New methods for numerical evaluation of ultra-high degree and order associated Legendre functions

MEHDI GOLI[1], ISMAEL FOROUGHI[2] AND PAVEL NOVÁK[3]

1 Shahrood University of Technology, Faculty of Civil Engineering, Iran

2 Department of Earth and Space Science and Engineering, York University, Toronto, ON, Canada (foroughi.ismael@gmail.com, iforough@yorku.ca)

3 New Technologies for the Information Society, University of West Bohemia, Pilsen, Czech Republic

Received: May 3, 2022; Revised: July 23, 2022; Accepted: August 26, 2022

ABSTRACT

We improve the precision and computation speed of the fully-normalized associated Legendre functions (fnALFs) for ultra-high degrees and orders of spherical harmonic transforms. We take advantage of their numerical behaviour of and propose two new methods for solving an underflow/overflow problem in their calculation. We specifically discuss the application of the two methods in the fixed-order increasing-degree recursion computation technique. The first method uses successive ratios of fnALFs and the second method, called the Midway method, starts iteration from tiny initial values, which are still in the range of the IEEE double-precision environment, rather than from sectorial fnALFs. The underflow/overflow problem in the successive ratio method is handled by using a logarithm-based method and the extended range arithmetic. We validate both methods using numerical tests and compare their results with the X-number method in terms of precision, stability, and speed. The results show that the relative precision of the proposed methods is better than 10^{-9} for the maximum degree of 100000, compared to results derived by the high precision Wolfram's Mathematica software. Average CPU times required for evaluation of fnALFs over different latitudes demonstrate that the two proposed methods are faster by about 10–30% and 20–90% with respect to the X-number method for the maximum degree in the range of 50–65000.

Keywords: associated Legendre function, underflow/overflow problem, successive ratio, recursive formula, spherical harmonic analysis/synthesis, X-number method

1. INTRODUCTION

The spherical approximation of the Earth facilitates the representation of its external gravitational field using a series expansion of spherical harmonics. While spherical harmonics represent basis functions, associated numerical coefficients can be estimated using observable quantities of the gravity field. This process is known as the spherical harmonic analysis. Reconstructing the gravitational potential (or its functionals) using the estimated numerical coefficients is then called the spherical harmonic synthesis. The spherical harmonic transform, which refers to both the synthesis and analysis, is the most used method for representation and spectral analysis of gravitational fields of the Earth and other planets, see, e.g., *Wieczorek and Meschede (2018)*.

In the spherical harmonic synthesis, a harmonic function f defined on a sphere is represented by the following series (*Heiskanen and Moritz, 1967, Sect. 1-13*):

$$f(\theta,\lambda)=\sum_{n=0}^{\infty}\sum_{m=0}^{n}\left(C_{nm}\cos m\lambda+S_{nm}\sin m\lambda\right)P_{n,m}(\cos\theta), \quad (1)$$

where λ and θ are the spherical longitude and co-latitude, C_{nm} and S_{nm} are numerical coefficients of the spherical harmonic series of degree n and order m, and $P_{n,m}$ are associated Legendre functions (ALFs) of the first kind. The maximum degree and order of the series expansion depend on the angular resolution of observable quantities used for the spherical harmonic analysis. With more datasets available from satellite, airborne, and terrestrial measurements, high degree/order spherical harmonic transforms (e.g., $n \geq 1000$) are often required in recent geoscience and planetary applications. However, evaluation of the ALFs is numerically challenging for high degrees and orders. In geodesy and some branches of geophysics, fully-normalized ALFs (fnALFs), denoted as $\bar{P}_{n,m}$, are used because of their simplicity (of course, respective numerical coefficients must also be normalized) compared to ALFs. Using the so-called geodetic norm, the fnALfs read (*Heiskanen and Moritz, 1967, Sec. 1-14*):

$$\bar{P}_{n,m}(t)=\sqrt{\left(2-\delta_{m,0}\right)(2n+1)\frac{(n-m)!}{(n+m)!}}P_{n,m}(t), \quad 0\leq m\leq n, \quad (2)$$

with Kronecker's delta $\delta_{m,0}$ and $t = \cos\theta$. Numerical evaluation of fnALFs for very-high degrees, i.e., for n larger than 1000, is a time-consuming and for the polar regions numerically unstable process. Different evaluation methods have been proposed to speed up their calculations while avoiding numerical instabilities due to under- or overflow problems. A commonly used approach for fast computation of fnALFs is a fixed-order increasing-degree (FOID) formula (*Holmes and Featherstone, 2002*)

$$\begin{gathered}\bar{P}_{n,m}(t)=a_{nm}t\,\bar{P}_{n-1,m}(t)-b_{nm}\bar{P}_{n-2,m}(t), \quad n\geq 2, \quad 0\leq m\leq n-2, \\ a_{nm}=\sqrt{\frac{(2n-1)(2n+1)}{(n-m)(n+m)}}, \quad b_{nm}=\sqrt{\frac{(2n+1)(n+m-1)(n-m-1)}{(2n-3)(n-m)(n+m)}},\end{gathered} \quad (3)$$

with initial values of sectorial fnALFs $\bar{P}_{n,n}$ computed using the following recursive formula with $\bar{P}_{0,0}=1$ and $u = \sin\theta$

$$\bar{P}_{n,n}(t) = \sqrt{\frac{2n+1}{2n}} u \bar{P}_{n-1,n-1}(t), \quad n \geq 1 . \tag{4}$$

The FOID recursion has been used for numerical evaluation of fnALFs in many studies (e.g., *Colombo, 1981*; *Smith et al., 1981*; *Prézeau and Reinecke, 2010*; *Balmino et al., 2011*; *Fukushima, 2012*; *Ishioka, 2018*). Most software packages developed for the spherical harmonic synthesis and analysis of the gravitational field parameters have implemented this numerical method, e.g., Libpsht (*Reinecke, 2011*); SHTns (*Schaeffer, 2013*); ISPACK (*Ishioka, 2018*); and SHTOOLS (*Wieczorek and Meschede, 2018*).

