Journal of Geodesy (2002) 76: 279–299
DOI 10.1007/s00190-002-0216-2

Journal of Geodesy

A unified approach to the Clenshaw summation and the recursive computation of very high degree and order normalised associated Legendre functions

S. A. Holmes, W. E. Featherstone

Department of Spatial Sciences, Curtin University of Technology, GPO Box U1987, Perth, WA 6845, Australia
e-mail: holmes@vesta.curtin.edu.au/w.featherstone@curtin.edu.au; Tel.: +61-8-9266-2218; Fax: +61-8-9266-2703

Received: 30 June 2000 / Accepted: 12 June 2001

Abstract. Spherical harmonic expansions form partial sums of fully normalised associated Legendre functions (ALFs). However, when evaluated increasingly close to the poles, the ultra-high degree and order (e.g. 2700) ALFs range over thousands of orders of magnitude. This causes existing recursion techniques for computing values of individual ALFs and their derivatives to fail. A common solution in geodesy is to evaluate these expansions using Clenshaw's method, which does not compute individual ALFs or their derivatives. Straightforward numerical principles govern the stability of this technique. Elementary algebra is employed to illustrate how these principles are implemented in Clenshaw's method. It is also demonstrated how existing recursion algorithms for computing ALFs and their first derivatives are easily modified to incorporate these same numerical principles. These modified recursions yield scaled ALFs and first derivatives, which can then be combined using Horner's scheme to compute partial sums, complete to degree and order 2700, for all latitudes (except at the poles for first derivatives). This exceeds any previously published result. Numerical tests suggest that this new approach is at least as precise and efficient as Clenshaw's method. However, the principal strength of the new techniques lies in their simplicity of formulation and implementation, since this quality should simplify the task of extending the approach to other uses, such as spherical harmonic analysis.

Keywords: Spherical Harmonic Expansions – Fully Normalised Associated Legendre Functions – Clenshaw Summation – Recursion – Horner's Scheme

1 Introduction

Current geodetic practice is witnessing an increase in the construction and use of ultra-high-degree spherical harmonic expansions of the geopotential or topography. For example, Wenzel (1998) released coefficients up to degree 1800, which were empirically derived to describe the gravitational potential of the Earth. Wenzel (1998) states that the maximum degree of 1800 for the spherical harmonic model was set by the numerical stability of the recursion algorithm adopted to compute the required fully normalised associated Legendre functions (ALFs).

The recent interest in synthetic Earth gravity models, used for comparing and validating gravity field determination techniques, has already seen the use of ultra-high-degree spherical harmonic expansions. These have taken the form of simple effects models (see e.g. Featherstone 1999; Novák et al. 2001) for which synthetic geopotential coefficients up to degree and order 2700 and 2160, respectively, were produced without reference to a mass distribution. There is also interest in source models in which synthetic geopotential coefficients are generated by analytical or numerical Newtonian integration over a synthetic global density distribution and topography (see e.g. Pail 1999). Hybrids of source and effects models also exist. For example, Haagmans (2000) combines empirically determined coefficients with synthetic ones derived from numerical integration over isostatically compensated source masses to degree and order 2160. Lastly, other scientific disciplines, such as meteorology, quantum physics and electronic engineering, are also showing increased interest in high-degree spherical harmonic modelling and analysis.

The numerical means for including the necessary ALFs constitutes the principal challenge to evaluating ultra-high-degree spherical harmonic expansions. Therefore, it is timely to critically examine the accuracy and numerical efficiency of algorithms that compute individual ALFs and their partial sums.

1.1 Spherical harmonic expansions

Truncated spherical harmonic expansions of a function, or its derivatives, reduce to sums $S^{(d)}$ of ALFs, or their derivatives, respectively. These are

Correspondence to: S.A. Holmes

$$S^{(d)} = c \sum_{m=0}^{M} \Omega_m^{(d)} \tag{1}$$

where

$$\Omega_m^{(d)} = \sum_{\alpha=1}^{2} \begin{cases} X_{m\alpha}^{(d)} \cos m\lambda & \text{for } \alpha = 1 \\ X_{m\alpha}^{(d)} \sin m\lambda & \text{for } \alpha = 2 \end{cases} \tag{2}$$

and

$$X_{m\alpha}^{(d)} = \sum_{n=\mu}^{M} \bar{E}_{nm\alpha} \bar{P}_{nm}^{(d)}(\theta) \tag{3}$$

For arguments of spherical polar coordinates (r, λ, θ) and for integer degree $n \geq 0$ and order $0 \leq m \leq n$: M is the maximum finite degree of the spherical harmonic expansion; μ is an integer that may vary with m; c is a real-numbered constant; $\bar{E}_{nm\alpha}$ is a real number incorporating the fully normalised spherical harmonic coefficients, $\bar{C}_{nm1}$ and $\bar{C}_{nm2}$; $\bar{P}_{nm}(\theta)$ are the fully normalised ALFs; the superscript (d) indicates the dth derivative with respect to θ, or definite integration ($d = -1$) between two parallels. This paper deals only with undifferentiated functions ($d = 0$) or first derivatives of these functions ($d = 1$). For $d = 0$, the superscript (d) is omitted. Thus $S^{(0)}$, $\Omega_m^{(0)}$, $X_{m\alpha}^{(0)}$ and $\bar{P}_{nm}^{(0)}(\theta)$ are written S, Ω_m, $X_{m\alpha}$ and $\bar{P}_{nm}$, respectively. First derivatives of these quantities are written $S^{(1)}$, $\Omega_m^{(1)}$, $X_{m\alpha}^{(1)}$ and $\bar{P}_{nm}^{(1)}(\theta)$, respectively. The general notation $S^{(d)}$, $\Omega_m^{(d)}$, $X_{m\alpha}^{(d)}$ and $\bar{P}_{nm}^{(d)}(\theta)$ is used whenever a textual or mathematical reference applies to both the undifferentiated quantities and the first derivatives simultaneously.

The example of a truncated spherical harmonic expansion of the gravitational potential $V(r, \theta, \lambda)$ is instructive here. Often, it is written as

$$V(r,\theta,\lambda) = \frac{GM}{r} + \frac{GM}{r} \sum_{n=2}^{M} \left(\frac{a}{r}\right)^n \sum_{m=0}^{n} (\bar{C}_{nm1} \cos m\lambda + \bar{C}_{nm2} \sin m\lambda) \bar{P}_{nm}(\theta) \tag{4}$$

where GM is the product of the Universal gravitational constant and the mass of the Earth. Alternatively, Eq. (4) may be written as

$$V(r,\theta,\lambda) = \frac{GM}{r} + \frac{GM}{r} \sum_{m=0}^{M} \left[\cos m\lambda \sum_{n=\mu}^{M} \left(\frac{a}{r}\right)^n \bar{C}_{nm1} \bar{P}_{nm}(\theta) + \sin m\lambda \sum_{n=\mu}^{M} \left(\frac{a}{r}\right)^n \bar{C}_{nm2} \bar{P}_{nm}(\theta) \right] \tag{5}$$

where μ is either 2 or m, whichever is the greater. Relating Eq. (5) to the form of Eqs. (1)–(3) yields

$$\bar{E}_{nm\alpha} = \begin{cases} \left(\frac{a}{r}\right)^n \bar{C}_{nm1}, & \text{for } \alpha = 1 \\ \left(\frac{a}{r}\right)^n \bar{C}_{nm2}, & \text{for } \alpha = 2 \end{cases} \tag{6}$$

and

$$X_{m\alpha} = \sum_{n=\mu}^{M} \left(\frac{a}{r}\right)^n \bar{C}_{nm\alpha} \bar{P}_{nm}(\theta) \tag{7}$$

When evaluating gravimetric quantities (e.g. disturbing potential, geoid heights, gravity anomalies, etc.) in a sequence of points for which r and θ are constant (i.e. along a geodetic parallel), the form of Eq. (5) is numerically more efficient than that of Eq. (4) (cf. Tscherning et al. 1983). This is because each $X_{m\alpha}$ in Eq. (3) is independent of λ, and thus need only be evaluated once for each parallel. If all such computation points are equally spaced in longitude, further numerical efficiencies can be achieved through application of the recursion algorithm developed by Rizos (1979). Abd-Elmotaal (1997) contains a re-derivation of this algorithm which demonstrates that, contrary to the approach of Rizos (1979), the algorithm can be applied in full without prior rotation of the geopotential coefficients.

1.2 Numerical considerations when evaluating Eq. (1)

The simplest approach to evaluating Eq. (1) is to use standard recursion relations, such as those found in Colombo (1981) and described in Sect. 2.1 of this paper, to compute the required values of $\bar{P}_{nm}^{(d)}(\theta)$. These values can then be multiplied by the corresponding values of $\bar{E}_{nm\alpha}$ to yield the intermediate values for $X_{m\alpha}^{(d)}$ in Eq. (3), and hence $\Omega_m^{(d)}$ in Eq. (2), which are then used to compute the final sums $S^{(d)}$ in Eq. (1).

The principal problem with this approach is that, for ultra-high values of M (e.g. 2700), the absolute values of $\bar{P}_{nm}(\theta)$ will range over thousands of orders of magnitude. For example, Fig. 1 shows that, for $M = 2700$, $|\bar{P}_{nm}^{(d)}(\theta)|$ ranges over $\sim$5000 orders of magnitude towards the poles ($\theta \to 0^\circ$ or $\theta \to 180^\circ$, i.e. $\cos\theta \to 1$). This is impractical, because the Institute of Electrical and Electronic Engineers' (IEEE) standard 754 for binary floating-point arithmetic (cf. Coonen 1980) only allocates eight bytes to store each double precision floating-point number (R). Thus, $|R|$ may only take values within the range of $\sim 10^{-310} < |R| < \sim 10^{310}$. Any computed value where $|R| < \sim 10^{-310}$ will

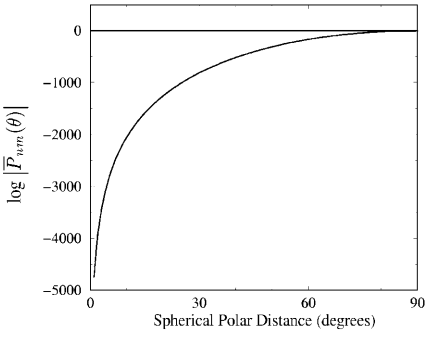

Fig. 1. Logarithm plot of maximum (*upper line*) and minimum (*lower line*) values of $|\bar{P}_{nm}(\theta)|, \forall n, m \leq 2700$

'underflow' and be set to zero, whilst any computed values for which $|R| > \sim 10^{310}$ will 'overflow' and be designated 'not a number' (NaN), such that any subsequent computation which employs this R will also be so designated.

Underflow in the computation of any $\bar{P}_{nm}^{(d)}(\theta)$ excludes the corresponding (matching degree and order) coefficients from contributing to $S^{(d)}$, whereas an overflow in the computation of one or more $\bar{P}_{nm}^{(d)}(\theta)$ prevents any result for $S^{(d)}$ from being achieved at all. Thus, IEEE double precision only permits a maximum range of $\sim$620 orders of magnitude within which to compute and store the required $\bar{P}_{nm}^{(d)}(\theta)$ values. Figure 1 shows that, for high latitudes, the range of values taken by the $\bar{P}_{nm}(\theta)$ values for $M = 2700$ will eventually exceed the range of magnitudes capable of being stored within the IEEE double-precision format. Scaling all of the computations upwards by a factor of 10^{200} allowed Wenzel (1998) to compute values of $\bar{P}_{nm}(\theta)$ to a maximum $M = 1900$ for $\sim 20^\circ \leq \theta \leq \sim 160^\circ$ in IEEE double precision.

A numerically more stable alternative to these standard recursion relations is the Clenshaw (1955) method (cf. Tscherning and Poder 1982; Gleason 1985; Deakin 1998). In Sect. 3, it will be shown that this can be used to evaluate S in Eq. (1) up to $M = 2700$ $(0^\circ \leq \theta \leq 180^\circ)$, as well as $S^{(1)}$ up to $M = 2700$ $(0^\circ < \theta < 180^\circ)$. Standard derivations of Clenshaw's method (cf. Gleason 1985; Deakin 1998) utilise matrix algebra, and generally focus on the means by which this method can be used to evaluate partial sums, $S^{(d)}$, without computing individual values of $\bar{P}_{nm}^{(d)}(\theta)$. Such derivations are complete, concise and rigorous, but they also obscure the numerical principles upon which the stability of the Clenshaw summation is based. These principles are quite simple, both in concept and in application.

This paper shows how existing algorithms for computing ALFs and first derivatives are easily modified to incorporate these same numerical principles. The modified algorithms can be used to compute scaled ALFs and their first derivatives, which can then be combined using Horner's scheme (cf. Harris and Stocker 1998) to yield values for the required partial sums, S in up to $M = 2700$ $(0^\circ \leq \theta \leq 180^\circ)$, as well as $S^{(1)}$ up to $M = 2700$ $(0^\circ < \theta < 180^\circ)$. Straightforward examples and elementary algebra are then used to illustrate the means by which these numerical principles are implemented in Clenshaw's method.

Results from numerical tests, presented in Sect. 4, suggest that the modified algorithms are at least as efficient and precise as the standard Clenshaw techniques for evaluating partial sums of $\bar{P}_{nm}(\theta)$ or $\bar{P}_{nm}^{(1)}(\theta)$. However, it is the intuitive simplicity of the new approaches, as well as the fact that they compute individual scaled values of $\bar{P}_{nm}^{(d)}(\theta)$, which constitutes their principal strength over the standard Clenshaw methods. These two properties should simplify the extension of these new approaches to other tasks, such as stabilising current techniques for spherical harmonic analysis (e.g. Lesur and Gubbins 1999).

2 Forward recursions for the calculation of ALFs

The most direct approach for evaluating $S^{(d)}$ [Eq. (1)] employs a recursive algorithm to compute $\bar{P}_{nm}(\theta)$. Values of $\bar{P}_{nm}^{(1)}(\theta)$, if required, are then computed directly from two previously computed values of $\bar{P}_{nm}(\theta)$. These values of $\bar{P}_{nm}^{(d)}(\theta)$ are multiplied by the corresponding $\bar{E}_{nm\alpha}$ terms to yield the required series values of $X_{m\alpha}^{(d)}$ [Eq. (3)], which subsequently yield $\Omega_m^{(d)}$ [Eq. (2)] and hence $S^{(d)}$ [Eq. (1)].

The recursion relations for $\bar{P}_{nm}(\theta)$ can be obtained by fully normalising standard relations for (un-normalised) $P_{nm}(\theta)$, which can be found, for example, in Magnus et al. (1966) or Abramowitz and Stegan (1972). The full normalisation is given by [adapted from Heiskanen and Moritz 1967, Eq. (1–73)

$$\bar{P}_{nm}(\theta) \;=\; \sqrt{\frac{k(2n+1)(n-m)!}{(n+m)!}}\, P_{nm}(\theta) \tag{8}$$

where $k = 1$ for $m = 0$ and $k = 2$ for $m > 0$. Similarly, quasi-normalised values of $\tilde{P}_{nm}(\theta)$ are related to $P_{nm}(\theta)$ by (cf. Tscherning and Poder 1982)

$$\tilde{P}_{nm}(\theta) \;=\; \sqrt{\frac{(n-m)!}{(n+m)!}}\, P_{nm}(\theta) \tag{9}$$

Inspection of Eqs. (8) and (9) shows that

$$\bar{P}_{nm}(\theta) \;=\; \sqrt{k(2n+1)}\, \tilde{P}_{nm}(\theta) \tag{10}$$

The relationships in Eqs. (8), (9) and (10) also hold for all dth derivatives $P_{nm}^{(d)}(\theta)$, $\bar{P}_{nm}^{(d)}(\theta)$ and $\tilde{P}_{nm}^{(d)}(\theta)$. However, unlike $\bar{P}_{nm}(\theta)\left\{\begin{smallmatrix}\cos m\lambda\\ \sin m\lambda\end{smallmatrix}\right\}$, $\bar{P}_{nm}^{(d\neq 0)}(\theta)\left\{\begin{smallmatrix}\cos m\lambda\\ \sin m\lambda\end{smallmatrix}\right\}$ is not normalised in the correct usage of the word since its average squared value integrated over the unit sphere is not unity. This paper focuses solely on the computation of fully normalised ALFs and their first derivatives. No numerical tests were conducted for the above quasi-normalisations. However, it is a trivial task to apply Eq. (10) to the algorithms presented in this paper.

