Conical: An extended module for computing a numerically satisfactory pair of solutions of the differential equation for conical functions☆

T.M. Dunster [a], A. Gil [b,a,*], J. Segura [c,a], N.M. Temme [d,1]

[a] Department of Mathematics and Statistics, San Diego State University, 5500 Campanile Drive San Diego, CA, USA
[b] Depto. de Matemática Aplicada y Ciencias de la Comput, Universidad de Cantabria, 39005-Santander, Spain
[c] Depto. de Matemáticas, Estadística y Comput, Universidad de Cantabria, 39005-Santander, Spain
[d] IAA, 1825 BD 18, Alkmaar, The Netherlands

ARTICLE INFO

ABSTRACT

Conical functions appear in a large number of applications in physics and engineering. In this paper we describe an extension of our module **Conical** (Gil et al., 2012) for the computation of conical functions. Specifically, the module includes now a routine for computing the function $R^m_{-\frac{1}{2}+i\tau}(x)$, a real-valued numerically satisfactory companion of the function $P^m_{-\frac{1}{2}+i\tau}(x)$ for $x > 1$. In this way, a natural basis for solving Dirichlet problems bounded by conical domains is provided.

The module also improves the performance of our previous algorithm for the conical function $P^m_{-\frac{1}{2}+i\tau}(x)$ and it includes now the computation of the first order derivative of the function. This is also considered for the function $R^m_{-\frac{1}{2}+i\tau}(x)$ in the extended algorithm.

Program summary

Program Title: Module **Conical**
Program Files doi: http://dx.doi.org/10.17632/rpw5d8gdkg.1
Licensing provisions: CC by 4.0
Programming language: Fortran 90
External routines/libraries: The module **Conical** uses a Fortran 90 version of the routine **dkia** (developed by the authors) for computing the modified Bessel functions $K_{ia}(x)$ and its derivative. This routine is available at http://toms.calgo.org.
Nature of problem: These functions are the natural function basis for solving, for example, the Laplace's problem in spherical coordinates for two intersecting cones or for regions bounded by two intersecting spheres, or by one or two confocal hyperboloids of revolution when using toroidal coordinates. The conical function $P^m_{-\frac{1}{2}+i\tau}(x)$ is also used in the Mehler–Fock integral transform for problems in potential and heat theory,
Solution method: The algorithm uses different methods of computation depending on the function under consideration ($P^m_{-\frac{1}{2}+i\tau}(x)$ or $R^m_{-\frac{1}{2}+i\tau}(x)$) and the values of x, τ and m: numerical quadrature, asymptotic expansions in terms of elementary functions, asymptotic expansions in terms of Bessel functions, asymptotic expansions for tau large and backward/forward recursion of three-term recurrence relations.
Restrictions: In order to avoid underflow/overflow problems in standard IEEE double precision arithmetic, the admissible parameter ranges for computing the conical function $P^m_{-\frac{1}{2}+i\tau}(x)$ in the routine **conicp** are:

$$-1 < x < 1, \ 0 < \tau <= 100, \ 0 \le m \le 40$$
$$1 < x \le 100, 0 < \tau \le 100, 0 \le m \le 100.$$

When using the routines **conicr** and **conicpr**, the admissible parameter ranges for computing the functions $P^m_{-\frac{1}{2}+i\tau}(x)$ and $R^m_{-\frac{1}{2}+i\tau}(x)$ are

$$1 < x \le 100, \ 0 < \tau \le 100, \ 0 \le m \le 100.$$

1. Introduction

Conical or Mehler functions are involved in a large number of applications in different areas of physics. In particular, these functions appear when solving the Laplace equation in spherical coordinates for two intersecting cones [1] or for regions bounded by two intersecting spheres, or by one or two confocal hyperboloids of revolution when using toroidal coordinates.

An extended version of our module **Conical** [2] for the computation of conical functions is presented in this paper. The new module includes now a routine for computing the function $R^m_{-\frac{1}{2}+i\tau}(x)$, a real-valued numerically satisfactory companion of the function $P^m_{-\frac{1}{2}+i\tau}(x)$ for $x > 1$. The module also improves our previous algorithm for the conical function $P^m_{-\frac{1}{2}+i\tau}(x)$ by considering now more coefficients in some of the asymptotic expansions used for computing the function in the region $x > 1$. The computation of the first order derivatives of $P^m_{-\frac{1}{2}+i\tau}(x)$ and $R^m_{-\frac{1}{2}+i\tau}(x)$ is also included in the new module.