Equation (4) implies that values of the sectorial fnALFs uniformly decrease with the increasing degree n, particularly when u is small, i.e., in the polar regions. For large degrees n and small values of the parameter u, numerical evaluation of high-degree sectorial fnALFs faces an underflow error, i.e., their numerical values cannot be represented by a double-precision floating number. According to the IEEE754-2008 standard (*IEEE, 2008*), the allowed range of double-precision floating numbers is $[2.2 \times 10^{-308}, 1.8 \times 10^{+308}]$. Figure 1 represents a logarithm of the sectorial fnALFs for two selected co-latitudes $\theta = 5°$ and $30°$. This figure demonstrates that the underflow error occurs for $\theta = 5°$ and $30°$ at $n > 290$ and $n > 1025$, respectively. Figure 1 also implies that using a quadruple-precision environment with the range of $[6.5 \times 10^{-4966}, 1.2 \times 10^{+4932}]$ does not represent an alternative for evaluation of ultra-high degree fnALFs as the underflow error occurs again at $n > 4688$ when $\theta = 5°$.

Different schemes have been proposed to overcome the underflow problem (e.g., *Hauser, 1996*). Among them, extended-range arithmetic (ERA) and scaling methods guarantee acceptable numerical precisions within reasonable computation times *(Smith et al., 1981*; *Prézeau and Reinecke, 2010*; *Nesvadba, 2011*; *Fukushima, 2012*). *Smith et al. (1981)* applied the ERA method to compute ultra-high degree (e.g., $n > 10000$) fnALFs over the range $\varepsilon < \theta < \pi - \varepsilon$, where ε is the machine epsilon. Their software increases the

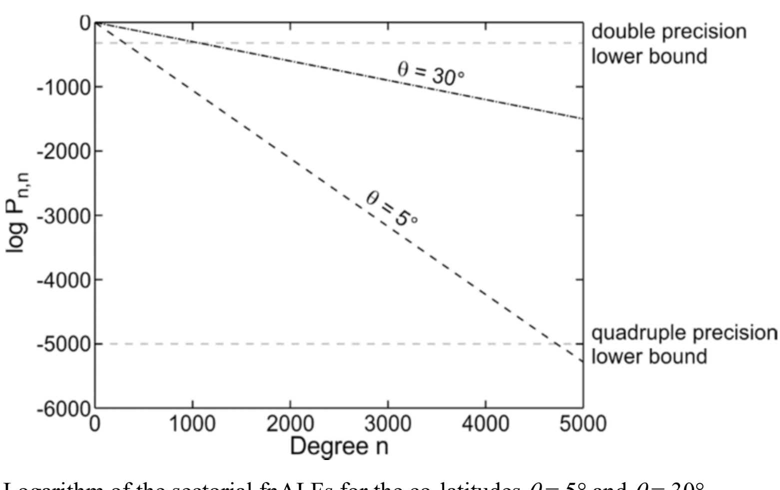

Fig. 1. Logarithm of the sectorial fnALFs for the co-latitudes $\theta = 5°$ and $\theta = 30°$.

computation time by the factor of 2 with respect to an ordinary recursive formula. *Fukushima (2012)* improved the ERA method by selecting an appropriate radix and automatic switching from the ERA mode to the ordinary mode whenever fnALF values return to the range of the IEEE standard. Fukushima's software can compute fnALFs of an arbitrary large degree while the execution time is only 10% longer compared to the ordinary formula.

The scaling method for computation of fnALFs up to the degree 1800 was used by *Wenzel (1998)*. *Holmes and Featherstone (2002)* developed their computational method based on *Libbrecht (1985*) by replacing $\bar{P}_{n,m}$ with $H_{n,m} = \bar{P}_{n,m} \big/ u^m$. According to their computational method, the term u^m, which is the main cause of the underflow problem in computing high-degree fnALFs, is removed from the FOID formula. However, increasing the degree n in $(2n+1)/2n > 1$ will still cause an overflow error in Eq. (4); therefore, they scaled all coefficients by the factor of 10^{-280}. Using this technique, fnALFs can safely be computed up to the degree of 2700 without any overflow errors. This technique is used in the Synth.f software (*Holmes and Pavlis, 2006*) and also in the SHTOOLS library (*Wieczorek and Meschede, 2018*). *Balmino et al. (2012)* utilized another scale technique to compute fnALFs up to the degree of 30000. They started with the factor of 10^{-280}, which was re-used whenever an overflow error occurred. Values of $H_{n,m}$ are then converted back to fnALFs using a logarithmic function which saves the computational time (*Novikova and Dmitrenko, 2016*).

Dmitrenko (2012) transformed the FOID recurrence formula into the logarithmic mode. *Novikova and Dmitrenko (2016)* reported that evaluation of fnALFs using the logarithmic method is 6 times slower than using the ERA method for the degree $n = 8000$ as exponential and logarithmic functions need extensive computation times. *Xing et al. (2019)* proposed a four-term recursive column-wise approach (fixed-degree increasing-order). They showed that their method was 15–22% faster than *Fukushima (2012)* software. Their method has a very simple mathematical formulation and a single fnALF can be computed using $\bar{P}_{n-1,m-1}$, $\bar{P}_{n-1,m}$, and $\bar{P}_{n-1,m+1}$. In another words, the computation of fnALFs of the order m depends on fnALFs of the order $m-1$. Thus, parallel computations cannot be employed for evaluation of a single fnALF. In conclusion, the methods discussed above are either too slow when removing the under- or overflow problems or cannot be used for parallel computations. This is the main motivation of our study.

In this article, we first discuss the behaviour of fnALFs in Section 2. Then we propose two new methods for numerical evaluation of high-degree fnALFs in Section 3 which are superior to the existing methods in terms of precision and speed. The new methods are tested numerically in Section 4, and Section 5 summarizes main findings and concludes the article.

2. NUMERICAL BEHAVIOUR OF fnALFs

We study first the behaviour pattern in numerical values of fnALFs computed by the FOID recursive formula in order to find its efficient evaluation method. Numerical analyses show that ALFs of different orders m have for $0 < \theta < \pi$ same numerical properties. Sectorial ALFs ($P_{n,n}$) have always the smallest absolute values among all degrees and are positive according to Eq. (4). By increasing the degree n, values of $\bar{P}_{n,m}$ monotonically increase until reaching their global maxima, then oscillate around stable values. Assuming N_{max} is the maximum degree of the spherical harmonic expansion, the degree band $m \leq n \leq N_{max}$ may be split into a monotonically-increasing region, in which values of fnALFs are positive, and an oscillating region. Figure 2 depicts examples of the fnALFs behaviour for different values of m and θ.