2.1 Standard forward column methods

The most popular recursive algorithm used for computing $\bar{P}_{nm}(\theta)$ in geodesy can be obtained by fully normalising, for example, Magnus et al. [1966, Eq. 4.3.3(2)]. This full normalisation yields a recursion that computes non-sectoral (i.e. $n > m$) $\bar{P}_{nm}(\theta)$ from previously computed $\bar{P}_{nm}(\theta)$. This recursion is given as (cf. Colombo 1981)

$$\bar{P}_{nm}(\theta) \;=\; a_{nm}\; t\; \bar{P}_{n-1,m}(\theta) - b_{nm}\; \bar{P}_{n-2,m}(\theta), \;\; \forall\; n > m \tag{11}$$

where $t = \cos\theta$

$$a_{nm} \;=\; \sqrt{\frac{(2n-1)(2n+1)}{(n-m)(n+m)}} \text{ and}$$

$$b_{nm} \;=\; \sqrt{\frac{(2n+1)(n+m-1)(n-m-1)}{(n-m)(n+m)(2n-3)}} \tag{12}$$

The sectoral (i.e. $n = m$) $\bar{P}_{mm}(\theta)$ serve as seed values for the recursion in Eq. (11). These are computed using the initial values $\bar{P}_{0,0}(\theta) = 1$ and $\bar{P}_{1,1}(\theta) = \sqrt{3}u$, where $u = \sin\theta$. The higher degree and order values of $\bar{P}_{mm}(\theta)$ are then computed using the recursion (cf. Colombo l981)

$$\bar{P}_{mm}(\theta) \;=\; u\,\sqrt{\frac{2m+1}{2m}}\,\bar{P}_{m-1,m-1}(\theta),\;\; \forall\; m > 1 \tag{13}$$

such that

$$\bar{P}_{mm}(\theta) \;=\; u^m\;\sqrt{3}\prod_{i=2}^{m}\sqrt{\frac{2i+1}{2i}}\;,\;\; \forall\; m \geq 1 \tag{14}$$

where $\prod$ is the product symbol (see e.g. Abramowitz and Stegan 1972).

The complete recursion process in Eqs. (11) and (13) may be visualised using the lower triangular matrix in Fig. 2, where each circle corresponds to a particular combination of n and m. Thus, each circle represents a value of $\bar{P}_{nm}(\theta)$, as well as the corresponding pair of recursive terms $(a_{nm}t)$ and b_{nm}. Note that in Fig. 2, the degree increases in rows down, the order increases in columns to the right, and the diagonal elements of the matrix are the sectoral values. The recursion in Eq. (11) computes $\bar{P}_{nm}(\theta)$ of constant m (a 'column' in Fig. 2) and sequentially increasing n [or down and away (i.e. 'forward') from the diagonal in Fig. 2]. Thus, Eq. (11) will be referred to as a standard forward column recursion. This nomenclature will be employed throughout the paper.

It appears from Eq. (11) that the computation of the first value of $\bar{P}_{m+1,m}(\theta)$ 'forward' from the sectoral diagonal (Fig. 2) requires a value of $\bar{P}_{m-1,m}(\theta)$ to be multiplied by the recursive term $b_{m+1,m}$. This $\bar{P}_{m-1,m}(\theta)$ does not exist for ordinary $\bar{P}_{nm}(\theta)$. However, the corresponding $b_{m+1,m}$ coefficient in Eq. (11) is always zero, thereby allowing the (non-existant) $\bar{P}_{m-1,m}(\theta)$ to be disregarded.

For a forward column computation of $\bar{P}^{(1)}_{nm}(\theta)$, normalisation of Magnus et al. [1966, Eq. 4.3.3(9)] gives (cf. Colombo 1981)

$$\bar{P}^{(1)}_{nm}(\theta) = \frac{1}{u}\left(nt\bar{P}_{nm}(\theta) \;-\; f_{nm}\bar{P}_{n-1,m}(\theta)\right),\;\; \forall\; n \geq m \tag{15}$$

where

$$f_{nm} \;=\; \sqrt{\frac{(n^2-m^2)(2n+1)}{(2n-1)}} \tag{16}$$

For all sectoral $\bar{P}^{(1)}_{mm}(\theta)$, $f_{mm} = 0$ and Eq. (15) reduces to

$$\bar{P}^{(1)}_{mm}(\theta) \;=\; m\;\frac{t}{u}\;\bar{P}_{mm}(\theta),\;\; \forall\; m \geq 0 \tag{17}$$

The first derivative of Eq. (13) with respect to θ also gives Eq. (17). For all $n \leq M$, each $\bar{P}_{nm}(\theta)$ of a given m can be computed using Eqs. (11) and (13). These values can be substituted into Eq. (15) to compute, without the need for any further recursion, all $\bar{P}^{(1)}_{nm}(\theta)$ of the same order m and $\forall\; n \geq m$.

2.2 Standard forward row methods

The next approach is termed the standard forward row recursion (Fig. 3), and appears to be rarely used in geodesy. As with the standard forward column recursion (Sect. 2.1), the sectoral $\bar{P}_{mm}(\theta)$ serve as seed values for the standard forward row recursion, and can be computed using Eq. (13). However, the standard forward row recursion computes non-sectoral $\bar{P}_{nm}(\theta)$ of constant n (a 'row' in Fig. 3) and sequentially decreasing m [to the left (i.e. 'forward') from the diagonal in Fig. 3]. Full normalisation of Magnus et al. [Eq. 4.3.3(1)] and substituting $\bar{P}_{nm}(\theta) = (-1)^m\bar{P}^m_n(\theta)$ yields

$$\bar{P}_{nm}(\theta) = \frac{1}{\sqrt{j}}\left(g_{nm}\;\frac{t}{u}\;\bar{P}_{n,m+1}(\theta) \;-h_{nm}\bar{P}_{n,m+2}(\theta)\right),\;\; \forall\; n > m \tag{18}$$

where $j = 2$ for $m = 0$ and $j = 1$ for $m > 0$, and

$$g_{nm} \;=\; \frac{2(m+1)}{\sqrt{(n-m)(n+m+1)}}$$

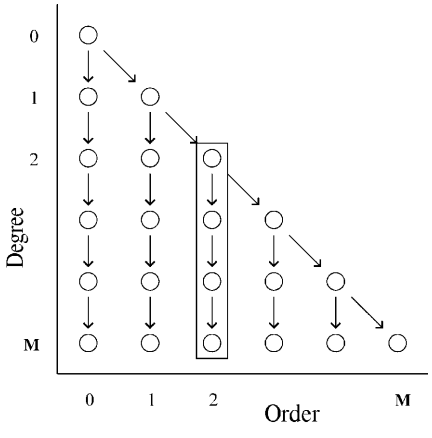

Fig. 2. Schematic of the recursion sequences employed in the standard, first modified and second modified forward column algorithms to compute $\bar{P}_{nm}(\theta)$, $(\bar{P}_{nm}(\theta))/u^m$ and $(\bar{P}_{nm}(\theta))/(P_{mm}(\theta))$ respectively

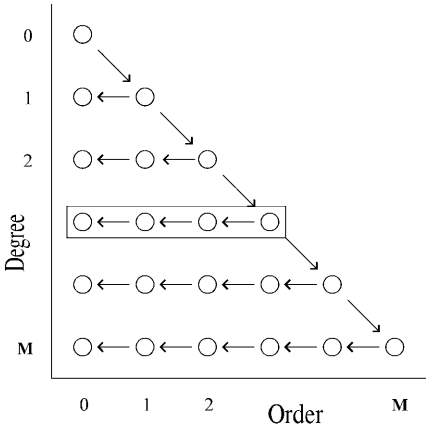

Fig. 3. Schematic of the recursion sequences employed in the standard and modified forward row algorithms to compute $\bar{P}_{nm}(\theta)$ and $(\bar{P}_{nm}(\theta))/u^m$, respectively

and

$$h_{nm} = \sqrt{\frac{(n+m+2)(n-m-1)}{(n-m)(n+m+1)}} \tag{19}$$

Using the same argument to that introduced for the forward column recursion, the non-existant value of $\bar{P}_{n,n+1}(\theta)$ required to compute $\bar{P}_{n,n-1}(\theta)$ in Eq. (18) may be disregarded because the corresponding recursion coefficient, $h_{n,n-1}$, is always zero.

Note that, to compute $\bar{P}_{nm}(\theta)$ using the forward row recursion, Eq. (18) uses the corresponding sectoral values of the same n, rather than the same m, as seed values. In this case, these sectoral values are more correctly denoted by $\bar{P}_{nn}(\theta)$, which may be written in the form of Eq. (14) as

$$\bar{P}_{nn}(\theta) = u^n \sqrt{3} \prod_{i=2}^{n} \sqrt{\frac{2i+1}{2i}}, \ \forall\ n \geq 1 \tag{20}$$

The $\bar{P}^{(1)}_{nm}(\theta)$ are obtained directly from $\bar{P}_{nm}(\theta)$ of matching n by fully normalising Abramowitz and Stegan [1972, Eq. (8.5.2)] to yield

$$\bar{P}^{(1)}_{nm}(\theta) = m\frac{t}{u}\bar{P}_{nm}(\theta) - e_{nm}\bar{P}_{n,m+1}, \ \forall\ n \geq m \tag{21}$$

where

$$e_{nm} = \sqrt{\frac{(n+m+1)(n-m)}{j}} \tag{22}$$

For all sectoral $\bar{P}^{(1)}_{mm}(\theta)$, or equivalently $\bar{P}^{(1)}_{nn}(\theta)$, $e_{mm} = 0$ and Eq. (21) reduces to Eq. (17).

2.3 Numerical problems with the standard forward methods

Even when applied in IEEE double precision, both the standard forward column [Eq. (11)] and standard forward row [Eq. (18)] recursions will underflow for $M > 1900$ in the co-latitude range $\sim 20^\circ \leq \theta \leq \sim 160^\circ$. The numerical instability of both these forward recursions is noted in the geodetic literature (e.g. Gleason 1985) and elsewhere (e.g. Libbrecht 1985). The cause of this instability is revealed by examining Eq. (14), which is first partitioned into the factors u^m and Π_m, such that

$$\bar{P}_{mm}(\theta) = u^m\ \Pi_m, \ \forall\ m \geq 1 \tag{23}$$

where $u^m = \sin^m \theta$, and

$$\Pi_m = \sqrt{3} \prod_{i=2}^{m} \sqrt{\frac{2i+1}{2i}}\ \forall\ m \geq 1 \tag{24}$$

Inspection of Fig. 4 shows that the Π_m factor introduces no computational difficulties for an arbitrarily ultra-high value of $m = 5400$. In contrast, the u^m term in Eq. (23) becomes increasingly small as $u \rightarrow 0$ (i.e. towards the poles) and as m increases. Accordingly, the high degree and order values of $\bar{P}_{mm}(\theta)$ will exceed the range of magnitudes capable of being stored in IEEE double precision, thereby resulting in an underflow. The failure to compute and store values of $\bar{P}_{mm}(\theta)$ means that these cannot serve as seed values for the standard forward column and forward row recursions (Sect. 2.1 and 2.2, respectively). This ultimately limits the ranges of θ and M over which these recursions can be used, thereby restricting the practical application of spherical harmonic expansions of ultra-high M at high latitudes.

2.4 Other normalisations and the Edmonds recursion

Belikov (1991) and Belkov and Taybatorov (1991) present a suite of recursive algorithms for computing the quantities $\hat{P}^{(d)}_{nm}$, where $\hat{P}^{(d)}_{nm}$ are related to unnormalised $P^{(d)}_{nm}$ according to the modified normalisation

$$\hat{P}_{nm}(\theta) = 2^m \frac{n!}{(n+m)!} P_{nm}(\theta) \tag{25}$$

However, this approach is also subject to numerical limitations. As the computation approaches the poles, the range of $|\hat{P}^{(d)}_{nm}(\theta)|$ is comparable to that of $|\bar{P}^{(d)}_{nm}(\theta)|$, thereby resulting in an underflow in IEEE double precision. For example, the sectoral values are given by

$$\hat{P}_{mm}(\theta) = u^m = \sin^m(\theta) \tag{26}$$

which for $\theta = 1^\circ$ and $M = 2700$ yields values that range from 1 to $\sim 10^{-4747}$. Therefore, employing the normalisation in Eq. (25) cannot solve the numerical problems discussed in Sect. 1.2.

Risbo (1996) claims that the Edmonds (1957) recursion for D-matrices can be used to compute fully normalised ALFs up to degree 200 000. However, the description of the test results which support this claim indicates that these computations were only performed at the equator, although the point is not clear. Nevertheless, Risbo (1996) includes a Fortran 77 subroutine for this recursion. It was found that, when implemented in IEEE double precision, the Risbo subroutine underflows for all polar distances $\theta < 50^\circ$ for $M = 2700$. This was not suprising, given that the corresponding ALFs cannot even be stored in IEEE double precision (Sect. 1.2), irrespective of the algorithm used to compute

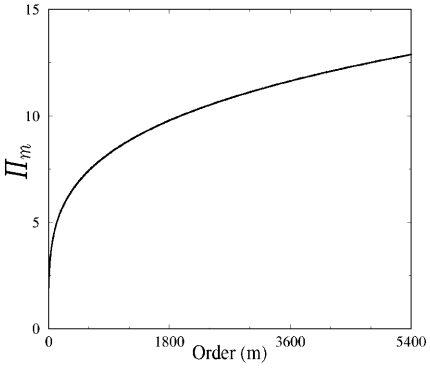

Fig. 4. Variation of Π_m [Eq. (24)] with order (m)

them. Therefore, although it may be possible to incorporate the new approaches presented here into the Risbo (1996) approach, it is clear that the Edmonds recursion alone does not solve the numerical problems encountered towards the poles as described in Sect. 1.2.

2.5 The modified forward row method

A simple, yet effective, method by which this problem of underflowing $\bar{P}_{mm}(\theta)$ may be avoided is to eliminate the u^m term from the recursion process in Eq. (13). To this end, Libbrecht (1985) adapted the standard forward row recursion (Sect. 2.2) to yield a modified forward row recursion that computes the quantities $(\bar{P}_{nm}(\theta))/u^m$. A recursive algorithm which computes the non-sectoral $(\bar{P}_{nm}(\theta))/u^m$ is obtained by dividing Eq. (18) by u^m to give

$$\frac{\bar{P}_{nm}(\theta)}{u^m} = \frac{1}{\sqrt{j}}\left(g_{nm}t\frac{\bar{P}_{n,m+1}(\theta)}{u^{m+1}} - h_{nm}u^2\frac{\bar{P}_{n,m+2}(\theta)}{u^{m+2}}\right), \quad \forall\ n > m \tag{27}$$

Equation (27) is seeded by the sectoral $\frac{\bar{P}_{mm}(\theta)}{u^m}$, the recursive algorithm for which is obtained by dividing Eq. (13) by u^m to give

$$\frac{\bar{P}_{mm}(\theta)}{u^m} = \sqrt{\frac{2m+1}{2m}}\ \frac{\bar{P}_{m-1,m-1}(\theta)}{u^{m-1}},\ \forall\ m > 1 \tag{28}$$

The value $(\bar{P}_{1,1}(\theta))/u^1 = \sqrt{3}$ serves as the seed for Eq. (28). Equation (23) yields

$$\frac{\bar{P}_{mm}(\theta)}{u^m} = \Pi_m,\ \forall\ m \geq 1 \tag{29}$$

Libbrecht's (1985) original formula for computing $(\bar{P}_{nm}(\theta))/u^m$ differs from Eq. (27) in that it does not include the $1/\sqrt{j}$ term on the right-hand side. This is because Libbrecht (1985) employs a different 'normalisation' of $P_{nm}(\theta)$ in which $k = 1,\ \forall\ m$. Moreover, Libbrecht (1985) focuses solely on the actual computation of the values of $(\bar{P}_{nm}(\theta))/u^m$. This paper shows how Libbrecht's (1985) method is easily extended to compute the quantities $(\bar{P}^{(1)}_{nm}(\theta))/u^m$. It also provides simple means by which values of $(\bar{P}^{(d)}_{nm}(\theta))/u^m$ can be applied in practice without dealing with the unmanageably small u^m terms for $u \to 0$.

The modified forward row recursion for computing $(\bar{P}^{(1)}_{nm}(\theta))/u^m$ is obtained by dividing Eq. (21) by u^m to yield

$$\frac{\bar{P}^{(1)}_{nm}(\theta)}{u^m} = m\frac{t}{u}\frac{\bar{P}_{nm}(\theta)}{u^m} - e_{nm}u\frac{\bar{P}_{n,m+1}(\theta)}{u^{m+1}}\ \ \forall\ n \geq m \tag{30}$$

such that all $(\bar{P}^{(1)}_{nm}(\theta))/u^m$ may be computed directly from previously computed $(\bar{P}_{nm}(\theta))/u^m$ of the same m.

