COMPARISON AMONG SPHERICAL HARMONIC SYNTHESIS METHODS FOR FUNCTIONALS OF THE GRAVITY FIELD

Elena FANTINO[1], Stefano CASOTTO[1,2]
[1] Dipartimento di Astronomia, Università di Padova, vicolo dell'Osservatorio 2, 35122 Padova, Italy; e-mails: fantino@pd.astro.it, casotto@pd.astro.it
[2] CISAS "G.Colombo", Università di Padova, via Venezia 15, 35131 Padova, Italy

Abstract
Four well-known and qualified algorithms for the computation of the Earth's gravitational potential and its first and second gradients are examined and compared. Their characteristics are analysed with reference to the computational requirements of the global geopotential estimation process from satellite gradiometry. Numerical tests have been performed in order to assess and compare the efficiency, accuracy and precision of the algorithms. The algorithm of Clenshaw has shown to be by far the most efficient and therefore the most suitable for implementation and use when dealing with large amounts of observations.

Keywords: Spherical harmonic synthesis; Associated Legendre Functions; Helmholtz Polynomials; Clenshaw summation method

1 Introduction

The computation of the Earth's gravitational potential and its derivatives is a problem of interest to many applications in geophysics and geodesy. With reference to satellite gravity gradiometry, the phase of harmonic synthesis and least squares adjustment for geopotential and gravity gradient estimation is of major concern when dealing with large amounts of observations (of the order of 10^8) and high resolution (of the order of $9 \cdot 10^4$ Stokes coefficients) gravity models to be solved for: aspects such as numerical accuracy and computational performance are an issue of primary importance and must be carefully and critically addressed.

Following Bettadpur et al. (1992), we can identify three areas of concern in general spherical harmonic synthesis: the selection of a potential formulation (including the choice of the coordinate system describing the position of the observation points), the selection of the algorithms for the recursive computation of the Associated Legendre Functions (ALFs) and, finally, the issue of arrangement of the computations for the best performance on a given computer architecture.

In the past great attention has been given by several authors to the algorithms for the computation of the geopotential and its derivatives. In this work we have selected, examined and compared four among the most qualified and well-known analytical methods for the computation of the gravitational potential and its derivatives. They are: the traditional forward column recursion in spherical coordinates; the method proposed by Pines (1973) based on Helmholtz Polynomials (HPs) in rectangular Earth-fixed cartesian coordinates (and here presented as a forward column recursion method); the method developed by Cunningham (1970) and more recently reintroduced by Metris et al. (1999) consisting in a downward recursion on a complex operator of derivation in rectangular Earth-fixed cartesian coordinates; a backward recurrence method based on the Clenshaw summation formula (Press et al., 1992; Tscherning and Poder, 1982) in spherical coordinates. Note that the two methods in cartesian Earth-fixed coordinates (i.e., Pines and Cunningham-Metris) are singularity-free at the poles.

After a general introduction to the geopotential and its derivatives and the definition of the coordinate systems and transformations (Section 2), a description of the four algorithms is provided (Sections 3 to 6). Section 7 presents and describes a set of numerical tests applied to each algorithm, aiming at assessing numerical efficiency, relative numerical precision and numerical accuracy. The results of the comparison are discussed in Section 8.

2 The Earth's gravitational potential and its gradients

The gravitational potential V at a point P above the surface of the Earth can be written as a spherical harmonic expansion (Heiskanen and Moritz, 1967)

$$V(P) = \frac{\mu}{r} \sum_{n=0}^{\infty} \left(\frac{a}{r}\right)^n \sum_{m=0}^{n} \overline{P}_{nm}(\sin\varphi) \left(\overline{C}_{nm} \cos m\lambda + \overline{S}_{nm} \sin m\lambda\right) \tag{1}$$

where r, φ and λ are the spherical coordinates of P representing geocentric radius, latitude and longitude respectively, a is the mean Earth's radius, μ is the Earth's gravitational parameter ($\mu = GM_{\oplus}$), $\overline{C}_{nm}$ and $\overline{S}_{nm}$ are the fully normalized Stokes coefficients of the gravity field and $\overline{P}_{nm}(\sin\varphi)$ is the fully normalized Associated Legendre Function (fnALF) of the first kind of degree n and order m:

$$\overline{P}_{nm}(\sin\varphi) = \overline{P}_{nm}(\varphi) = N_{nm} \left[\frac{1}{n!2^n} \frac{d^{n+m}}{d(\sin\varphi)^{n+m}} (\sin^2\varphi - 1)^n\right] \tag{2}$$

with the full normalization factor

$$N_{nm} = \sqrt{\frac{(2-\delta_{o,m})(2n+1)(n-m)!}{(n+m)!}}. \tag{3}$$

In practice the expansion (1) is truncated at a maximum degree N, which determines the resolution of the gravity model. In this work we set $N = 360$. For issues such as the numerical stability of the recurrence relations employed for evaluating spherical harmonics of ultra-high degree and order (e.g., $N = 2700$) the reader is referred to Holmes and Featherstone (2002).

The gravitational potential V is a scalar field and as such can be regarded as a tensor of rank 0. The first gravity gradient ∇V is a tensor of rank 1 (a vector), the second gravity gradient $\nabla\nabla V$ is a tensor of rank 2. Now let us consider two generalized coordinate systems u^p and x^q. The transformations that affect a covariant and a contravariant tensor of rank 1, T_k and T^k respectively, under change of coordinates

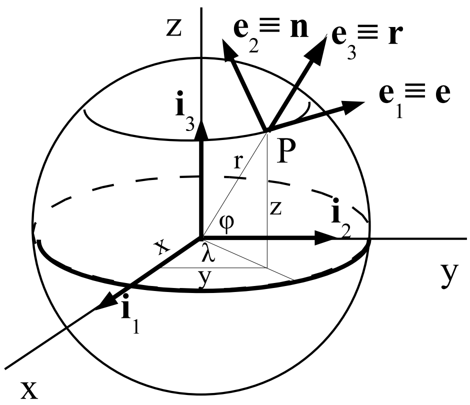

Figure 1: Relation between the spherical local coordinate system and the rectangular Earth-fixed coordinate system.

from u^p to x^q are

$$\begin{aligned} T_k &= \left(\frac{\partial u^m}{\partial x^k}\right) T'_m \\ T^k &= \left(\frac{\partial x^k}{\partial u^m}\right) T'^m. \end{aligned} \tag{4}$$

Similarly, for a covariant and a contravariant rank 2 tensor, T_{kl} and T^{kl} respectively, we have:

$$\begin{aligned} T_{kl} &= \left(\frac{\partial u^m}{\partial x^k}\right)\left(\frac{\partial u^n}{\partial x^l}\right) T'_{mn} \\ T^{kl} &= \left(\frac{\partial x^k}{\partial u^m}\right)\left(\frac{\partial x^l}{\partial u^n}\right) T'^{mn}. \end{aligned} \tag{5}$$

In this work we use two reference frames, the first a cartesian, Earth-centred, Earth-fixed frame with orthonormal vector basis ($\mathbf{i}_1$,$\mathbf{i}_2$,$\mathbf{i}_3$); the second a spherical local frame, centred at $P(\lambda,\varphi,r)$ with orthonormal vector basis ($\mathbf{e}_1$,$\mathbf{e}_2$,$\mathbf{e}_3$) which coincide with the East ($\mathbf{e}$), North ($\mathbf{n}$) and radial directions ($\mathbf{r}$) respectively (see Fig. 1).