One possible method for the stable numerical evaluation of fnALFs without underflow errors is starting recurrence from a very small fnALF which is still in the range of the IEEE standard for double-precision numbers. There are two key issues that must be

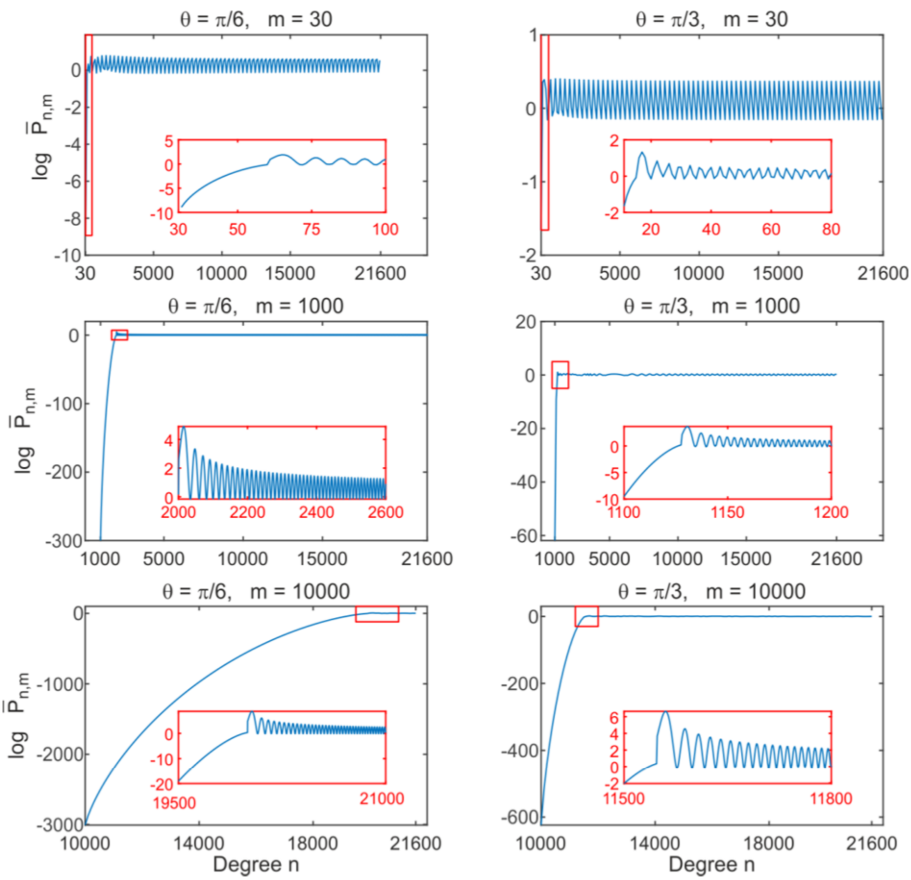

Fig. 2. Behaviour of $\bar{P}_{n,m}$ for co-latitude $\theta = \pi/6$ (left) and $\pi/3$ (right); order m is fixed and degree $n = m$:21600. The insets show finer details at the boundary between the monotonically-increasing and oscillating regions.

solved: one is the choice of the appropriate degree n for starting the recursion and the other one is the calculation of seed values. To tackle the first issue, we use the property of fnALFs at the boundaries between the two regions, see Fig. 1. According to this figure, the variability of fnALF values decreases immediately before the end of the monotonically-increasing region. Thus, assuming $\bar{P}_{n,m} \approx \bar{P}_{n-1,m} \approx \bar{P}_{n-2,m}$ for a sufficiently large degree n, Eq. (3) may be reduced to

$$1 = a_{nm} t - b_{nm} , \tag{5}$$

i.e., we assume

$$\frac{2n+1}{2n-3} \approx 1 , \quad n+m-1 \approx n+m , \quad n-m+1 \approx n-m , \quad (2n+1)(2n-1) \approx 4n^2 .$$

In this case, $a_{nm} \approx 2n \big/ \sqrt{n^2 - m^2}$ and $b_{nm} \approx 1$. Substituting a_{nm} and b_{nm} in Eq. (5) yields

$$m \approx n\, u . \tag{6}$$

Equation (6) shows a linear relation at the boundary between the monotonically-increasing and oscillating regions. *Jekeli et al. (2007)* derived the same formula for the fixed-degree increasing-order recursion. Figure 2 shows that the global maxima of fnALFs appear at the end of the monotonically-increasing regions. To validate Eq. (6), the global maxima of fnALFs are compared for different values of the order m and θ in Table 1. As expected, the degree derived by Eq. (6) is always smaller than the degree which corresponds to the global maximum of fnALF.

To find the degree n_0 with a sufficiently small numerical value of fnALF, say 10^{-30}, the parameter μ is applied to Eq. (6) as follows:

$$n_0 = \frac{m - \mu}{u} . \tag{7}$$

The parameter μ depends on m, N_{max}, and θ. It would be difficult to find an explicit formula that works well for any arbitrary m, N_{max}, and θ. *Jekeli et al. (2007)* assumed that values of ALFs are almost equal to zero for $m > (n_0 + \mu)u$ and for $\mu = 15$. They used this condition to overcome the instability of the fixed-degree increasing order recurrence method for $N_{max} = 10800$. *Balmino et al. (2012)* proposed that the FOID is carried over all orders up to

$$m = n\, u + 20 + 8 \times 10^{-3}\, N_{max} . \tag{8}$$

Table 1. Location of $\max\left(\bar{P}_{n,m}\right)$ and $n = m\, u^{-1}$.

θ	m	$\mathrm{true}(N_{max})$	$N_{max} = m\, t^{-1}$
$\frac{\pi}{6}$	30	64	60
	1000	2014	2000
	10000	20031	20000
$\frac{\pi}{3}$	30	36	35
	1000	1160	1155
	10000	11559	11547

Table 2. The maximum values of $\bar{P}_{n,m}(t)$ for $\theta \in (0, \pi/2)$ using Eqs (7) and (8).

n	64800	43200	21600	10800	5400	2160	360
Eq. (7), $\mu = 15$	4.96	4.04	2.81	1.88	1.16	4.99e^{-01}	2.61e^{-02}
Eq. (8)	1.23e^{-32}	2.10e^{-22}	3.45e^{-12}	3.62e^{-07}	9.91e^{-05}	1.88e^{-03}	6.87e^{-04}

To validate Eqs (7) and (8), Table 2 summarizes the largest values of ALFs corresponding to Eqs (7) and (8) over the values $\theta \in (0, \pi/2)$, with the 30″ step. According to Table 2, neither of the formulas is suitable for arbitrary values of the degree N_{max} and co-latitude θ. Therefore, the appropriate degree n_0 must be computed using a trial and error approach.

