Numerical Computation of Inverse Complete Elliptic Integrals of First and Second Kind

Toshio Fukushima

Abstract

We developed the numerical procedures to evaluate the inverse functions of the complete elliptic integrals of the first and second kind, $K(m)$ and $E(m)$, with respect to the parameter $m$. The evaluation is executed by inverting eight sets of the truncated Taylor series expansions of the integrals in terms of $m$ or of $-\log(1-m)$. The developed procedures are (1) so precise that the maximum absolute errors are less than 3-5 machine epsilons, and (2) 30-40\% faster than the evaluation of the integrals themselves by the fastest procedures (Fukushima 2009a, 2011).

Keywords: complete elliptic integral; inversion

1. Introduction

1.1. Complete elliptic integrals of first and second kind

The complete elliptic integrals of the first and the second kind are defined [1, Section 19.2(i)] as

\[ K(m) \equiv \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - m \sin^2 \theta}}, \quad E(m) \equiv \int_0^{\pi/2} \sqrt{1 - m \sin^2 \theta} \, d\theta, \] (1)

where $m$ is called the parameter. They appear in various fields of mathematical physics and engineering [2, 3, 4, 5]. In discussing the real-valued integrals, we assume that $m$ is in the standard domain,

\[ 0 \leq m < 1. \] (2)
Refer to [6, 7] for the process to reduce the other cases to the standard one. See Figure 1 for the functional dependence of $K(m)$ and $E(m)$ on the parameter $m$ in the standard domain.

1.2. Numerical computation of complete elliptic integrals

Many methods have been developed to compute the values of $K(m)$ and/or $E(m)$ when $m$ is given. The popular iterative schemes are those using (1) the arithmetic-geometric mean [1, Section 19.8(i)], (2) the Landen transformation [8, 9], (3) the Bartky transformation [10, 11], or (4) the duplication theorems [12, 13]. They are useful in computing the integrals in arbitrary precision arithmetic.

For practical purposes, however, their Chebyshev approximations [14, 15, 16, 17, 18] are sufficient. Recently, we developed a method based on their Taylor series expansions [6]. It is as precise as the Chebyshev approximation and runs twice faster than that. Also, we extended it to the case of a general complete elliptic integral of the second kind, $\alpha K(m) + \beta E(m)$ [7]. The extended method is as precise as Bulirsch’s cel2 and Carlson’s rf and rd and runs 5-18 times faster than them.

1.3. Necessity of inverse complete elliptic integrals

In practical applications, we sometimes encounter with the inversion problem of $K(m)$ or $E(m)$, namely the determination of $m$ when $K(m)$ or $E(m)$ is given. Especially, that of $K(m)$ is required in estimating a physical quantity related to $m$ from $4K(m)$, the observed period of a certain physical phenomenon expressed by the Jacobian elliptic functions, $\text{sn}(u|m)$, $\text{cn}(u|m)$, and $\text{dn}(u|m)$ [1, Chapter 22]. A classic example of such phenomenon is the motion of a simple gravity pendulum [1, Section 22.19(i)]. Another is the torque-free rotation of an asymmetrical rigid body [19].

1.4. Outline of article

Despite such practical needs, the inversion problem of $K(m)$ and $E(m)$ has not been studied comprehensively. This might be due to their logarithmic singularity around $m = 1$. In this short article, we report a numerical method to invert the integrals by adopting the same approach in our previous work of their evaluation [6, 7]: the piecewise polynomial approximation based on
the Taylor series expansion. In §2, we discuss the difficulties of the inversion problem and find a clue to resolve them. In §3, we present the details of the method we developed based on the hint we found in §2. In §4, we show the results of the numerical experiments in order to illustrate the computational error and the CPU times of the new method.