In order to evaluate Eq. (1), the values of $(\bar{P}^{(d)}_{nm}(\theta))/u^m$ are multiplied by the corresponding values of $\bar{E}_{nm\alpha}$ [Eq. (3)] and the resulting products summed to yield values of $X^{(d)}_{m\alpha}/u^m$ and $\Omega^{(d)}_m/u^m$, instead of $X^{(d)}_{m\alpha}$ and $\Omega^{(d)}_m$, respectively. In order to compute $S^{(d)}$, Eq. (1) is factorised using Horner's scheme in terms of u

$$S^{(d)} = c\sum_{m=0}^{M}\Omega^{(d)}_m = c\left[\left\{\cdots\left(\left[\left\{\frac{\Omega^{(d)}_M}{u^M}\right\}u + \frac{\Omega^{(d)}_{M-1}}{u^{M-1}}\right]u + \frac{\Omega^{(d)}_{M-2}}{u^{M-2}}\right)u + \cdots + \frac{\Omega^{(d)}_2}{u^2}\right\}u + \frac{\Omega^{(d)}_1}{u^1}\right]u + c\Omega^{(d)}_0 \tag{31}$$

From Eq. (31), the running total is repeatedly multiplied by u upon the addition of each $\Omega^{(d)}_m/u^m$ term. This allows the sum $S^{(d)}$ to be computed directly from the values of $\Omega^{(d)}_m/u^m$, and so avoids the need to compute underflowing values of u^m as $u \to 0$ and m increases. This will be demonstrated numerically in Sect. 4.

2.6 Modified Forward Column Method

Values of $(\bar{P}^{(d)}_{nm}(\theta))/u^m$ may also be computed using what will be termed the first modified forward column recursion. To effect this computation, Eq. (28) is retained to compute the sectoral values of $(\bar{P}_{mm}(\theta))/u^m$. Equations (11) and (15) are divided by u^m, to give, respectively

$$\frac{\bar{P}_{nm}(\theta)}{u^m} = a_{nm}t\frac{\bar{P}_{n-1,m}(\theta)}{u^m} - b_{nm}\frac{\bar{P}_{n-2,m}(\theta)}{u^m},\ \ \forall\ n > m \tag{32}$$

$$\frac{\bar{P}^{(1)}_{nm}(\theta)}{u^m} = \frac{1}{u}\left(nt\frac{\bar{P}_{nm}(\theta)}{u^m} - f_{nm}\frac{\bar{P}_{n-1,m}(\theta)}{u^m}\right),\ \ \forall\ n \geq m \tag{33}$$

A variation of the first modified forward column recursion, herein termed the second modified forward column recursion, is to compute values of $(\bar{P}^{(d)}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ in which the entire sectoral value of $\bar{P}_{mm}(\theta)$ has been eliminated, rather than just the problematic u^m component. An immediate result is that all the sectoral values of $(\bar{P}_{mm}(\theta))/(\bar{P}_{mm}(\theta))$ equal 1. Dividing Eqs. (11) and (15) by $\bar{P}_{mm}(\theta)$ gives, respectively

$$\frac{\bar{P}_{nm}(\theta)}{\bar{P}_{mm}(\theta)} = a_{nm}t\frac{\bar{P}_{n-1,m}(\theta)}{\bar{P}_{mm}(\theta)} - b_{nm}\frac{\bar{P}_{n-2,m}(\theta)}{\bar{P}_{mm}(\theta)},\ \ \forall\ n > m \tag{34}$$

$$\frac{\bar{P}^{(1)}_{nm}(\theta)}{\bar{P}_{mm}(\theta)} = \frac{1}{u}\left(n\ t\ \frac{\bar{P}_{nm}(\theta)}{\bar{P}_{mm}(\theta)} - f_{nm}\ \frac{\bar{P}_{n-1,m}(\theta)}{\bar{P}_{mm}(\theta)}\right),\ \ \forall\ n \geq m \tag{35}$$

Comparison of Eq. (11) with Eqs. (32) and (34) shows these recursions to be of identical form; similarly for the comparison of Eq. (15) with Eqs. (33) and (35). That is, for Eqs. (32) and (34), and for Eqs. (33) and (35), the entire computation has simply been divided by u^m or $\bar{P}_{mm}(\theta)$, respectively. Thus any computer program that already employs Eqs. (11) and (15) to compute $\bar{P}^{(d)}_{nm}(\theta)$ is quickly adapted to compute $(\bar{P}^{(d)}_{nm}(\theta))/u^m$ or $(\bar{P}^{(d)}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ by simply altering the sectoral subroutine to return $(\bar{P}_{mm}(\theta))/u^m = \Pi_m$ or $(\bar{P}_{mm}(\theta))/(\bar{P}_{mm}(\theta)) = 1$, respectively.

To evaluate $S^{(d)}$ in Eq. (1), the required $(\bar{P}^{(d)}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ of the same m may be multiplied by their

corresponding values of $\bar{E}_{nm\alpha}$ [Eq. (3)] and the results added to give values of $X^{(d)}_{m\alpha}/(\bar{P}_{mm}(\theta))$ and subsequently $\Omega^{(d)}_m/(\bar{P}_{mm}(\theta))$. These $\Omega^{(d)}_m/(\bar{P}_{mm}(\theta))$ can be multiplied by Π_m in Eq. (14) to yield $\Omega^{(d)}_m/u^m$ for use in Horner's scheme in Eq. (31). Alternatively, inspection of Eq. (14) shows that $\bar{P}_{mm}(\theta)$ may be factorised into

$$\bar{P}_{mm}(\theta) \;=\; u^m\sqrt{3}\prod_{i=2}^{m}\sqrt{\frac{2i+1}{2i}} \;=\prod_{i=1}^{m} U_i,\;\; \forall\; m \geq 1 \qquad (36)$$

where

$$U_i \;=\begin{cases} \sqrt{3}u, & i=1 \\ \sqrt{\frac{2i+1}{2i}}u, & \forall\; i>1 \end{cases} \qquad (37)$$

Thus, Eq. (31) may be written using Horner's scheme in terms of U_m

$$\begin{aligned} S^{(d)} &= c\sum_{m=0}^{M}\Omega^{(d)}_m \\ &= c\left[\left\{\cdots\left(\left[\left\{\frac{\Omega^{(d)}_M}{\bar{P}_{M,M}(\theta)}\right\}U_M \;+\frac{\Omega^{(d)}_{M-1}}{\bar{P}_{M-1,M-1}(\theta)}\right]\right.\right. \\ &\quad\left.\left.\times U_{M-1}+\frac{\Omega^{(d)}_{M-2}}{\bar{P}_{M-2,M-2}(\theta)}\right)U_{M-2}+\cdots+\frac{\Omega^{(d)}_2}{\bar{P}_{2,2}(\theta)}\right\} \\ &\quad\times U_2+\frac{\Omega^{(d)}_1}{\bar{P}_{1,1}(\theta)}\Bigg]U_1+c\Omega^{(d)}_0 \end{aligned} \qquad (38)$$

Due to the numerically stable behaviour of Π_m (Fig. 4), using Π_m in this way, rather than multiplying the values of $\Omega^{(d)}_m/(\bar{P}_{mm}(\theta))$ by Π_m to yield $\Omega^{(d)}_m/u^m$ for use in Eq. (31), gives identical results for $S^{(d)}$ when performed in IEEE double precision. Moreover, the first and second modified forward column recursions are essentially the same, differing only in the treatment of Π_m, which is irrelevant to the numerical stability of each algorithm. The second modified forward column recursion [Eqs. (34) and (35)] and the implementation of Horner's scheme in Eq. (38) are introduced here primarily because of their relevance to the Clenshaw-based methods, discussed in Sect. 3. For the purposes of numerical testing (Sect. 4.2, 4.3 and 4.4), the first and second modified forward column methods will be treated as a single modified forward column recursion.

2.7 IEEE overflows and global scale factors in forward methods

The entire ranges of maximum and minimum values taken by $|(\bar{P}_{nm}(\theta))/u^m|$ and $|(\bar{P}^{(1)}_{nm}(\theta))/u^m|$, for $M = 2700$, are shown in Figs. 5 and 6, respectively. Inspection of Figs. 5 and 6 indicates that further factorisation is required to prevent the computations from overflowing in IEEE double precision. Overflows can be prevented for all of the modified forward methods introduced thus far simply by scaling all of the computations downwards by a global scale factor of 10^{-280}. This is achieved for all of the forward methods by simply multiplying the sectoral values of $(\bar{P}_{mm}(\theta))/u^m$ or $(\bar{P}_{mm}(\theta))/(\bar{P}_{mm}(\theta))$ by 10^{-280}, and using these scaled sectoral values as the recursive seeds in place of the original values. As such, this scale factor propagates linearly through all subsequent computations, thereby generating values of $((\bar{P}^{(d)}_{nm}(\theta))/u^m)\times 10^{-280}$ or $((\bar{P}^{(d)}_{nm}(\theta))/(\bar{P}_{mm}(\theta)))\times 10^{-280}$, respectively. These scaled values are used to form $(\Omega^{(d)}_m/u^m)\times 10^{-280}$ or $(\Omega^{(d)}_m/(\bar{P}_{mm}(\theta)))\times 10^{-280}$ which, when used in place of $\Omega^{(d)}_m/u^m$ or $\Omega^{(d)}_m/(\bar{P}_{mm}(\theta))$ in Horner's scheme in Eqs. (31) or (38), respectively, will yield values of $S^{(d)}\times 10^{-280}$. These are multiplied by 10^{280} to yield $S^{(d)}$.

Importantly, this global scaling allows, in IEEE double precision, the computation of spherical harmonic series S for $0^\circ \leq \theta \leq 180^\circ$ and $S^{(1)}$ for $0^\circ < \theta < 180^\circ$ up to $M = 2700$. Note that the spherical coordinate system renders partial sums $S^{(1)}$ indeterminate at the poles, since here the meridian tangents no longer uniquely define the direction of the derivative. There are useful ways around this problem (see e.g. Tscherning 1976), but for the sake of continuity they will not be considered here. Thus, for the remainder of this paper, no partial sums $S^{(1)}$ will be computed at the poles.

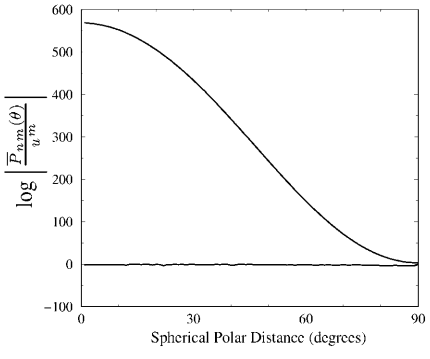

Fig. 5. Logarithm plot of maximum (*upper line*) and minimum (*lower line*) values of $|(\bar{P}_{nm}(\theta))/u^m|, \forall\; n, m \leq 2700$

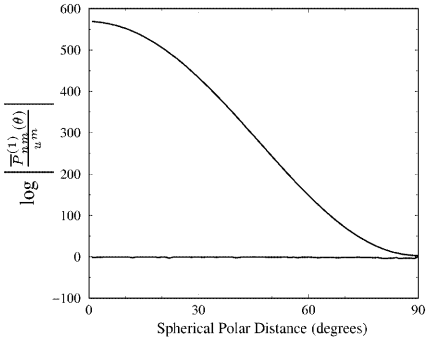

Fig. 6. Logarithm plot of maximum (*upper line*) and minimum (*lower line*) values of $|(\bar{P}^{(1)}_{nm}(\theta))/u^m|, \forall\; n, m \leq 2700$

3 Clenshaw-based methods

3.1 The forward column factorisation

It is instructive at this stage to consider a simple example of the summation described in Eq. (3). The rectangle in Fig. 2 contains circles representing values of $\bar{P}_{nm}(\theta)$ and $\bar{E}_{nm\alpha}$ for $m = 2$ and $2 \le n \le 5$. A summation of these elements in the form of Eq. (3) may be expanded as

$$X_{2,\alpha} = \sum_{n=2}^{5} \bar{E}_{n,2,\alpha}\bar{P}_{n2}(\theta) = \bar{E}_{2,2,\alpha}\bar{P}_{2,2}(\theta) + \bar{E}_{3,2,\alpha}\,\bar{P}_{3,2}(\theta) + \bar{E}_{4,2,\alpha}\bar{P}_{4,2}(\theta) + \bar{E}_{5,2,\alpha}\bar{P}_{5,2}(\theta) \tag{39}$$

The recursion relation in Eq. (11) gives the $\bar{P}_{nm}(\theta)$ required in Eq. (39); these are

$$\bar{P}_{3,2}(\theta) = \left[a_{3,2}t\right]\bar{P}_{2,2}(\theta) \tag{40}$$

$$\bar{P}_{4,2}(\theta) = a_{4,2}t\bar{P}_{3,2}(\theta) - b_{4,2}\bar{P}_{2,2}(\theta) \tag{41}$$

$$\bar{P}_{5,2}(\theta) = a_{5,2}t\bar{P}_{4,2}(\theta) - b_{5,2}\bar{P}_{3,2}(\theta) \tag{42}$$

Substitution of Eq. (40) into Eq. (41) yields

$$\bar{P}_{4,2}(\theta) = \left[a_{4,2}a_{3,2}t^2 - b_{4,2}\right]\bar{P}_{2,2}(\theta) \tag{43}$$

Likewise, substitution of Eqs. (40) and (43) into Eq. (42) gives

$$\bar{P}_{5,2}(\theta) = \left[a_{5,2}a_{4,2}a_{3,2}t^3 - a_{5,2}b_{4,2}t - a_{3,2}b_{5,2}t\right]\bar{P}_{2,2}(\theta) \tag{44}$$

Therefore, each $\bar{P}_{n,2}(\theta)$ can be factored into two components: the seed $\bar{P}_{2,2}(\theta)$ value and the aggregation of $(a_{l,2}t)$ and $b_{l,2}$ recursive terms [within the square brackets in Eqs. (40), (43) and (44)], which constitute polynomials in $t = \cos\theta$. This factorisation will be used in Sect. 3.2 to introduce the reverse column algorithms. Note that l rather than n is used here to denote the degree of the recursive terms, since each $\bar{P}_{n,2}(\theta)$ of degree n is comprised of an aggregation of all of the $(a_{l,2}t)$ and $b_{l,2}$ for $3 \le l \le n$. In general, any $\bar{P}_{nm}(\theta)$ may be factored into $\bar{P}_{mm}(\theta)$ and an aggregation of all of the $(a_{lm}t)$ and b_{lm} terms for $(m+1) \le l \le n$. Inspection of Eqs. (40), (43) and (44) shows that these aggregations are simply values of $(\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$, which are generated using Eq. (34). Substitution of Eqs. (40), (43) and (44) into Eq. (39) gives

$$X_{2,\alpha} = \left\{\bar{E}_{2,2,\alpha} + \bar{E}_{3,2,\alpha}\left[a_{3,2}t\right] + \bar{E}_{4,2,\alpha}\left[a_{4,2}a_{3,2}t^2 - b_{4,2}\right] + \bar{E}_{5,2,\alpha}\left[a_{5,2}a_{4,2}a_{3,2}t^3 - a_{5,2}b_{4,2}t - a_{3,2}b_{5,2}t\right]\right\}\bar{P}_{2,2}(\theta) \tag{45}$$

The quantity in curly brackets in Eq. (45) equates to $X_{2,\alpha}/(\bar{P}_{2,2}(\theta))$, which is used to form $\Omega_2/(\bar{P}_{2,2}(\theta))$ or Ω_2/u^2 (Sect. 2.6) for use in Eqs. (31) or (38), respectively.

3.2 Reverse column method

Results from timing tests presented in Sect. 4.3 show the reverse column methods, described below, to be highly inefficient in comparison with the other approaches presented in this paper for evaluating the required partial sums. The primary reason for describing the reverse column methods here is because these methods incorporate characteristics of both the modified forward column methods (Sect. 2.6) and the standard Clenshaw methods (Sect. 3.3). Thus, the reverse column methods are used here to highlight the basic similarities and differences between these two approaches.

To compute any value of $(\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$, the second modified forward column recursion [Eq. (34)] aggregates the necessary $(a_{lm}t)$ and b_{lm} recursive terms in the sequence of increasing degree l (sequentially down each column in Fig. 2). An alternative is to reverse this process and apply these same recursive terms in the sequence of decreasing degree l (sequentially up each column in Fig. 2). That is, a recursion may be employed whereby the $(a_{lm}t)$ and b_{lm} recursive terms for which $l = n$ (i.e. the largest value of l) are applied first and the recursive terms for which $l = m + 1$ (i.e. the smallest value of l) are applied last. This will be called a reverse column recursion, and is illustrated schematically in Fig. 7.