3. NEW METHODS

3.1. Successive ratio method

The successive ratio property of fnALFs can be obtained by dividing Eq. (1) by $\bar{P}_{n,m-1}$, i.e.,

$$\frac{\bar{P}_{n,m}(t)}{\bar{P}_{n-1,m}(t)} = a_{nm}t - b_{nm}\frac{\bar{P}_{n-2,m}(t)}{\bar{P}_{n-1,m}(t)}, \quad n \geq 2, \quad 0 \leq m \leq n-2 . \tag{9}$$

Assuming $d_{n,m}(t) = \bar{P}_{n,m}(t) / \bar{P}_{n-1,m}(t)$, we get

$$d_{n,m}(t) = a_{nm}t - \frac{b_{nm}}{d_{n-1,m}(t)}, \quad n \geq 2, \quad 0 \leq m \leq n-2 . \tag{10}$$

The recurrence in Eq. (10) starts from $\bar{P}_{m+1,m}(t) / \bar{P}_{m,m}(t) = a_{mm}t$. Thus, fnALFs can easily be calculated using:

$$\bar{P}_{n,m}(t) = \bar{P}_{m,m}(t) d_{m+1,m}(t) d_{m+2,m}(t) \ldots d_{n,m}(t), \tag{11}$$

where $d_{n,m}$ is computed using Eq. (10). As shown in Fig. 2, values of fnALFs are so small in the underflow region, so they do not have to be computed there at all. Indeed, we require only two $\bar{P}_{n_0,m}$ and $\bar{P}_{n_0+1,m}$ greater than the threshold η (η is a small number close to the lower bound of the IEEE precision, e.g., $\eta = 10^{-280}$). Assuming $\bar{P}_{n_0,m}$ is the first fnALF greater than η, then:

$$\bar{P}_{n_0,m}(t) = \bar{P}_{m,m}(t) \prod_{k=m+1}^{n_0} d_{k,m}(t). \tag{12}$$

The key property of our method is fast and stable evaluation of Eq. (12) for very small values of fnALFs, i.e., to avoid underflow errors, as well as for large values of the product of the two ratios, i.e., to avoid overflow errors.

The FOID recursion in Eq. (4) is faster than the recursion of ratios of two ALFs as the latter contains a division operator which slows the computations down but handles the underflow issue. To avoid the division operator in Eq. (10), the product of ratios can be evaluated using the following recursive formula

$$M_{n+2,m}\left(t\right)=a_{n+2,m}tM_{n+1,m}\left(t\right)-b_{n+2,m}M_{n,m}\left(t\right),\quad n\geq m\,, \tag{13}$$

where $M_{n,m}=\prod_{k=m+1}^{n}d_{n,m}$ and the seed values are $M_{m,m}=1$ and $M_{m+1,m}=a_{m+1,m}t$. Equation (13) is obtained by dividing both sides of Eq. (3) by $\overline{P}_{m,m}$. The underflow issue in computation of sectorial fnALFs can be managed by using logarithmic functions. It was mentioned in the previous section that values of fnALFs are always positive within the monotonically-increasing region. Therefore, from Eq. (12) we have

$$\log\overline{P}_{n_0,m}\left(t\right)=\log\overline{P}_{m,m}\left(t\right)+\log M_{n_0,m}\left(t\right). \tag{14}$$

The logarithm of sectorial fnALFs is calculated using the following recurrence:

$$\log\overline{P}_{m,m}\left(t\right)=\log\overline{P}_{m-1,m-1}\left(t\right)+\log u+\log\sqrt{\frac{2m+1}{2m}}\,,\quad m>0\,, \tag{15}$$

where $\log\overline{P}_{0,0}\left(t\right)=0$. The overflow issue in numerical evaluation of $\log M_{n_0,m}$ can be handled by applying a simple rescaling method. In the recursive formula of Eq. (13), $M_{n,m}$, $M_{n+1,m}$, and $M_{n+2,m}$ are multiplied by a suitable factor such as $\mu=10^{-10}$ whenever $M_{n+2,m}>\mu$ and $-\log\mu$ should be added to $\log\overline{P}_{m,m}$. This technique solves the overflow problem, that occurs in Eq. (13), and reduces time-consuming logarithmic computations to only one instance. This algorithm can be used for numerical evaluation of fnALFs for arbitrarily large degrees and orders, and for $0<\theta<\pi$ without over- or underflow issues. To increase its computational efficiency, calculations are performed using FOID whenever $\log\overline{P}_{n_0,m}>\log\eta$.

A simple ERA method is a possible alternative for the evaluation of the expression in Eq. (12). This method is equivalent to the extended-range arithmetic method presented by *Fukushima (2012)*. In this method, sectorial fnALFs are calculated by an ERA number as follows:

$$X=xB^{ix}\,, \tag{16}$$

where x is a float number and ix is an integer. The parameter B is a large number. Here $B=\eta^{-1/2}=10^{140}$. As all values of fnALFs are positive in the monotonically-increasing region, their transformation is easy. For example, sectorial fnALFs computed by recursion in Eq. (5) are compared with B^{-1}. Whenever $\bar{P}_{m,m} < B^{-1}$, then $ix = ix + 1$ and $x = x/\eta$.

To evaluate the expression in Eq. (12), the product of ratios $M_{n,m}$ is computed using recurrence in Eq. (13) with the initial values $M_{n,m} = x$ and $M_{n+1,m} = xa_{n+1,m}t$, and x is the floating number part of $\bar{P}_{m,m}$ in the ERA presentation. As $M_{n,m} > 1$ and the numerical values of the functions $M_{n,m}$, $M_{n+1,m}$, and $M_{n+2,m}$ do not change significantly, one addition and two multiplications of recurrence in Eq. (13) are performed as ordinary operators. The overflow issue of Eq. (13) can be solved in a simple manner. For any degree n, if $M_{m+2,m} > B$, then $ix = ix - 1$ and $M_{m+2,m}x = M_{m+2,m}x/\eta$. This process continues for $M_{n,m}$ until $ix < 2$ and $M_{n,m} > \eta$. In this case, the $M_{n,m}$ becomes larger than η and the final value of the expression in Eq. (12) is an ordinary float number. Remaining calculations are performed using the standard FOID formula. The proposed algorithm is simple as it requires only very few extra additions and multiplication with respect to the standard form. As fnALFs do not need to be computed in the underflow region, extra computations, such as the sum of ERA numbers and the transformation of a floating point to the ERA number and vice-versa, are avoided. This is the main difference between our method and other methods previously reported (*Smith et al., 1981*; *Fukushima, 2012*).