In order to perform numerical comparisons among the algorithms we will apply the transformation of the first and second gravity gradient tensors from cartesian Earth-fixed coordinates to spherical coordinates. Introduce the transformation $x^p = x^p\,(u^q)$ between spherical local and rectangular Earth-fixed coordinates:

$$\begin{aligned} x &= r\cos\varphi\cos\lambda \\ y &= r\cos\varphi\sin\lambda \\ z &= r\sin\varphi. \end{aligned} \tag{6}$$

The associated Jacobian matrix $\partial x^p/\partial u^q$ is

$$\begin{pmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ \cos\varphi\cos\lambda & \cos\varphi\sin\lambda & \sin\varphi \end{pmatrix}. \tag{7}$$

The transformations (4) and (5) from spherical to cartesian Earth-fixed coordinates as applied to the first and second gravity gradient tensors are provided in Tables 1

Table 1: The first gravity gradient: rotation of the physical components from cartesian Earth-fixed to spherical local coordinates.

$$
\begin{aligned}
V_\lambda &= -V_x \sin\lambda + V_y \cos\lambda \\
V_\varphi &= -V_x \cos\lambda \sin\varphi - V_y \sin\lambda \sin\varphi + V_z \cos\varphi \\
V_r &= V_x \cos\lambda \cos\varphi + V_y \cos\varphi \sin\lambda + V_z \sin\varphi
\end{aligned}
$$

Table 2: The second gravity gradient: rotation of the physical components from cartesian Earth-fixed to spherical local coordinates.

$$
\begin{aligned}
V_{\lambda\lambda} &= V_{xx} \sin^2\lambda - V_{xy} \sin 2\lambda + V_{yy} \cos^2\lambda \\
V_{\lambda\varphi} &= (V_{xx} - V_{yy}) \sin\lambda \cos\lambda \sin\varphi - V_{xy} \cos 2\lambda \sin\varphi + (V_{yz} \cos\lambda - V_{xz} \sin\lambda) \cos\varphi \\
V_{\varphi\varphi} &= \left(V_{xx} \cos^2\lambda + V_{xy} \sin 2\lambda + V_{yy} \sin^2\lambda\right) \sin^2\varphi - (V_{xz} \cos\lambda + V_{yz} \sin\lambda) \sin 2\varphi + V_{zz} \cos^2\varphi \\
V_{\lambda r} &= [-V_{xx} \sin\lambda \cos\varphi + V_{yz} \sin\varphi + V_{yy} \sin\lambda \cos\varphi] \cos\lambda + V_{xy} \cos\varphi \cos 2\lambda - V_{xz} \sin\lambda \sin\varphi \\
V_{\varphi r} &= \left(V_{zz} - V_{xx} \cos^2\lambda - V_{xy} \sin 2\lambda - V_{yy} \sin^2\lambda\right) \sin 2\varphi/2 + (V_{xz} \cos\lambda + V_{yz} \sin\lambda) \cos 2\varphi \\
V_{rr} &= \left(V_{xx} \cos^2\lambda + V_{yy} \sin^2\lambda + V_{xy} \sin 2\lambda\right) \cos^2\varphi + (V_{xz} \cos\lambda + V_{yz} \sin\lambda) \sin 2\varphi + V_{zz} \sin^2\varphi
\end{aligned}
$$

and 2: there the tensor components are physical components, i.e., components with respect to the orthonormal vector basis of the corresponding coordinate system.

3 Forward column recursion in spherical coordinates

The most direct and traditional approach for evaluating the gravitational potential uses recursions on Associated Legendre Functions. The physical components of the first and second gravity gradients have the form reported in Table 3. The partial derivatives that appear there are obtained by differentiating Eq. (1) with respect to λ, φ and r accordingly. Their full expressions can be found in Xu (2003) and Ditmar and Klees (2002).

The standard way of evaluating the fnALFs and their derivatives is by recursion. There are numerous recurrence relations, reproduced for example in Abramowitz and Stegun (1964): there are recurrences on n alone, on m alone, and on both n and

Table 3: Physical components of the first and second gravity gradients in local spherical coordinates.

$$
\begin{aligned}
V_\lambda &= \frac{1}{r\cos\varphi}\partial_\lambda V & V_\varphi &= \frac{1}{r}\partial_\varphi V \\
V_r &= \partial_r V & V_{\lambda\lambda} &= \frac{1}{r^2\cos^2\varphi}\left(\partial_{\lambda\lambda} V + r\cos^2\varphi\,\partial_r V - \frac{1}{2}\sin 2\varphi\,\partial_\varphi V\right) \\
V_{\lambda\varphi} &= \frac{1}{r^2\cos\varphi}\left(\partial_{\lambda\varphi} V + \tan\varphi\,\partial_\lambda V\right) & V_{\varphi\varphi} &= \frac{1}{r^2}\left(\partial_{\varphi\varphi} V + r\partial_r V\right) \\
V_{\lambda r} &= \frac{1}{r\cos\varphi}\left(\partial_{\lambda r} V - \frac{1}{r}\partial_\lambda V\right) & V_{\varphi r} &= \frac{1}{r}\left(\partial_{\varphi r} V - \frac{1}{r}\partial_\varphi V\right) \\
V_{rr} &= \partial_{rr} V
\end{aligned}
$$

m simultaneously. Most of the recurrences involving m are unstable and therefore dangerous for numerical work. The most commonly adopted procedure to produce a set of fnALFs complete to degree and order N follows the very well known forward column recursion (FCR) scheme on n (Hobson, 1965), which starts from the initialization $\overline{P}_{00}(\varphi) = 1$ and then computes the sectorial terms ($m = n$ with $n \geq 1$) by means of

$$\overline{P}_{nn}(\varphi) = f_n \cos\varphi \overline{P}_{n-1,n-1}(\varphi) \tag{8}$$