It is now necessary to introduce the recursive algorithm

$$s_{lm\alpha} = a_{l+1,m}\,t\,s_{l+1,m,\alpha} - b_{l+2,m}s_{l+2,m,\alpha} + y_{lm\alpha} \tag{46}$$

where $y_{lm\alpha}$ are predetermined, real-numbered constants (described later), and the subscript α ($\alpha = 1, 2$) is included here as it will be referred to later when discussing the standard Clenshaw methods. Equation (46) is used as follows. The recursion begins at the computation of a chosen $s_{nm\alpha}$. For this initial computation, $s_{n+1,m,\alpha}$ and $s_{n+2,m,\alpha}$ are set to predetermined values and then used in the first recursion to yield $s_{nm\alpha}$. Equation (46) is then used to achieve the recursive

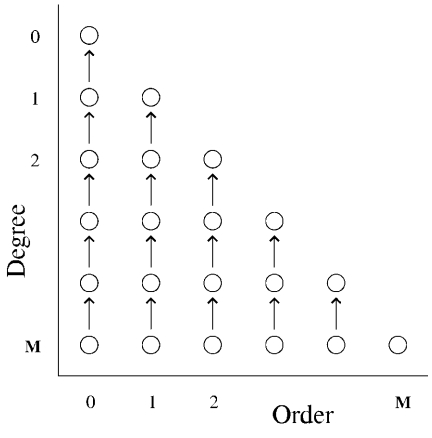

Fig. 7. A schematic of the recursion sequences employed in the reverse column, and the first and second Clenshaw algorithms to compute the quantities $(\bar{P}_{nm}(\theta)/\bar{P}_{mm}(\theta), \mathrm{d}(\mathbf{P}_{\mathrm{nm}}(\theta)/\mathbf{P}_{\mathrm{mm}}(\theta))/\mathrm{dt})$, $(X_{m\alpha}/\bar{P}_{mm}(\theta), \mathrm{d}(\mathbf{X}_{\mathrm{m}\alpha}/\mathbf{P}_{\mathrm{mm}}(\theta))/\mathrm{dt})$ and $(X_{m\alpha}/\bar{P}_{mm}(\theta)q^m, \mathrm{d}(\mathbf{X}_{\mathrm{m}\alpha}/\mathbf{P}_{\mathrm{mm}}(\theta)\mathrm{q}^{\mathrm{m}})/\mathrm{dt})$, respectively

computation of all $s_{lm\alpha}$, of constant m (a 'column' in Fig. 7), and sequentially increasing l [upwards and towards the diagonal (i.e. 'reverse') in Fig. 7], until the recursion is terminated at the computation of the sectoral $s_{mm\alpha}$ (on the diagonal in Fig. 7).

Recursion algorithms resembling Eq. (46) form part of the standard Clenshaw methods (cf. Tscherning and Poder 1982) for evaluating the partial sums $S^{(d)}$ without computing individual scaled values of $\bar{P}^{(d)}_{nm}(\theta)$ (Sect. 3.3). However, Eq. (46) can be used in a different context to compute individual values of $(\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$. Setting $s_{n+1,m,\alpha} = 0$, $s_{nm\alpha} = 1$, and all $y_{lm\alpha} = 0$ allows the recursive computation of all $s_{lm\alpha}$, of constant m and sequentially decreasing l, from $s_{n-1,m,\alpha}$ to $s_{mm\alpha}$. The effect of using the recursion in Eq. (46) in this way is to sequentially aggregate the $(a_{lm}t)$ and b_{lm} recursive terms, in the sequence of decreasing l, until the recursion terminates at the computation of $s_{mm\alpha} = (\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$.

One drawback of this approach is that the intermediate values of $s_{lm\alpha}$ in the recursion do not constitute actual values of $(\bar{P}_{lm}(\theta))/(\bar{P}_{mm}(\theta))$. Instead, each $(\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ value must be computed in isolation from the others using $(n-m)$ recursions of Eq. (46). This necessitates $\sim((M-m)^2)/2$ recursions to compute all non-sectoral values of $(\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ of order m and degree $(m+1) \le n \le M$. This contrasts with the second modified forward column recursion, where these same values of $(\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ can be computed using only $(M-m)$ recursions of Eq. (34). The relative numerical efficiency of these approaches is tested in Sect. 4.3.

To compute values of $(\bar{P}^{(1)}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$, the first derivative of Eq. (46) with respect to $t = \cos\theta$ gives

$$\dot{s}_{lm\alpha} = a_{l+1,m}\left(\dot{s}_{l+1,m,\alpha}t + s_{l+1,m,\alpha}\right) - b_{l+2m}\dot{s}_{l+2,m,\alpha} \qquad (47)$$

where $\dot{s}_{lm\alpha} = \mathrm{d}s_{lm\alpha}/\mathrm{d}t$. The seed values for the recursion in Eq. (47) (i.e. $s_{n,m,\alpha}$ and $s_{n+1,m,\alpha}$) are differentiated with respect to t to give $\dot{s}_{nm\alpha} = \dot{s}_{n+1,m,\alpha} = 0$. These initial values allow the recursion in Eq. (47) to compute all $\dot{s}_{lm\alpha}$, of constant m and sequentially decreasing l, from $\dot{s}_{n-1,m,\alpha}$ to $\dot{s}_{mm\alpha}$. When used in this way, the recursion in Eq. (47) terminates at the computation of

$$\dot{s}_{mm\alpha} = \frac{\mathrm{d}\left(\frac{\bar{P}_{nm}(\theta)}{\bar{P}_{mm}(\theta)}\right)}{\mathrm{d}t}$$

Since sectoral

$$\frac{\mathrm{d}\left(\frac{\bar{P}_{nm}(\theta)}{\bar{P}_{mm}(\theta)}\right)}{\mathrm{d}t}$$

are zero, their corresponding $\dot{s}_{mm\alpha}$ are simply set to zero without the need for any recursion. Application of the product rule and the chain rule to

$$\frac{\mathrm{d}\left(\frac{\bar{P}_{nm}(\theta)}{\bar{P}_{mm}(\theta)}\right)}{\mathrm{d}t}$$

yields

$$\frac{\bar{P}^{(1)}_{nm}(\theta)}{\bar{P}_{mm}(\theta)} = m\frac{t}{u}\, s_{mm\alpha} - u\dot{s}_{mm\alpha},\ \forall\, n \ge m \qquad (48)$$

where, as above, $s_{mm\alpha} = \frac{\bar{P}_{nm}(\theta)}{\bar{P}_{mm}(\theta)}$ and

$$\dot{s}_{mm\alpha} = \frac{\mathrm{d}\left(\frac{\bar{P}_{nm}(\theta)}{\bar{P}_{mm}(\theta)}\right)}{\mathrm{d}t}$$

To compute all non-sectoral values of $(\bar{P}^{(1)}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ of order m and degree $(m+1) \le n \le M$, the reverse column technique requires $\sim((M-m)^2)/2$ recursions of Eqs. (46) and (47), as well as $(M-m)$ applications of Eq. (48). In the same manner as the reverse column computation of $(\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ [Eq. (46)], the numerical efficiency of this approach (Sect. 4.3) contrasts poorly against the second modified forward column recursion (Sect. 2.6) in which the $(\bar{P}^{(1)}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ are obtained from previously computed values of $(\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ through only $(M-m)$ applications of Eq. (35).

As with the modified forward column and forward row recursion techniques (Sect. 2.6 and 2.5, respectively), a global scale factor must be applied to the reverse column computations to prevent an overflow in the computation of $(\bar{P}^{(d)}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$. This is achieved by simply setting the initial value, $s_{nm\alpha}$, to 10^{-280} (rather than 1) for use in Eq. (46). This scale factor of 10^{-280} will propagate linearly through the subsequent computations to yield values of $(\bar{P}^{(d)}_{nm}(\theta)/(\bar{P}_{mm}(\theta)) \times 10^{-280}$. Importantly, this global scaling allows, in IEEE double precision, the computation of spherical harmonic series S for $0^\circ \le \theta \le 180^\circ$ and $S^{(1)}$ for $0^\circ < \theta < 180^\circ$ up to $M = 2700$.

3.3 Standard Clenshaw methods

The standard Clenshaw methods, summarised below, closely resemble the reverse column recursions (Sect. 3.2). The Clenshaw (1955) approach, which was formulated originally to evaluate partial sums of Chebyshev polynomials, was adapted for use in geodesy by Gulick (1970) to compute partial sums of $\bar{P}^{(d)}_{nm}(\theta)$. Section 3.3.1 introduces a simple implementation of the Clenshaw (1955) approach, whilst Sect. 3.3.2 presents the implementation that is used more commonly in geodesy (cf. Gleason 1985).

3.3.1 The first Clenshaw method

The simplest implementation of the Clenshaw (1955) technique, herein termed the first Clenshaw method, uses the recursions in Eqs. (46) and (47) to compute, directly, the intermediate sums $X^{(d)}_{m\alpha}/(\bar{P}_{mm}(\theta))$, without evaluating individual values of $(\bar{P}^{(d)}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$. Undifferentiated values of $X_{m\alpha}/(\bar{P}_{mm}(\theta))$ may be computed using Eq. (46). Setting $s_{M+1,m,\alpha} = s_{M+2,m,\alpha} = 0$ and all $y_{lm\alpha} = \bar{E}_{lm\alpha}$ [$y_{lm\alpha} = 0, \forall l < \mu$ in Eq. (3)], allows the recursive computation of all $s_{lm\alpha}$, of constant m (a column in Fig. 7), and sequentially increasing l (upwards and towards the diagonal in Fig. 7), from $s_{Mm\alpha}$ to $s_{mm\alpha}$. As in the reverse column recursion (Sect. 3.2), the recursive process terminates at the computation of $s_{mm\alpha}$ (on the diagonal in Fig. 7), except that, in this case, the sectoral $s_{mm\alpha} = X_{m\alpha}/(\bar{P}_{mm}(\theta))$.

The first Clenshaw method also may be extended to compute partial sums $S^{(1)}$, $S^{(2)}$ and $S^{(-1)}$, where '$d=-1$' denotes definite integration (cf. Tscherning and Poder 1982). However, this study is confined to the computation of $S^{(1)}$. For this task, the first Clenshaw method uses the recursion in Eq. (47). The first derivative with respect to $t=\cos\theta$ of the seed values used above ($s_{M+1,m,\alpha}=s_{M+2,m,\alpha}=0$) gives $\dot{s}_{M+1,m,\alpha}=\dot{s}_{M+2,m,\alpha}=0$. These seed values allow the recursive computation of all $\dot{s}_{lm\alpha}$, of constant m and sequentially decreasing l, from $\dot{s}_{Mm\alpha}$ to $\dot{s}_{mm\alpha}$. As in the reverse column recursion (Sect. 3.2), this recursive process terminates at the computation of $\dot{s}_{mm\alpha}$, except that in this case

$$\dot{s}_{mm\alpha}=\frac{\mathrm{d}\left(\frac{X_{m\alpha}}{\bar{P}_{mm}(\theta)}\right)}{\mathrm{d}t}$$

Application of the product rule and the chain rule to the quantity

$$\frac{\mathrm{d}\left(\frac{X_{m\alpha}}{\bar{P}_{mm}(\theta)}\right)}{\mathrm{d}t}$$

yields (adapted from Gleason 1985)

$$\frac{X^{(1)}_{m\alpha}}{\bar{P}_{mm}(\theta)}=m\frac{t}{u}s_{mm\alpha}-u\dot{s}_{mm\alpha} \tag{49}$$

where, as above, $s_{mm\alpha}=X_{m\alpha}/\bar{P}_{mm}(\theta)$ and

$$\dot{s}_{mm\alpha}=\frac{\mathrm{d}\left(\frac{X_{m\alpha}}{\bar{P}_{mm}(\theta)}\right)}{\mathrm{d}t}$$

3.3.2 The second Clenshaw method

In geodesy, what is termed the second Clenshaw method in this paper is more commonly used to evaluate spherical harmonic expansions such as Eq. (5) (e.g. Gleason 1985; Deakin 1998). For this task, Eq. (38) is reformulated using Horner's scheme in terms of $(U_m\ q)$, where $q=a/r$

$$\begin{aligned}S^{(d)}&=c\sum_{m=0}^{M}\Omega_m^{(d)}\\&=c\left[\left\{\cdots\left(\left[\left\{\frac{\Omega_M^{(d)}}{\bar{P}_{M,M}(\theta)q^M}\right\}U_Mq+\frac{\Omega_{M-1}^{(d)}}{\bar{P}_{M-1,M-1}(\theta)q^{M-1}}\right]\right.\right.\\&\quad\times U_{M-1}q+\frac{\Omega_{M-2}^{(d)}}{\bar{P}_{M-2,M-2}(\theta)q^{M-2}}\Bigg)\\&\quad\left.\times U_{M-2}q+\cdots+\frac{\Omega_2^{(d)}}{\bar{P}_{2,2}(\theta)q^2}\right\}\\&\quad\left.\times U_2q+\frac{\Omega_1^{(d)}}{\bar{P}_{1,1}(\theta)q}\right]U_1q+c\Omega_0^{(d)}\end{aligned} \tag{50}$$

Note that in the geodetic literature [see e.g. Gleason 1985, Eqs. (2.38), (2.39) and (2.40)], the use of Horner's scheme in Eq. (50) is presented as an implementation of Eq. (46).

For undifferentiated values of S, the required $\Omega_m/(\bar{P}_{mm}(\theta)q^m)$ quantities are computed using a modified version of the reverse recursion algorithm in Eq. (46); this is (Gleason 1985)

$$s_{lm\alpha}\ =a_{l+1,m}\,t\,qs_{l+1,m,\alpha}-b_{l+2,m}q^2s_{l+2,m,\alpha}+y_{lm\alpha} \tag{51}$$

To produce the required values of $X_{m\alpha}/(\bar{P}_{mm}(\theta)q^m)$, the recursion in Eq. (46) is initiated exactly as for the first Clenshaw method ($s_{M+1,m,\alpha}=s_{M+2,m,\alpha}=0$), except that all $y_{lm\alpha}$ are set to $\bar{E}_{lm\alpha}/q^l$ [$y_{lm\alpha}=0,\forall l<\mu$ in Eq. (3)], rather than $\bar{E}_{lm\alpha}$. Thus, to evaluate the truncated expansion of geopotential in Eq. (5), these values of $y_{lm\alpha}$ are set to $\bar{C}_{lm\alpha}$, rather than $(a/r)^l\bar{C}_{lm\alpha}$, as for the first Clenshaw method. This allows the recursive computation of all $s_{lm\alpha}$, of constant m (a column in Fig. 7), and sequentially increasing l (upwards and towards the diagonal in Fig. 7), from $s_{Mm\alpha}$ to $s_{mm\alpha}$, where $s_{mm\alpha}=X_{m\alpha}/(\bar{P}_{mm}(\theta)q^m)$.

Similarly, values of $X^{(1)}_{m\alpha}/(\bar{P}_{mm}(\theta)q^m)$ may be computed by differentiating Eq. (51) with respect to t to give (Gleason 1985)

$$\dot{s}_{lm\alpha}=a_{l+1,m}q(\dot{s}_{l+1,m,\alpha}t+s_{l+1,m,\alpha})-b_{l+2m}q^2\dot{s}_{l+2,m,\alpha} \tag{52}$$

Differentiation (with respect to t) of the seed values used above ($s_{M+1,m,\alpha}=s_{M+2,m,\alpha}=0$) gives $\dot{s}_{M+1,m,\alpha}=\dot{s}_{M+2,m,\alpha}=0$. These values allow the recursive computation of all $\dot{s}_{lm\alpha}$, of the same m and sequentially decreasing l, from $\dot{s}_{Mm\alpha}$ to $\dot{s}_{mm\alpha}$. The recursive process terminates at the computation of $\dot{s}_{mm\alpha}$, where

$$\dot{s}_{mm\alpha}=\frac{\mathrm{d}\left(\frac{X_{m\alpha}}{\bar{P}_{mm}(\theta)q^m}\right)}{\mathrm{d}t}$$

Application of the product rule and the chain rule to the quantity

$$\frac{\mathrm{d}\left(\frac{X_{m\alpha}}{\bar{P}_{mm}(\theta)q^m}\right)}{\mathrm{d}t}$$

yields [adapted from Gleason (1985)]

$$\frac{X^{(1)}_{m\alpha}}{\bar{P}_{mm}(\theta)q^m}\ =m\frac{t}{u}s_{mm\alpha}-u\dot{s}_{mm\alpha} \tag{53}$$

where, as above, $s_{mm\alpha}=X_{m\alpha}/(\bar{P}_{mm}(\theta)q^m)$ and

$$\dot{s}_{mm\alpha}=\frac{\mathrm{d}\left(\frac{X_{m\alpha}}{\bar{P}_{mm}(\theta)q^m}\right)}{\mathrm{d}t}$$

Values of $X^{(d)}_{m\alpha}/(\bar{P}_{mm}(\theta)q^m)$, obtained from the second Clenshaw method, are used in place of $X^{(d)}_{m\alpha}$ in Eq. (2) to form $\Omega_m^{(d)}/(\bar{P}_{mm}(\theta)q^m)$. The standard approach (cf. Gleason 1985; Deakin 1998) is to combine these values using the implementation of Horner's scheme given in Eq. (50). Alternatively, values of $\Omega_m^{(d)}/(\bar{P}_{mm}(\theta)q^m)$ can either be multiplied by q^m to yield $\Omega_m^{(d)}/(\bar{P}_{mm}(\theta))$, which are combined using the implementation of Horner's scheme in Eq. (38), or multiplied by $q^m\Pi_m$ to yield values of $\Omega_m^{(d)}/u^m$, which are combined using the implementation of Horner's scheme given in Eq. (31).