3.2. Midway method

The basic idea of this method is to use Eq. (7), i.e., a linear relation that separates the monotonically-increasing and oscillating regions of fnALFs. In this method, the recursion starts from an appropriate degree n_0 for which the fnALF value is very small but still in the range of the IEEE standards. Equation (7) is then used to find an approximate value of n_0. The appropriate value is determined by a trial and error method in the vicinity of n_0. When it is estimated, seed values of $\bar{P}_{n_0,m}$ and $\bar{P}_{n_0+1,m}$ are required to start the FOID recurrence. These functions can simply be computed by a stable formula using $\bar{P}_{n,m-1}$ and $\bar{P}_{n-1,m-1}$, i.e., using values from the previous column:

$$\bar{P}_{n,m}(t) = -\frac{t}{u}\sqrt{\frac{n-m+1}{n+m}}\bar{P}_{n,m-1}(t) + \frac{1}{u}\sqrt{\frac{(2n+1)(n+m-1)}{(2n-1)(n+m)}}\bar{P}_{n-1,m-1}(t), \quad n, m > 0 . \quad (17)$$

The validity of Eq. (17) is proven in Appendix A.

In the Midway method, the seed values for the order m are obtained from $m-1$ when the underflow error occurs. Therefore, parallel computations of a single fnALF cannot be used over different orders m unless performed over one latitude. This limitation is the main disadvantage of the method.

4. NUMERICAL TESTS

In this section, the numerical precision and computational cost of the methods discussed above are examined. Tested methods include 1) successive ratio method with logarithm, 2) successive ratio method with ERA, and 3) Midway method. The former two methods are denoted as Ratio_log and Ratio_ERA, respectively. Their results are also compared against values computed using the code provided by *Fukushima (2012)* called herein the X-number method. All codes are written in Fortran 90 using the standard double-precision environment. To compile and run the software, a laptop with the Ryzen 7 5700u processor was used. The GCC compiler of gfortran v.11.2 under the command –O3 –march=native was employed.

4.1. Spot verification

Numerical values of fnALFs computed using the discussed methods were also compared with results computed using *Wolfram (2003)* Mathematica software. For each method, the relative numerical error is computed as follows:

$$\bar{\varepsilon} = \left| \frac{\bar{P}_{n,m}(t) - \bar{P}^{true}_{n,m}(t)}{\bar{P}^{true}_{n,m}(t)} \right| , \tag{18}$$

where $\bar{P}^{true}_{n,m}$ is computed by Wolfram's Mathematica with 35 significant digits. Since computations of $\bar{P}^{true}_{n,m}$ are rather time-consuming, comparisons are limited to θ of $\pi/36$, $\pi/6$, $\pi/3$, and to N_{max} in the range from 1000 to 10000. The order m equals to $N_{max}/2$ for θ of $\pi/6$ and $\pi/3$, and to $N_{max}/20$ for $\theta = \pi/36$. Note that $P_{n,n/2}(\cos(\pi/36)) = 0$ for $N_{max} > 1000$ in the double-precision environment.

Figure 3 shows relative precisions of the proposed methods. As expected, relative errors increase for increasing N_{max} and decreasing θ. In general, numerical precisions of the X-number method and the Ratio_ERA method are comparable while the precision of the Ratio_log method is about three orders of magnitude worse for $\theta = \pi/6$. For small $\theta = \pi/36 = 5^\circ$, results of all methods are about the same. Their relative precision is better than 10^{-9}, which is sufficient for applications of spherical harmonics syntheses and analyses in geosciences.

The relative precision of fnALFs is estimated using the following identity (*Holmes and Featherstone, 2002*):

$$\delta I = \left| \frac{I}{(N+1)^2} - 1 \right| , \quad I = \sum_{n=0}^{N} \sum_{m=0}^{n} \bar{P}_{n,m}(t)^2 = (N+1)^2 . \tag{19}$$

