrt{n-|m|}\sqrt{n+|m|+1} \times \sqrt{\frac{2-\delta_{|m|,0}}{2-\delta_{|m|+1,0}}} \quad (A.16)$$

$$f^1_{nm} = \frac{m(n+2)}{2|m|}\sqrt{n+|m|}\sqrt{n+|m|-1} \times \sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|-1,0})(2n-1)}} \quad (A.17)$$

$$f^2_{nm} = \frac{m(n+2)}{2|m|}\sqrt{n-|m|}\sqrt{n-|m|-1} \times \sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|+1,0})(2n-1)}} \quad (A.18)$$

$$g^1_{nm} = \frac{m(n+2)}{2|m|}\sqrt{n-|m|+1}\sqrt{n-|m|+2} \times \sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|-1,0})(2n+3)}} \quad (A.19)$$

$$g^2_{nm} = \frac{m(n+2)}{2|m|}\sqrt{n+|m|+1}\sqrt{n+|m|+2} \times \sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|+1,0})(2n+3)}}. \quad (A.20)$$

Appendix B

The constant coefficients of Eqs. 4a–4d:

$$h^1_{nm} = \frac{1}{2}\sqrt{\frac{2-\delta_{|m|,0}}{2-\delta_{|m|-2,0}}}\sqrt{n+|m|}\sqrt{n+|m|-1} \times \sqrt{n-|m|+1}\sqrt{n-|m|+2} \quad (B.1)$$

$$h^2_{nm} = m^2 \quad (B.2)$$

$$h^3_{nm} = \frac{1}{2}\sqrt{\frac{2-\delta_{|m|,0}}{2-\delta_{|m|+2,0}}}\sqrt{n-|m|}\sqrt{n-|m|-1} \times \sqrt{n+|m|+1}\sqrt{n+|m|+2} \quad (B.3)$$

$$k^1_{nm} = \frac{m}{2|m|}\sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|-2,0})(2n+3)}}\sqrt{n+|m|} \times \sqrt{n-|m|+3}\sqrt{n-|m|+2} \times \sqrt{n-|m|+1} \quad (B.4)$$

$$k^2_{nm} = \frac{m}{2}\sqrt{\frac{(2n+1)(n+m+1)}{(2n+3)(n-m+1)}} \times [(n+m+2)(n-m+1)-(n-m+2)(n-m+1)] \quad (B.5)$$

$$k^3_{nm} = -\frac{m}{2|m|}\sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|+2,0})(2n+3)}}\sqrt{n-|m|} \times \sqrt{n+|m|+3}\sqrt{n+|m|+2} \times \sqrt{n+|m|+1} \quad (B.6)$$

$$l^1_{nm} = \frac{m}{2|m|}\sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|-2,0})(2n-1)}}\sqrt{n+|m|} \times \sqrt{n+|m|-3}\sqrt{n+|m|-2}\sqrt{n-|m|+1} \quad (B.7)$$

$$l^2_{nm} = -m\sqrt{\frac{2n+1}{2n-1}}\sqrt{n-|m|}\sqrt{n+|m|} \quad (B.8)$$

$$l^3_{nm} = -\frac{m}{2|m|}\sqrt{\frac{(2-\delta_{|m|,0})(2n+1)}{(2-\delta_{|m|+2,0})(2n-1)}}\sqrt{n-|m|} \times \sqrt{n-|m|-1}\sqrt{n-|m|-2}\sqrt{n+|m|+1} \quad (B.9)$$

and $e'_{nm}1 = e^1_{nm}/(n+2)$, $e'_{nm}2 = e^2_{nm}/(n+2)$, $g'_{nm}1 = g^1_{nm}/(n+2)$, $g'_{nm}2 = g^2_{nm}/(n+2)$, $f'_{nm}1 = f^1_{nm}/(n+2)$, $f'_{nm}2 = f^2_{nm}/(n+2)$.

References

Balmino G (2009) Efficient propagation of error covariance matrices of gravitational models: application to GRACE and GOCE. J Geodesy 83:989–995

Balmino G, Barriot JP, Koop R, Middle B, Thong NC, Vermeer M (1991) Simulation of gravity gradients: a comparison study. Bull Géod 65:218–229

Bethencourt A, Wang J, Rizos C, Kearsley AHW (2005) Using personal computers in spherical harmonic synthesis of high degree Earth geopotential models, Dynamic Planet 2005, Cairns, Australia, 22–26 August

Bettadpur SV, Schutz BE, Lundberg JB (1992) Spherical harmonic synthesis and least-squares computations in satellite gravity gradiometry. Bull Géod 66:261–271

Ditmar P, Klees R, Kostenko F (2003) Fast and accurate computation of spherical harmonic coefficients from satellite gravity gradiometry data. J Geodesy 76:690–705

ESA (1999) Gravity Field and Steady-State Ocean Circulation Mission. ESA SP-1233(1), Report for mission selection of the four candidate earth explorer missions, ESA Publications Division, p 217

Eshagh M (2008) Non-singular expression for vector and tensor of gravitation in a geocentric frame. Comput Geosci 34:1762–1768

Eshagh M (2009a) On satellite gravity gradiometry, PhD thesis in Geodesy, Royal Institute of Technology (KTH), Stockholm, Sweden

Eshagh M (2009b) Impact of vectorization on global synthesis and analysis in gradiometry. Acta Geod Geophys Hung 44:323–342

Eshagh M (2009c) Alternative expressions for gravitational gradients in local north oriented frame and tensor spherical harmonics. Acta Geophys 58:215–243

Gelderen M, Rummel R (2001) The solution of the general boundary value problem by least-squares. J Geodesy 75:1–11

Heiskanen WA, Moritz H (1967) Physical geodesy. Freeman, San Francisco

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 Geodesy 76:279–299

Kiamehr R, Eshagh M (2008) EGMlab, a scientific software for determining the gravity and gradient components from global geopotential models. Earth Sci Inf 1:93–103

Lemoine FG, Kenyon SC, Factor JK, Trimmer RG, Pavlis NK, Chinn D, Cox CM, Klosko SM, Luthcke SB, Torrence MH, Wang YM, Williamson RG, Pavlis EC, Rapp RH, Olson TR (1998) Geopotential model EGM96. NASA/TP-1998-206861. Goddard Space Flight Center, Greenbelt

Martinec Z (2003) Green's function solution to spherical gradiometric boundary-value problems. J Geodesy 77:41–49

Pavlis NK, Holmes SA, Kenyon SC, Factor JK (2008) An Earth gravitational model to degree 2160: EGM2008, Presented at the 2008 General Assembly of the European Geosciences Union, Vienna, Austria, April 13–18

Petrovskaya MS, Vershkov AN (2006) Non-singular expressions for the gravity gradients in the local north-oriented and orbital reference frames. J Geodesy 80:117–127

Rummel R, Sanso F, Gelderen M, Koop R, Schrama E, Brovelli M, Migiliaccio F, Sacerdote F (1993) Spherical harmonic analysis of satellite gradiometry. Publ Geodesy, New Series, No. 39 Netherlands Geodetic Commission, Delft

Sneeuw N (1994) Global spherical harmonic analysis by least-squares and numerical quadrature methods in historical perspective. Geophys J Int 118:707–716

Tscherning CC, Poder K (1982) Some Geodetic application of Clenshaw summation, Bollettino di geodesia e scienze affini No. 4