The resulting numerical values of $S^{(d)}$ are identical in all cases (Sect. 4.2).

3.4 IEEE overflows and global scale factors in the standard Clenshaw methods

Overflows can be prevented in the both first and second Clenshaw methods by multiplying all $y_{lm\alpha}$ by the global scale factor of 10^{-280}. This scale factor propagates linearly through subsequent recursions to finally produce values of $(X^{(d)}_{m\alpha}/(\bar{P}_{mm}(\theta))) \times 10^{-280}$ and $(X^{(d)}_{m\alpha}/(\bar{P}_{mm}(\theta)q^m)) \times 10^{-280}$. These are used in place of $X^{(d)}_{m\alpha}$ in Eq. (2) to form $(\Omega^{(d)}_m/(\bar{P}_{mm}(\theta))) \times 10^{-280}$ and $(\Omega^{(d)}_m/(\bar{P}_{mm}(\theta)q^m)) \times 10^{-280}$, respectively. Again, this global scaling allows, in IEEE double precision, the computation of spherical harmonic series S for $0^\circ \le \theta \le 180^\circ$ and $S^{(1)}$ for $0^\circ < \theta < 180^\circ$ up to $M = 2700$.

4 Numerical tests

4.1 Viable methods

The previous derivations have presented six apparently viable methods for computing $S(0^\circ \le \theta \le 180^\circ)$ and $S^{(1)}(0^\circ < \theta < 180^\circ)$ for $M \le 2700$. These algorithms are summarised in Table 1.

As mentioned in Sect. 2.6, the first (MFC-1) and second (MFC-2) modified forward column recursions are, essentially, a single method. Therefore, for the remainder of this section they will be treated as one algorithm, termed the modified forward column recursion. This leaves, to this point, five separate methods for computing spherical harmonic expansions. The purpose of this section is to provide an initial, general assessment of the relative merits of these algorithms using tests of precision, numerical efficiency and accuracy. The tests of precision will compare partial sums, $S^{(d)}$, computed in IEEE double precision and IEEE extended double precision. The tests of numerical efficiency will compare the execution times of the algorithms. The tests of numerical accuracy will use analytic solutions for the sum of the square of $\bar{P}^{(d)}_{nm}(\theta)$, to compare the modified forward row and modified forward column algorithms only.

Table 1. Summary of algorithms for computing partial sums $S^{(d)}$

Method[a]	Section	Formulae		Computed quantities
		for S	for $S^{(d)}$	
MFC-1	2.6	(28) (32)	(33)	$\frac{\bar{P}^{(d)}_{nm}(\theta)}{u^m}$
MFC-2	2.6	(34)	(35)	$\frac{\bar{P}^{(d)}_{nm}(\theta)}{\bar{P}_{mm}(\theta)}$
MFR	2.5	(28) (27)	(30)	$\frac{\bar{P}^{(d)}_{nm}(\theta)}{u^m}$
RC	3.2	(46)	(47) (48)	$\frac{\bar{P}^{(d)}_{nm}(\theta)}{\bar{P}_{mm}(\theta)}$
CLEN-1	3.3.1	(46)	(47) (49)	$\frac{X^{(d)}_{nm\alpha}}{\bar{P}_{mm}(\theta)}$
CLEN-2	3.3.2	(51)	(52) (53)	$\frac{X^{(d)}_{nm\alpha}}{\bar{P}_{mm}(\theta)q^m}$

[a] MFC-1: first modified forward column; MFC-2: second modified forward column; MFR: modified forward row; RC: reverse column; CLEN-1: first Clenshaw; CLEN-2: second Clenshaw

4.2 Relative numerical precision

The first step in comparing different methods for computing the partial sums $S^{(d)}$ in Eq. (1) is to choose some appropriate values for $\bar{E}_{nm\alpha}$ and λ. For $\bar{E}_{nm\alpha}$, we might use empirically generated coefficients such as EGM96 (Lemoine et al. 1998) and/or GPM98B (Wenzel 1998) to compute the lower degree $\bar{E}_{nm\alpha}$. Higher-degree coefficients could be generated synthetically to conform with the predicted spectral characteristics of the Earth's gravity field (e.g. Tscherning and Rapp 1974). However, the ubiquity of high-degree spherical harmonic expansions across multiple scientific disciplines favours a more general approach. For this reason, the following comparisons have employed the testing regime utilised by Gleason (1985), which is to set all $\bar{E}_{nm\alpha}$ equal to 1 and λ equal to 0. This testing regime has the advantage of being straightforward to present, as well as being sufficiently general for a first assessment of the new algorithms. In this approach, the computed estimates $s^{(d)}$ of the partial sums $S^{(d)}$ [Eq. (1)] reduce to

$$s^{(d)} = \sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}^{(d)}_{nm}(\theta) \tag{54}$$

Each algorithm in Table 1 was evaluated in IEEE double precision on a Sun Ultra-10 workstation. The algorithms were encoded in Fortran 77 and compiled using the Sparkworks^{TM} (v3.0.1) Fortran compiler. The values of s were computed for integer values of colatitude $0^\circ \le \theta \le 180^\circ$ and values of $s^{(1)}$ were computed for integer values of colatitude $1^\circ \le \theta \le 179^\circ$. These were compared with the corresponding 'control' values, obtained from the second Clenshaw summation (Sect. 3.3.2) which was implemented in IEEE extended double precision (i.e. 16 bytes to store each floating-point number) (cf. Coonen 1980). The relative precision (RP) for each θ was calculated using

$$RP = \left| \frac{s^{(d)}(\text{double}) - s^{(d)}(\text{extended})}{s^{(d)}(\text{extended})} \right| \tag{55}$$

where $s^{(d)}$(double) is the value of Eq. (54) for the summation computed in double precision by each respective method and $s^{(d)}$(extended) is the result for the same sum computed using the second Clenshaw method in extended double precision. The values of RP are computed under the assumption that the results obtained from IEEE extended double precision are correct to at least one significant figure more than those obtained from IEEE double precision.

The term precision is used here since any systematic errors common to computations in both the double precision and extended double precision formats will not be revealed during such a comparison. Moreover, to

insure that such systematic errors did not bias the control $s^{(d)}$(extended) values in favour of the $s^{(d)}$(double) values computed using the second Clenshaw method, a second set of control values was computed in extended double precision using the modified forward column method. The two sets of $s^{(d)}$ (extended) control values agreed to a minimum of 25 significant figures.

For each of the algorithms, the computed quantities (Table 1) were globally scaled to yield values of $(\Omega_m/u_m) \times 10^{-280}, (\Omega_m/(\bar{P}_{mm}(\theta))) \times 10^{-280}$ or

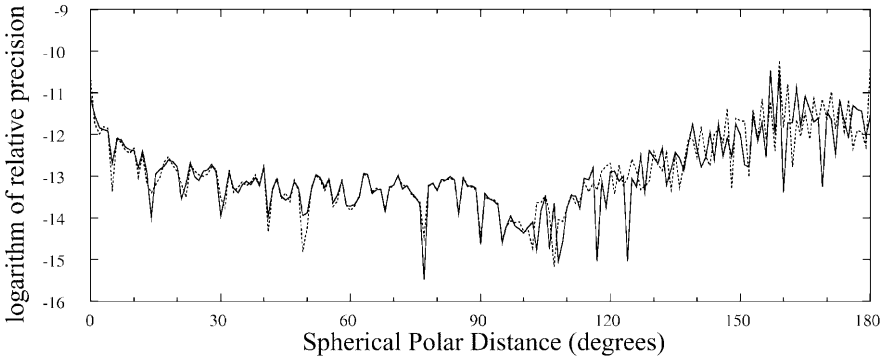

Fig. 8. Logarithm of the relative precision [Eq. (54)] to evaluate $\sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}(\theta)$ using the first Clenshaw (*solid line*) and the second Clenshaw (*dashed line*) methods

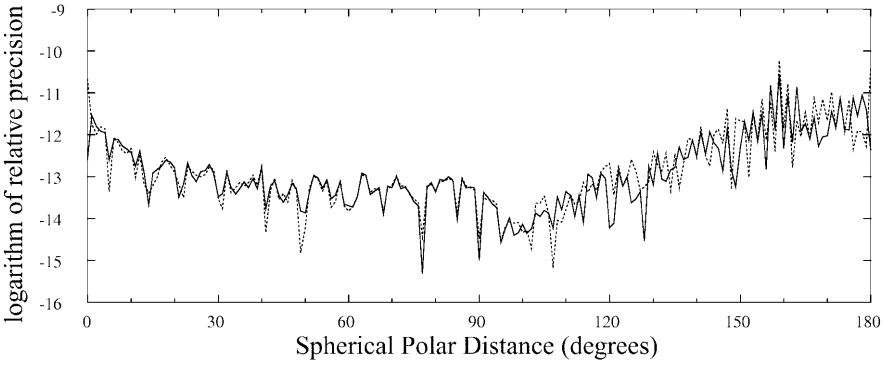

Fig. 9. Logarithm of the relative precision [Eq. (54)] to evaluate $\sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}(\theta)$ using the modified forward column (*solid line*) and the second Clenshaw (*dashed line*) methods

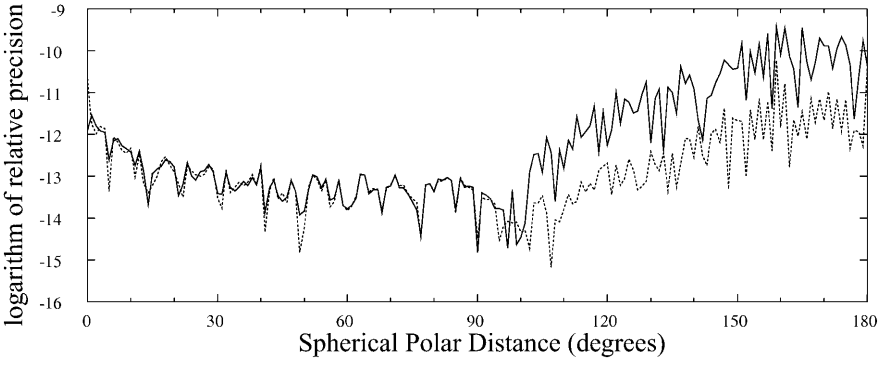

Fig. 10. Logarithm of the relative precision [Eq. (54)] to evaluate $\sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}(\theta)$ using the reverse column (*solid line*) and second Clenshaw (*dashed line*) methods

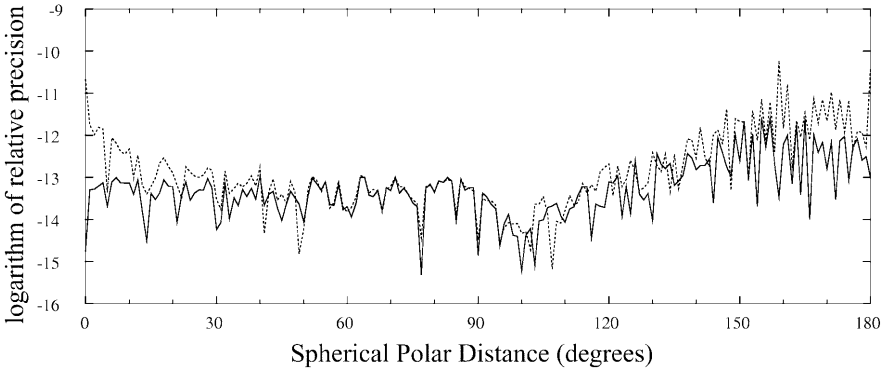

Fig. 11. Logarithm of the relative precision [Eq. (54)] to evaluate $\sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}(\theta)$ using the modified forward row (*solid line*) and second Clenshaw (*dashed line*) methods

$(\Omega_m/(\bar{P}_{mm}(\theta)\ q^m)) \times 10^{-280}$, which were then substituted into the implementations of Horner's scheme in Eqs. (31), (38) and (50), respectively, to yield values of $s^{(d)} \times 10^{-280}$ and then $s^{(d)}$. Recall that $\bar{E}_{nm\alpha} = 1$ and $\lambda = 0$, so Eqs. (31),(38) and (50) reduce to Eq. (54). It was found that, once the values of $(\Omega_m/u^m) \times 10^{-280}$, $(\Omega_m/(\bar{P}_{mm}(\theta))) \times 10^{-280}$ or $(\Omega_m/(\bar{P}_{mm}(\theta)q^m)) \times 10^{-280}$ were computed, they could be appropriately factorised for use in any of the three implementations of Horner's scheme with no change to the final computed value of $s^{(d)}$. That is, all combinations of the five algorithms with the three implementations of Horner's scheme showed that the choice of implementation of Horner's scheme was irrelevant to the observed precision of the algorithm.

Of the five algorithms to be tested in this way, the second Clenshaw method is the most widely used in geodesy. Thus, it is useful to employ the *RP* error signature of this method as a benchmark against which the performance of the other five methods can be assessed. To test the second Clenshaw method, an extreme, but realistic, value for q was chosen to provide a contrast to the first Clenshaw method. Thus, q was set to ${}^{b}\text{GRS80}/{}^{a}\text{GRS80} = 6\ 356\ 752.3141/6\ 378\ 137$, where ${}^{a}\text{GRS80}$ and ${}^{b}\text{GRS80}$ are the semi-major and semi-minor axes of the GRS80 ellipsoid (Mortiz 1980).

The relative precision signatures of the five algorithms for computing s are shown in Figs. 8 to 11, whilst the relative precision signatures of the five algorithms for computing $s^{(1)}$ are shown in Figs. 12 to 15. To facilitate easier visual comparisons, the error signatures obtained from the second Clenshaw method in the computation of s and $s^{(1)}$ have been superimposed (*dashed line*) over the corresponding error signatures obtained from each of the other four methods.

For the computation of the quantities $s = \sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}(\theta)$, inspection of Figs. 8 and 9 shows no sys-

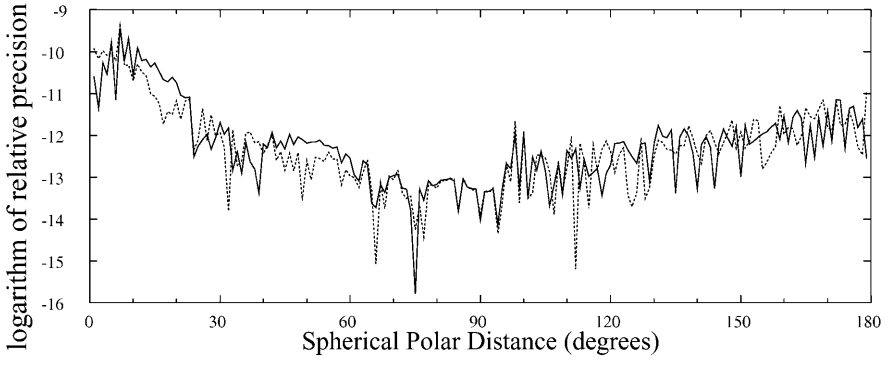

Fig. 12. Logarithm of the relative precision [Eq. (54)] to evaluate $\sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}^{(1)}(\theta)$ using the first Clenshaw (*solid line*) and the second Clenshaw (*dashed line*) methods

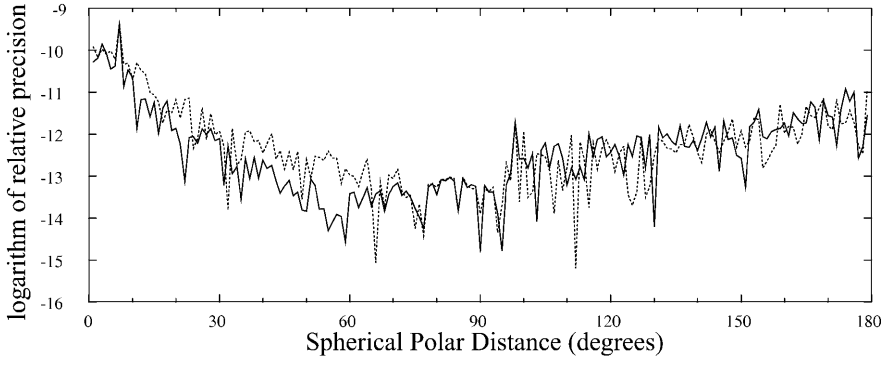

Fig. 13. Logarithm of the relative precision [Eq. (54)] to evaluate $\sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}^{(1)}(\theta)$ using the modified forward column (*solid line*) and second Clenshaw (*dashed line*) methods