with

$$f_n = \sqrt{\frac{(2n+1)(1+\delta_{1,n})}{2n}}. \tag{9}$$

The remaining terms (zonals and tesserals with $0 \leq m \leq n-1$) are obtained from

$$\overline{P}_{nm}(\varphi) = g_{nm} \sin\varphi \overline{P}_{n-1,m}(\varphi) - h_{nm}\overline{P}_{n-2,m}(\varphi), \quad n > m \tag{10}$$

where

$$\begin{aligned} g_{nm} &= \sqrt{\frac{(2n+1)(2n-1)}{(n+m)(n-m)}} \\ h_{nm} &= \sqrt{\frac{(2n+1)(n-m-1)(n+m-1)}{(2n-3)(n+m)(n-m)}} = \frac{g_{nm}}{g_{n-1,m}}, \end{aligned} \tag{11}$$

while the fnALFs with $m > n$ are defined to be zero. The first derivatives of the fnALFs with respect to φ are given by (Ilk, 1983):

$$\frac{d\overline{P}_{nm}(\varphi)}{d\varphi} = m \tan\varphi \overline{P}_{nm}(\varphi) - k_{nm}\overline{P}_{n,m+1}(\varphi) \tag{12}$$

where

$$k_{nm} = \sqrt{\frac{(2-\delta_{o,m})(n-m)(n+m+1)}{2}}. \tag{13}$$

Differentiating Eq. (12) with respect to φ yields the second derivatives:

$$\begin{aligned} \frac{d^2\overline{P}_{nm}(\varphi)}{d\varphi^2} =& \, m\left(m\tan^2\varphi + \sec^2\varphi\right)\overline{P}_{nm}(\varphi) - (2m+1)k_{nm}\tan\varphi\,\overline{P}_{n,m+1}(\varphi) \\ &+ j_{nm}\overline{P}_{n,m+2}(\varphi) \end{aligned} \tag{14}$$

in which

$$j_{nm} = \sqrt{\frac{(2-\delta_{o,m})(n-m)(n-m-1)(n+m+2)(n+m+1)}{2}}. \tag{15}$$

The input required by the algorithm is the gravity field model ($\overline{C}_{nm}$, $\overline{S}_{nm}$ with $n, m = 0, ..., N$) and the position of K observation points (P_k: λ_k, φ_k, r_k with $k = 1, ..., K$ and $K \geq 1$). Optimum performance is reached by first evaluating and storing the normalization coefficients f_{nm}, g_{nm} ,h_{nm}, j_{nm} and k_{nm} [Eqs. (9), (11), (13) and (15)], then the latitude-dependent functions, i.e., the $\overline{P}_{nm}$ and their derivatives $d\overline{P}_{nm}/d\varphi$ and $d^2\overline{P}_{nm}/d\varphi^2$ [Eqs. (8), (10), (12) and (14)]; finally, for the given λ_k, the sums which yield the various partial derivatives of V are accumulated. This arrangement ensures

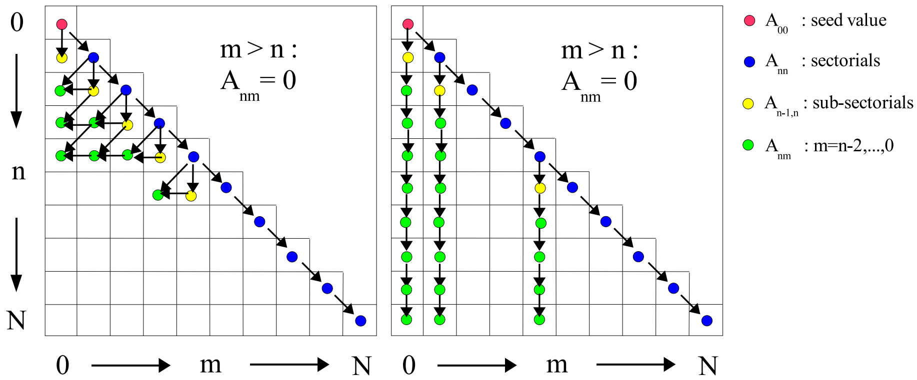

Figure 2: Left: mixed strides recursion algorithm. Right: forward column recursion scheme.

higher efficiency when dealing with data uniformly distributed on a spherical grid and accessed per parallel.

We implemented this algorithm in a Fortran90 computer code, named *LEGENDRE_a*, that, given the fully normalized Stokes coefficients of the gravity model and the position of the observation points, computes the gravitational potential and its first and second gradients in the ($\mathbf{e}_1$,$\mathbf{e}_2$,$\mathbf{e}_3$) orthonormal basis. Another implementation of the same algorithm was kindly provided by F. Sansó (Politecnico di Milano, PoliMi) in the form of a dynamic link library (DLL) running under Matlab. We named it *LEGENDRE_b*.

4 The method of Helmholtz polynomials in Earth-fixed cartesian coordinates

In the work published by S. Pines in 1973 and later revised by Spencer (1976) the potential and its first and second gradients are computed in rectangular Earth-fixed coordinates. The algorithm developed by Pines uses Helmholtz Polynomials $A_{nm}(u)$ (Balmino et al., 1990) instead of ALFs

$$A_{nm}(u) = \frac{1}{n!2^n} \frac{d^{n+m}}{du^{n+m}} (u^2 - 1)^n \tag{16}$$

and the coordinates of the observation points are given by the components s, t and u of the geocentric unit position vector, supplemented by the geocentric distance r. The k^{th}derivative of the Helmholtz Polynomial of degree n and order m is the Helmholtz Polynomial of degree n and order $(m+k)$:

$$\frac{d^k A_{nm}(u)}{du^k} = A_{n,m+k}(u). \tag{17}$$

The author presented a recursion for the computation of the A_{nm} which, starting from the initialization $A_{00}(u) = 1$ (which follows from $P_0 = 1$), first fills in the diagonal (sectorial terms, $m = n$ with $n \geq 1$) and then yields the remaining zonal and tesseral terms ($0 \leq m \leq n-2$) through a “mixed strides” recursion: away from the diagonal each polynomial is a linear combination of polynomials of different degree/order, therefore not on the same stride (row/column) (Fig. 2). This recipe is presented as a stable recursion. However, since we are basing our computations on fully normalized Stokes coefficients, appropriate normalization factors must be applied. The fully normalized Helmholtz Polynomial (fnHPs) $\bar{A}_{nm}(u)$ of degree n and order m is

$$\bar{A}_{nm}(u) = \sqrt{\frac{(2-\delta_{o,m})(2n+1)(n-m)!}{(n+m)!}} A_{nm}(u). \tag{18}$$

From our numerical experiments, we found that Pines' recursion scheme, as applied to the fnHPs, is strongly unstable. Fig. 3 illustrates a comparison with the results obtained by application of the traditional FCR scheme ($N = 360$), which on the contrary is very stable. As a consequence, we replaced the MSR scheme proposed by Pines with the FCR algorithm.

