Computer Physics Communications 124 (2000) 104–122

Computer Physics Communications

www.elsevier.nl/locate/cpc

Evaluation of toroidal harmonics

J. Segura [1], A. Gil [2]

Instituto de Bioingeniería, Universidad Miguel Hernández, Edificio La Galia, 03202 Elche, Alicante, Spain

Received 12 April 1999

Abstract

Three algorithms to evaluate toroidal harmonics, i.e., Legendre functions of integral order and half-odd degree of the first and second kinds for real arguments larger than one, are presented.

The first algorithm (DTORH1) allows the evaluation of the set $\{P^m_{n-1/2}(x), Q^m_{n-1/2}(x)\}$ for fixed (integer and positive) values of m and $n = 0, 1, \ldots, N$. The algorithms DTORH2 and DTORH3 extend the method used in DTORH1 to obtain the set $\{P^m_{n-1/2}(x), Q^m_{n-1/2}(x)\}$ for $m = 0, \ldots, M$ and $n = 0, 1, \ldots, N$.

The output of DTORH2 is equivalent to the result of the application of DTORH1 M times ($m = 0, 1, \ldots, M$). However, due to the organization of the algorithm, DTORH2 is faster than DTORH1 when several orders and degrees are calculated. DTORH2 is better suited than DTORH3 when high orders and degrees are needed. On the other hand DTORH3, though more restrictive on the maximum evaluable degrees N, is even faster than DTORH2.

Our tests of accuracy, show that the three codes achieve a precision of one part in 10^{12}. We discuss the performance of our codes, their speed and their range of validity. The application of the algorithms to solve Laplace's and Poisson's equations in toroidal coordinates is discussed and an explicit numerical example is shown.

Keywords: Toroidal harmonics; Legendre functions; Laplace's equation; Toroidal coordinates

PROGRAM SUMMARY

Title of program: DTORH1, DTORH2, DTORH3

Catalogue identifier: ADKV

Program Summary URL:
http://www.cpc.cs.qub.ac.uk/cpc/summaries/ADKV

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Computers for which the program is designed an others on which it is operable: Hewlett Packard, Model 715/100; SUN Enterprise 3000; PC Pentium MMX 200 MHz

Operating systems under which the program has been tested: UNIX, Linux

Programming language used: Standard FORTRAN 77

No. of bytes in distributed program, including test data, etc.: 27 060 bytes

Distribution format: uuencoded compressed tar file

[1] E-mail: segura@flamenco.ific.uv.es.
[2] E-mail: amparo@condor.ific.uv.es.

0010-4655/00/$ – see front matter
PII: S0010-4655(99)00428-2

Keywords: Toroidal harmonics; Legendre functions; Laplace's equation; toroidal coordinates

Nature of the physical problem

We include three codes to evaluate toroidal harmonics: ◦ DTORH1: This code evaluates toroidal harmonics (TH) of the first and second kinds for a given order m, from the lowest (positive) degree ($n = 0$) to a maximum degree $n = N$ in the same run.

◦ DTORH2: This code evaluates toroidal harmonics of the first and second kinds $P^m_{n-1/2}(x)$ and $Q^m_{n-1/2}(x)$ for orders $m = 0, \ldots, M$ and degrees $n = 0, \ldots, N$. In this code, for each given order m, TH up to the maximum degree reached by the computer for such order m ($N(m)$) can be calculated.

◦ DTORH3: This code evaluates toroidal harmonics of the first and second kinds for orders $m = 0, \ldots, M$ and degrees $n = 0, \ldots, N$. TH up to the smallest maximum degree reachable for the different values of $m = 0, 1, \ldots, M$ can be calculated.

The algorithms find their application in problems with toroidal geometry. In particular, the potential problem for a torus is solved using both Ps and Qs.

Method of solution

The codes are based on the application of recurrence relations both over m and n. The recurrence over n is applied forward for the Ps and backward for the Qs while, over m, the situation is the contrary: the Ps are evaluated backwards and the Qs forward.

The forward and backward recursions (over n or over m) are linked through continued fractions for the ratio of minimal solutions and Wronskian relations.

The recurrences are fed with the two starting values $Q^0_{1/2}$ and $Q^1_{-1/2}$, which are evaluated using Carlson's duplication formula for elliptic integrals.

Restrictions on the complexity of the problem

The maximum degree (order) that can be reached with our method, for a given order (degree) $m(n)$ and for a fixed real positive value of x, is provided by the maximum real number defined in our machine. The user can choose two different relative accuracies (10^{-8} or 10^{-12}) in the interval $1.001 < x < 1000$ for all available values of the orders and degrees. For more details see text: LONG WRITE-UP, Section 3.

Typical running time

Depends on the values of the argument x, the orders (m) and the degrees (n). As an example, to evaluate the set $\{P^m_{n-1/2}(1.1),\ Q^m_{n-1/2}(1.1), m = 0, \ldots, 10,\ n = 0, \ldots, 50\}$ the code DTORH3 spends 2.6 ms in a HP 715/100.

For more details see text: LONG WRITE-UP, Section 5.

LONG WRITE-UP

1. Introduction

In Ref. [1] a code to evaluate Legendre functions of order 0 and integer degrees ($P^m_n, Q^m_n,\ m = 0,\ n = 0, 1, 2, \ldots$) or half-integer degrees ($P^m_{n-1/2}, Q^m_{n-1/2},\ m = 0,\ n = 0, 1, 2, \ldots$) was presented. Later [2], the method of evaluation of Legendre functions of integer degrees (n) was generalized to account for natural orders (m).

In the present work, the series of previous articles concerning the evaluation of Associated Legendre functions for arguments larger than one is completed with the corresponding generalization for half-integer degrees and natural orders. The resulting associated Legendre functions, called toroidal harmonics arise in the solution of Laplace's equation in toroidal coordinates [3] and find their application in boundary value problems with toroidal symmetry (see, for instance, Refs. [4,5]).

The first and only codes (written in ALGOL 60) for the numerical evaluation of toroidal harmonics (TH) were built by Gautschi in 1965 [6], based on three-term recurrence relations and Miller's algorithm. In the present work, the evaluation of TH is revisited. The key point of our algorithms, as in Gautschi's, will be the character of the functions as minimal or dominant solutions of three term recurrence relations. However, the organization of our algorithms improve considerably the range of calculation of the method and its accuracy. Besides our method is also faster, particularly when several orders and degrees for each value of the argument are to be considered. This, as we will see, is the case when solving Dirichlet problems in toroidal coordinates.

A reliable numerical evaluation of toroidal harmonics becomes much harder than in the case of Legendre functions of integer degree (prolate and oblate spheroidal harmonics). The main reason is the absence of simple and closed form expressions to start the recurrences, contrary to prolate and oblate spheroidal harmonics for which P^m_m

and P^m_{m+1} could be used to start the forward recurrence over the degrees. For toroidal harmonics, we will need two different recurrences to evaluate toroidal functions for all the orders and degrees reachable (within overflow limits). The recurrence over degrees (n) is applied forward for the Ps and backward for the Qs while over orders (m) the situation is the contrary; the Ps are evaluated backwards while the Qs are obtained forward. The forward and backward recurrences (over n or m) will be linked through continued fractions (evaluated using Lentz–Thompson algorithm) for the ratio of minimal solutions and Wronskian relations. The recurrences are fed with the two starting values $Q^0_{1/2}$ and $Q^1_{-1/2}$, which are evaluated using Carlson's duplication formula for elliptic integrals.

The present codes, together with those in Refs. [1,2], fill a gap in the libraries of numerical software in which only Legendre functions on the cut (arguments $-1 < x < 1$) have been taken into account.

We will present some applications of toroidal harmonics. We will discuss how the simple method employed in [5] to evaluate toroidal harmonics is surpassed by our codes in accuracy, speed and in the evaluable ranges of the parameters. In consequence, the use of our programs would improve the performance of the programs presented in [5] to solve Poisson's equation for a point charge near a dielectric toroidal surface. In addition, we will show another (simpler) application of toroidal harmonics. We will discuss how to evaluate the potential due to a toroidal conductor at a potential $V(\phi)$, where ϕ is the azimuthal angle, and give explicit numerical results for the case $V(\phi) = \cos m\phi$.