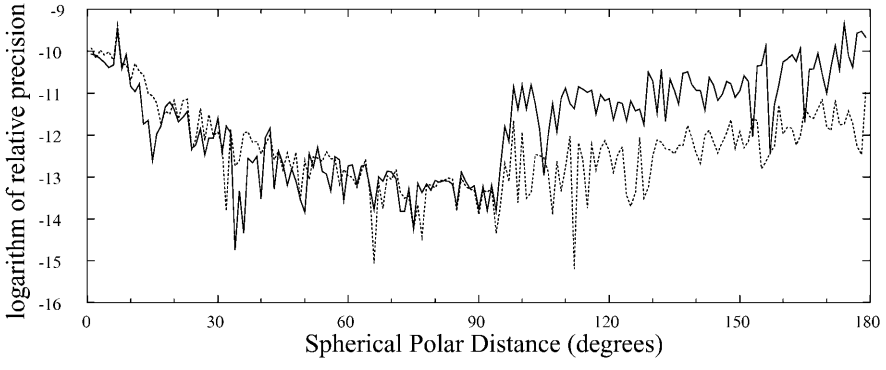

Fig. 14. Logarithm of the relative precision [Eq. (54)] to evaluate $\sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}^{(1)}(\theta)$ using the reverse column (*solid line*) and second Clenshaw (*dashed line*) methods

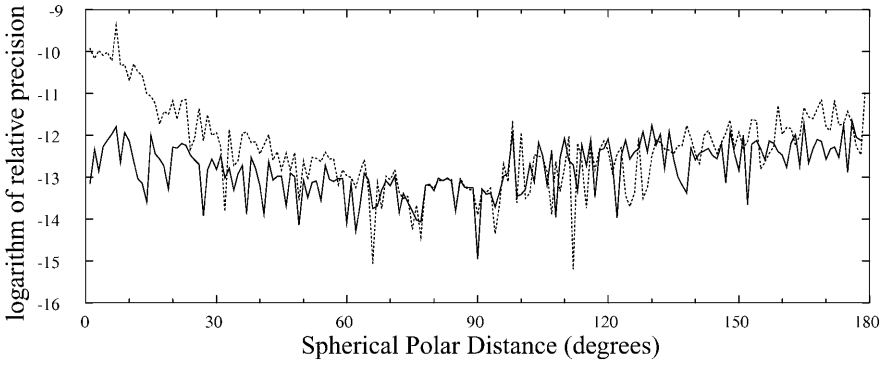

Fig. 15. Logarithm of the relative precision [Eq. (54)] to evaluate $\sum_{n=0}^{2700}\sum_{m=0}^{n}\bar{P}_{nm}^{(1)}(\theta)$ using the modified forward row (*solid line*) and second Clenshaw (*dashed line*) methods

tematic differences between the performance of the Clenshaw methods and the modified forward column method. One interesting feature is that the relative precision signatures obtained from all of the column methods for the northern latitudes are almost identical, whilst the signatures for the southern latitudes are not (cf. Figs. 8 to 10). This is particularly evident for the relative precision signature for the reverse column recursion (Fig. 10), which contrasts poorly, in the southern latitudes, against the signature from the second Clenshaw method. Nevertheless, the relative precision is still $< 10^{-9}$.

Inspection of the relative precision signature for the modified forward row recursion (Fig. 11) reveals a slight, but clear, improvement in precision over the other algorithms. This is particularly evident as computation approaches the poles.

For the computation of the quantities $s^{(1)} = \sum_{n=0}^{2700}\sum_{m=0}^{n}\bar{P}_{nm}^{(1)}(\theta)$, Fig. 12 reveals no clear differences between the relative precision signatures of the first and second Clenshaw methods. Figure 13 shows, in the northern latitudes only, a moderately improved relative precision signature for the modified forward column method over that of the second Clenshaw method. Similarly, the relative precision signature for the modified forward row recursion (Fig. 15) shows an increasing improvement over the second Clenshaw method towards the north pole. There is also a slight improvement towards the south pole. As with the relative precision signatures for s, the reverse column method gives a relative poor relative precision signature for $s^{(1)}$ in the southern latitudes (Fig. 14). This result is to be expected since the values of $(\bar{P}_{nm}^{(1)}(\theta))/u^m$ are computed from $(\bar{P}_{nm}(\theta))/u^m$.

None of the five recursion tested delivered a relative error of $>10^{-9}$ in the computation of either s or $s^{(1)}$. No analysis was conducted to explain the hemisphere-dependent performance observed (Figs. 8 to 15) during these tests. However, this phenomenon is not observed in the accuracy tests for the two modified forward algorithms (Sect. 4.4). These both produced relative accuracy signatures (Figs. 16 and 17) that are noticeably more symmetric about the equator than the precision signatures (Figs. 9, 13 and 11, 15) for the corresponding algorithms.

For each algorithm, the accuracy tests used the same values of $(\bar{P}_{nm}^{(d)}(\theta))/u^m$ as those used in the precision tests. However, the accuracy tests squared these values before combining them using Horner's scheme. This suggests that, at least for the two modified forward algorithms, the actual computation of $(\bar{P}_{nm}(\theta))/u^m$ and $(\bar{P}_{nm}^{(1)}(\theta))/u^m$ is performed equally well in both hemispheres. In this case, the lack of symmetry in the relative precision signatures returned by these algorithms might result from the change in sign, across the equator, of half the $(\bar{P}_{nm}(\theta))/u^m$ and $(\bar{P}_{nm}^{(1)}(\theta))/u^m$. This would create different cancelling effects in each hemisphere when these values are combined using Horner's scheme. Such an effect would not be present when squared terms are combined. Further work may validate this explanation.

4.3 Numerical efficiency

The five methods that successfully computed $S^{(d)}$, for $M = 2700$ and for integer values of θ to the poles, were tested for their relative numerical efficiency. Considerable attention was given to eliminating all redundant computations from each algorithm. For example, the square roots and inverted square roots required to construct the recursion coefficients were computed once by each algorithm and then stored for multiple use in the synthesis subroutines. However, the latitude-independent components of the recursion coefficients were not computed and stored in this way, but were generated from the square roots and inverted square roots as they were required. This was done so that the efficiency results would be applicable to the many PCs, still widely used, which possess insufficient RAM to store $\sim\tilde{7}.3$ million recursion coefficients.

Table 2 shows the CPU time required by each of the five algorithms to computer the following:

(1) $s = \sum_{n=0}^{2700}\sum_{m=0}^{n}\bar{P}_{nm}(\theta)$, for integer values of $0^\circ \le \theta \le 180^\circ$
(2) s as above and $s^{(1)} = \sum_{n=0}^{2700}\sum_{m=0}^{n}\bar{P}_{nm}^{(1)}(\theta)$, for integer values of $1^\circ \le \theta \le 179^\circ$ simultaneously.

The CPU times for the reverse column algorithm are excessively large and so were extrapolated from the computation times for a single parallel. All computations were performed, once again, on a Sun Ultra-10(333 MHz) workstation that uses a virtual or 'swapped' RAM configuration, which is slower than actual RAM.

It should be noted that these CPU times, in addition to showing the relative efficiency of each approach, are

Table 2. CPU time required to compute $S = \sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}(\theta)$, $\forall$ integer values of $0^\circ \le \theta \le 180^\circ$, and the CPU time required to compute both S, $\forall$ integer values of $1^\circ \le \theta \le 179^\circ$, and $S^{(1)} = \sum_{n=0}^{2700} \sum_{m=0}^{n} \bar{P}_{nm}^{(1)}(\theta)$, $\forall$ integer values of $1^\circ \le \theta \le 179^\circ$, together

Task	S (sec)	S and $S^{(1)}$ (sec)
CLEN-1	192	282
CLEN-2	192	282
MFC	186	408
MFR	174	258
RC	41 862	67 778

[a] CLEN-1: first Clenshaw; CLEN-2: second Clenshaw; MFC: modified forward column; MFR: modified forward row; RC: reverse column

also functions of the computer architecture, compiler and programming language employed, as well as the programmer's implementation of these algorithms. Variations in any of these, particularly algorithmic implementation, can slightly improve or worsen the relative performance of each algorithm. However, the results presented in Table 2 are sufficient for the current purpose, which is to demonstrate that all algorithms, except the reverse column algorithm, appear to be of comparable numerical efficiency in evaluating the required parital sums. Lastly, for the reasons outlined in Sect. 3.2, the reverse column algorithm is extremely inefficient when compard with the other four approaches, and thus will be excluded from further examination.

4.4 Accuracy

As mentioned, the numerical evaluations presented in Sect. 4.2 are tests of precision only, since both the tested algorithms and the 'control' algorithm computed in IEEE extended double precision may contain shared systematic errors. These could be due to any one of compiler, computer architecture or programming errors, for example. Therefore, it is prudent to supplement tests of precision with accuracy assessments that utilise exact identities (i.e. analytic results) incorporating the computed quantities. For this purpose, starting with the well-known identity

$$\sum_{m=0}^{n} (\bar{P}_{nm}(\theta))^2 = 2n + 1, \ \forall\ \theta \tag{56}$$

this can be shown to yield

$$\underset{M}{\Sigma} = \sum_{n=0}^{M} \sum_{m=0}^{n} (\bar{P}_{nm}(\theta))^2 = (M+1)^2, \ \forall\ \theta \tag{57}$$

Eq. (56) gives

$$\sum_{m=0}^{n} \left(\bar{P}_{nm}^{(1)}(\theta)\right)^2 = \left(\frac{n(n+1)(2n+1)}{2}\right), \ (0^\circ < \theta < 180^\circ) \tag{58}$$

from which results

$$\overset{*}{\underset{M}{\Sigma}} = \sum_{n=0}^{M} \sum_{m=0}^{n} \left(\bar{P}^{(1)}(\theta)\right)^2 = \left(\frac{M(M+1)^2(M+2)}{4}\right), \ (0^\circ < \theta < 180^\circ) \tag{59}$$

For $M = 2700$, Eq. (57) gives $\Sigma_{2700} = 7\,295\,401$ and Eq. (59) gives $\Sigma^*_{2700} = 13\,305\,717\,113\,850$. However, these analytic values of Σ_{2700} and Σ^*_{2700} cannot be used to verify the accuracy of the first and second Clenshaw methods, because these methods do not compute individual, scaled values of $\bar{P}_{nm}^{(d)}(\theta)$. An alternative test for the standad Clenshaw approaches is to compute partial sums of second derivatives, $S^{(2)}$, and then use these to evaluate Laplace's equation ($\Delta f = 0$). However, this study does not extend to the computation of second derivatives, and so will be confined to testing the accuracy of the modified forward column approach and the modified forward row approach only.

To implement this accuracy test, both methods were applied, in IEEE double precision, to compute, for $M = 2700$, integer values of $((\bar{P}_{nm}(\theta))/u^m) \times 10^{-280} (0^\circ \le \theta \le 180^\circ)$ and $((\bar{P}_{nm}^{(1)}(\theta))/u^m) \times 10^{-280} (1^\circ \le \theta \le 179^\circ)$. However, these values cannot be squared in IEEE double precesion without an underflow. This necessitates that the squaring of $((\bar{P}_{nm}^{(d)}(\theta))/u^m) \times 10^{-280}$, as well as their combination using Horner's scheme, be performed in IEEE extended double precision. This means that the results from these tests will only reflect the accuracy with which $(\bar{P}_{nm}^{(d)}(\theta))/u^m \times 10^{-280}$ are computed, since this is the only operation which can be performed in IEEE double precision for $M = 2700$.

The values of $((\bar{P}_{nm}^{(d)}(\theta))/u^m) \times 10^{-280}$ were converted to IEEE extended double precision, squared, and then rescaled by 10^{560} to yield values of $((\bar{P}_{nm}^{(d)}(\theta))^2)/u^{2m}$. All the values of $((\bar{P}_{nm}^{(d)}(\theta))^2)/u^{2m}$ of the same m were summed to give the quantities

$$\Gamma_m = \sum_{n=m}^{2700} \frac{(\bar{P}_{nm}(\theta))^2}{u^{2m}}$$

and

$$\Gamma^*_m = \sum_{n=m}^{2700} \frac{(\bar{P}_{nm}^{(1)}(\theta))^2}{u^{2m}}$$

These Γ_m were then combined using Horner's scheme

$$\underset{2700}{\Sigma}(\text{comp}) = \left[\left\{\cdots\left(\left[\left\{\frac{\Gamma_{2700}}{u^{5400}}\right\}u^2 + \frac{\Gamma_{2699}}{u^{5398}}\right]u^2 + \frac{\Gamma_{2698}}{u^{5396}}\right)u^2 + \cdots + \frac{\Gamma_2}{u^4}\right\}u^2 + \frac{\Gamma_1}{u^2}\right]u^2 + \Gamma_0 \tag{60}$$

where $\Sigma_{2700}(\text{comp})$ are the computed estimates of Σ_{2700}. Exchanging Γ^*_m for Γ_m in Eq. (60) yields Σ^*_{2700} (comp), which are the computed estimates of Σ^*_{2700}.

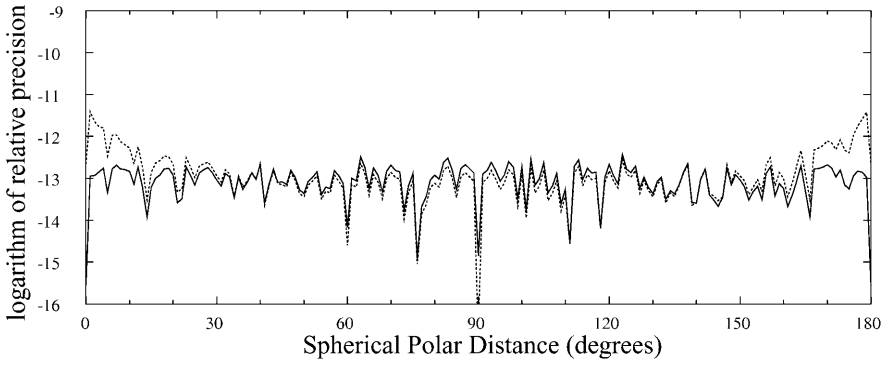

Fig. 16. Logarithm of the relative accuracy [Eq. (61)] to evaluate $\sum_{n=0}^{2700}\sum_{m=0}^{n}(\bar{P}_{nm}(\theta))^2$ using the modified forward row (*solid line*) and modified forward column (*dashed line*) algorithms

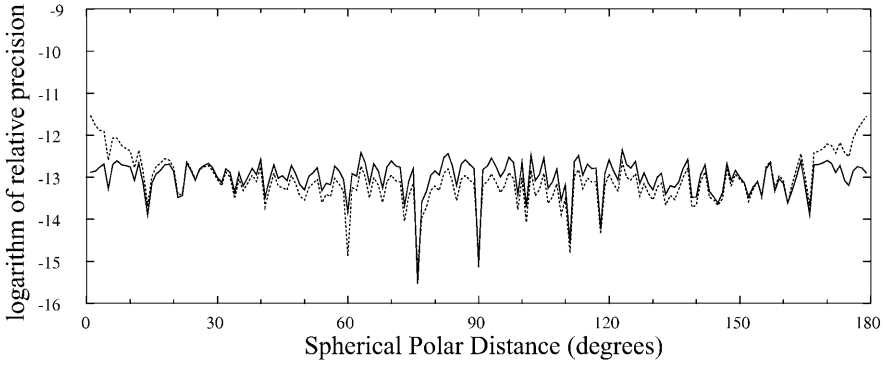

Fig. 17. Logarithm of the relative accuracy [Eq. (62)] to evaluate $\sum_{n=0}^{2700}\sum_{m=0}^{n}(\bar{P}_{nm}^{(1)}(\theta))^2$ using the modified forward row (*solid line*) and modified forward column (*dashed line*) algorithms

The numerical accuracy (NA) of $\bar{P}_{nm}(\theta)$ was calculated using the relation

$$NA = \left(\frac{\Sigma_{2700}(\text{comp}) - 7\ 295\ 401}{7\ 295\ 401}\right) \tag{61}$$

whereas the numerical accuracy (NA^*) of $\bar{P}_{nm}(\theta)$ was calculated using the relation

$$NA = \left(\frac{\Sigma^*_{2700}(\text{comp}) - 13\ 305\ 717\ 113\ 850}{13\ 305\ 717\ 113\ 850}\right) \tag{62}$$

The resulting Σ_{2700}(comp) error signatures of both algorithms are plotted in Fig. 16 for integer values of $0^\circ \le \theta \le 180^\circ$. The Σ^*_{2700}(comp) error signatures of both algorithms are plotted in Fig. 17 for integer values of $1^\circ \le \theta \le 179^\circ$.

Figures 16 and 17 show the accuracy of both methods to be almost identical. In the computation of both Σ_{2700}(comp) (Fig. 16) and Σ^*_{2700}(comp) (Fig. 17), the modified forward column algorithm performs slightly, but consistently, better than the modified forward row algorithm in the lower latitudes. However, the modified forward column algorithm becomes increasingly less accurate than the modified forward row algorithm as the computation approaches the poles. This observation is consistent with the results presented in Sect. 4.2, in which, near the poles, the modified forward row algorithms remained relatively stable in comparison with the other methods tested. Lastly, note that neither the modified forward row method, nor the modified forward column method, delivered values of NA or NA^* greater than 10^{-11}. These results support those obtained for these two algorithms in the precision tests (Sect. 4.2).