Written as a series expansion (truncated at a maximum degree N) in fnHPs, the Earth's gravitational potential V at a point P (x, y, z) in outer space is:

$$V(P) = V(x,y,z) \equiv V(s,t,u,r) = \sum_{n=0}^{N} \rho_n \sum_{m=0}^{n} \overline{A}_{nm}(u) \overline{D}_{nm}(s,t) \tag{19}$$

ρ_n being a recursively-defined function of r and $\overline{D}_{nm}(s,t)$ a so-called fully normalized mass coefficient function which depends on some trigonometric functions r_m and i_m of λ and the Stokes coefficients. By application of the chain rule of differentiation, the operators of derivation of the gravitational potential with respect to the cartesian Earth-fixed coordinates x, y, z of P can be written as a function of the operators of derivation with respect to s, t, u and r. Finally, the first and second gravity gradients, ∇V and $\nabla\nabla V$, in Earth-fixed cartesian coordinates are given by

$$\begin{aligned} V_x &= a_1 + s a_4 \\ V_y &= a_2 + t a_4 \\ V_z &= a_3 + u a_4 \end{aligned} \tag{20}$$

$$\begin{aligned} V_{xx} &= a_{11} + 2s a_{14} + a_4/r + s^2 a_{44} - s^2 a_4/r \\ V_{xy} &= a_{12} + st a_{44} + s a_{24} + t a_{14} - st a_4/r \\ V_{yy} &= a_{22} + 2t a_{24} + a_4/r + t^2 a_{44} - t^2 a_4/r \\ V_{xz} &= a_{13} + su a_{44} + s a_{34} + u a_{14} - su a_4/r \\ V_{yz} &= a_{23} + tu a_{44} + t a_{34} + u a_{24} - tu a_4/r \\ V_{zz} &= a_{33} + 2u a_{34} + a_4/r + u^2 a_{44} - u^2 a_4/r \end{aligned} \tag{21}$$

where a_j $(j = 1, 4)$ and a_{ij} $(i, j = 1, 4)$ are partial sums, functions of four additional mass coefficient functions and their derivatives. For the details of the method we refer the reader to the original work by Pines (1973).

We implemented this algorithm in a Fortran90 computer code, named *PINES*, that, given the fully normalized Stokes coefficients of the gravity model and the Earth-fixed rectangular coordinates x, y, z of K observation points P_k ($k = 1, ..., K$ and $K \geq 1$), determines the gravitational potential and its first and second gradients. Best performance is reached by first evaluating and storing the various normalization coefficients, the functions ρ_n, r_m and i_m; then, the latitude dependent quantities, i.e.,

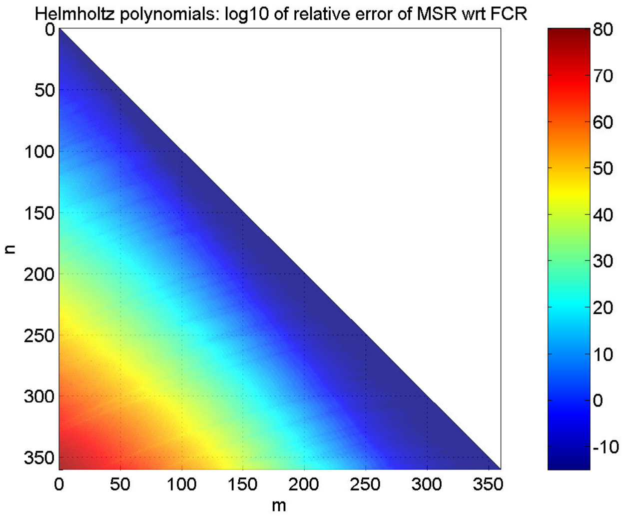

Figure 3: Log_{10} of the relative difference between the Helmholtz polynomials as computed with the FCR algorithm and with the MSR scheme: the difference grows larger with increasing degree (downwards) and decreasing order (backwards) proving that the MSR algorithm is highly unstable.

the Helmholtz polynomials $\overline{A}_{nm}$ and their derivatives $d\overline{A}_{nm}/du$ and $d^2\overline{A}_{nm}/du^2$ are computed; finally, for each s and t, the mass coefficient functions and their derivatives are formed, and the partial sums a_j and a_{ij} are accumulated. This arrangement offers higher efficiency when dealing with data distributed on a spherical grid and accessed per parallel.

5 The method of Cunningham-Metris

The algorithm developed by Cunningham (1970) is based on the representation of the gravitational potential in solid spherical harmonics V_{nm}

$$V_{nm} \equiv \frac{\overline{P}_{nm}(\sin\varphi)(\cos m\lambda + \mathrm{i}\sin m\lambda)}{r^{n+1}} \tag{22}$$

which enter the expression of V in the following way:

$$V = \mathrm{Re}\left[\mu \sum_{n=0}^{N}\sum_{m=0}^{n} a^n \left(\overline{C}_{nm} - \mathrm{i}\overline{S}_{nm}\right) V_{nm}\right] \tag{23}$$

i being the imaginary unit. The V_{nm} can be expressed through the operators of derivation $\partial/\partial z$ and $(\partial/\partial x + \mathrm{i}\partial/\partial y)$ and obey a downward recursion (Cunningham, 1970). Thanks to these two characteristics, any derivative of V_{nm} can be written as a simple linear combination of other V_{nm}.

The work recently published by Metris et al. (1999) makes one step forward by providing expressions for the derivatives of any order of V itself.

G.Metris and his collaborators implemented this algorithm into a Fortran77 computer programme which, given a set of fully normalized Stokes coefficients for the gravity model and the positions of the observation points, determines the gravitational potential and its gradients of arbitrary order with respect to the Earth-fixed coordinates x, y and z. The source code of this computer programme is available on the web http://wwwrc.obs-azur.fr/cerga/mecanique/potential. We upgraded it from single to double floating point precision and from Fortran77 to Fortran90. In the following it will be referred to as *METRIS*.

6 The method of Clenshaw summation

The algorithm developed by Clenshaw and known under the name of *Clenshaw summation* is a general method for computing sums of products between indexed coefficients and functions which obey a three-term recurrence relation. It belongs to the class of downward recurrence formulas. A general description of the algorithm can be found in Press et al. (1992); its application to geodetic problems was discussed by Tscherning and Poder (1982). Here we provide a derivation of the Clenshaw summation algorithm applied to the evaluation of the geopotential and its first and second gradients.