2. Theoretical background

Conical functions are solutions of the second order differential equation

$$(1-x^2)\frac{d^2w}{dx^2} - 2x\frac{dw}{dx} - \left(\tau^2 + \frac{1}{4} + \frac{\mu^2}{1-x^2}\right)w = 0. \tag{1}$$

We will restrict to integer positive values of the parameter μ ($\mu = m \in \mathbb{Z}^+$).

When $-1 < x < 1$, a real-valued satisfactory pair of solutions of (1) is $P^m_{-\frac{1}{2}+i\tau}(x)$ and $P^{-m}_{-\frac{1}{2}+i\tau}(x)$. Both functions can be computed using our algorithm for $P^m_{-\frac{1}{2}+i\tau}(x)$ implemented in **conicp**;

for computing $P^{-m}_{-\frac{1}{2}+i\tau}(x)$ the following relation can be used

$$P^{-m}_{-\frac{1}{2}+i\tau}(x) = \frac{\pi}{\cosh(\pi\tau)\left|\Gamma(m+\frac{1}{2}+i\tau)\right|^2}P^m_{-\frac{1}{2}+i\tau}(x). \tag{2}$$

When $x > 1$, a real-valued satisfactory pair of solutions of (1) is $P^m_{-\frac{1}{2}+i\tau}(x)$ and $R^m_{-\frac{1}{2}+i\tau}(x) \equiv \Re\left\{e^{-i\pi m}Q^m_{-\frac{1}{2}+i\tau}(x)\right\}$ (the function $Q^m_{-\frac{1}{2}+i\tau}(x)$ is complex-valued).

The Wronskian relation for $P^m_{-\frac{1}{2}+i\tau}(x)$ and $R^m_{-\frac{1}{2}+i\tau}(x)$, useful for testing, is given by

$$W\left\{P^m_{-\frac{1}{2}+i\tau}(x), R^m_{-\frac{1}{2}+i\tau}(x)\right\}$$
$$= \frac{\pi(e^{-\tau\pi} + \sinh(\tau\pi))}{\left|\Gamma\left(-m+\frac{1}{2}+i\tau\right)\right|^2(\cosh^2(\tau\pi))(1-x^2)}. \tag{3}$$

The algorithm for computing the conical function $P^m_{-\frac{1}{2}+i\tau}(x)$ was described in [2]. In the new module **Conical** we also compute the first order derivatives for $P^m_{-\frac{1}{2}+i\tau}(x)$ and $R^m_{-\frac{1}{2}+i\tau}(x)$: the first order derivative of $P^m_{-\frac{1}{2}+i\tau}(x)$ can be obtained using the relation

$$\frac{d}{dx}P^m_{-\frac{1}{2}+i\tau}(x) = -\frac{1}{\sqrt{x^2-1}}P^{m+1}_{-\frac{1}{2}+i\tau}(x) + \frac{mx}{x^2-1}P^m_{-\frac{1}{2}+i\tau}(x). \tag{4}$$

The first order derivative of $R^m_{-\frac{1}{2}+i\tau}(x)$ satisfies the same relation (4) with $R^{m+1}_{-\frac{1}{2}+i\tau}(x)$ and $R^m_{-\frac{1}{2}+i\tau}(x)$.

A preliminary algorithm for computing the function $R^m_{-\frac{1}{2}+i\tau}(x)$ was presented in [3] although the final algorithm in finite precision arithmetic implemented in the routine **conicr** presents some differences with respect to the first algorithm. We have also changed the notation of the function with respect to the one used in that reference; we are using now $R^m_{-\frac{1}{2}+i\tau}(x)$ instead of $\widetilde{Q}^m_{-\frac{1}{2}+i\tau}(x)$ for simplicity.

Next we summarize the methods for computing the conical function $R^m_{-\frac{1}{2}+i\tau}(x)$.