2. Consideration

2.1. Slow convergence of inverted Maclaurin series

If $m$ is expected to be small, we may compute it by inverting the Maclaurin series expansion of the integrals by the Lagrange inversion theorem \[1, \text{Section 1.10(vii)}\]. The Maclaurin series of the integrals are expressed in terms of Gauss’s hypergeometric series \[1, \text{Formulas 19.5.1 and 19.5.2}\] as

$$K(m) = \frac{\pi}{2} F\left(\frac{1}{2}, \frac{1}{2}; 1; m\right), \quad E(m) = \frac{\pi}{2} F\left(\frac{1}{2}, -\frac{1}{2}; 1; m\right).$$

(3)

Their inversion series around $K(0) = E(0) = \pi/2$ are given as

$$m \approx 4x - 9x^2 + \frac{31}{2}x^3 - \frac{185}{8}x^4 + \frac{507}{16}x^5 + \cdots,$$

(4)

$$m \approx -4y - 3y^2 + \frac{1}{2}y^3 - \frac{5}{8}y^4 + \frac{15}{16}y^5 + \cdots,$$

(5)

where

$$x \equiv \frac{2K(m)}{\pi} - 1, \quad y \equiv \frac{2E(m)}{\pi} - 1.$$  

(6)

One may obtain the higher order expressions by issuing a command in Mathematica \[21\] such as

$$\text{InverseSeries[Series[EllipticK[m], \{m, 0, 65\}]]}$$

The convergence of the hypergeometric series to compute $K(m)$ and $E(m)$ is slow when $m$ increases \[22\]. The same is that of the inverted series. Figure 2 illustrates the order of these inverted power series needed to assure a certain level of accuracy, say 14 digits in this case, as functions of $m$. Apparently, they significantly increase according as $m \to 1$.  

3
2.2. Difficulty in root finding

Another approach of the inversion is the application of a root finding algorithm [5, Chapter 9] to the defining equations

\[ K(m) = K, \quad \text{or} \quad E(m) = E. \tag{7} \]

Although the precise and fast evaluation of \( K(m) \) and \( E(m) \) is established [6, 7], it is not enough. We must find a robust and sufficiently accurate approximate solution such that the following repetition of the update formulas such as the Newton method surely converges.

However, this is a difficult task. The feature of the curves depicted in Figure 1 support this conjecture. Also, the convergence of the iteration becomes slow when \( m \approx 1 \) because the absolute magnitude of the derivatives of \( K(m) \) and \( E(m) \) grow infinitely when \( m \to 1 \).

2.3. Logarithmic singularity

The difficulties explained in the previous subsections are caused by the logarithmic singularity of \( K(m) \) and \( E(m) \) at \( m = 1 \) [1, Formulas 19.12.1 and 19.12.2]. Their simple expressions around the singularity are given [7] as

\[ K(m) = K_1 - K_X \log m_c, \quad E(m) = E_1 - E_X \log m_c, \tag{8} \]

where \( m_c \equiv 1 - m \) is the complementary parameter and \( K_1 \) through \( E_X \) are the quantities being regular around \( m_c = 0 \) as

\[ K_1 \equiv \frac{K(m_c)}{\pi} \log \left( \frac{m_c}{q(m_c)} \right) = 2 \log 2 + \left( \frac{\log 2}{2} - \frac{1}{4} \right) m_c + \cdots, \tag{9} \]

\[ K_X \equiv \frac{K(m_c)}{\pi} = \frac{1}{2} + \frac{m_c}{8} + \cdots, \tag{10} \]

\[ E_1 \equiv \frac{1}{2K_X} + \left( 1 - \frac{E(m_c)}{K(m_c)} \right) K_1 = 1 + \left( \log 2 - \frac{1}{4} \right) m_c + \cdots, \tag{11} \]

\[ E_X \equiv \left( 1 - \frac{E(m_c)}{K(m_c)} \right) K_X = \frac{m_c}{4} + \cdots, \tag{12} \]

where \( q(m) \) is Jacobi’s nome [1, Formula 19.5.5]. These are the base expressions of the Chebyshev approximations [14, 16].
2.4. Variable transformation

The asymptotic forms of the integrals around \( m = 1 \) shown in the previous subsection trigger us to change the main variable from \( m \) to

\[
p \equiv -\log m_c.
\]  

(13)

The domain of \( p \) is semi-infinite since \( p \to +\infty \) when \( m \to 1 \). However, its practical domain can be limited since we may set the meaningful maximum of \( p \) as \( p_{\text{MAX}} \equiv -\log \epsilon \) where \( \epsilon \) is the machine epsilon. It is at most a few tens as \( p_{\text{MAX}} \approx 16.6 \) and \( \approx 36.7 \) in the single and the double precision environments, respectively.