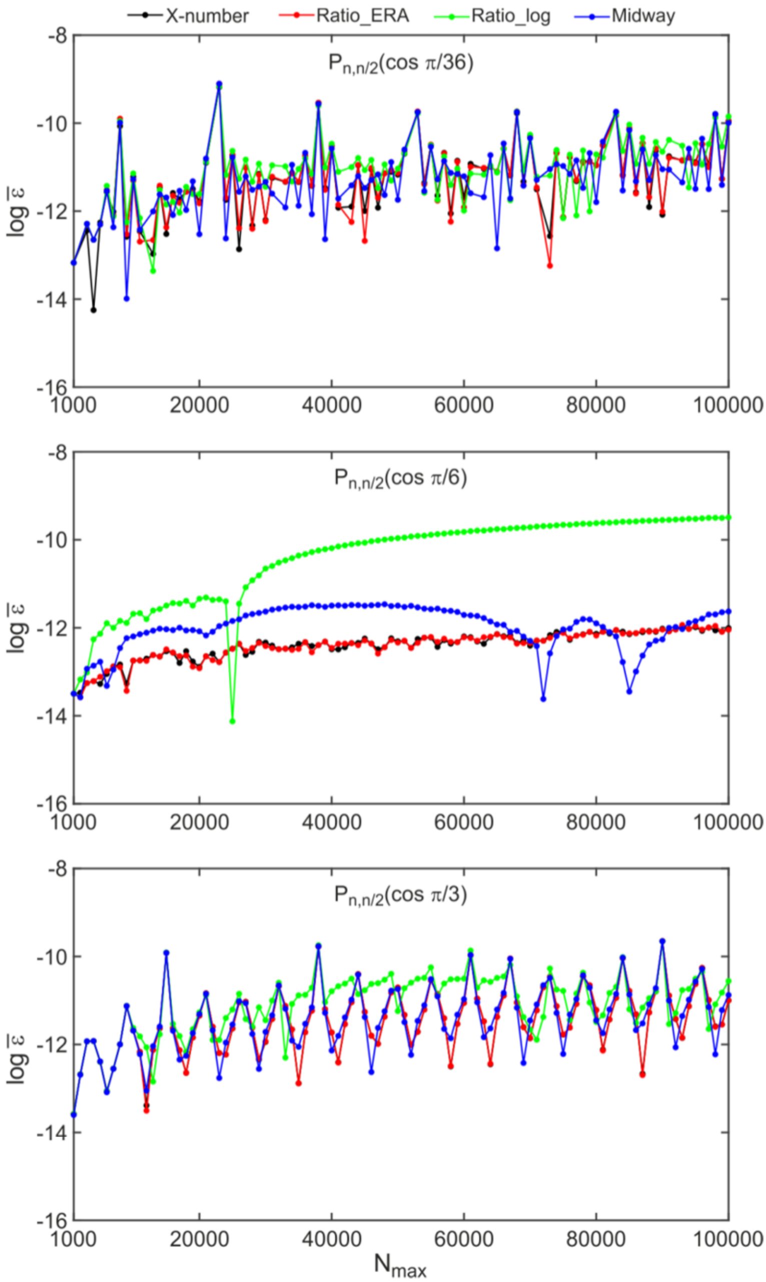

Fig. 3. Logarithms of relative error $\overline{\varepsilon}$ (Eq. (18)) of the tested methods compared to Wolfram's Mathematica for $P_{n,n/2}(\cos\theta)$ with co-latitudes θ of $\pi/36$, $\pi/6$ and $\pi/3$.

Typically, the precision of the recursive methods decays for the increasing degree and for points close to the poles. To investigate the effect of high latitudes on the computational precision, the relative error δI is computed on the uniform grid defined as follows:

$$\theta_i = \left[\Delta_0 , \Delta_0 + 15', \Delta_0 + 30' + \ldots + 90^\circ - \Delta_0 \right], \qquad (20)$$

where $\Delta_0 = 180^\circ / N_{max}$, and the number of the grid points is 360. To guarantee that the above methods provide sufficient accuracy for computation of fnALFs as part of spherical harmonic transforms up to degree and order N_{max}, the first point of the grid in Eq. (20) was considered equal to Δ_0. Figure 4 shows the variation of δI corresponding to maximum degrees 2160, 21600, and 64800. According to Fig. 4, the general behaviour of any method is similar for different degrees and, as expected, the larger the degree N_{max}, the larger the value of δI. The precision of all methods depends on the latitude, the precision is reduced for points close to the poles. The precision of the Ratio_log method is one-two orders of magnitude worse than that of the other two methods for mid-latitudes and for N_{max} of 21600 and 64800. The precision of all methods, but the Ratio_log method, is almost the same.

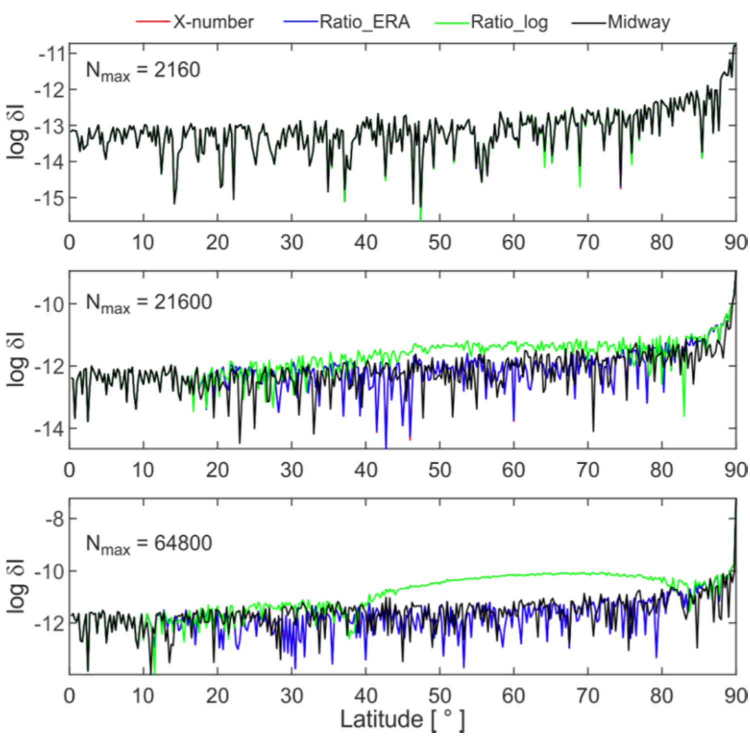

Fig. 4. Logarithms of the relative precision δI (Eq. (19)) for different latitudes and three values of N_{max}.

4.2. Time analysis

In this subsection, the efficiency of the methods in terms of the CPU time is investigated. For the X-number and ratio methods, computations dynamically switch to float numbers whenever this is possible which improves the computational speed significantly. Figure 5 shows the normalized computation times for different N_{max}. This time is obtained by dividing the sum of the computation times for the uniform grid $\theta \in (0, \pi/2)$ by the number of fnALFs, i.e., $(N_{max}+1)(N_{max}+2)/2$. As expected, the Midway method is the fastest as it cancels out the computation of all ALFs with η smaller than 10^{-280}. The X-number method is the slowest and the computation time of the Ratio_log method is slightly shorter than that of the ratio_ratio method for N_{max} from 500 to 65000. The Midway method is 20–90% faster than the X-number method for the same range of maximum degrees. The ratio-based methods are about 10–30% faster than the X-number method for N_{max} from 500 to 65000. The trend of computation times in Fig. 5 shows that for degree n above 65000, the efficiency of the Midway and ratio-based methods with respect to the X-number method has increased.

4.3. Polar region optimization

As discussed above, fnALF values decay fast in the polar regions which allows for optimizing their computations by cancelling out those fnALFs which are smaller than some threshold value such as 10^{-50}. For all methods based on Eq. (7), we can limit the degree n to $m_0 = \min(N_{max}, N_{max}u + u)$, while keeping the same precision. Figure 6 shows the cancel-out area for $\theta = 15°$. Using this method, a large amount of computations can be avoided for decreasing u, i.e., in the polar regions. The ratio of eliminated functions to total functions required for computations depends on the latitude θ. For the grid defined in Eq. (20), this ratio may reach 20% for $N_{max} \geq 2000$.