2.1. Computation of $R^m_{-\frac{1}{2}+i\tau}(x)$ for values of x close to 1

2.1.1. Small or moderate values of τ

To compute $R^0_{-\frac{1}{2}+i\tau}(x)$ and $R^1_{-\frac{1}{2}+i\tau}(x)$ we will use the expansions

$$R^0_{-\frac{1}{2}+i\tau}(x) = \Re\left(\sum_{k=0}^{\infty}\frac{\left(\frac{1}{2}-i\tau\right)_k\left(\frac{1}{2}+i\tau\right)_k}{(k!)^2}z^k\left(\psi(k+1)\right.\right.$$
$$\left.\left. - \psi(\tfrac{1}{2}+i\tau) - \ln w\right)\right),$$

$$R^1_{-\frac{1}{2}+i\tau}(x) \tag{5}$$
$$= \Re\left(\sqrt{x^2-1}\sum_{k=0}^{\infty}\frac{\left(\frac{1}{2}-i\tau\right)_k\left(\frac{1}{2}+i\tau\right)_k}{(k!)^2}\tfrac{1}{2}z^{k-1}\left(\tfrac{1}{2}\left(-1+w^2\right)\right.\right.$$
$$\left.\left. + k\left(\psi(k+1) - \psi(\tfrac{1}{2}+i\tau) - \ln w\right)\right)\right),$$

where z and w are given by

$$z = \tfrac{1}{2}(1-x), \ w = \sqrt{\frac{x-1}{x+1}}, \tag{6}$$

and $\psi(\alpha) = \Gamma'(\alpha)/\Gamma(\alpha)$.

The only complex quantity in (5) is the function $\psi(\frac{1}{2} + i\tau)$. For computing this function we use an algorithm which computes separately the real and imaginary parts of $\psi(\frac{1}{2} + i\tau)$ avoiding, in this way, the use of complex arithmetics when computing (5). The algorithm is based on the use of the asymptotic expansion

$$\psi(\alpha) \sim \ln\alpha - \frac{1}{2\alpha} - \frac{1}{12\alpha^2} + \frac{1}{120\alpha^4} - \frac{1}{252\alpha^6} + \cdots, \tag{7}$$

valid for $\alpha \to \infty$ in $|\mathrm{ph}\,\alpha| < \pi$. We use this expansion if $|\alpha| \ge 12$ with 8 terms of the series (or less), and we use the recurrence relation $\psi(\alpha) = \psi(\alpha+1) - 1/\alpha$ for smaller values of $|\alpha|$.

When $m \geq 2$ we will use the recursion relation

$$R_{-\frac{1}{2}+i\tau}^{m+1}(x) - \frac{2mx}{\sqrt{x^2-1}} R_{-\frac{1}{2}+i\tau}^m(x)$$
$$+ \left((m-\tfrac{1}{2})^2 + \tau^2 \right) R_{-\frac{1}{2}+i\tau}^{m-1}(x) = 0 \tag{8}$$

in the direction of increasing m.

2.1.2. Large values of τ

A representation in terms of Kummer $U-$functions is used in this case:

$$R_{-\frac{1}{2}+i\tau}^{\mu}(x) \sim \sqrt{\pi/2} \, \alpha^{\mu+\frac{1}{2}} (x^2-1)^{-\frac{1}{4}}$$
$$\times \sum_{k=0}^{\infty} f_k \left(\Re \Phi_k \cos \phi + \Im \Phi_k \sin \phi \right), \tag{9}$$

where $\phi = \tau \log(x + \sqrt{x^2-1})$, $\alpha = \ln \left(\frac{z+1}{z} \right)$ and z is given by

$$z = \frac{1}{2\sqrt{x^2-1} \left(x + \sqrt{x^2-1} \right)}. \tag{10}$$

The functions Φ_k are given in terms of Kummer $U-$functions as follows

$$\Phi_k = \left(\tfrac{1}{2} - \mu \right)_k \omega^{2\mu-k} U \left(\tfrac{1}{2} + \mu, 1 + 2\mu - k, \alpha\omega \right), \quad \omega = i\tau. \tag{11}$$

The functions can be also written in terms of the Hankel functions $H_{\mu}^{(2)}(z)$. This representation makes simple separating the real and imaginary parts of Φ_k by using

$$H_{\mu}^{(2)}(z) = J_{\mu}(z) - iY_{\mu}(z). \tag{12}$$

For the computation of the Bessel functions $J_{\mu}(z)$, $Y_{\mu}(z)$ we use an algorithm which combines the use of series expansions, Debye's asymptotic expansions, asymptotic expansions for large z, Airy-type asymptotic expansions and three-term recurrence relations. This algorithm is implemented in the module **BesselJY** and it is also included in the software package.