If \( m_c \) is sufficiently small, say less than \( \epsilon \), we may retain only the leading terms in Eqs (9) and (10). Thus, we obtain an asymptotic expression of \( K \equiv K(m) \) as

\[
K = 2 \log 2 + \frac{p}{2} + \cdots.
\]  

(14)

This leads to the inversion in terms of \( p \) as

\[
p = 2K - 4 \log 2 + \cdots,
\]  

(15)

Namely, \( p \) is almost linear with respect to \( K \) when \( K \) is sufficiently large, say larger than 2. Figure 3 confirms this conjecture.

In the case of \( E \equiv E(m) \), we must keep up to the first order terms in Eqs (11) and (12). Then, we arrive at a little complicated expression as

\[
E = 1 + (p + 4 \log 2 - 1) \frac{e^{-p}}{4} + \cdots.
\]  

(16)

We rewrite this as

\[
r \equiv -\log (E - 1) = p - \log \left( \frac{p - 1}{4} + \log 2 \right) + \cdots.
\]  

(17)

The variability of the second term in the right hand side is relatively small as

\[-2.264 < -\log \left( \frac{p - 1}{4} + \log 2 \right) < -0.814,\]  

(18)
when $0 < p < 36.7$. Thus, we ignore its contribution and obtain a solution as

$$p \approx r.$$  \hspace{1cm} (19)

This is a crude approximation. A better solution would be obtained by directly solving Eq. (16) with help of $W_{-1}(x)$, the secondary branch of Lambert W-function [21]. Since the numerical evaluation of $W_{-1}(x)$ is itself a difficult problem [23, 24, 25], we do not go further in this direction.

At any rate, Figure 3 indicates that $K$ and $r$ are almost linear with respect to $p$. Then, they may be candidates of the functions to be expanded in terms of the new variable, $p$.

3. Method

3.1. Strategy

Based on the discussions in §2, we developed a method to invert $K$ and $E$ with respect to $m$ in the standard domain, $0 \leq m < 1$. Let us denote the inverted solutions by $m_K(K)$ and $m_E(E)$, respectively. They are defined as the functions to satisfy the relations

$$K(m_K(x)) = x, \quad E(m_E(x)) = x.$$  \hspace{1cm} (20)

The key idea is splitting the integral value domain into two regions: that corresponding to not-so-large $m$, say less than 0.9 or so, and the rest.

In the former region, we approximate $m$ by a piecewise polynomial of $K$ or $E$. Each piece of the approximate polynomials is obtained by inverting the Taylor series expansion of $K(m)$ and $E(m)$ around a certain reference value of $m$. In the second region, we approximate $p \equiv -\log m_c$ by a piecewise polynomial of $K$ or $r \equiv -\log(E-1)$. Similarly, each piece of the approximate polynomials is obtained by inverting the Taylor series expansion of $K(m(p))$ and $-(E(m(p))-1)$ around a certain reference value of $p$ where

$$m(p) \equiv 1 - e^{-p}.$$  \hspace{1cm} (21)

Once $p$ is obtained, we can numerically evaluate $m$ by this relation.

In the double precision environment, we limit the order of the approximate polynomials less than 19. This is in order to reduce the typical computational time of the inverted solutions down to the level of that of elementary functions like $\exp$. Under this restriction, we try to minimize the number of pieces by adjusting the reference values in terms of $m$ or $p$. 

6
After the approximate polynomials are fixed, we try to apply the Lanczos economization [5, Section 5.11] to them. Namely, we rewrite each piece of them into a finite series of the Chebyshev polynomials of the first kind [26]. Except the cases near the singularity, the magnitude of the series coefficients smoothly decreases according as the degree of the Chebyshev polynomial. In that case, by removing the Chebyshev polynomials of some highest degrees from the series, we can reduce the degree of the original approximate polynomials significantly while keeping their precision practically unchanged.

It is well known that the finite linear sum of the Chebyshev polynomials can be efficiently evaluated by Clenshaw’s summation algorithm [26, Section 3.2]. See also [27, Section 3.7]. Nevertheless, the fastest evaluation is achieved by Horner’s method of the power series expanded around the middle point of the defining interval. The coefficients of the rearranged power series are rigorously obtained from the Chebyshev polynomial coefficients by a conversion procedure such as given in Appendix A.2.

In place of the economized power series, however, we present here the Chebyshev polynomial coefficients such that one can truncate them at an arbitrary precision. This will be useful to construct the single precision procedures. Skipping the details, we show the final results below.