2. Toroidal harmonics

Toroidal harmonics $\{P^m_{n-1/2}(x),\ Q^m_{n-1/2}(x);\ n, m = 0, 1, \ldots \text{ and } x > 1\}$ are solutions of the differential equation [7, Eq. (8.1.1)]; [3, Eq. (7.12.8)]

$$\left(1-x^2\right)u'' - 2xu' + \left[\left(n^2 - \frac{1}{4}\right) - \frac{m^2}{1-x^2}\right]u = 0. \tag{1}$$

We will consider the standard notation for the Ps and the Qs [7,3].

Associated Legendre functions (and, in particular, toroidal functions) satisfy, among others, the following recurrence relations [7, Eqs. (8.5.1) and (8.5.3)]:

$$(\nu - m + 1)P^m_{\nu+1}(x) - (2\nu+1)xP^m_\nu(x) + (\nu+m)P^m_{\nu-1}(x) = 0, \tag{2}$$

$$P^{m+2}_\nu(x) + \frac{2(m+1)x}{(x^2-1)^{1/2}}P^{m+1}_\nu(x) - (\nu-m)(\nu+m+1)P^m_\nu(x) = 0, \tag{3}$$

with ν a real number and m a positive integer. The Q^m_νs satisfy the same recurrence relations.

In terms of hypergeometric series, $Q^m_\nu(x)$ can be expressed as (Eq. (8.1.3) of Ref. [7])

$$Q^m_\nu(x) = 2^{-\nu-1}\frac{(-1)^m\sqrt{\pi}\,\Gamma(\nu+m+1)}{\Gamma(\nu+\frac{3}{2})}x^{-\nu-m-1}\left(x^2-1\right)^{m/2} \times F\left(\frac{\nu+m+2}{2}, \frac{\nu+m+1}{2}; \nu+\frac{3}{2}; \frac{1}{x^2}\right), \quad x > 1. \tag{4}$$

The Ps can also be expressed in terms of hypergeometric functions [7, Eq. (8.1.2)].

The Wronskian relating Ps and Qs reads [7, Eq. (8.1.8)]; [3, Eq. (7.12.18)]

$$W\left\{P^m_\nu(x), Q^m_\nu(x)\right\} = \frac{\Gamma(\nu+m+1)}{\Gamma(\nu-m+1)}\frac{(-1)^m}{1-x^2}, \tag{5}$$

which can be used to write the following two relations

$$P^m_\nu(x)Q^m_{\nu-1}(x) - P^m_{\nu-1}(x)Q^m_\nu(x) = \frac{\Gamma(\nu+m)}{\Gamma(\nu-m+1)}(-1)^m \tag{6}$$

and

$$P_\nu^m(x)Q_\nu^{m+1}(x) - P_\nu^{m+1}(x)Q_\nu^m(x) = \frac{\Gamma(\nu+m+1)}{\Gamma(\nu-m+1)}\frac{(-1)^{m+1}}{\sqrt{x^2-1}}. \tag{7}$$

2.1. Recurrence relations and stability

The application of Perron's theorem [8,9] for $x > 1$ tells us that the recurrence (2) admits a minimal solution. Q_ν^m is the minimal solution of recurrence (2), while P_ν^m is a dominant solution [9]. Only the forward recursion should be applied for the dominant solution and only backward recurrence for the minimal one.

Re-writing the recurrence relation for the Qs as

$$\frac{Q_\nu^m(x)}{Q_{\nu-1}^m(x)} = \frac{1}{(2\nu+1)/(\nu+m)\cdot x - (\nu-m+1)/(\nu+m)\cdot Q_{\nu+1}^m(x)/Q_\nu^m} \tag{8}$$

and iterating this expression, we obtain a continued fraction representation for the ratio $Q_\nu^m(x)/Q_{\nu-1}^m(x)$ which, according to Pincherle's theorem [10,11], is convergent for $x > 1$.

Repeating the same kind of arguments for the recurrence relation (3), it is easy to check that the minimal solution for such recurrence is $P_\nu^m(x)$, for which Eq. (3) has to be applied backwards. The Qs are, in this case, dominant and must be evaluated forward. Analogously as before, one can write Eq. (3) as

$$\frac{P_\nu^m}{P_\nu^{m-1}} = \frac{(\nu-m+1)(\nu+m)}{2mx/\sqrt{x^2-1} + P_\nu^{m+1}/P_\nu^m} \tag{9}$$

and iterating this expression we obtain a continued fraction representation for the ratio $P_\nu^m(x)/P_\nu^{m-1}(x)$.

2.2. Evaluation of $P_{\pm1/2}^m$

The starting point of the algorithm DTORH1 will be the evaluation of the initial values for the forward recurrence over $\nu \equiv n - 1/2$, n integer: $P_{-1/2}^m$ and $P_{1/2}^m$. Then, we will consider a similar scheme to that of Refs. [1,2]: starting from $P_{\pm1/2}^m$ we will apply the forward recurrence relation (2) up to a given maximum degree; then we will use a CF (Eq. (8)) and a Wronskian relation (Eq. (6)) to get the maximum degrees Qs; and then backward recurrence over the degree will be used for the Qs until the values $Q_{\pm1/2}^m$ are reached.

The additional difficulty of the present code with respect to those in Refs. [1,2] is the evaluation of the starting values $P_{\pm1/2}^m$ for which two different approaches are considered depending on the range of ν and x.

The sub-algorithm to evaluate $P_{\pm1/2}^m$ first calculates $Q_{-1/2}^0$ and $Q_{-1/2}^1$ through their relation with elliptic integrals [7]

$$\begin{aligned} Q_{-1/2}^0 &= \sqrt{\frac{2}{x+1}}K\left(\sqrt{\frac{2}{x+1}}\right), \\ Q_{-1/2}^1 &= \frac{-1}{\sqrt{2(x-1)}}E\left(\sqrt{\frac{2}{x+1}}\right). \end{aligned} \tag{10}$$

We use SLATEC routines, based on Carlson's duplication theorem [12] to evaluate the elliptic integrals involved. Then we apply the forward recurrence relation (3) for the Qs until we obtain $Q_{-1/2}^m$ and $Q_{-1/2}^{m-1}$.

- Evaluation of $P_{-1/2}^m$

  The next step is the evaluation of $P_{-1/2}^m$ for which, depending on the value of x and the ratio x/m, two approaches are considered.

(i) In the first approach, we evaluate the ratio $P^m_{-1/2}/P^{m-1}_{-1/2}$ by means of the continued fraction obtained from the iteration of Eq. (9) with $\nu = -1/2$

$$H_P(x) = \frac{P^m_{-1/2}}{P^{m-1}_{-1/2}} = \frac{-(m-1/2)^2}{2mx/\sqrt{x^2-1}-} \frac{(m+1/2)^2}{2(m+1)x/\sqrt{x^2-1}-} \frac{(m+3/2)^2}{2(m+2)x/\sqrt{x^2-1}-} \cdots \tag{11}$$

which, combined with the Wronskian relation (7) leads to

$$P^m_{-1/2}(x) = -\frac{\Gamma(m-1/2)^2}{\pi} \frac{1}{\sqrt{x^2-1}} \frac{1}{Q^m_{-1/2}/H_P(x) - Q^{m-1}_{-1/2}}. \tag{12}$$

The continued fraction tends to converge more slowly as x increases. This fact can be proven from Pincherle's theorem [10] and is illustrated in Table 1 (Section 5.4). To avoid slow convergence we will use a series expansion (13) in powers of $1/x^2$ instead of the CF whenever it converges faster, namely for $x > 5$. However, for low x/m such series shows large roundoff errors, as checked by comparing the series in double precision arithmetic with a version in quadruple precision. Then, we will keep using the CF for $x > 5$ when x/m is small ($x \leqslant 0.12$ for a precision of 10^{-8} and $x \leqslant 0.22$ for 10^{-12}).