The functions Φ_0, Φ_1 are given by

$$\Phi_0 = -\tfrac{1}{2} i \sqrt{\pi} (\tau/\alpha)^{\mu} e^{\frac{1}{2}i\alpha\tau} H_{\mu}^{(2)} \left(\tfrac{1}{2}\alpha\tau \right),$$
$$\Phi_1 = \tfrac{1}{4}\alpha \sqrt{\pi} (\tau/\alpha)^{\mu} e^{\frac{1}{2}i\alpha\tau} \left(iH_{\mu}^{(2)} \left(\tfrac{1}{2}\alpha\tau \right) + H_{\mu-1}^{(2)} \left(\tfrac{1}{2}\alpha\tau \right) \right). \tag{13}$$

For computing Φ_n for $n = 2, \ldots$ we can use a recurrence relation for the Kummer $U-$functions which gives

$$\omega \Phi_{n+1} = (n - 2\mu - \alpha\omega)\Phi_n + \alpha(n - \tfrac{1}{2} - \mu)\Phi_{n-1}. \tag{14}$$

From this, the following recurrence relations for both the real and imaginary parts of Φ_{n+1} can be obtained:

$$\Re \Phi_{n+1} = \frac{n-2\mu}{\tau} \Im \Phi_n - \alpha \Re \Phi_n + \frac{\alpha}{\tau}(n - \tfrac{1}{2} - \mu)\Im \Phi_{n-1},$$
$$\Im \Phi_{n+1} = -\frac{n-2\mu}{\tau} \Re \Phi_n - \alpha \Im \Phi_n - \frac{\alpha}{\tau}(n - \tfrac{1}{2} - \mu)\Re \Phi_{n-1}. \tag{15}$$

The first coefficients f_k in (9) are given by

$$f_0 = 1,$$
$$f_1 = \frac{b}{2d} (2dz + d - 2z),$$
$$f_2 = \frac{b}{24d^2} \left(12z^2 + 12bz^2 + d^2 - 12d^2z - 12d^2z^2 - 24bdz^2 \right.$$
$$\left. + 12bd^2z + 12bd^2z^2 + 3bd^2 - 12bdz \right), \tag{16}$$

where $b = -\mu - \tfrac{1}{2}$ and $d = z\alpha$.

2.2. Computation of $R_{-\frac{1}{2}+i\tau}^m(x)$ for moderate or large values of x

In this case we use the expansion

$$R_{-\frac{1}{2}+i\tau}^{\mu}(x) = \Re \left(\sqrt{\pi/2} \, (x^2-1)^{-\frac{1}{4}} G(\mu,\tau) e^{-i\phi} \right.$$
$$\left. \times \sum_{k=0}^{\infty} \left(\tfrac{1}{2} + \mu \right)_k \left(\tfrac{1}{2} - \mu \right)_k \frac{u_k(\tau) + iv_k(\tau)}{w_k(\tau)} \frac{(-z)^k}{k!} \right), \tag{17}$$

where z is given in Eq. (10), $\phi = \tau \log \left(x + \sqrt{x^2-1} \right)$,

$$G(\mu,\tau) = \frac{\Gamma \left(\tfrac{1}{2} + \mu + i\tau \right)}{\Gamma (1 + i\tau)}, \tag{18}$$

and

$$\frac{u_k(\tau) + iv_k(\tau)}{w_k(\tau)} = \frac{1}{(1 + i\tau)_k}, \quad k = 0, 1, 2, \ldots. \tag{19}$$

We can compute $u_k(\tau)$, $v_k(\tau)$ and $w_k(\tau)$ from the recurrence relations

$$u_{k+1}(\tau) = (k+1)u_k(\tau) + \tau v_k(\tau),$$
$$v_{k+1}(\tau) = (k+1)v_k(\tau) - \tau u_k(\tau),$$
$$w_{k+1}(\tau) = \left((k+1)^2 + \tau^2 \right) w_k(\tau), \tag{20}$$

with $u_0(\tau) = 1$, $v_0(\tau) = 0$, $w_0(\tau) = 1$.