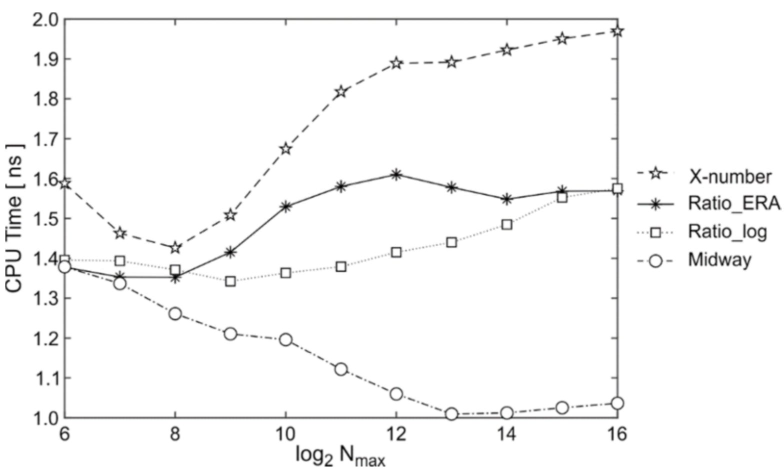

Fig. 5. CPU times of the four evaluation methods for different N_{max}.

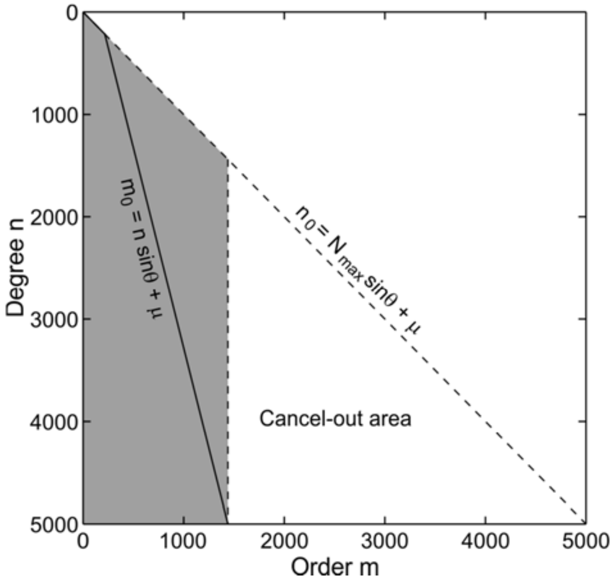

Fig. 6. Scheme of the optimization area for latitude $\theta = 15°$.

Figure 7 shows the normalized computation times of the methods after their polar optimization. Similar to the results reported in the previous subsection, the X-number and Midway method is the slowest and fastest, respectively. Numerical results show that the polar optimization may reduce computation times by about 60%, 35%, 35% and 10% for the X-number, Ratio_ERA, Ratio_log, and Midway methods, respectively. It is easy to conclude that the Midway method cannot fully be optimized using Eq. (7) as this method

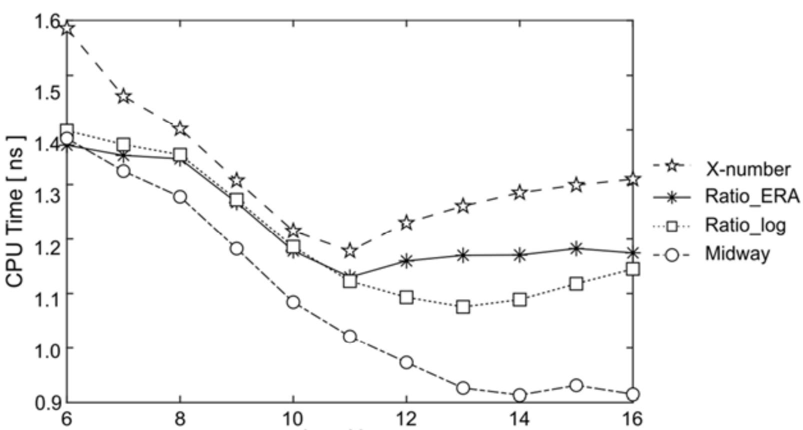

Fig. 7. The same as in Fig. 5, but after the polar optimization.

is already inherently optimized for the polar regions. The percentage improvement for the three former methods is greater than the percentage of eliminated fnALFs (about 20%) because computations of small fnALFs for higher degrees and orders are more expensive due to extra computations required to handle the underflow problem.

The optimized CPU times imply that the Midway method is 15−45% faster than the X-number method for N_{max} from 2000 to 65000. The ratio-based methods are about 5−10% faster than the X-number method for the same range of N_{max}.

5. CONCLUSIONS

We introduced two new methods to overcome the under- and over-flow error in the computation of high-degree fnALFs. Both methods were applied for numerical evaluation of the high-degree/order fnALFs using the FOID recursive formula. We found from investigations of the numerical behaviour of fnALFs for the fixed order m and different degrees $n = m : N_{max}$, that fnALF values increase monotonically from the sectorial fnALFs to their global maxima, then oscillate around a stable value while their values in this region are all positive. The monotonically-increasing and oscillating regions are separated by a simple linear relationship $n \approx m/u$. For sufficiently small θ and large N_{max}, the underflow problem occurs in the monotonically-increasing region. Two different methods are proposed that avoid evaluations of unnecessary fnALFs in the monotonically-increasing region which solves the underflow problem and reduces the CPU time.

The first method takes advantage of the successive ratios of fnALFs. For small sectorial ALFs, multiplications of the successive ratios of fnALFs suffer from an overflow error in the monotonically-increasing region. In other words, under- and overflow errors occur while evaluating sectorial ALFs and successive ratios of fnALF. Both under- and overflow problems are solved using logarithm or a simple form of extended-range arithmetic.

In the second method, an iteration process starts from the degree n_0 for which the fnALF value is a very small double-precision number rather than from sectorial and semi-sectorial fnALFs. New seed values are computed by a proposed stable formula using the previous column, i.e., $\bar{P}_{n_0,m}$.

Numerical results of this study indicate that the relative precision of the two proposed methods compared to Wolfram's Mathematica results is at least 10^{-9} for N_{max} up to 100000. Also, the relative precision of the two methods is better than 10^{-10} for $N_{max} = 60000$.

The comparison of CPU times of the proposed methods with the X-number methods shows that the ratio-based methods speed up the computations by 10-30% for N_{max} from 50 to 60000. The corresponding values are about 20–90% for the Midway method.

In the last part of the study, we optimize the computation of fnALFs over the polar regions. We conclude that polar optimization works best for the ratio-based methods. Nevertheless, the Midway method is the fastest in both optimized and ordinary computations. It has a very simple formulation but evaluation of fnALFs for a single order is not independent. Therefore, parallel computations cannot be used.