(ii) For $x > 5$ and large enough x/m ($x/m > 0.12$ for a precision of 10^{-8} and $x/m > 0.22$ for 10^{-12}) it is better to use a series expansion for $P^m_{-1/2}$ in powers of $1/x^2$. The series reads [13, p. 211]

$$P^m_{-1/2}(x) = \frac{2(-1)^m}{\pi^{3/2}} \left(1 - \frac{1}{x^2}\right)^{m/2} \sum_{r=0}^{\infty} \left\{ \log(2x) - \Psi(m+2r+1/2) + \Psi(r+1) \right\} \times \frac{\Gamma(1/2+m+2r)}{\Gamma^2(r+1)} \frac{1}{(2x)^{2r+1/2}}, \tag{13}$$

where ψ is the logarithmic derivative of the Γ function.

The limiting values $x/m = 0.12$ for a precision of 10^{-8} (respectively, $x/m = 0.22$ for 10^{-12}) are obtained by comparing the values obtained for $P^m_{-1/2}$ from procedures (i) and (ii). Both precision limits are determined by the loss in accuracy of the series, as checked by comparing both types of evaluation (i) and (ii) as used in our programs (double precision) with their corresponding version in quadruple precision. Indeed, the series, in powers of x^{-2}, converges slowly for small x while for large m, drastic cancellations between negative terms (first terms) and positive terms take place, thereby increasing the rounding errors. We have tested the validity of such compromise by comparing procedures (i) and (ii) in the reachable range (limited by overflows) for the argument x and the corresponding orders m (using MODE 2, see Section 3): $x \leqslant 115$ for $x/m = 0.12$ (accuracy 10^{-8}) and $x \leqslant 225$ for $x/m = 0.22$ (accuracy 10^{-12}).

- Evaluation of $P^m_{1/2}$

Once $P^m_{-1/2}$ and $Q^m_{-1/2}$ have been evaluated we combine the CF for $Q^m_{1/2}/Q^m_{-1/2}$

$$H_Q \equiv \frac{Q^m_{1/2}}{Q^m_{-1/2}} = \frac{1}{2x/(m+1/2)-} \frac{(3/2-m)/(1/2+m)}{4x/(3/2+m)-} \frac{(5/2-m)/(3/2+m)}{6x/(5/2+m)-} \cdots \tag{14}$$

with the Wronskian relation (6) to get $P^m_{1/2}$:

$$P^m_{1/2} = P^m_{-1/2} H_Q + \frac{1}{Q^m_{-1/2}} \frac{\Gamma(3/2+m)}{\Gamma(3/2-m)} \frac{(-1)^m}{(m+1/2)}. \tag{15}$$

This last step finishes the (non trivial) finding of $P^m_{\pm 1/2}$.

In Ref. [5] a simpler approach is considered to evaluate $P^m_{\pm 1/2}$. A single series expansion, based on a hypergeometric function of argument $\xi = 1 - 1/x^2$ [14, p. 129, Eq. (24)], is considered. Such hypergeometric series converge very slowly for large x, since $\xi \to 1$ as $x \to \infty$. Then, as we will see, for $x \sim 10$ or larger values, such series suffer from slow convergence and loss of accuracy.

3. Algorithms for toroidal harmonics

3.1. *Algorithm* 1: $P^m_{n-1/2}$, $Q^m_{n-1/2}$, *fixed* m *and* $n = 0, \ldots, N$

In this section, we will use the abbreviated notation: $p^m_n \equiv P^m_{n-1/2}(x)$ and $q^m_n \equiv Q^m_{n-1/2}(x)$, with $n = 0, 1, \ldots, N$. This is closer to the actual notation used in our codes.

Our code evaluates q^m_n and p^m_n from the lowest degrees up to a maximum N (for a fixed m) defined by the user but limited by the overflow number.

The algorithm is organized as follows:

(A) Evaluation of the starting values p^m_0 and p^m_1.
This first part of the algorithm evaluates p^m_0 and p^m_1 as explained in Section 2.2.
- First, we calculate q^0_0 and q^1_0 by means of Eq. (10). The elliptic integrals are evaluated by using SLATEC routines (based on Carlson's duplication theorem).
- Then, we use the forward recurrence relation over the orders for q^k_0 (applying Eq. (3) for the qs) and we evaluate it forward up to $k = m$. Notice that q^m_0 has then been calculated.
- We evaluate p^m_0 following two different approaches depending on the range of x/m.
  (i) If x/m is small (lower than 0.12 for a precision of 10^{-8} and lower than 0.22 for a precision of 10^{-12}) we evaluate the continued fraction (11) $H_P = p^m_0/p^{m-1}_0$ using the Lentz–Thompson algorithm [15,11]. The Wronskian relation (7), combined with H_P, is used to get p^m_0.
  (ii) If x/m is larger than in case (i) we evaluate p^m_0 from a series expansion (Eq. (13)).
- Once q^m_0 and p^m_0 have been obtained, p^m_1 is calculated combining these values with the CF $H_{Q_0} = q^m_1/q^m_0$ (Lentz–Thompson method) and the Wronskian relation (6) to get p^m_1.

(B) Evaluation of the set $\{p^m_n, q^m_n,\ n = 0, 1, \ldots, N\}$.
From the values p^m_0, p^m_1 evaluated in (A) the code calculates and stores the set $\{p^m_n, q^m_n,\ n = 0, 1, \ldots, N\}$ using a procedure analogous to those described in Refs. [1,2]:
- We apply the forward recurrence relation (2) for p^m_n starting from p^m_0 and p^m_1 up to a given degree N.
- We evaluate the continued fraction (6) H_Q using the Lentz–Thompson algorithm [15].
- Considering the calculated values of p^m_N, p^m_{N-1}, the ratio H_Q and the relation (6) we obtain q^m_{N-1} and q^m_N.
- The evaluation of the set $\{q^m_N,\ N = N, N-1, \ldots, 1, 0\}$ is performed considering the calculated values of q^m_{N-1} and q^m_N and using the backward recurrence relation over degrees (2) for the qs.

We have introduced two additional modes (default is MODE = 0) to extend the range of calculation.
- MODE = 0 evaluates the set

$$\left\{P^m_{n-1/2}, Q^m_{n-1/2}\right\}.$$

- If MODE = 1 is selected, the code evaluates the set

$$\left\{P^m_{n-1/2}/\Gamma(m+1/2), Q^m_{n-1/2}/\Gamma(m+1/2)\right\}$$

and uses this definition both in part (A) and (B). The normalization for MODE = 1 has the advantage of extending the range of evaluable orders m (part (A)). Both the recurrence and the continued fraction are unaltered but the Ps and Qs are renormalized to smaller quantities, reducing overflows. Also, the application of the Wronskian for the new set of functions avoids overflow problems due to the factorials. In this way, the range of calculation over m is extended (for examples, see the Section 5.2).

For MODE = 0 and MODE = 1, the range of calculation is restricted by the following condition: if the ratio x/m is smaller than 0.12 for a precision of 10^{-8} or smaller than 0.22 for a precision of 10^{-12}, then x must be smaller than 20. This means that the evaluation of the CF in the forbidden regions in Fig. 1 is avoided so as not to slow down the code since the CF converges more slowly as x becomes larger.

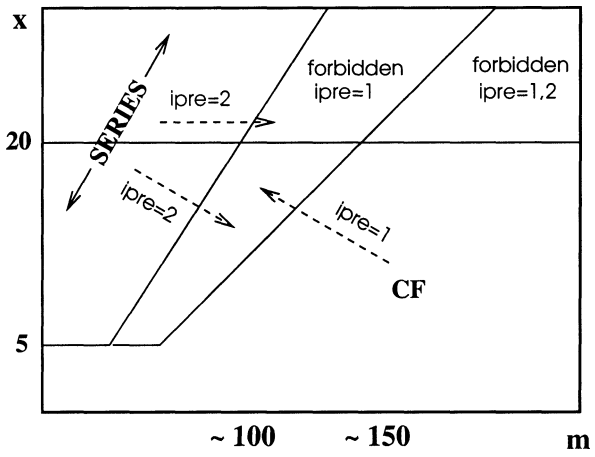

Fig. 1. The (x, m) plane divided according to the method of evaluation of $P^m_{1/2}$. IPRE $= 1$ corresponds to a best accuracy of 10^{-12}, while IPRE $= 2$ corresponds to 10^{-8} (see Section 4).

If MODE $= 2$ is chosen, the code evaluates the same set of toroidal harmonics as for MODE $= 1$ but the previous restriction on the range of x is removed. The CF is also used in the forbidden regions in Fig. 1; however, the code may eventually become slow.