5 Summary, conclusions and recommendations

This paper has shown that standard Clenshaw methods for evaluating high-degree spherical harmonic expansions derive their stability from simple numerical principles. IEEE underflows are avoided by first eliminating the numerically problematic u^m term from the fundamental recursive algorithms, and then employing Horner's scheme to gradually reintroduce this term into the final computed value for the paritial sums $S^{(d)}$. Moreover, existing algorithms for computing individual values of $P_{nm}(\theta)$ and $P^{(1)}_{nm}(\theta)$ are easily modified to incorporate these two fundamental characteristics of the standard Clenshaw methods.

This last statement is strongly supported by the results of numerical tests. These show that the two new algorithms (the modified forward row and the modified forward column methods) can be applied in IEEE double precision to compute the partial sums S, up to $M = 2700(0^\circ \le \theta \le 180^\circ)$, as well as $S^{(1)}$, up to $M = 2700(0^\circ < \theta < 180^\circ)$, without IEEE underflow or overflow. Moreover, the results also suggest that the new algorithms are equivalent to the standard Clenshaw methods in both precision and efficiency. No doubt a more rigorous testing regime, specific to geodesy, will incorporate realistic geopotential coefficients into the computed partial sums $S^{(d)}$. The relatively stable performance of the modified forward row method, near the poles, might also warrant further examination.

Perhaps the most interesting characteristic of the new methods is their relative simplicity. Unlike the standard Clenshaw methods, both the modified forward row and modified forward column algorithms are easily for-

mulated using elementary algebra. More importantly, the mechanisms within the computation process are highly intuitive and transparent. These qualities should simplify the process of adapting these approaches to other tasks, such as evaluating partial sums of even higher degree and order (e.g. 5400), for all latitudes tested in this study. Two other useful adaptions include, for $M = 2700$, evaluating partial sums of second derivatives, and evaluating quantities that have been integrated over geographic squares bordered by meridians and parallels.

Another potentially useful property of the new methods is the fact that they compute individual, scaled values of $\bar{P}_{nm}^{(d)}(\theta)$. This property, in conjunction with the inherent simplicity of the principles presented here, renders the new approach an attractive starting point for extending the maximum M over which existing algorithms for spherical harmonic analysis can be applied.

Acknowledgements. The authors would like to offer their sincerest thanks to Christian Tscherning and the other two reviewers. Their suggestions were of considerable use to the authors in allowing them to produce a much more rigorous and integrated document.

Appendix A

This Appendix deals with some miscellaneous points on implementing the new methods on a computer.

A.1 RAM conservation for the modified forward row method

The modified forward row method (Sect. 2.5) computes $(\bar{P}_{nm}^{(d)}(\theta))/u^m$ of matching degree. However, efficient evaluation of the partial sums $S^{(d)}$ requires that $(\bar{P}_{nm}^{(d)}(\theta))/u^m$ of matching *order* be combined to yield $\Omega_m^{(d)}/u^m$ for use using Horner's scheme in Eq. (31). A straightforward approach is to precompute and store, for all degrees and orders, all the $(\bar{P}_{nm}^{(d)}(\theta))/u^m$ and then combine the $(\bar{P}_{nm}^{(d)}(\theta))/u^m$ of matching order. However, many PCs do not have sufficient RAM to store, say, $\sim$3.7 million values of $(\bar{P}_{nm}(\theta))/u^m$ and/or $(\bar{P}_{nm}^{(1)}(\theta))/u^m$ for $M = 2700$.

The alternative is to use the modified forward row method to compute $(\bar{P}_{nm}^{(d)}(\theta))/u^m$ for one order at a time. The sectoral $(\bar{P}_{nm}^{(d)}(\theta))/u^m$ are computed and stored as before. Referring again to Fig. 3, computation of all non-sectoral $(\bar{P}_{nm}^{(d)}(\theta))/u^m$, of order m and degree $(m+1) \le n \le M$ (a complete 'column' in Fig. 3), requires only that previously computed $(\bar{P}_{n,m+1}(\theta))/u^{m+1}$ and $(\bar{P}_{n,m+2}(\theta))/u^{m+2}$ have been stored.

Three arrays are used in 'rotation'. Arrays 1 and 2 contain all the previously computed values of $(\bar{P}_{n,m+1}(\theta))/u^{m+1}$ and $(\bar{P}_{n,m+2}(\theta))/u^{m+2}$, respectively. These are used in Eq. (27) to compute all the required non-sectoral $(\bar{P}_{nm}(\theta))/u^m$, which are stored in array X. Values of $(\bar{P}_{nm+1}(\theta))/u^{m+1}$ are then used in Eq. (30) to compute all the required $(\bar{P}_{nm}^{(1)}(\theta))/u^m$, which are stored in array 2. Ω_m/u^m and $\Omega_m^{(1)}/u^m$ are obtained from the values of $(\bar{P}_{n,m}(\theta))/u^m$ and $(\bar{P}_{n,m}^{(1)}(\theta))/u^m$, repectively. For the next round of computation $(\bar{P}_{n,m-1}^{(d)}(\theta))/u^{m-1}$, (corresponding to the next 'column' to the left in Fig. 3), the old array X becomes the new array 1, the old array 1 becomes the new array 2, and the old array 2 becomes the new array X.

This method employs the same number of mathematical operations as that which precomputes and stores all the required $(\bar{P}_{n,m}(\theta))/u^m$ and $(\bar{P}_{n,m}^{(1)}(\theta))/u^m$. However, the method of three 'rotated' arrays requires a RAM allocation of only $3M$ array elements to compute and use the required non-sectoral $(\bar{P}_{n,m}(\theta))/u^m$ and $(\bar{P}_{n,m}^{(1)}(\theta))/u^m$. This contrasts with the 'pre-compute' approach, which requires a RAM allocation of $(M(M+1))/2$ array elements for the same task.

A.2 Underflows from problematic coefficients

For spherical harmonic synthesis of very high degree (e.g. $M = 2700$), all of the methods presented here will underflow for sufficiently small values of $E_{nm\alpha}$ in Eq. (3). For example, all methods will report underflows ($\forall\theta$) when EGM96 coefficients are employed for the lower degrees. The sole cause of this is $\bar{C}_{360,360} \approx -4.5 \times 10^{-25}$, which underflows when combined with $(\bar{P}_{360,360}(\theta))/u^{360} \times 10^{-280}$ using Horner's scheme. In this case, setting $\bar{C}_{360,360}$ to zero prevents the underflow message and yields an error which is undetectable in IEEE double precision. Of course, an entire coefficient set that is relatively homogenous in magnitude can be scaled upwards or downwards as needed. Otherwise, a set of coefficients which differ by 20 orders of magnitude or more can be partitioned, according to magnitude, into subsets. Each subset is then scaled as a whole and then used to compute a corresponding partial sum. The resulting partial sums are then rescaled and combined to yield the final result.

A.3 Combining components by degree

The modified forward column and modified forward row recursions are immediately more versatile than the standard Clenshaw methods, since they do not automatically combine quantities of the same order m. This feature is necessary, for example, to form intermediate sums of quantities which share the same n, rather than the same m. For example, consider the spherical harmonic expansion of gravitational potential V in Eq. (4). Set

$$V(r,\theta,\lambda) = \frac{GM}{r} + \frac{GM}{r}\sum_{n=2}^{M}\left(\frac{a}{r}\right)^n \Omega_n^{(d)} \tag{A1}$$

where

$$\Omega_n^{(d)} = \sum_{m=0}^{n} Z_m^{(d)} \tag{A2}$$

and

$$Z_m^{(d)} = (\bar{C}_{nm1}\cos m\lambda + \bar{C}_{nm2}\sin m\lambda)\bar{P}_{nm}^{(d)}(\theta) \tag{A3}$$

For ultra-high values of M, the modified forward column or modified forward row algorithms are used to generate values of $(\bar{P}_{nm}^{(d)}(\theta))/u^m$. These replace $\bar{P}_{nm}^{(d)}(\theta)$ in Eq. (A3) to yield $Z_m^{(d)}/u^m$, which are combined using Horner's scheme

$$\begin{aligned}\Omega_n^{(d)} &= \sum_{m=0}^{n} Z_m^{(d)} \\ &= \left[\left\{\cdots\left(\left[\left\{\frac{Z_m^{(d)}}{u^m}\right\}u + \frac{Z_{m-1}^{(d)}}{u^{m-1}}\right]u + \frac{Z_{m-2}^{(d)}}{u^{m-2}}\right)u \right.\right. \\ &\quad \left.\left. + \cdots + \frac{Z_2^{(d)}}{u^2}\right\}u + \frac{Z_1^{(d)}}{u^1}\right]u + Z_0^{(d)}\end{aligned} \tag{A4}$$

to give each separate $\Omega_n^{(d)}, \forall\ 2 \le n \le M$. Inspection of Eq. (A1) shows that this algorithm provides an efficient means for computing multiple values of V (or any other gravimetric quantity) at multiple points along the geocentric radial through θ and λ.

Appendix B

In addition to the forward row recursion in Eq. (27), Libbrecht (1985) provides a second algorithm for computing values of $(\bar{P}_{nm}(\theta))/u^m$. Fully normalising Magnus et al. [1966, Eq. 4.3.3(6)] and then dividing by u^m gives

$$\begin{aligned}\frac{\bar{P}_{nm}(\theta)}{u^m} &= \frac{\sqrt{2n+1}}{\sqrt{(2n-1)(n+m)}}\left(\sqrt{n-m}\;t\,\frac{\bar{P}_{n-1,m}(\theta)}{u^m}\right. \\ &\quad \left. + \sqrt{\rho(n+m-1)}\,\frac{\bar{P}_{n-1,m-1}(\theta)}{u^{m-1}}\right),\ \forall\ n > m,\ m > 0\end{aligned} \tag{A5}$$

where $\sqrt{\rho}$ accounts for the k term [Eq. (8)] in the full normalisation of the $P_{nm}(\theta)$, and is given by $\rho = 2$ for $m = 1$ and $\rho = 1\ \forall\ m > 1$. Note that, similar to the modified forward row recursion [Eq. (27), Sect. 2.5], Eq. (A5) is presented in Libbrecht [1985, Eq. (4)] without the $\sqrt{\rho}$ term, due to a different 'normalisation' which uses $k = 1, \forall\ m > 0$. A schematic of this staggered recursion algorithm is given in Fig. B1.

To use the recursion in Eq. (A5), both the sectoral $(\bar{P}_{mm}(\theta))/u^m, \forall\ m \le M$ (upper diagonal in Fig. (31), and zonal $(\bar{P}_{n,0}(\theta))/u^0 = \bar{P}_{n,0}(\theta), \forall\ n \le M$ (leftmost column in Fig. B1), must be computed independently beforehand. The sectoral values are computed as before using Eq. (28), whilst the zonal values may be computed using the modified forward column algorithm [Eq. (32)]. Eq. (A5) can then be used to compute each of the remaining $(\bar{P}_{nm}(\theta))/u^m$ from the adjacent $(\bar{P}_{n-1,m}(\theta))/u^m$ (immediately above in Fig. B1), and $(\bar{P}_{n-1,m-1}(\theta))/u^{m-1}$ (diagonally above and to the left in Fig. B1). That is, once all of the sectoral and zonal values are known, this is sufficient to compute all $(\bar{P}_{n,1}(\theta))/u^1$ (second column to the right in Fig. B1), which may then be used to compute all $(\bar{P}_{n,2}(\theta))/u^2$ (third column to the right) and so on up to $m = M - 1$.

Libbrecht (1985, p. 372) claims that, provided that the zonal $\bar{P}_{n,0}(\theta)$ values are computed with sufficient accuracy using a "...rapidly converging trigonometric expansion...", "...one would have to go up to a very high l [degree] and m [order] indeed before roundoff errors became a problem". That is, the claim seems to be that the overall accuracy of this computing values of $(\bar{P}_{nm}(\theta))/u^m$ is superior to that of implied modified forward row recursion. However, no numerical results are provided by Libbrecht (1985) for this staggered algorithm. Moreover, results from numerical tests conducted in the current study contradict this claim. For these tests, sectoral $(\bar{P}_{mm}(\theta))/u^m$ were computed using Eq. (28), whilst the zonal $\bar{P}_{nm}(\theta)$ were computed using the standard forward column recursion [Eq. (32)] applied in IEEE extended double precision. A procedure identical to that used for the precision trials of the other recursion methods (Sect. 4.2) was employed to yield a relative precision signature for the computation of

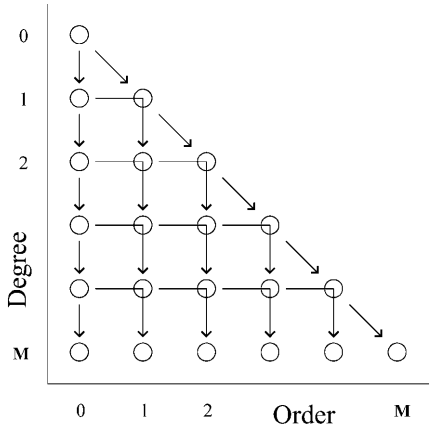

Fig. B1. Schematic of the recursion sequences employed in staggered algorithm to compute $(\bar{P}_{nm}(\theta))/u^m$

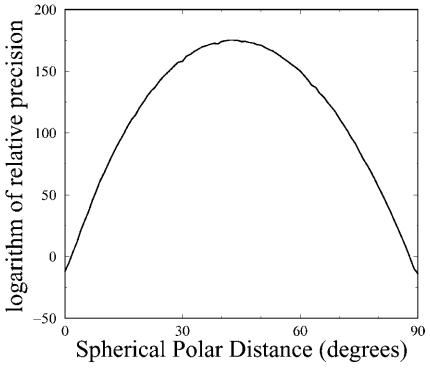

Fig. B2. Logarithm of the relative precision [Eq. (54)] to evaluate $\sum_{n=0}^{2700}\sum_{m=0}^{n}\bar{P}_{nm}(\theta)$ using the staggered algorithm

$\sum_{n=0}^{2700}\sum_{m=0}^{n}\bar{P}_{nm}(\theta)$. The plot of the relative precision statistic is shown in Fig. B2.

Inspection of Fig. B2 shows that the staggered algorithm for computing $(\bar{P}_{nm}(\theta))/u^m$ is highly unstable, except for points proximal to the poles and the equator. However, the relative precision of this algorithm close to the poles does not exceed that of any of the approaches presented in Sect. 4.2. Therefore, the staggered recursion algorithm should not be used to compute ultra-high degree and order spherical harmonic expansions.

Appendix C

This Appendix explores the possibility of formulating Clenshaw methods, based on row recursions rather than column recursions, for evaluating the partial sums $S^{(d)}$. These formulations have proven, at present, less successful than the other approaches presented in the main body of the paper. The general approach in outlined below in the event that it may yet prove useful for future developments in this area.

C.1 Two forward row factorisations

In Sect. 2.6, each of the standard [Eq. (11)], first modified [Eq. (32)] and second modified [Eq. (34)] forward column recursion algorithms are of identical form such that only the initial sectoral seed values differ. That is, Eqs. (11), (32) and (34) all employ the same recursive terms ($a_{lm}t$) and b_{lm} but use different sectoral seed values. In contrast, the modified forward row recursion [Eq. (27)] for computing $(\bar{P}_{nm}(\theta))/u^m$ differs in form from the standard row recursion [Eq. (18)] for computing $\bar{P}_{nm}(\theta)$ because the denominator u^m varies with the order of each $(\bar{P}_{nm}(\theta))/u^m$ in any given row. That is, the recursive terms, $g_{nm}(t/u)$ and h_{nm}, used in Eq. (18), are different from the equivalent terms, $g_{nm}t$ and $h_{nm}u^2$, used in Eq. (27). Thus, the standard forward row recursion and the modified forward row recursion will each yield a separate factorisation of $\bar{P}_{nm}(\theta)$. These factorisations serve to introduce the quantities $(\bar{P}_{nm}(\theta))/(\bar{P}_{nn}(\theta))$ and $(\bar{P}_{nm}(\theta))/(u^m\prod_n)$, which will be examined in the following two subsections.