Our implementation of the Clenshaw summation is based on the recursion relations (10) and (8) that form the backbone of the FCR algorithm for fnALFs. By considering terms of equal order (therefore dropping the subscript m), the functions p_n are defined as

$$p_n = \overline{P}_{nm}(u)\left(\frac{a}{r}\right)^n \tag{24}$$

in which $u = \sin\varphi$. After some algebraic manipulation, the three-term recurrence relation (10) becomes

$$p_n = \alpha_n p_{n-1} - \beta_n p_{n-2} \tag{25}$$

with initialization $p_0 = 1$ and $p_1 = \alpha_1$, where $\alpha_n = u g_{nm} a/r$ and $\beta_n = (a/r)^2 h_{nm}$, with g_{nm} and h_{nm} given by Eq. (11). For every value of m there exists a set of coefficients $y_n^{(i)}$ such that

$$\begin{aligned} y_{N+1}^{(i)} &= 0 \\ y_N^{(i)} &= c_N^{(i)} \\ &\dots \\ y_n^{(i)} &= c_n^{(i)} + \alpha_{n+1} y_{n+1}^{(i)} - \beta_{n+2} y_{n+2}^{(i)} \\ \dots \\ y_m^{(i)} &= c_m^{(i)} + \alpha_{m+1} y_{m+1}^{(i)} - \beta_{m+2} y_{m+2}^{(i)} \end{aligned} \tag{26}$$

where

$$c_n^{(i)} = \begin{cases} \overline{C}_{nm} \text{ if } i = 1 \\ \overline{S}_{nm} \text{ if } i = 2 \end{cases} \tag{27}$$

On defining the functions $v_m^{(i)}$ as

$$v_m^{(i)} = \sum_{n=m}^{N} c_n^{(i)} p_n, \tag{28}$$

it can be shown that

$$v_m^{(i)} = y_m^{(i)} p_m \tag{29}$$

and

$$V(r,\varphi,\lambda) = \frac{GM}{r}\sum_{m=0}^{N}\left(v_m^{(1)}\cos m\lambda + v_m^{(2)}\sin m\lambda\right). \tag{30}$$

Note that only the sectorial fnALFs need to be computed and stored, and each term in the sum over the orders [Eq. (30)] is the product of only two factors, i.e., the fnALF of degree m and order m and the output $y_m^{(i)}$ of the recursion (26). This makes the evaluation of the gravitational potential very efficient.

The first and second order partial derivatives of the gravitational potential V with respect to the local spherical coordinates λ, φ and r of the observation point P can be written in a form similar to Eq. (30) by introducing recursions on appropriate "y" coefficients (Table 4) and defining the corresponding "v" terms (Table 5). Note that some partial derivatives with respect to λ make use of y coefficients and v terms specifically developed for other derivatives. The final expressions for the first and second order partial derivatives of V with respect to spherical coordinates are given in Table 6.

Table 4: Seed values (first and second column) and general three-term recursion (third column) for each set of y coefficients. The subscript $_{..,uu}$ means for example that the corresponding coefficient is employed in the computation of the second derivative of V with respect to u. The quantity q replaces the ratio a/r.

$y_{N+1}^{(i)} = 0$	$y_N^{(i)} = c_N^{(i)}$	$y_n^{(i)} = c_n^{(i)} + uqg_{n+1,n}y_{n+1}^{(i)} - q^2h_{n+2,m}y_{n+2}^{(i)}$
$y_{N+1,r}^{(i)} = 0$	$y_{N,r}^{(i)} = c_N^{(i)}(N+1)$	$y_{n,r}^{(i)} = c_n^{(i)}(n+1) + uqg_{n+1,m}y_{n+1,r}^{(i)} - q^2h_{n+2,m}y_{n+2,r}^{(i)}$
$y_{N+1,u}^{(i)} = 0$	$y_{N,u}^{(i)} = 0$	$y_{n,u}^{(i)} = qg_{n+1,m}y_{n+1,u}^{(i)} + uqg_{n+1,m}y_{n+1}^{(i)} - q^2h_{n+2,m}y_{n+2,u}^{(i)}$
$y_{N+1,uu}^{(i)} = 0$	$y_{N,uu}^{(i)} = 0$	$y_{n,uu}^{i} = 2qg_{n+1,m}y_{n+1,u}^{(i)} + uqg_{n+1,m}y_{n+1,uu}^{(i)} - q^2h_{n+2,m}y_{n+2,uu}^{(i)}$
$y_{N+1,ur}^{(i)} = 0$	$y_{N,ur}^{(i)} = 0$	$y_{n,ur}^{(i)} = qg_{n+1,m}y_{n+1,r}^{(i)} + uqg_{n+1,m}y_{n+1,ur}^{(i)} - q^2h_{n+2,m}y_{n+2,ur}^{(i)}$
$y_{N+1,rr}^{(i)} = 0$	$y_{N,rr}^{(i)} = c_N^{(i)}(N+1)(N+2)$	$y_{n,r}^{(i)} = c_n^{(i)}(n+1)(n+2) + uqg_{n+1,m}y_{n+1,rr}^{(i)} - q^2h_{n+2,m}y_{n+2,rr}^{(i)}$

Table 5: Definition of the v terms involved in the sums over the orders to obtain V and its derivatives. The subscript $_{..,uu}$ means for example that the corresponding term is employed in the computation of the second derivative of V with respect to u.

$v_m^{(i)} = y_m^{(i)}p_m$	$v_{m,u}^{(i)} = y_m^{(i)}\dfrac{dp_m}{du} + y_{m,u}^{(i)}p_m$
$v_{m,r}^{(i)} = y_{m,r}^{(i)}p_m$	$v_{m,\lambda u}^{(i)} = m\left(y_{m,u}^{(i)}p_m + y_m^{(i)}\dfrac{dp_m}{du}\right)$
$v_{m,uu}^{(i)} = y_{m,uu}^{(i)}p_m + \dfrac{d^2p_m}{du^2}y_m^{(i)} + 2y_{m,u}^{(i)}\dfrac{dp_m}{du}$	$v_{m,\lambda r}^{(i)} = my_{m,r}^{(1)}p_m$
$v_{m,ur}^{(i)} = y_{m,r}^{(1)}\dfrac{dp_m}{du} + y_{m,ur}^{(1)}p_m$	$v_{m,rr}^{(i)} = y_{m,rr}^{(i)}p_m$

We implemented this algorithm into a Fortran90 computer programme named *CLENSHAW* that, given the fully normalized Stokes coefficients of the gravity model and

Table 6: The geopotential V and its first and second order partial derivatives with respect to spherical coordinates according to the Clenshaw summation scheme.