3.2. *Algorithm* 2: $P^m_{n-1/2}$, $Q^m_{n-1/2}$, $m = 0, \ldots, M$; $n = 0, \ldots, N$

One could apply Algorithm 1 M times to evaluate the set $\{P^m_{n-1/2}, Q^m_{n-1/2},\ m = 0, \ldots, M,\ n = 0, \ldots, N\}$. However, for this purpose a better option is evaluating $\{P^m_{-1/2}, P^m_{+1/2},\ m = 0, \ldots, M\}$ as a first step in the algorithm, and later using such values to start the recurrences over n for each value of m. Algorithm 2 considers this strategy which improves the speed of the method when several degrees and orders are wanted.

As before, we are using the notation $p^m_n \equiv P^m_{n-1/2}(x)$ and $q^m_n \equiv Q^m_{n-1/2}(x)$, with $m = 0, 1, \ldots, M$ and $n = 0, 1, \ldots, N$.

The algorithm is organized as follows.

(A) Evaluation of the set $\{p^m_0, p^m_1,\ m = 0, 1, \ldots, M\}$
- We calculate q^0_0 and q^1_0 using their relation with elliptic integrals, as explained for Algorithm 1.
- Then we apply the forward recurrence relation for q^m_0s (Eq. (3)) up to $m = M$. When the input value M is such that the recurrence over m suffers from overflow problems, the program fixes M to be the maximal evaluable order.
- We evaluate p^M_0 depending on the range of x/M, following one of the two different approaches (i) and (ii) mentioned in Algorithm 1.
- We evaluate p^{M-1}_0. If approach (i) was used in the previous step, we calculate p^{M-1}_0 using the CF $H_P = p^M_0/p^{M-1}_0$ (Eq. (9)) and the calculated value of p^M_0. If approach (ii) was employed, the calculation of p^{M-1}_0 is made using the Wronskian relation (7) and the calculated values of q^M_0, q^{M-1}_0 and p^M_0 (p^M_0 from series).
- We calculate p^M_1 (and p^{M-1}_1) as we did in Algorithm 1.
- We apply, twice, the backward recurrence relation over m (Eq. (3));
  (i) starting from p^M_0, p^{M-1}_0 to generate p^m_0, $m = 0, \ldots, M$; and
  (ii) starting from p^M_1, p^{M-1}_1 to generate p^m_0, $m = 0, \ldots, M$.

(B) Evaluation of the set $\{p_n^m, q_n^m;\ m = 0, \ldots, M,\ n = 0, \ldots, N\}$
Once the set $\{p_0^m, p_1^m;\ m = 0, 1, \ldots, M\}$ has been calculated, the evaluation of $\{p_n^m, q_n^m\}$, $n = 0, \ldots, N$, *for each* $m = 0, 1, \ldots, M$, is made as explained in Algorithm 1.
If overflows take place in the forward recursion over n, the algorithm fixes N for such m to be the maximal evaluable one.

In summary, the Algorithm 2 evaluates q_n^m and p_n^m from the lowest orders and degrees to the maximum chosen order M and degree N. For each m, the recurrence over n could be limited by overflows. If they take place for a given order m, the code avoids them by setting $N = N(m)$, where $N(m)$ is the highest evaluable degree for such order m. Then, this algorithm can give the highest reachable degrees for each m.

3.3. *Algorithm* 3: $P_{n-1/2}^m$, $Q_{n-1/2}^m$, $m = 0, \ldots, M,\ n = 0, \ldots, N$

Algorithm 3 performs a similar task as Algorithm 2 but the organization is different, resulting in an improvement in the execution time. The improvement comes from avoiding the evaluation of the continued fraction $Q_{N-1/2}^m/Q_{N-3/2}^m \equiv q_N^m/q_{N-1}^m$ for each $m = 0, 1, \ldots, M$. The disadvantage comes from the fact that the range of evaluable values of n is restricted since in this case the maximum degree evaluable will be $N = \min\{N(m),\ m = 0, 1, \ldots, M\}$, where the $N(m)$s are the highest evaluable degrees for each order m.

We will use same notation as for Algorithm 2.

The algorithm is organized as follows:

(A) Evaluation of the set $\{P_{\pm 1/2}^m,\ m = 0, 1, \ldots, M\} \equiv \{p_0^m, p_1^m,\ m = 0, 1, \ldots, M\}$
This evaluation is performed as in Algorithm 2.

(B) Evaluation of the set $\{p_n^m,\ m = 0, \ldots, M,\ n = 0, \ldots, N\}$
For each m and starting from p_0^m and p_1^m we apply the forward recurrence over n (Eq. (2)). For each m we test overflows in the forward recurrence over n. If an overflow is going to happen we set N to be the maximum degree for which the evaluation is safe. Then, after all forward recurrences (for each m) have been performed, we will have $N = \min\{N(m),\ m = 0, 1, \ldots, M\}$.

(C) Evaluation of the set $\{q_n^m,\ m = 0, \ldots, M,\ n = 0, \ldots, N\}$
  (i) From p_N^0 and p_{N-1}^0 and using the CF for q_N^0/q_{N-1}^0 and the Wronskian relation (6) we evaluate q_N^0 and q_{N-1}^0.
  (ii) Similarly, we evaluate q_N^1 and q_{N-1}^1 from p_N^1 and p_{N-1}^1.
  (iii) Starting from q_N^0 and q_N^1 we evaluate the set $\{q_N^m,\ m = 0, \ldots, M\}$ using the forward recurrence relation over m.
  (iv) We also evaluate the set $\{q_{N-1}^m,\ m = 0, \ldots, M\}$, starting from q_{N-1}^0 and q_{N-1}^1, using forward recursion over m.
  (v) *For each* $m = 0, 1, \ldots, M$ we start from the values q_N^m and q_{N-1}^m and apply the backward recurrence over the degree n to obtain the set $\{q_n^m,\ m = 0, \ldots, N,\ m = 0, \ldots, M\}$.

4. Subprogram specification

In this section we introduce the subroutines DTORH1, DTORH2 and DTORH3 which are used to compute the toroidal harmonics that we have analyzed.

It should be noted that the three subroutines have to be linked with ROUT which includes the routines (referenced in DTORH1, DTORH2 and DTORH3) for the evaluation of the continued fractions, the series expansion and the elliptic integrals.

The subroutine signatures for DTORH1, DTORH2, DTORH3 are described below.

SUBROUTINE DTORH1(Z, M, NMAX, PL, QL, NEWN)

INPUT:

Z: argument of the toroidal harmonics.

M: order of the toroidal harmonics.

NMAX: maximum degree of the toroidal harmonics: we get toroidal harmonics of all degrees below MIN(NMAX, NEWN). NEWN is defined below.

OUTPUT:

PL(N): toroidal harmonics p_N^M, N = 0, 1, ..., MIN(NMAX, NEWN). We store these values in an array.

QL(N): toroidal harmonics q_N^M, N = 0, 1, ..., MIN(NMAX, NEWN). We store these values in an array.

NEWN: maximum degree of toroidal harmonics calculated when PL(NMAX) is larger than 1/TINY. TINY is defined below.

PARAMETERS:

MODE: enables the enlargement of the range of degrees that can be evaluated. When MODE = 0 the code evaluates plain toroidal harmonics. When MODE = 1, the program evaluates PL/GAMMA(M + 1/2), QL/GAMMA(M + 1/2). If MODE = 2 is selected, the code performs as for MODE = 1, but the restriction Z < 20 for the evaluation of the continued fraction is not considered.

EPS: controls the accuracy of the continued fractions. (EPS = 10**(−14) in this version of the code).

IPRE: parameter which limits the precision attainable by the code, depending on the range of application of series for $P_{-1/2}^M \equiv$ PL(0). The precision of the code is P = MAX(10^{-8}, EPS) for IPRE = 1 and P = MAX(10^{-12}, EPS) for IPRE = 2. IPRE = 1 in this version of the code.

TINY: small parameter close to the underflow limit. In this version of the code,

TINY = 1.D–290.

SUBROUTINE DTORH2(Z, MDIM, NDIM, MMAX, NMAX, PL, QL, NEWM, NEWN)

INPUT:

Z: argument of the toroidal harmonics.