The real part of (17) can be obtained by writing

$$G(\mu,\tau) = H(\mu,\tau)e^{i\rho(\mu,\tau)}, \quad u_k(\tau) + iv_k(\tau) = r_k(\tau)e^{i\sigma_k(\tau)}, \tag{21}$$

which gives

$$R_{-\frac{1}{2}+i\tau}^{\mu}(x) = \sqrt{\pi/2} \, H(\mu,\tau)(x^2-1)^{-\frac{1}{4}}$$
$$\times \sum_{k=0}^{\infty} \left(\tfrac{1}{2} + \mu \right)_k \left(\tfrac{1}{2} - \mu \right)_k \frac{r_k(\tau)}{w_k(\tau)} \frac{(-z)^k}{k!} \cos(\psi_k), \tag{22}$$

where

$$\psi_k = \tau \log \left(x + \sqrt{x^2-1} \right) - \rho(\mu,\tau) - \sigma_k(\tau). \tag{23}$$

The computation of the ratio of two gamma functions in (18) is made using an algorithm for computing the gamma function for complex values of the argument. The algorithm adapts for complex arguments the scheme used for real values described in [4].

3. Overview of the software structure

The Fortran 90 package includes the main module **Conical**, which includes the routines **conicp**, **conicr** and **conicpr**.

In the module **Conical**, the auxiliary module **Someconstants** is used. This is a module for the computation of the main constants used in the different routines. The module **BesselJY** (for the computation of Bessel functions) and **AiryFunction** (for the computation of Airy functions) are used. The routines included in **auxil.f90** are also used in the module **Conical**.

4. Description of the individual software components

The Fortran 90 module **Conical** includes the public routine **conicp**, which computes the conical functions $P_{-\frac{1}{2}+i\tau}^m(x)$ for $x > -1$, $m \geq 0$ and $\tau > 0$; the routine **conicr**, which computes the function $R_{-\frac{1}{2}+i\tau}^m(x)$, for $x > 1$, $m \geq 0$ and $\tau > 0$ and the routine **conicpr**, which computes both functions $P_{-\frac{1}{2}+i\tau}^m(x)$, $R_{-\frac{1}{2}+i\tau}^m(x)$
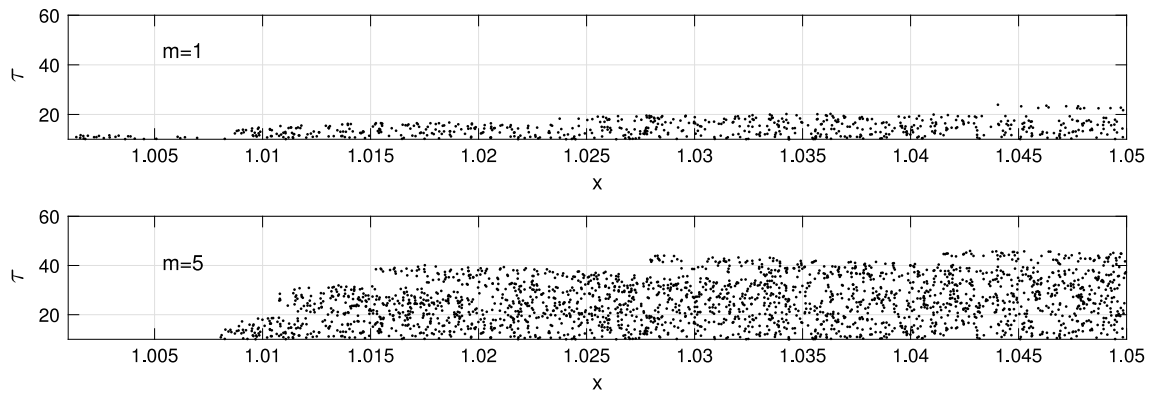
Fig. 1. Test of the performance of the expansion (9). The points where the value of the error when testing the recurrence relation (8) is greater than 10^{-12} are plotted.