$V = \frac{\mu}{r}\sum_{m=0}^{N}\left(v_m^{(1)}\cos m\lambda + v_m^{(2)}\sin m\lambda\right)$	$\partial_\lambda V = -\frac{\mu}{r}\sum_{m=0}^{N} m(v_m^{(1)}\sin m\lambda - v_m^{(2)}\cos m\lambda)$
$\partial_\varphi V = \frac{\mu\cos\varphi}{r}\sum_{m=0}^{N}\left(v_{m,u}^{(1)}\cos m\lambda + v_{m,u}^{(2)}\sin m\lambda\right)$	$\partial_r V = -\frac{\mu}{r^2}\sum_{m=0}^{N}\left(v_{m,r}^{(1)}\cos m\lambda + v_{m,r}^{(2)}\sin m\lambda\right)$
$\partial_{\lambda\lambda} V = -\frac{\mu}{r}\sum_{m=0}^{N} m^2(v_m^{(1)}\cos m\lambda + v_m^{(2)}\sin m\lambda)$	$\partial_{\lambda\varphi} V = -\frac{\mu\cos\varphi}{r}\sum_{m=0}^{N}\left(v_{m,\lambda u}^{(1)}\sin m\lambda - v_{m,\lambda u}^{(2)}\cos m\lambda\right)$
$\partial_{\varphi\varphi} V = \frac{\mu\cos^2\varphi}{r}\sum_{m=0}^{N}\left(v_{m,uu}^{(1)}\cos m\lambda + v_{m,uu}^{(2)}\sin m\lambda\right) - \frac{\mu\sin\varphi}{r}\sum_{m=0}^{N}\left(v_{m,u}^{(1)}\cos m\lambda + v_{m,u}^{(2)}\sin m\lambda\right)$	$\partial_{\lambda r} V = \frac{\mu}{r^2}\sum_{m=0}^{N}\left(v_{m,\lambda r}^{(1)}\sin m\lambda - v_{m,\lambda r}^{(2)}\cos m\lambda\right)$
$\partial_{\varphi r} V = -\frac{\mu\cos\varphi}{r^2}\sum_{m=0}^{N}\left(v_{m,ur}^{(1)}\cos m\lambda + v_{m,ur}^{(2)}\sin m\lambda\right)$	$\partial_{rr} V = \frac{\mu}{r^3}\sum_{m=0}^{N}\left(v_{m,rr}^{(1)}\cos m\lambda + v_{m,rr}^{(2)}\sin m\lambda\right)$

the local spherical coordinates λ_k, φ_k, r_k of K observation points P_k ($k = 1, ..., K$ and $K \geq 1$), determines the gravitational potential and its first and second gradients. Best performance is reached by first evaluating and storing the normalization coefficients f_n, g_{nm} and h_{nm}, and then computing the latitude-dependent quantities, i.e., the sectorials $\overline{P}_{mm}$, their derivatives $d\overline{P}_{mm}/du$ and $d^2\overline{P}_{mm}/du^2$ and the various recursions on the y coefficients (Table 4); then, for each λ_k the sums listed in Table 6 are accumulated. This arrangement offers higher efficiency when dealing with data distributed on a spherical grid and accessed per parallel.

7 Numerical tests

We performed a set of numerical tests on the following computer programmes:

1a. *LEGENDRE_a* (Fantino and Casotto, FC) implementing the FCR algorithm based on fnALFs in Fortran90 (Section 3);

1b. *LEGENDRE_b* (PoliMi) implementing the FCR algorithm based on fnALFs as a DLL running under Matlab (Section 3);

2. *PINES* (FC) implementing the FCR algorithm based on fnHPs in Fortran90 (Section 4);

3. *METRIS* (Metris et al.) implementing a downward recursion on solid spherical harmonics in Fortran77 (Section 5);

4. *CLENSHAW* (FC) implementing the Clenshaw summation on fnALFs in Fortran90 (Section 6).

All programmes have been provided with an interface which allows to select the simulation parameters: the intervals (N_1, N_2) and (M_1, M_2) of spherical harmonic degrees and orders to be considered, the gravity model (as a set of fully normalized Stokes coefficients) and the type of input for the observation points, to be chosen between:

- spherical grid at constant heigth h, extending over the longitude and latitude intervals (λ_s, λ_f) and (φ_s, φ_f) with resolution $\Delta\lambda$ and $\Delta\varphi$ respectively;
- previously computed positions over a satellite orbit and provided in the form of binary input data.

Three simulation configurations have been adopted:

a) GRID: spherical grid of 64440 uniformly distributed observation points; simulation parameters: $h = 250\ km$, $\lambda_s = -179^o$, $\lambda_f = 180^o$, $\Delta\lambda = 1^o$, $\varphi_s = -89^o$, $\varphi_f = 89^o$, $\Delta\varphi = 1^o$; gravity model: EGM96; $N_1 = 0$, $N_2 = 360$, $M_1 = 0$, $M_2 = 360$; simulated quantities: V, the three physical components of ∇V and the six physical components of $\nabla\nabla V$;

b) ORBIT1: positions over a simulated orbit (Fig. 4) with semimajor axis of 6628 km, eccentricity of 0.002, inclination of 96.5^o - corresponding to the orbital parameters of the GOCE mission (ESA, 2000; Albertella et al., 2002) - and angular elements precessing due to J_2; the considered mission extends over $2 \cdot 10^6$ s and is sampled at an interval of 31 s; gravity model: EGM96; $N_1 = 0$, $N_2 = 360$, $M_1 = 0$, $M_2 = 360$; number of field points: 64440; simulated quantities: V, the three physical components of ∇V and the six physical components of $\nabla\nabla V$;

c) ORBIT2: same as ORBIT1 but with a sampling rate of 1 observation per second, as in the GOCE mission, resulting in $2 \cdot 10^6$ simulation points.

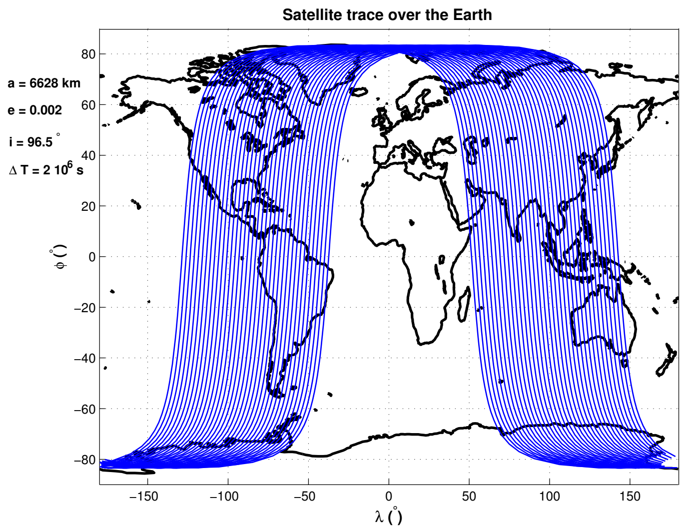

Figure 4: Ground track of a simulated satellite orbit over the surface of the Earth.