MDIM, NDIM: dimension of the arrays PL(MDIM, NDIM), QL(MDIM, NDIM). MDIM must be greater than MMAX. NDIM must be greater than NMAX. When several calls to DTORH2 are needed NDIM (MDIM) must be larger than the largest chosen value of NMAX (MMAX).

MMAX: maximum order of the toroidal harmonics. We get toroidal harmonics of all orders below MIN(MMAX, NEWM). NEWM is defined below.

NMAX: maximum degree of the toroidal harmonics. We get toroidal harmonics of all degrees below MIN(NMAX, NEWN(M)). NEWN(M) is defined below.

OUTPUT:

PL(M, N): toroidal harmonics p_N^M. We store these values in an array.

QL(M, N): toroidal harmonics q_N^M. We store these values in an array.

NEWM: maximum order of toroidal harmonics calculated when QL(MMAX, 0) is larger than 1/TINY. TINY is defined below.

NEWN(M): maximum degree of toroidal harmonics calculated for a given order M when PL(M, NMAX) is larger than 1/TINY. TINY is defined below.

PARAMETERS:

MODE: enables the enlargement of the range of degrees that can be evaluated. When MODE = 0 the code evaluates plain toroidal harmonics. When MODE = 1, the program evaluates PL/GAMMA(M + 1/2), QL/GAMMA $(M + 1/2)$. If MODE = 2 is selected, the code performs as for MODE = 1, but the restriction Z < 20 for the evaluation of the continued fraction is not considered.

EPS: controls the accuracy of the continued fractions. (EPS = 10**(−14) in this version of the code.)

IPRE: parameter which limits the precision attainable by the code, depending on the range of application of series for $P_{-1/2}^{\mathrm{MMAX}} \equiv \mathrm{PL(MMAX, 0)}$. The precision of the code is $\mathrm{P} = \mathrm{MAX}(10^{-8}, \mathrm{EPS})$ for IPRE = 1 and $\mathrm{P} = \mathrm{MAX}(10^{-12}, \mathrm{EPS})$ for IPRE = 2. IPRE = 1 in this version of the code.

SUBROUTINE DTORH3(Z, MDIM, NDIM, MMAX, NMAX, PL, QL, NEWM, NEWN)

INPUT:

Z: argument of the toroidal harmonics.

MDIM, NDIM: dimension of the arrays PL(MDIM, NDIM), QL(MDIM, NDIM). MDIM must be greater than MMAX. NDIM must be greater than NMAX. When several calls to DTORH2 are needed NDIM (MDIM) must be larger than the largest chosen value of NMAX (MMAX).

MMAX: maximum order of the toroidal harmonics. We get toroidal harmonics of all orders below MIN(MMAX, NEWM). NEWM is defined below.

NMAX: maximum degree of the toroidal harmonics. We get toroidal harmonics of all degrees below MIN(NMAX, NEWN). NEWN is defined below.

OUTPUT:

PL(M, N): toroidal harmonics p_N^M. We keep these values in an array.

QL(M, N): toroidal harmonics q_N^M. We keep these values in an array.

NEWM: maximum order of toroidal harmonics calculated when QL(MMAX, 0) is larger than 1/TINY. TINY is defined below.

NEWN: maximum degree of toroidal harmonics calculated when PL(M, NMAX) is larger than 1/TINY for some M = 0, 1, . . . , NEWM. TINY is defined below.

PARAMETERS:

MODE: enables the enlargement of the range of degrees that can be evaluated. When MODE = 0 the code evaluates plain toroidal harmonics. When MODE = 1, the program evaluates PL/GAMMA(M + 1/2), QL/GAMMA(M + 1/2) and if MODE = 2 is selected, the code performs as for MODE = 1, but the restriction Z < 20 for the evaluation of the continued fraction is not considered.

EPS: controls the accuracy of the continued fractions. (EPS = 10**(−14) in this version of the code).

IPRE: parameter which limits the precision attainable by the code, depending on the range of application of series for $P_{-1/2}^{\mathrm{MMAX}} \equiv \mathrm{PL(MMAX, 0)}$. The precision of the code is $\mathrm{P} = \mathrm{MAX}(10^{-8}, \mathrm{EPS})$ for IPRE = 1 and $\mathrm{P} = \mathrm{MAX}(10^{-12}, \mathrm{EPS})$ for IPRE = 2. IPRE = 1 in this version of the code.

TINY: small parameter close to the underflow limit. In this version of the code,

TINY = 1.D–290.

5. Performance of the codes

As commented before, all the codes have been prepared to work with two different precisions: 10^{-8} and 10^{-12}. In the following, we will refer to a precision of 10^{-12}. The precision parameters considered are: IPRE = 1, EPS = 10^{-14} (see Section 4). For this selection of parameters, and as we are next discussing, the accuracy of the algorithms is better than 10^{-12} in all cases; similarly, for IPRE = 2 and EPS < 10^{-8} the accuracy is always better than 10^{-8}.

We will also comment on the reachable ranges for n, m and x.

5.1. Accuracy of the codes

Several internal and external checks have been performed in order to test the accuracy of the codes. By *internal checks* we mean all those tests that use relations already used by the algorithm (but in a different step of the procedure). *External checks* will those which are based on relations not used in any part of the codes.

The main limitation in the precision of our algorithms is the roundoff error in the series used to evaluate $P^m_{-1/2}$. Such series, as explained before, are used to improve the speed of the algorithm when the convergence of the CF for $P^m_{-1/2}/P^{m-1}_{-1/2}$ becomes slow. All the tests considered show that when IPRE $= 1$ (precision 10^{-12} in the series) the precision is better than 10^{-12} taking EPS $< 10^{-12}$, while an accuracy of at least 10^{-8} is obtained when IPRE $= 2$ (precision 10^{-8} in the series) and EPS $< 10^{-8}$.

Internal checks:

We have compared the continued fractions from Eqs. (8) and (9) with the corresponding ratios of functions given by the algorithm. Also, we have checked the deviation of our calculated functions from Eqs. (6) and (7). In all cases, the deviation was of the order of $\sim 10^{-13}$ (or lower) for $1.001 < x < 1000$ regardless of the order m and degree n. This means that for the evaluable ranges of the parameters the accumulation of roundoff errors is not a problem in achieving a precision better than 10^{-12}.

External checks:

As external checks we have considered addition formulae and series expansions for the Qs. We have also compared the values of $P^0_{n-1/2}$ with the values obtained from a previous algorithm [1] for Legendre functions. This last check is also external since no recurrence on m is considered in such earlier algorithm for Legendre functions.

- We have tested the following two addition formulae [6]:

$$Q^m_{-1/2}(x) + 2\sum_{n=1}^{\infty} Q^m_{n-1/2}(x) = (-1)^m \sqrt{\frac{\pi}{2}}\,\Gamma(m+1/2)(x-1)^{-1/2}\left(\frac{x+1}{x-1}\right)^{m/2} \tag{16}$$

and

$$\frac{P_{n-1/2}(x)}{\Gamma(n+1/2)} + 2\sum_{m=1}^{\infty} \frac{P^m_{n-1/2}(x)}{\Gamma(m+n+1/2)} = \frac{(x+(x^2-1)^{1/2})^{n-1/2}}{\Gamma(n+1/2)}\,. \tag{17}$$

  We have tested such sums in the intervals $1.01 \leqslant x \leqslant 10$, $0 \leqslant n \leqslant 100$, $0 \leqslant m \leqslant 100$ concluding that our results agree with a precision better than 10^{-13}.

  For larger values of x one needs many terms in Eq. (17). Therefore, it is better to perform other tests since the convergence of the series is slow (as discussed in [8]), large roundoff errors can accumulate and overflows in the Ps can take place. Similarly, Eq. (16) is not convenient to test the algorithm for small x. For values of x outside the range $1.01 < x < 10$ there is a gradual loss of accuracy when evaluating the expressions (16) and (17). This we have checked by comparing with quadruple precision arithmetics. Both expressions are used in [6] as renormalizing conditions. Contrary, the loss of accuracy in the continued fractions used by our algorithms is much milder and overflows do not take place. In addition, the CF is replaced by series when it converges slowly, improving the speed of the algorithm. These facts explain in part why our method can be applied for larger ranges of the arguments compared to Gautschi's.
- As another test, we have compared the values of the Qs obtained using our codes with those calculated using the series given in Eq. (4). We have extensively checked the range $0 \leqslant m, n \leqslant 100$, $1.01 < x < 1000$ and found our results in agreement with the series with a deviation of $\sim 10^{-13}$ in the worst cases.
- Finally, we have compared our results, evaluated in MODE $= 2$, for the toroidal harmonics $P^0_{n-1/2}$, $Q^0_{n-1/2}$ with the values obtained from a previous code for Legendre functions (see Ref. [1]) letting the parameters m and n vary in the full reachable range for $1.001 \leqslant x \leqslant 1000$. The discrepancy was, in the worst cases, of the order of 10^{-12}.