3.2. Case of small parameter

When the solution $m$ is not expected to be large, say less than 0.9 or so, we evaluate $m_K$ and $m_E$ by their piecewise approximate polynomials derived from the inverted Taylor series expansions. The piecewise polynomials consist of four polynomials corresponding to the four sub domains of the integral values denoted by A, B, C, and D. Table 1 shows the integral values to separate these sub domains. For example, as $m_K$, we use the approximate polynomial obtained in the sub domain A if $K_A \leq K < K_B$, that in the sub domain B if $K_B \leq K < K_C$, and so on.

In constructing the piecewise polynomials, we set $m_0$, the reference value of $m$ in each sub domain, as 0.2, 0.5, 0.7, and 0.85 for $K$ and 0.2, 0.5, 0.7, and 0.83 for $E$. In each sub domain, we first obtained the approximate polynomials in the form of truncated Taylor series such as

$$m_K = \sum_{n=0}^{N} a_n (K - K_0)^n,$$  \hspace{1cm} (22)

where $K_0 \equiv K (m_0)$ is the reference integral value. We determined the power
Table 1: Separation values of $K$ and $E$. Listed are the values of $K$ and $E$ to select the best approximate polynomial. The index means that, for $K$, the sub domain $A$ is the interval $[K_A, K_B)$, the sub domain $B$ is the interval $[K_B, K_C)$, $\cdots$, and the sub domain $H$ is the interval $[K_H, \infty)$. On the other hand, for $E$, the sub domain $A$ is the interval $(E_B, E_A]$, the sub domain $B$ is the interval $(E_C, E_B]$, $\cdots$, and the sub domain $H$ is the interval $[1, E_H]$. Also, we added the values of $r \equiv -\log(E - 1)$ for the indices $E$ through $H$.

<table>
<thead>
<tr>
<th>Index</th>
<th>$K$</th>
<th>$E$</th>
<th>$r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.5707963267948966</td>
<td>1.5707963267948966</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1.7539969906494259</td>
<td>1.4111237670965148</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1.9459577498060259</td>
<td>1.290768683852396</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>2.1998873957149437</td>
<td>1.187833138534634</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>2.5687780604965515</td>
<td>1.1199612906642078</td>
<td>2.128631785870677</td>
</tr>
<tr>
<td>F</td>
<td>3.6189656655262064</td>
<td>1.0556008611359877</td>
<td>2.8895565897882066</td>
</tr>
<tr>
<td>G</td>
<td>5.3616789503774974</td>
<td>1.0048203911429256</td>
<td>5.3349002042275533</td>
</tr>
<tr>
<td>H</td>
<td>8.7012952137792582</td>
<td>1.0000443476373730</td>
<td>10.0234511229381364</td>
</tr>
</tbody>
</table>

series coefficients, $a_n$, by issuing commands of Mathematica [21] like

\[
\text{InverseSeries[Series[EllipticK[m], \{m, 0.2, 17\}]}]
\]

which gives those of $m_K$ in the sub domain $A$.

Next, we examined the errors of the piecewise polynomial, $m_K(K)$ or $m_E(E)$, by comparing the corresponding integral values, $K\left(m_K(K)\right)$ or $E\left(m_E(E)\right)$, with the original input values of $K$ or $E$, respectively. We conducted the comparison in the quadruple precision by using qce1, the quadruple precision extension of Bulirsch’s cel [11]. In the errors of some pieces, we found a constant offset of the order of a few machine epsilons. It seems to be caused by the accumulation of round-off errors in the inversion process. At any rate, we adjusted $a_0$ in that case.

Third, by taking the balance between the measured errors of the approximate polynomials obtained in the adjacent sub domains, we determined their separation values in terms of $m$ as 0.365, 0.600, 0.773, and 0.898 for $K$ and 0.375, 0.614, 0.786, and 0.881 for $E$. These are translated into the corresponding integral values as $K_B = K(0.365)$, $\cdots$, $K_E = K(0.898)$, and $E_B = E(0.375)$, $\cdots$, $E_E = E(0.881)$. They are listed in Table 1 together with the lower end values, $K_A \equiv K(0) = \pi/2$ and $E_A \equiv E(0) = \pi/2$. 