All numerical tests have been run on a Pentium IV personal computer with 500 MB of RAM and 1.3 GHz of processor speed.

Following the example of Holmes and Featherstone (2002), we have assessed the relative merits of the five programmes by testing their efficiency, precision and accuracy:

Table 7: Execution times of the five codes for the three chosen simulation scenarios. Note that *LEGENDRE_b* is interpreted, while the remaining software codes are all compiled; this may explain its low performance. The execution times of ORBIT2 have been extrapolated by scaling the performance of ORBIT1 according to the different number of observation points.

CPU execution times						
	GRID		ORBIT1		ORBIT2	
Programme	seconds	hours	seconds	hours	seconds	hours
LEGENDRE_a (FC)	2390	0.7	2079	0.6	64449	17.9
LEGENDRE_b (PoliMi)	9793	2.7	9975	2.8	309225	85.9
PINES (FC)	5275	1.5	6306	1.8	195486	54.3
METRIS (Metris et al.)	3271	0.9	3418	0.9	105958	29.4
CLENSHAW (FC)	8	-	992	0.3	30752	8.5

the tests of efficiency compare the execution (CPU) times; the tests of numerical precision compare the numerical values of the gravitational potential V, the three components of ∇V and the six components of $\nabla\nabla V$ as computed in quadruple (REAL*16) and double (REAL*8) floating point precision; the tests of accuracy employ exact analytic identities to perform self-validation of each individual code.

7.1 Numerical efficiency

The execution times of the first two simulation scenarios (i.e., GRID and ORBIT1), supplemented by extrapolated estimates for the CPU times of the third case (ORBIT2), are shown in Table 7, which gives the performance of each algorithm in seconds and hours. These execution times include I/O operations, which, however, are arranged in an efficient way: the gravity tensor quantities are stored in binary files made up of 179 records of 360 double precision numbers each and the output is updated per record. The use of binary output is efficient also from the viewpoint of storage space. The performance of the algorithm of Clenshaw on the GRID is three orders of magnitude better than all the others: this is due to the very favourable arrangement of the computations, which requires partial sums over the orders only.

7.2 Numerical precision

The numerical agreement among the results produced by each code has been verified by first expressing all tensor components in the same coordinate system: since three codes provide results in spherical coordinates and two work in cartesian coordinates, we have chosen to transform from cartesian to spherical cooordinates (Tables 1 and 2).

The assessment of numerical precision has been obtained by compiling the method of Clenshaw in quadruple floating point precision (*CLENSHAW**16) and then computing the relative differences with each algorithm in double floating point precision (i.e., *LEGENDRE_a**8, *LEGENDRE_b**8, *PINES**8, *METRIS**8 and *CLENSHAW**8). The adopted simulation configuration is GRID. The number of coinciding decimal digits in each pair is shown in Figs. 5 to 9: the level of agreement within each pair

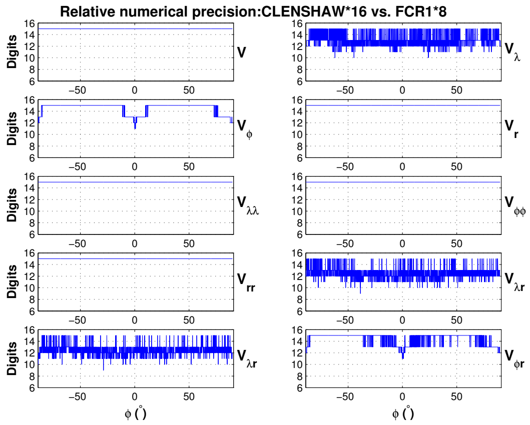

Figure 5: Relative numerical precision between *CLENSHAW*∗16 and *LEGENDRE_a*∗8 (denoted FCR1∗8 in the plot).

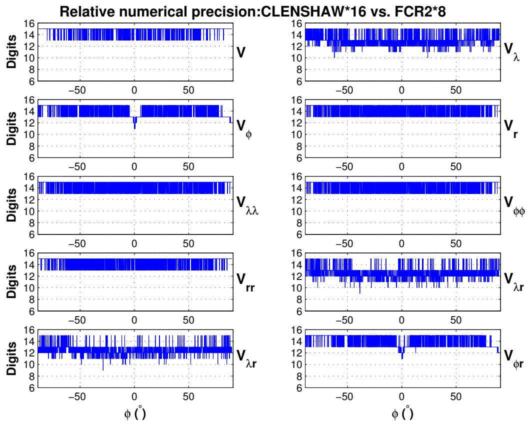

Figure 6: Relative numerical precision between *CLENSHAW*∗16 and *LEGENDRE_b*∗8 (denoted FCR2∗8 in the plot).

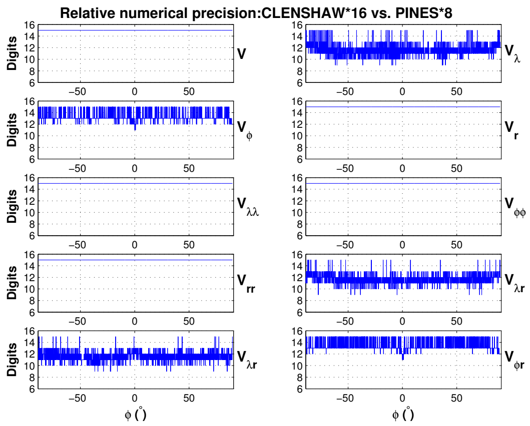

Figure 7: Relative numerical precision between *CLENSHAW*∗16 and *PINES*∗8.

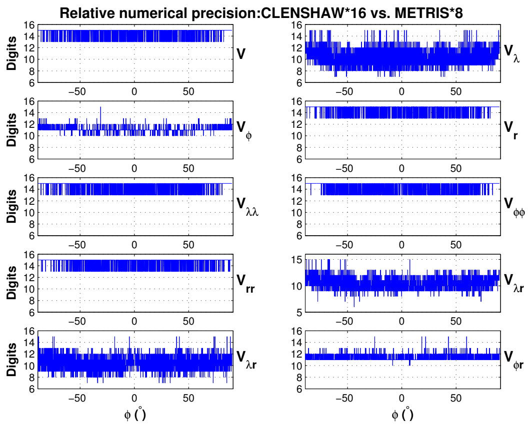

Figure 8: Relative numerical precision between *CLENSHAW*∗16 and *METRIS*∗8.