Table 1
CPU times (in 1/1000 s) corresponding to DTORH1 for IPRE = 1 (precision = 10^{-12}). For each x and M, NEWN(0) represents the maximum degree that can be evaluated in MODE = 0 and NEWN(1) is the maximum degree evaluated in MODE = 1. The last row shows the CPU time spent in evaluating up to the N ≡ NMAX = 10 degree; the use of the CF or the series for $P^m_{-1/2}$ is specified

x	M	NEWN(0)	NEWN(1)	$10^3 \times$ CPU-t. N = NEWN(0)	$10^3 \times$ CPU-t. N = NEWN(1)	$10^3 \times$ CPU-t. N = 10
1.1	5	1416	1425	4.61″	4.64″	0.32″ (cf)
	50	760	1051	2.70″	4.05″	0.40″ (cf)
10.	5	213	215	0.97″	0.87″	0.25″ (s)
	50	143	187	0.95″	1.08″	0.53″ (cf)
100.	5	121	122	0.58″	0.58″	0.23″ (s)
	50	87	111	0.62″	0.64″	0.35″ (s)
1000.	5	85	85	0.48″	0.46″	0.23″ (s)
	50	64	81	0.53″	0.54″	0.32″ (s)

Table 2
CPU times (in 1/100 s) spent in evaluating PL(M, N), QL(M, N), M = 0, 1, . . . , 50 and N = 0, 1, . . . , 50 with the codes DTORH2 and DTORH3

	DTORH2	DTORH3
x	$10^2 \times$ CPU-t. M ⩽ 50; N ⩽ 50	$10^2 \times$ CPU-t. M ⩽ 50; N ⩽ 50
1.1	1.78″	1.41″
10.	1.54″	1.43″
100.	1.53″	1.41″
1000.	1.58″	1.41″

5.2. Available ranges of the parameters and CPU time

Table 1 illustrates both the speed and the ranges of evaluable degrees n depending on the argument x and the order m. As shown, the available range for n, due to overflow/underflow limitations, is smaller as the values of m and x are larger.

Concerning the CPU-times, it is worth noting (see last row in Table 1) that the CF (11) is indeed slower as x increases while the series (13) becomes faster for large x.

In Table 2 the CPU times spent by DTORH2 and DTORH3 to evaluate $P^m_{n-1/2}$, $Q^m_{n-1/2}$, $m = 0, 1, \ldots, 50$ and $n = 0, 1, \ldots, 50$, are compared. As there are no overflow problems for such limited ranges of orders and degrees (when $1.1 \leqslant x \leqslant 1000$), both CPU times correspond to the evaluation of the same number of functions. The differences are related to the different organization of the algorithms. DTORH2 evaluates in this case, 49 more CFs (Eq. (8)) than DTORH3. Since the convergence of the CF (8) is slower as x is smaller, the difference is bigger for small x.

In Table 3 we show the ranges of evaluable orders m in MODE = 0 (first row) and MODE = 1 (second row) for DTORH2 and DTORH3. Note that for the larger evaluable orders, the maximal evaluable degree will be quite small.

As seen from Table 2, in cases where the demanded range for n and m is not too large, DTORH3 becomes more convenient than DTORH2 since the execution time is shorter. However, when restrictions on the degree n are found, DTORH2 could become more convenient (being slower) since the available ranges can be larger

Table 3
Available ranges in m for DTORH2 and DTORH3 in MODE = 0 (first row) and MODE = 1 (second row). The numbers with (∗) are values given by the limit $x/m > 0.22$ (when $x > 20$), considered to avoid slow convergence of the CF. The rest of values are due to the overflow limitation (for an overflow number 10^{290})

x	NEWM(0)	NEWM(1)
1.1	125	441
10.	160	6705
100.	163	454 (∗)
1000.	163	4545 (∗)

Table 4
Available ranges of the degree (NEWN(m)) depending on the order m for the code DTORH2. Also, the order of magnitude of PL(m, NMAX(m)) and PL(m, 0) are shown. INPUT variables: $x = 1.1$, MMAX = 125 (= NEWM), NMAX = 500

m	NEWN(m)	PL(m, NMAX(m))	PL(m, 0)
50	500	$\sim 10^{227}$	10^{29}
100	305	$\sim 10^{287}$	10^{89}
125	209	$\sim 10^{288}$	10^{124}

Table 5
Available range of the degree NEWN for the code DTORH3. Also, the order of magnitude of PL(m, NMAX) and PL(m, 0) are shown. INPUT variables: $x = 1.1$, MMAX = 125 (= NEWM), NMAX = 500

m	NEWN	PL(m, NMAX)	PL(m, 0)
50	209	$\sim 10^{148}$	10^{29}
100	209	$\sim 10^{245}$	10^{89}
125	209	$\sim 10^{288}$	10^{124}

than in DTORH3. As an example, Tables 4 and 5 show the available range considering as input parameters M ≡ MMAX = 125, N ≡ NMAX = 500, $x = 1.1$. Also, the order of magnitude of PL(m, NEWN) and PL(m, 0) are given for comparison; recall that, in this version of the code TINY = 10^{-290} and then the overflow number to test the forward recurrences is 10^{290}.

6. Test run

At the end of the paper, we show a test program for the calculation of toroidal harmonics together with its corresponding output.

The program first calls the subroutine DTORH1; the functions shown, for different values of the argument x, are $Q^m_{n-1/2} \equiv \mathrm{QL}(n)$ and $P^m_{n-1/2} \equiv \mathrm{PL}(n)$ for $m = 120$ and the highest order (n) that can be evaluated together with an intermediate order ($n = 10$); also $P^0_{-1/2} \equiv \mathrm{PL}(0)$ is shown. Parameters: MODE = 1, IPRE = 1, EPS = 1.D–14.

The second and third calls are to DTORH2 and DTORH3 and the functions shown in both cases are $Q^m_{n-1/2} \equiv \mathrm{QL}(m,n)$ and $P^m_{n-1/2} \equiv \mathrm{PL}(m,n)$ for $m = 10$ and $m = 50$ and the corresponding maximal degrees (N ≡ NEWN); also the values of $P^0_{-1/2} \equiv \mathrm{PL}(0,0)$ are shown. Parameters: MODE = 0, IPRE = 1, EPS = 1.D–14.

7. Physical examples

7.1. Laplace's equation with Dirichlet boundary conditions on a torus

Toroidal harmonics appear in the solution of Laplace's equation in toroidal coordinates and are suitable to solve the potential problem in regions bounded by tori. As an example, we will obtain the electrostatic potential due to a toroidal surface (see Fig. 2) at a potential of the type $V = \cos m\phi$, $m = 0, 1, \ldots$, being ϕ the axial toroidal coordinate defined below.

The toroidal coordinates (α, β, ϕ), with $0 \leqslant \alpha < \infty$, $-\pi < \beta \leqslant \pi$ and $-\pi < \phi \leqslant \pi$, are related to cylindrical coordinates through the expression

$$z + ir = ic \coth\left(\frac{\alpha + i\beta}{2}\right), \tag{18}$$

where c is a scale factor. We are considering the same notation as in [3].