and their first order derivatives for $x > 1$, $m \geq 0$ and $\tau > 0$. The calling sequences of these routines are

```
CALL  conicp(x,mu,tau,pm,ierr)
```

where the input data are: *x*, *mu* and *tau* (arguments of the functions). The outputs are the error flag *ierr* and the function value *pm*. The possible values of the error flag are: *ierr* = 0, successful computation; *ierr* = 1, computation failed due to overflow/underflow; *ierr* = 2, arguments out of range.

```
CALL  conicr(x,mu,tau,rm,ierr)
```

where the input data are: *x*, *mu* and *tau* (arguments of the functions). The outputs are the error flag *ierr* and the function value *rm*. The possible values of the error flag are: *ierr* = 0, successful computation; *ierr* = 1, computation failed due to overflow/underflow; *ierr* = 2, arguments out of range.

```
CALL  conicpr(x,mu,tau,pm,pmd,rm,rmd,ierr)
```

where the input data are: *x*, *mu* and *tau* (arguments of the functions). The outputs are the error flag *ierr*, the function values *pm*, *rm* and the first order derivatives *pmd*, *rmd*. The possible values of the error flag are: *ierr* = 0, successful computation; *ierr* = 1, computation failed.

5. Testing the algorithm

For testing the accuracy of the expansions used to compute the conical function $R^m_{-\frac{1}{2}+i\tau}(x)$, we have first checked (8) written in the form

$$\frac{\left(2mx/\sqrt{x^2-1}\right)R^m_{-\frac{1}{2}+i\tau}(x) - \left((m-\frac{1}{2})^2+\tau^2\right)R^{m-1}_{-\frac{1}{2}+i\tau}(x)}{R^{m+1}_{-\frac{1}{2}+i\tau}(x)} = 1.$$

(24)

This check fails close to the zeros of $R^{m+1}_{-\frac{1}{2}+i\tau}(x)$; in this case, we can consider the alternative test

$$\frac{R^{m+1}_{-\frac{1}{2}+i\tau}(x) + \left((m-\frac{1}{2})^2+\tau^2\right)R^{m-1}_{-\frac{1}{2}+i\tau}(x)}{\left(2mx/\sqrt{x^2-1}\right)R^m_{-\frac{1}{2}+i\tau}(x)} = 1.$$

(25)

Because the zeros of $R^{m+1}_{-\frac{1}{2}+i\tau}(x)$ and $R^m_{-\frac{1}{2}+i\tau}(x)$ are interlaced, both tests will not fail simultaneously. We can therefore take the minimum of both errors. We have considered these tests for the expansions described in Sections 2.1.2 (*x* small) and 2.2 (*x* large). Fig. 1 shows the points where the minimum value of the error of the tests (24) and (25) when using (9) is greater than 10^{-12}. In

the algorithm, we have fixed $N = 7$ the number of terms used in the expansion. Random points have been generated in the domain $(x, \tau) \in (1.001, 1.05) \times (15, 100)$. As can be seen, for $m = 1$ (upper figure) the use of the expansion (9) allows to compute the function values with an accuracy better than 10^{-12} for values of τ greater than 20 when *x* is close to 1. The accuracy of the expansion worsens as *m* increases, as can be seen also in Fig. 1 (lower figure) where the same test is considered for $m = 5$. Therefore, one has to use an alternative method of computation for moderate/large values of *m* as, for example, the use of the recurrence relation (8) starting from $R^0_{-\frac{1}{2}+i\tau}(x)$ and $R^1_{-\frac{1}{2}+i\tau}(x)$.

Fig. 2 shows the same tests (24) and (25) for the expansion (17) and for $\mu \equiv m = 1$. The domain where the random points have been generated is now $(x, \tau) \in (1.2, 100) \times (0, 100)$. As can be seen in the figure, there is some loss of accuracy when τ is large and *x* is moderate/large. In any case, we have checked that the accuracy was always better than $5 \, 10^{-12}$.

For testing the expansions for $R^0_{-\frac{1}{2}+i\tau}(x)$ and $R^1_{-\frac{1}{2}+i\tau}(x)$ of Section 2.1.1, we have used the Wronskian relation given in (3). In this case, we have

$$P^1_{-\frac{1}{2}+i\tau}(x)R^0_{-\frac{1}{2}+i\tau}(x) - P^0_{-\frac{1}{2}+i\tau}(x)R^1_{-\frac{1}{2}+i\tau}(x)$$
$$= \frac{e^{-\pi\tau} + \sinh(\pi\tau)}{\cosh(\pi\tau)\sqrt{x^2-1}}.$$

(26)

Fig. 3 shows the points where the value of the error in the Wronskian relation (26) is greater than 10^{-12}.