APPENDIX A
PROOF OF Eq. (17)

Considering the following varying degree and varying degree/order recurrences for non-normalized ALFs (*Arfken et al., 2013, Eqs 15.88 and 15.89*):

$$\left(2n+1\right)t\,P_{n,m-1}\left(t\right)=\left(n+m-1\right)P_{n-1,m-1}\left(t\right)+\left(n-m+2\right)P_{n+1,m-1}\left(t\right), \tag{A.1}$$

$$\begin{aligned}\left(2n+1\right)u\,P_{n,m}\left(t\right)&=\left(n-m+1\right)\left(n-m+2\right)P_{n+1,m-1}\left(t\right)\\&+\left(n+m\right)\left(n+m-1\right)P_{n+1,m-1}\left(t\right).\end{aligned} \tag{A.2}$$

Substituting the value of $P_{n+1,m-1}$ from Eq. (A.1) into Eq. (A.2) yields, after some simplifications, the new recurrence formula:

$$P_{n,m}\left(t\right)=\frac{t}{u}\left(n-m+1\right)P_{n,m-1}\left(t\right)+\frac{1}{u}\left(n+m-1\right)P_{n-1,m-1}\left(t\right). \tag{A.3}$$

After normalization, we obtain:

$$\begin{aligned}P_{n,m}\left(t\right)=-\frac{t}{u}\sqrt{\frac{n-m+1}{n+m}}P_{n,m-1}\left(t\right)+\frac{1}{u}\sqrt{\frac{\left(2n+1\right)\left(n+m-1\right)}{\left(2n-1\right)\left(n+m\right)}}P_{n-1,m-1}\left(t\right),\\ n,m>0\,.\end{aligned} \tag{A.4}$$

Computer Code Availability: All codes (called in a ALF_UHD software package) are written in Fortran 90 using the standard double-precision environment and available to download in a public domain at *https://github.com/iforough/ALF*. The GCC compiler of gfortran v.11.2 under the command –O3 –march=native is recommended. All the questions regarding the software codes can be directed to the corresponding author.

Acknowledgments: Ismael Foroughi was supported by Mitacs Application No. IT25134. Pavel Novák was supported by project No. 21-13713S of the Czech Science Foundation.

References

Arfken G.B., Weber H.J. and Harris F.E., 2013. Legendre functions. In: *Mathematical Methods for Physicists: A Comprehensive Guide.* Seventh Edition. Academic Press, Waltham, MA, 715–772, DOI: 10.1016/B978-0-12-384654-9.00015-3

Balmino G., Vales N., Bonvalot S. and Briais A., 2012. Spherical harmonic modelling to ultra-high degree of Bouguer and isostatic anomalies. *J. Geodesy*, **86**, 499–520, DOI: 10.1007/s00190-011-0533-4

Colombo O.L., 1981. *Numerical Methods for Harmonic Analysis on the Sphere*. Report No. 310. Department of Geodetic Science and Surveying, The Ohio State University, Columbus, OH

Fukushima T., 2012. Numerical computation of spherical harmonics of arbitrary degree and order by extending exponent of floating point numbers. *J. Geodesy*, **86**, 271–285, DOI: 10.1007/s00190-011-0519-2

Hauser J., 1996. Handling floating-point exceptions in numeric programs. *ACM Trans. Program. Lang. Syst.*, **18**, 139–174, DOI: 10.1145/227699.227701

Heiskanen W.A. and Mortiz H., 1967. *Physical Geodesy*. W.H. Freeman, San Francisco, CA

Holmes S.A. and Featherstone W.E., 2002. A unified approach to the Clenshaw summation and the recursive computation of very high degree and order normalised associated Legendre functions. *J. Geodesy*, **76**, 279–299, DOI: 10.1007/s00190-002-0216-2

Holmes S. and Pavlis N., 2006. NGA: Synth, EGM harmonic synthesis software and test coefficient sets. https://earth-info.nga.mil/GandG/wgs84/gravitymod/new_egm/new_egm.html

IEEE, 2008. *IEEE Standard for Floating-Point Arithmetic*. IEEE Std. 754-2008. IEEE Standards Association, Piscataway, NJ

Ishioka K., 2018. A new recurrence formula for efficient computation of spherical harmonic transform. *J. Meteor. Soc. Japan*, **96**, 241–249, DOI: 10.2151/jmsj.2018-019

Jekeli C., Lee J.K. and Kwon J.H., 2007. On the computation and approximation of ultra-high degree spherical harmonic series. *J. Geodesy*, **81**, 603–615, DOI: 10.1007/s00190-006-0123-z

Libbrecht K.G., 1985. Practical considerations for the generation of large-order spherical harmonics. *Solar Phys.*, **99**, 371–373, DOI: 10.1007/BF00157320

Novikova E. and Dmitrenko A., 2016. Problems and methods of calculating the Legendre functions of arbitrary degree and order. *Geod. Cartogr.*, **65**, 283–312, DOI: 10.1515/geocart-2016-0017

Prézeau G. and Reinecke M., 2010. Algorithm for the evaluation of reduced Wigner matrices. *Astrophys. J. Suppl. Ser.*, **190**, 267–274, DOI: 10.1088/0067-0049/190/2/267

Reinecke M., 2011. Libpsht - algorithms for efficient spherical harmonic transforms. *Astron. Astrophys.*, **526**, Art.No. A108, DOI: 10.1051/0004-6361/201015906

Schaeffer N., 2013. Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations. *Geochem. Geophys. Geosyst.*, **14**, 751–758, DOI: 10.1002/ggge.20071

Smith J.M., Olver F. and Lozier D., 1981. Extended-range arithmetic and normalized Legendre polynomials. *ACM Trans. Math. Softw.*, **7**, 93–105, DOI: 10.1145/355934.355940

Wenzel H.G., 1998. Ultra-high degree geopotential models GPM98A and GPM98B to degree 1,800. In: *Proceedings of Joint Meeting International Gravity Commission and International Geoid Commission*, Budapest, 10–14 March. Report 98:4, Finnish Geodetic Institute, Helsinki, Finland, 71–80

Wieczorek M.A. and Meschede M., 2018. SHTools: Tools for working with spherical harmonics. *Geochem. Geophys. Geosyst.*, **19**, 2574–2592, DOI: 10.1029/2018gc007529

Xing Z., Li S., Tian M., Fan D. and Zhang C., 2019. Numerical experiments on column-wise recurrence formula to compute fully normalized associated Legendre functions of ultra-high degree and order. *J. Geodesy*, **94**, Art.No. 2, DOI:10.1007/s00190-019-01331-0