The coordinates are chosen so that $\alpha = \alpha_0$ represents the toroidal surface. α_0 and c are then related to the geometric parameters a, l (see Fig. 2) of the torus as follows

$$c = \sqrt{l^2 - a^2}, \quad \cosh\alpha_0 = \frac{l}{a}. \tag{19}$$

The general solution of Laplace's equation $\vec{\nabla}^2\Psi = 0$ in toroidal coordinates can be written in the form [3,4]

$$\Psi(\alpha,\beta,\phi) = f(\alpha,\beta) \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \left(A_{mn} Q^m_{n-1/2}(\cosh\alpha) + B_{mn} P^m_n(\cosh\alpha)\right) \times \cos n(\beta - \beta_{mn}) \cos m(\phi - \phi_{mn}) \tag{20}$$

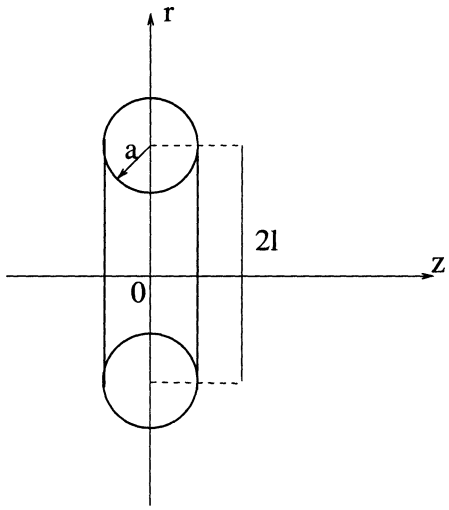

Fig. 2. Toroidal conductor of geometric parameters a and l.

Table 6
CPU times (in 1/100 s) and values of the electrostatic potential Ψ for several values of a and l; $M = 10$, $\phi = 0.5$, $\alpha = \alpha_0/2$ and $\beta = 0$. n is the maximum order required for an accuracy of 10^{-8}

a	l	α_0	n	Ψ	$100 \times$ CPU time
100.	100.1	4.4717×10^{-2}	207	1.3648503×10^{-4}	$0.65''$
100.	101.	0.14130	66	1.3795758×10^{-4}	$0.45''$
100.	110.	0.44357	22	1.5308774×10^{-4}	$0.37''$
100.	150.	0.96242	10	2.2886202×10^{-4}	$0.35''$

with

$$f(\alpha, \beta) \equiv \sqrt{\cosh\alpha - \cos\beta}. \tag{21}$$

The coefficients $(A_{mn}, B_{mn}, \beta_{mn}, \phi_{mn})$ have to be determined from the boundary condition.

In the particular case of solving the Dirichlet problem for a domain bounded by a torus of equation $\alpha = \alpha_0$, with boundary condition $\Psi(\alpha_0, \beta, \phi) = V(\beta, \phi)$ the regularity of the solution implies that $A_{mn} = 0$ when considering the exterior problem ($\alpha < \alpha_0$) while $B_{mn} = 0$ for the interior problem ($\alpha > \alpha_0$).

For the particular case

$$\Psi(\alpha_0, \beta, \phi) \equiv V(\phi) = \sum_{m=0}^{\infty} a_m \cos\bigl(m(\phi - \phi_m)\bigr) \tag{22}$$

the solution for the exterior problem can be written as

$$\Psi_{\text{ext}}(\alpha, \beta, \phi) = \frac{\sqrt{2}}{\pi} f(\alpha, \beta) \sum_{m=0}^{\infty} a_m \cos m(\phi - \phi_m) \times \left[\frac{Q_{-1/2}(\cosh\alpha_0)}{P^m_{-1/2}(\cosh\alpha_0)} P^m_{-1/2}(\cosh\alpha) + 2\sum_{n=1}^{\infty} \frac{Q_{n-1/2}(\cosh\alpha_0)}{P^m_{n-1/2}(\cosh\alpha_0)} P^m_{n-1/2}(\cosh\alpha)\cos n\beta \right] \tag{23}$$

as can be checked by comparing the Fourier series expansions (in β and ϕ) for both sides of Eq. (23) and taking into account the integral representation [3, Eq. (7.10.10)]

$$Q^0_{n-1/2}(\cosh\alpha_o) = \int_0^{\pi} \frac{\cos n\beta}{\sqrt{2\cosh\alpha_0 - 2\cos\beta}} d\beta. \tag{24}$$

The corresponding expression for the interior problem ($\alpha < \alpha_0 < \infty$) can be obtained from (23) by replacing $P^m_{n-1/2}$ by $Q^m_{n-1/2}$.

As an illustration, we show the numerical results for the electrostatic potential originated by a toroidal conductor at a potential $V(\phi) = \cos m\phi$, with $m = 10$.

In Table 6 we show the number of evaluated terms required for a precision of 10^{-8} for several values of a and l. We also show the values of the electrostatic field Ψ and the CPU time spent (in 1/100 s). In the calculation we fix $\alpha = \alpha_0/2$, $\beta = 0$, $M = 10$ and $\phi = 0.5$.

7.2. The Poisson's equation with Dirichlet boundary conditions on a torus: comment on Hoyle's et al. code

A further illustration of the applicability of toroidal harmonics can be found in Ref. [4], where the solutions of Poisson's equation with Dirichlet boundary conditions on a toroidal dielectric surface are considered in order to model the passage of an ion through a cellular membrane channel pore.

The solution is found by superposing the potential from Laplace's equation, Eq. (20), and the potential due to a point charge using toroidal coordinates. The main difficulty consists in imposing the boundary conditions and obtaining the integration constants. For such a purpose, as well as to compute the final solution, the evaluation of toroidal harmonics for different orders and degrees is needed. Recently, Fortran 90 codes to solve numerically this problem have been published which include the evaluation of toroidal harmonics with varying orders, degrees and arguments (module an_functions from Ref. [5]).

The method employed in [5] consists in applying series expansions, in terms of hypergeometric functions, to evaluate the starting values for the forward and backward recurrence relation over degrees n (the recurrence over m is not used); the starting values for the forward recurrence are $P^m_{\pm 1/2}(x)$ (for each m), evaluated from an hypergeometric series expansion in powers of $\xi \equiv 1-1/x^2$ [14, p. 129, Eq. (24)] while for the backward recurrence relation the starting values for each m are $Q^m_{N+1/2}(x)$, $Q^m_{N-1/2}(x)$ from an hypergeometric series expansion in powers of $\chi \equiv \frac{1}{2}(\sqrt{x^2-1}-x)/\sqrt{x^2-1}$ [14, p. 137, Eq. (44)].

Our method of evaluation of toroidal harmonics improves considerably the speed and the range of calculation. Notice that the evaluation (for each m) of the series expansion can be very slow, especially when $|\xi|$ or $|\chi|$ are close to one (x large or close to 1, respectively). The approach from series expansions fails for $x < 3/2\sqrt{2} \simeq 1.06$ for the Qs while 500 terms of the series are not enough for the Ps for a precision of 10^{-8} when $x > 10$. Regarding timing, choosing, for instance, $0 \leqslant m \leqslant 10$, $0 \leqslant n \leqslant 10$ and demanding a precision of 10^{-12} we have checked that for $x = 1.1, 10, 100$ our approach is faster by a factor 2, 60 and 200, respectively. Of course, taking larger m makes the comparison more favorable for our code. Also, as x approaches 1 or becomes larger and more terms of the series are needed, large roundoff errors in the series become unavoidable.

We conclude that the use of our codes would considerably improve the code presented in [5] for the solution of Poisson's equation, enlarging the available ranges of the parameters and improving both the speed and the accuracy of the codes.

Acknowledgements

The authors acknowledge the use of subroutines from the SLATEC Public Domain Library. gopher: archives.math.utk.edu/software/multi-platform/SLATEC.

References

[1] A. Gil, J. Segura, Comput. Phys. Comm. 105 (1997) 273.
[2] A. Gil, J. Segura, Comput. Phys. Comm. 108 (1998) 267.
[3] N.N. Lebedev, Special Functions and Their Applications (Dover Publications, Inc., 1972).
[4] S. Kuyucak, M. Hoyles, S.-H. Chung, Biophys. J. 74 (1998) 22.
[5] M. Hoyles, S. Kuyucak, S.-H. Chung, Comput. Phys. Comm. 115 (1998) 45.
[6] W. Gautschi, Comm. ACM 8 (8) (1965) 488.
[7] M. Abramowitz, I. Stegun, Handbook of Mathematical Functions (Dover Publications, Inc., 1972).
[8] W. Gautschi, SIAM Review 9 (1967) 24.
[9] N.M. Temme, Special Functions: An Introduction to the Classical Functions of Mathematical Physics (John Wiley & Sons, 1996) Chap. 13.
[10] C. Brezinski (Ed.), Continued Fractions and Padé Approximants (North-Holland, 1990) p. 142.
[11] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran (Cambridge University Press, 1992) Chap. 6.
[12] B.C. Carlson, Numer. Math. 33 (1–16) (1979);
B.C. Carlson, E.M. Notis, ACM Trans. Math. Soft. 7 (1981) 398–403.
[13] E.W. Hobson, The Theory of Spherical and Ellipsoidal Harmonics (Cambridge University Press, 1965) Chap. 5.
[14] A. Erdélyi, Higher transcendental functions, Bateman Manuscript Project (McGraw-Hill, 1953).
[15] I.J. Thompson, A.R. Barnett, J. Comput. Phys. 64 (1984) 490.