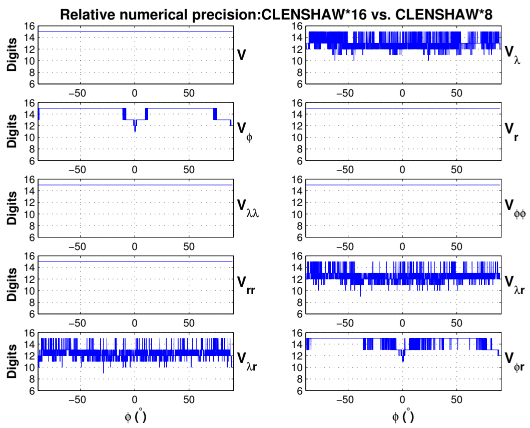

Figure 9: Relative numerical precision between *CLENSHAW*∗16 and *CLENSHAW*∗8.

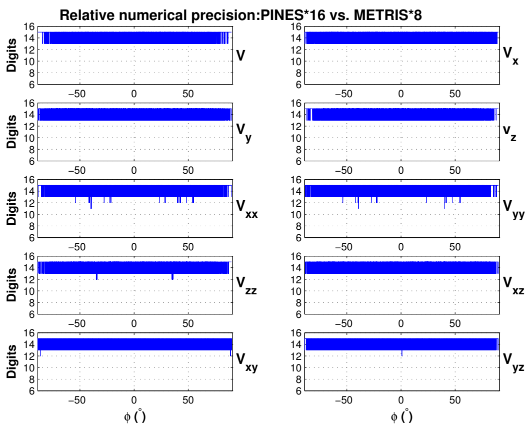

Figure 10: Relative numerical precision between *PINES*∗16 and *METRIS*∗8.

shows variations with the specific tensor component considered and ranges from 10 to 15 decimal digits in the case of the algorithms developed in spherical coordinates (*LEGENDRE_a**8, *LEGENDRE_b**8 and *CLENSHAW**8), while it can get as low as 7 decimal digits in the case of *METRIS**8. We believe that the poor agreement between *CLENSHAW**16 and *METRIS**8 is not due to error propagation resulting from the coordinate transformations. However, a direct comparison between *METRIS**8 and *PINES**8 shows that the two algorithms are in very good agreement (Fig. 10). We believe that the issue of numerical precision should be further investigated.

7.3 Numerical accuracy

The accuracy of the computations performed by each programme can be verified by computing the values of exact identities incorporating specific tensor components. The most important and well known of these is the Laplace identity:

$$\sum_{\alpha=1}^{3} V_{\alpha\alpha} = 0 \tag{31}$$

which relates the diagonal elements of the second gravity gradient tensor. Other exact relations involve finite sums of the ALFs and their derivatives; for example:

$$\sum_{n=0}^{N}\sum_{m=0}^{n}\left[\overline{P}_{nm}(\varphi)\right]^2 = (N+1)^2 \tag{32}$$

and

$$\sum_{n=0}^{N}\sum_{m=0}^{n}\left[\frac{\partial\overline{P}_{nm}(\varphi)}{\partial\varphi}\right]^2 = \frac{N(N+2)(N+1)^2}{4}. \tag{33}$$

We evaluated Eq. (31) with all five computer codes and Eqs. (32) and (33) with *LEGENDRE_a* only. We found that all algorithms verify Eq. (31) with results of the order of 10^{-18} or 10^{-19}; *LEGENDRE_a* reproduces the identities (32) and (33) with accuracies at the level of 10^{-14} and 10^{-13} respectively.

8 Conclusions

We have described, tested and compared four numerical algorithms for spherical harmonic synthesis applied to the evaluation of the gravitational potential and its first and second gradients.

The method of Pines is computationally heavy also when running on a spherical grid: this is due to the large number of longitude-dependent auxiliary functions to be evaluated. The apparently low performance of *LEGENDRE_b* both on the grid and on the orbit might simply be an effect of the mode (interpreted rather than compiled) in which it runs. The performance of the algorithm of Clenshaw on the GRID is three orders of magnitude better than all the others: this is due to the very favourable arrangement of the computations, which requires partial sums over the orders only. The precision and accuracy of the methods considered is good in most cases, although some aspects of the tests on numerical precision deserve to be investigated further.

We conclude that the algorithm of Clenshaw is by far the most efficient and therefore the most suitable for implementation and use when dealing with large amounts of observations.

Acknowledgements

We thank F. Sansò and M. Reguzzoni for providing their software and for helpful suggestions.

References

Abramowitz M, Stegun IA (1964) Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Dover Publications, New York

Albertella A, Migliaccio F, Sansò F (2002) GOCE: the Earth gravity field by space gradiometry. Cel Mech & Dynam Astron 83: 1-15

Balmino G, Barriot JP, Valès N (1990) Non-singular formulation of the gravity vector and gravity gradient tensor in spherical harmonics. Manuscr Geod 15: 11-16

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

Cunningham LE (1970) On the computation of the spherical harmonic terms needed during the numerical integration of the orbital motion of an artificial satellite. Cel Mech & Dynam Astron 2: 207-216

Ditmar P, Klees R (2002) A Method to Compute the Earth's Gravity Field from SGG/SST data to be Acquired by the GOCE Satellite. Technical report

ESA (2000) From Eötvös to Milligal. Final Report

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

Hobson EW (1965) The theory of spherical and ellipsoidal harmonics. Chelsea Publishing Company, New York

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: 279-299

Ilk KH (1983) Ein Beitrag zur Dynamik ausgedehnter Körper - Gravitationswechselwirkung. PhD thesis, Bayerischen Akademie der Wissenschaften, Munchen

Metris G, Xu J, Wytrzyszczak I (1999) Derivatives of the gravity potential with respect to rectangular coordinates. Cel Mech & Dynam Astron 71: 137-151

Pines S (1973) Uniform representation of the Gravitational Potential and its Derivatives. AIAA Journal 11/11: 1508-1511

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical Recipes in Fortran 77. Cambridge University Press

Spencer JL (1976) Pines' nonsingular gravitational potential derivation, description and implementation. Technical report, NASA Contract No. 9-13970

Tscherning CC, Poder K (1982) Some Geodetic applications of Clenshaw summation. Boll Geod Sci Aff, XLI/4: 349-375

Xu G (2003) GPS - Theory, Algorithms and Applications. Springer-Verlag, Berlin