The accuracy of the final algorithm for $R^m_{-\frac{1}{2}+i\tau}(x)$ has been tested by computing the Wronskian relation given in (3) for a very large number of random points in the parameter domain $(x, m\tau) \in (1.001, 100) \times (0, 100) \times (0, 100)$. The algorithm for the conical function $P^m_{-\frac{1}{2}+i\tau}(x)$ was improved by considering more coefficients in some of the asymptotic expansions used for computing the function in the region $x > 1$. We have checked that the accuracy of the Wronskian test (3) is close to 10^{-12} in the whole parameter domain and better than 10^{-13} for a large fraction of the tested parameter values.

6. Test run description

The Fortran 90 test program **testcon.f90** includes the computation of 20 values of the functions $P^m_{-\frac{1}{2}+i\tau}(x)$, $R^m_{-\frac{1}{2}+i\tau}(x)$ and their first order derivatives and their comparison with the corresponding pre-computed results.
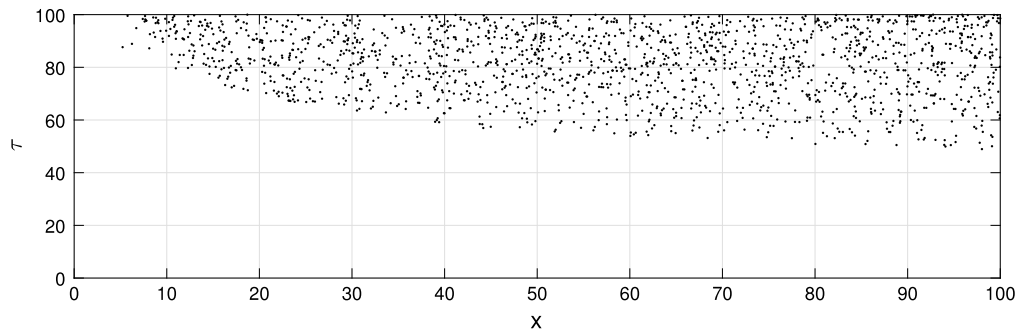
Fig. 2. Test of the performance of the expansion (17) for $\mu \equiv m = 1$. The points where the value of the error in the recurrence relation (8) is greater than 10^{-12} are plotted.
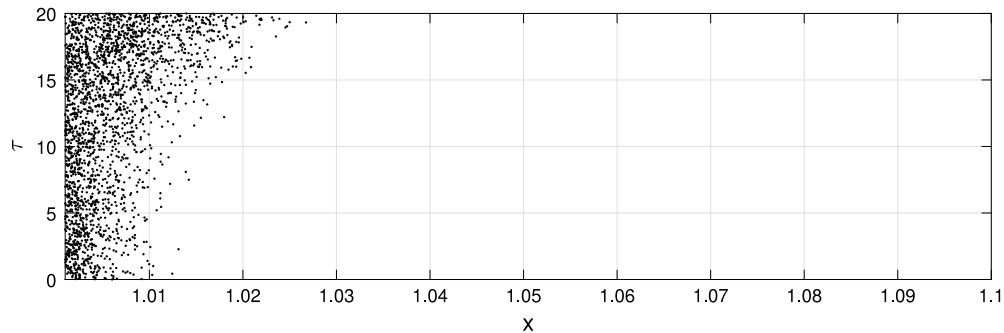

Fig. 3. Test of the performance of the expansions given in (5). The points where the value of the error in the Wronskian relation (26) is greater than 10^{-12} are plotted.

Acknowledgments

References

[1] E. Thebault, J.J. Schott, M. Mandea, J. Geophys. Res.: Solid Earth 111 (2006) B1.
[2] A. Gil, J. Segura, N.M. Temme, Comput. Phys. Comm. (ISSN: 0098-3500) 183 (2012) 794–799.
[3] T.M. Dunster, A. Gil, J. Segura, N.M. Temme, Numer. Algorithms (ISSN: 0308-2105) 68 (2015) 497–509.
[4] A. Gil, J. Segura, N.M. Temme, Comput. Phys. Comm. 191 (2015) 132–139.