TEST PROGRAM

```
      PROGRAM TOROID
      INTEGER MDIM,NDIM,NMAX,MMAX,NEWN,NEWM,M,NM,L1,L2,L3
      INTEGER NEWNM(0:51)
      DOUBLE PRECISION Z
      DOUBLE PRECISION PL(0:301),QL(0:301)
      DOUBLE PRECISION PL1(0:51,0:301),QL1(0:51,0:301)
      MDIM=51
      NDIM=301
      NMAX=300
      M=120
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C SET MODE=1 AND IPRE=1 IN ROUTINE DTORH1  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      OPEN(UNIT=10,FILE='TEST.OUT',STATUS='UNKNOWN')
      WRITE(10,*)' '
      WRITE(10,*)'TOROIDAL HARMONICS, M=120, NMAX=300, MODE 1'
      WRITE(10,*)' '
      WRITE(10,31)'Z','N','PL/GAMMA(M+1/2)','QL/GAMMA(M+1/2)',
     *   'PL/GAMMA(M+1/2)'
      WRITE(10,*)' '
      Z=1.5D0
      DO 20 L1=1,6
         CALL DTORH1(Z,M,NMAX,PL,QL,NEWN)
         WRITE(10,801) Z,NEWN,PL(NEWN),QL(NEWN),PL(0)
         NM=10
         WRITE(10,801) Z,NM,PL(NM),QL(NM),PL(0)
         Z=Z+1.6D0
20    CONTINUE
      WRITE(10,*)' '
      MMAX=50
      NMAX=300
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C SET MODE=0 AND IPRE=1 IN ROUTINE DTORH2  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      WRITE(10,*)'TOROIDAL HARMONICS, MMAX=50, NMAX=300, MODE 0'
      WRITE(10,*)' '
      WRITE(10,32)'Z','M','N', 'PL(M,N)', 'QL(M,N)', 'PL(0,0)'
      WRITE(10,*)' '
      Z=1.5D0
      DO 30 L2=1,6
         CALL DTORH2(Z,MDIM,NDIM,MMAX,NMAX,PL1,QL1,NEWM,NEWNM)
         NM=NEWNM(NEWM)
         WRITE(10,802) Z,NEWM,NM,PL1(NEWM,NM),QL1(NEWM,NM),
     *                 PL1(0,0)
         M=10
         NM=NEWNM(10)
         WRITE(10,802) Z,M,NM,PL1(M,NM),QL1(M,NM),PL1(0,0)
         Z=Z+1.6D0
30    CONTINUE
      WRITE(10,*)' '
```

```
      NMAX=300
      MMAX=50
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C SET MODE=0 AND IPRE=1 IN ROUTINE DTORH3  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      WRITE(10,*)'TOROIDAL HARMONICS, MMAX=50, NMAX=300, MODE 0'
      WRITE(10,*)' '
      WRITE(10,32)'Z','M','N', 'PL(M,N)', 'QL(M,N)', 'PL(0,0)'
      WRITE(10,*)' '
      Z=1.5D0
      DO 40 L3=1,6
         CALL DTORH3(Z,MDIM,NDIM,MMAX,NMAX,PL1,QL1,NEWM,NEWN)
         WRITE(10,803) Z,NEWM,NEWN,PL1(NEWM,NEWN),QL1(NEWM,NEWN),
     *                 PL1(0,0)
         M=10
         WRITE(10,803) Z,M,NEWN,PL1(M,NEWN),QL1(M,NEWN),PL1(0,0)
         Z=Z+1.6D0
40    CONTINUE
      WRITE(10,*)' '
   31 FORMAT (4X,A1,3X,A1,2X,A16,2(1X,A16))
   32 FORMAT (4X,A1,4X,A1,4X,A1,3(3X,A11))
  801 FORMAT (F6.1,1X,I4,1X,D16.9,1X,D16.9,1X,D16.9)
  802 FORMAT (F6.1,1X,I4,1X,I3,1X,D16.9,1X,D16.9,1X,D16.9)
  803 FORMAT (F6.1,1X,I4,1X,I4,1X,D16.9,1X,D16.9,1X,D15.9)
      END
```

TEST OUTPUT

```
TOROIDAL HARMONICS, M=120, NMAX=300, MODE 1

   Z   N      PL/GAMMA(M+1/2)  QL/GAMMA(M+1/2)  PL/GAMMA(M+1/2)

  1.5  300   .313054244+208   .463394336D-14   .334489883D-43
  1.5   10   .946614433D-43   .321309267D+41   .334489883D-43
  3.1  269   .240764689+288   .199296699-106   .106659413D-18
  3.1   10   .577935145D-18   .513451631D+16   .106659413D-18
  4.7  229   .781105908+288   .222279718-125   .158775542D-12
  4.7   10   .161860668D-11   .176266088D+10   .158775542D-12
  6.3  208   .197802978+288   .380130850-136   .131117077D-09
  6.3   10   .246445139D-08   .110043643D+07   .131117077D-09
  7.9  195   .200428866+288   .560475686-144   .671953487D-08
  7.9   10   .227849591D-06   .112236029D+05   .671953487D-08
  9.5  186   .443495288+288   .351518383-150   .897212349D-07
  9.5   10   .536667084D-05   .447110644D+03   .897212349D-07

TOROIDAL HARMONICS, MMAX=50, NMAX=300, MODE 0

   Z    M    N     PL(M,N)          QL(M,N)          PL(0,0)

  1.5   50  300   .111845003+246   .425981468D+00   .945006331D+00
  1.5   10  300   .253594778+149   .204117888-101   .945006331D+00
  3.1   50  223   .812883944+287   .270949179D-55   .829041884D+00
  3.1   10  300   .102410130+258   .192662145-210   .829041884D+00
  4.7   50  185   .194110231+288   .454421163D-64   .756654960D+00
  4.7   10  273   .426545201+287   .492158343-241   .756654960D+00
  6.3   50  166   .399253598+288   .262308690D-69   .704973539D+00
  6.3   10  242   .239698123+288   .653915175-243   .704973539D+00
  7.9   50  154   .558972520+288   .681225412D-73   .665271208D+00
  7.9   10  222   .114199070+288   .211327227-243   .665271208D+00
  9.5   50  145   .113750977+288   .561965770D-75   .633319116D+00
  9.5   10  208   .572664837+287   .101303188-243   .633319116D+00

TOROIDAL HARMONICS, MMAX=50, NMAX=300, MODE 0

   Z    M    N     PL(M,N)          QL(M,N)         PL(0,0)

  1.5   50  300   .111845003+246   .425981468D+00  .945006331D+00
  1.5   10  300   .253594778+149   .204117888-101  .945006331D+00
  3.1   50  223   .812883944+287   .270949179D-55  .829041884D+00
  3.1   10  223   .448132878+196   .156638276-151  .829041884D+00
  4.7   50  185   .194110231+288   .454421163D-64  .756654960D+00
  4.7   10  185   .615946382+200   .208640557-157  .756654960D+00
  6.3   50  166   .399253598+288   .262308690D-69  .704973539D+00
  6.3   10  166   .230793870+203   .523279474-161  .704973539D+00
  7.9   50  154   .558972520+288   .681225412D-73  .665271208D+00
  7.9   10  154   .130018700+205   .176858911-163  .665271208D+00
  9.5   50  145   .113750977+288   .561965770D-75  .633319116D+00
  9.5   10  145   .545226887+205   .111222130-164  .633319116D+00
```