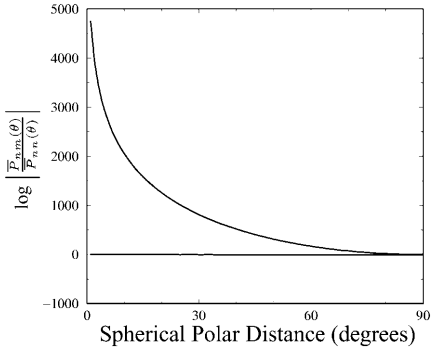

Fig. C1. Logarithm plot of maximum (*upper line*) and minimum (*lower line*) values of $|(\bar{P}_{nm}(\theta))/(\bar{P}_{nn}(\theta))|, \forall\, n, m \le 2700$

C.1.1 Standard forward row factorisation: $(\bar{P}_{nm}(\theta))/(\bar{P}_{nn}(\theta))$

The rectangle in Fig. 3 for the forward row recursions contains all $\bar{P}_{nm}(\theta)$, for which $n = 3$ and $0 \le m \le 3$. For these values of $\bar{P}_{nm}(\theta)$, the standard forward row recursion [Eq. (18)] gives

$$\bar{P}_{3,2}(\theta) = \left[g_{3,2}\left(\frac{t}{u}\right)\right]\bar{P}_{3,3}(\theta) \tag{A6}$$

$$\bar{P}_{3,1}(\theta) = \left[g_{3,1}g_{3,2}\left(\frac{t}{u}\right)^2 - h_{3,1}\right]\bar{P}_{3,3}(\theta) \tag{A7}$$

$$\bar{P}_{3,0}(\theta) = \left[\frac{1}{\sqrt{2}}\left(g_{3,0}g_{3,1}g_{3,2}\left(\frac{t}{u}\right)^3 - g_{3,0}h_{3,1}\left(\frac{t}{u}\right) - g_{3,2}h_{3,0}\left(\frac{t}{u}\right)\right)\right]\bar{P}_{3,3}(\theta) \tag{A8}$$

where the aggregations of $g_{np}(t/u)$ and h_{np} terms within the square brackets are equal to $(\bar{P}_{nm}(\theta))/(\bar{P}_{nn}(\theta))$. Here, p has been used instead of m to denote the order of each recursive term $g_{np}(t/u)$ and h_{np}, since each $\bar{P}_{3,m}(\theta)$ or order m is comprised of an aggregation of all $g_{np}(t/u)$ and h_{np} terms for $m \le p \le 2$. In general, any $\bar{P}_{nm}(\theta)$ may be factored into $\bar{P}_{nn}(\theta)$ and an aggregation of all of the $g_{np}(t/u)$ and h_{np} terms for $m \le p \le (n-1)$. Note that the denominator of $(\bar{P}_{nm}(\theta))/(\bar{P}_{nn}(\theta))$ is not $\bar{P}_{mm}(\theta)$, but $\bar{P}_{nn}(\theta)$, which is the sectoral value of the same n, rather than that of the same m.

The quantity of $(\bar{P}_{nm}(\theta))/(\bar{P}_{nn}(\theta))$ appears to be of no practical use. Inspection of Fig. C1, which gives the range of magnitudes taken by $(\bar{P}_{nm}(\theta))/(\bar{P}_{nn}(\theta))$, shows that there is no global scale factor, capable of storage in IEEE double precision, that will allow these quantities to be computed for $0^\circ < \theta < 180^\circ$ up to $M = 2700$. As such, the quantity $(\bar{P}_{nm}(\theta))/(\bar{P}_{nn}(\theta))$ will not be used to compute very high degree and order spherical harmonic expansions.

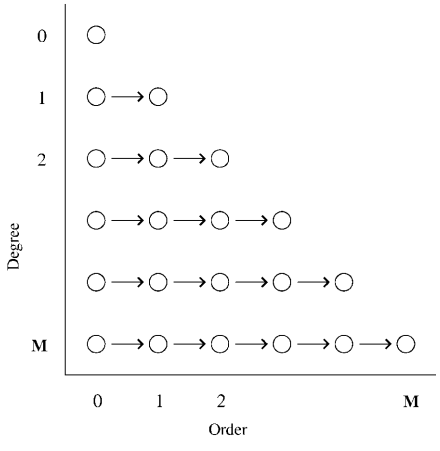

Fig. C2. A schematic of the recursion sequences employed in the reverse row algorithm to compute the quantities $\sqrt{j}\frac{\bar{P}_{nm}(\theta)}{\Pi_n u^m}$ and $\sqrt{j}\frac{\mathrm{d}(\bar{P}_{nm}(\theta))/(\Pi_n u^m)}{\mathrm{d}t}$

C.1.2 Modified forward row factorisation: $(\bar{P}_{nm}(\theta))/(u^m\Pi_n)$

Expanding on the example in Sect. C.1.1, recall that the rectangle in Fig. 3 for the forward row recursions contains $\bar{P}_{nm}(\theta)$ for which $n = 3$ and $0 \le m \le 3$. For these values of $\bar{P}_{nm}(\theta)$, the modified forward row recursion [Eq. (27)] gives

$$\frac{\bar{P}_{3,2}(\theta)}{u^2} = [g_{3,2}t]\Pi_{n=3} \tag{A9}$$

$$\frac{\bar{P}_{3,1}(\theta)}{u^1} = [g_{3,1}g_{3,2}t^2 - h_{3,1}u^2]\Pi_{n=3} \tag{A10}$$

$$\frac{\bar{P}_{3,0}(\theta)}{u^0} = \left[\frac{1}{\sqrt{2}}\left(g_{3,0}g_{3,1}g_{3,2}t^3 - g_{3,0}h_{3,1}tu^2 - g_{3,2}h_{3,0}tu^2\right)\right]\Pi_{n=3} \tag{A11}$$

where quantities in square brackets are $(\bar{P}_{nm}(\theta))/(u^m\Pi_n)$. Note that the Π_n term in the denominator renders any $(\bar{P}_{nm}(\theta))/(u^m\Pi_n)$ different from the corresponding $(\bar{P}_{nm}(\theta))/(u^m)$ (Fig. 5) by no more than one order of magnitude.

C.2 Reverse row methods

To compute any value of $(\bar{P}_{nm}(\theta))/(u^m)$, the modified forward row recursion begins with the seed value Π_m and then aggregates the necessary $(g_{np}t)$ and $(h_{np}u^2)$ recursive terms in order of decreasing p (sequentially left across each row in Fig. 3). In a manner similar to the reverse column recursion [Eq. (46)], an alternative to the modified forward row recursion [Eq. (27)] is to reverse this process and apply these same recursive terms in the sequence of increasing p (sequentially right across each row in Fig. C2). That is, a recursion may be employed whereby the $(g_{np}t)$ and $(h_{np}u^2)$ recursive terms, for which $p = m$, and are applied first and the recursive terms, for which $p = n - 1$, are applied last. This will be called a reverse row recursion and is illustrated schematically in Fig. C2.

The reverse row recursion is applied using the algorithm

$$s_{np\alpha} = g_{n,p-1}ts_{n,p-1,\alpha} - h_{n,p-2}u^2s_{n,p-2,\alpha}, +y_{np\alpha} \tag{A12}$$

To compute $(\bar{P}_{nm}(\theta))/(\Pi_n u^m)$, the recursion in Eq. (A12) is used as follows. The seed values are set to $s_{nm\alpha} = 1, s_{n,m-1,\alpha} = 0$, and all $y_{np\alpha} = 0$. This allows the recursive computation of all $s_{np\alpha}$, of constant n (a row in Fig. C2), and sequentially increasing p (across to the right and towards the diagonal in Fig. C2), from $s_{n,m+1,\alpha}$ to $s_{nn\alpha}$. The effect of using the recursion in Eq. (A12) in this way is to sequentially aggregate the $(g_{np}t$ and $(h_{np}u^2)$ terms, in the sequence of increasing p, until the recursion terminates at the computation of $s_{nn\alpha} = \sqrt{j}((\bar{P}_{nm}(\theta))/(\Pi_n u^m))$. Here, the j values (Sect. 2.2) is determined by the value of m in $\bar{P}_{nm}(\theta)$.

To compute $(\bar{P}^{(1)}_{nm}(\theta))/(u^m\Pi_n)$, differentiating Eq. (A12) with respect to $t = \cos\theta$ gives

$$\dot{s}_{np\alpha} = g_{n,p-1}(t\dot{s}_{n,p-1,\alpha} + s_{n,p-1,\alpha}) - h_{n,p-2}(u^2\dot{s}_{n,p-2,\alpha} - 2ts_{n,p-1,\alpha}) \tag{A13}$$

The seed values for the recursion in Eq. (A13), $s_{nm\alpha}$ and $s_{n,m-1,\alpha}$, are differentiated with respect to t to give $\dot{s}_{nm\alpha} = \dot{s}_{n,m-1,\alpha} = 0$. These seed values allow the recursion in Eq. (A13) to be used to compute all $\dot{s}_{np\alpha}$, of the same n, and sequentially increasing p, from $\dot{s}_{n,m+1,\alpha}$ to $\dot{s}_{nn\alpha}$. When used in this way, the recursion in Eq. (A13) will terminate at the computation of

$$\dot{s}_{nn\alpha} = \sqrt{j}\frac{\mathrm{d}\left(\frac{\bar{P}_{nm}(\theta)}{u^m\Pi_n}\right)}{\mathrm{d}t}$$

The sectoral

$$\frac{\mathrm{d}\left(\frac{\bar{P}_{nn}(\theta)}{u^n\Pi_n}\right)}{\mathrm{d}t}$$

are zero, and so their corresponding $\dot{s}_{nn\alpha}$ are simply set to zero without the need for any recursion. Application of the product rule and the chain rule to

$$\sqrt{j}\frac{\mathrm{d}\left(\frac{\bar{P}_{nm}(\theta)}{u^m\Pi_n}\right)}{\mathrm{d}t}$$

yields

$$\frac{\bar{P}^{(1)}_{nm}(\theta)}{u^m\Pi_n} = \frac{1}{\sqrt{j}}\left(m\frac{t}{u}s_{nn\alpha} - u\dot{s}_{nn\alpha}\right), \quad \forall\, n \ge m \tag{A14}$$

where, as above

$$s_{nn\alpha} = \sqrt{j}\frac{\bar{P}_{nm}(\theta)}{\Pi_n u^m}$$

and

$$\dot{s}_{nn\alpha} = \sqrt{j}\frac{\mathrm{d}\left(\frac{\bar{P}_{nm}(\theta)}{u^m\Pi_n}\right)}{\mathrm{d}t}$$

In terms of the number of recursions required, the efficiency of the reverse row techniques for computing values of $(\bar{P}_{nm}(\theta))/(u^m\Pi_n)$ and $(\bar{P}^{(1)}_{nm}(\theta))/(u^m\Pi_n)$ is the same as for the reverse column methods (Sect. 3.2) for computing values of $(\bar{P}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$ and $(\bar{P}^{(1)}_{nm}(\theta))/(\bar{P}_{mm}(\theta))$, respectively.

Similarly, overflows in the final computed values of $(\bar{P}^{(d)}_{nm}(\theta))/(u^m\Pi_n)$ are prevented in the reverse row method by setting the seed value $s_{nm\alpha}$ to 10^{-280} (rather than 1) for use in Eq. (A12). This scaling propagates linearly though subsequent computations of $\sqrt{j}((\bar{P}^{(d)}_{nm}(\theta))/(\Pi_n u^m))$ to generate values of $\sqrt{j}((\bar{P}^{(d)}_{nm}(\theta))/(\Pi_n u^m)) \times 10^{-280}$. However, while this will prevent an overflow, the reverse row method cannot be applied over

the same ranges of M and θ as the reverse column algorithm due to underflow problems during the computation. For example, for $M = 2700$, the reverse row method will underflow for $\theta < \sim 76^\circ$ and $\theta > \sim 104^\circ$. No investigation of this underflow was conducted.

C.3 Row Clenshaw methods

The standard Clenshaw methods do not translate well to row-type recursions. The first Clenshaw method uses the reverse column recursion in Eqs. (46) and (47) to compute $X^{(d)}_{m\alpha}/(\bar{P}_{mm}(\theta))$. An equivalent utilisation of the reverse row recursion in Eq. (A12) proves to be of little use. In this case, setting $s_{n,m-1,\alpha} = s_{n,m-2,\alpha} = 0$ and all $y_{np\alpha} = \bar{E}_{np\alpha}$ allows the recursive computation of all $s_{np\alpha}$, of constant n (a row in Fig. C2), and sequentially increasing p (across to the right and towards the diagonal in Fig. C2), from $s_{nm\alpha}$ to $s_{nn\alpha}$. In this case recursion terminates at

$$s_{nn\alpha} = \sum_{m=0}^{n} \sqrt{j}\left(\bar{E}_{nm\alpha}\frac{\bar{P}_{nm}(\theta)}{u^m \Pi_n}\right)$$

The $\sqrt{j}$ term notwithstanding, the principal problem with the quantity

$$s_{nn\alpha} = \sum_{m=0}^{n} \sqrt{j}\left(\bar{E}_{nm\alpha}\frac{\bar{P}_{nm}(\theta)}{u^m \Pi_n}\right)$$

is that the denominator of $(\bar{P}_{nm}(\theta))/(u^m \Pi_n)$ varies with both n and m. This means that this quantity has effectively summed components for which the scale factor is not constant, thereby preventing such sums from being combined and rescaled to achieve the final sums S.

References

Abd-Elmotaal HA (1997) An efficient technique for the computation of the gravimetric quantities from geopotential earth models. Paper presented to the IAG Scientific Assembly, 3–9 September, Rio de Janeiro

Abramowitz W, Stegan IA (1972) Handbook of mathematical functions, 9th edn. Dover Publications, New York

Belikov MV (1991) Spherical harmonic analysis and synthesis with the use of columnwise recurrence relations. Manuscr Geod 16: 384–410

Belikov MV, Taybatorov KA (1991) An efficient algorithm for computing the Earth's gravitational potential and its derivatives at satellite altitudes. Manuscr Geod 17: 104–116

Clenshaw CW (1955) A note on the summation of the Chebyshev series. Math Tab Automat Comput 9: 118–120

Colombo C (1981) Numerical methods for harmonic analysis on the sphere. Rep 310, Department of Geodetic Science and Surveying, The Ohio State University, Columbus

Coonen JT (1980) An implementation guide to a proposed standard for floating point arithmetic. Computer 13(1): 68–79

Deakin RE (1998) Derivatives of the Earth's potentials. Geom Res Aust 68: 31–60

Edmonds AR (1957) Angular momentum in quantum mechanics, Princeton University Press, Princeton, NJ

Featherstone WE (1999) Tests of two forms of Stokes's integral using a synthetic gravity field based on spherical harmonics. In: Krumm F, Schwarze VS (eds) Quo vadis geodesia. Institute of Geodesy, University of Stuttgart, pp 101–112

Gleason DM (1985) Partial sums of Legendre series via Clenshaw summation. Manuscr Geod 10: 115–130

Gulick LJ (1970) A comparison of methods for computing gravitational potential derivatives. Tech rep 40, Environmental Sciences Services Administration, US Department of Commerce, Washington, DC

Haagmans RRN (2000) A synthetic Earth for use in geodesy. J Geod 74: 531–551

Harris JW, Stocker H (1998) Handbook of mathematics and computational science. Springer, Berlin Heidelberg New York

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

Lemoine FG, Kenyon SC, Factor JK, Trimmer RG, Pavlis NK, Chinn DS, Cox CM, Klosko SM, Luthcke SB, Torrence MH, Wang YM, Williamson RG, Pavlis EC, Rapp RH, Olson TR (1998) The development of the joint NASA GSFC and the National Imagery and Mapping Agency (NIMA) geopotential model EGM96. NASA/TP-1998-206861, National Aeronautics and Space Administration, Washington, DC

Lesur V, Gubbins D (1999) Evaluation of fast spherical transforms for geophysical applications. Geophys J Int 139: 547–555

Libbrecht KG (1985) Practical considerations for the generation of large order spherical harmonics. Solar Phys 99: 371–373

Magnus W, Oberhettinger F, Soni RP (1966) Formulas and theorems for special functions of mathematical physics, 3rd edn. Springer, Berlin Heidelberg New York

Moritz H (1980) Geodetic reference system 1980. Bull Géod 54(4): 395–405

Novák P, Vanícek P, Véronneau M, Holmes SA, Featherstone WE (2001) On the accuracy of Stokes's integration in the precise high-frequency gravimetric geoid determination. J Geod 74: 644–654

Pail R (1999) Synthetic global gravity model for planetary bodies and applications in satellite gravity gradiometry. PhD thesis, Technical University of Graz

Risbo T (1996) Fourier transform summation of Legendere series and D-functions. J Geod 70: 383–396

Rizos C (1979) An efficient computer technique for the evaluation of geopotential from spherical harmonic models. Aust J Geod Photogram Surv 31: 161–169

Tscherning CC (1976) Computation of second-order derivatives of the normal potential based on the representation by a Legendre series. Manuscr Geod 1: 71–92

Tschering CC, Poder K (1982) Some geodetic applications of Clenshaw summation. Boll Geofis Sci Aff 4: 351–364

Tscherning CC, Rapp RH (1974) Closed covariance expressions for gravity anomalies, geoid undulations, and deflections of the vertical implied by anomaly degree variance models. Rep 208, Department of Geodetic Science and Surveying, The Ohio State University, Columbus

Tscherning CC, Rapp RH, Goad C (1983) A comparison of methods for computing gravimetric quantities from high degree spherical harmonic expansions. Manuscr Geod 8: 249–272

Wenzel G (1998) Ultra-high degree geopotential models GPM98A, B, and C to degree 1800. Paper presented to the joint meeting of the International Gravity Commission and International Geoid Commission, 7–12 September, Trieste