8
Finally, we transformed the obtained polynomials into the finite sum of Chebyshev polynomials as

$$m_K = \sum_{n=0}^{N} c_n T_n(t) = c_0 T_0(t) + c_1 T_1(t) + c_2 T_2(t) + \cdots,$$

(23)

where $N$ is the degree of the obtained polynomial, $T_n(t)$ is the Chebyshev polynomial of the first kind of degree $n$ [1, Table 18.3.1], and $c_n$ is its $n$-th coefficient. Appendix A.1 describes an algorithm to do the conversion from $a_n$ to $c_n$. Notice that we define the zeroth coefficient, $c_0$, without the divisor 2. This is different from the definition of the Chebyshev expansion in some literatures [26, 5, 27].

Meanwhile, the argument of the Chebyshev polynomials, $t$, which is in the range $[-1, +1]$, is computed from the integral values such as

$$t \equiv a(K - b),$$

(24)

where

$$a \equiv \frac{2}{K_U - K_L}, \quad b \equiv \frac{K_U + K_L}{2},$$

(25)

are the transformation coefficients determined from $K_U$ and $K_L$, the upper and lower ends of the sub domain. These end values are already listed in Table 1. We conducted exactly the same treatment for $E$. The resulting Chebyshev polynomial coefficients and the transformation coefficients are explicitly given in Tables 2 and 3 for $m_K$ and $m_E$, respectively.

3.3. Case of large parameter

If $K \geq K_E$ or $E \leq E_E$, the parameter $m$ becomes significantly large, say larger than 0.89 or so. In that case, we approximate $p \equiv -\log m_c$ by its piecewise polynomial in terms of $K$ or $r \equiv -\log(E - 1)$. In constructing the piecewise polynomials, we set $p_0$, the reference value of $p$ in each sub domain, as 3.3, 6.0, 10.5, and 17.6 for $K$ and 1.8, 4.2, 8.0, and 15.0 for $r$. In each sub domain, the approximate polynomials are written as

$$p_K = \sum_{n=0}^{N_K} p_n (K - K_0)^n, \quad p_E = \sum_{n=0}^{N_E} q_n (r - r_0)^n,$$

(26)

where

$$K_0 \equiv K \left(1 - e^{-p_0}\right), \quad r_0 \equiv -\log \left[E \left(1 - e^{-p_0}\right) - 1\right],$$

(27)
Table 2: Chebyshev polynomial coefficients of $m_K$: sub domains A through D. The numerical values of the coefficients, $c_n$, are multiplied by a certain powers of 10 such as $10^{17}$ in order to suppress the leading zeros. For example, the zeroth coefficients must read $+0.19\cdots, +0.49\cdots, +0.69\cdots, +0.84\cdots$. These zeroth coefficients are defined without the divisor 2. Namely, $m_K$ is approximated as $m_K = c_0T_0(t) + c_1T_1(t) + \cdots$, where $T_n$ is the Chebyshev polynomial of the first kind of degree $n$. Also, we listed $a$ and $b$, the transformation coefficients from $K$ to $t$, the argument of Chebyshev polynomials, such that $t = a(K - b)$. All the values are shown with a few more digits than necessary in the double precision computation. This is in order to avoid the round-off errors in the implementation.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>$10^{17}c_n$</td>
<td>$10^{17}c_n$</td>
<td>$10^{17}c_n$</td>
<td>$10^{17}c_n$</td>
</tr>
<tr>
<td>0</td>
<td>+19401242299345095</td>
<td>+4890719448270469</td>
<td>+69305133428229853</td>
<td>+84201552855821970</td>
</tr>
<tr>
<td>1</td>
<td>+18194753782220755</td>
<td>+11715040871065477</td>
<td>+8613490854834244</td>
<td>+62010362075457533</td>
</tr>
<tr>
<td>2</td>
<td>-1148955521044524</td>
<td>-739311643473559</td>
<td>-653420708906968</td>
<td>-64851448920333</td>
</tr>
<tr>
<td>3</td>
<td>+55159476593419</td>
<td>+349074745999398</td>
<td>+36485606978491</td>
<td>+48795524452071</td>
</tr>
<tr>
<td>4</td>
<td>-2283696557743</td>
<td>-14060191209690</td>
<td>-1709827595538</td>
<td>-3029715923753</td>
</tr>
<tr>
<td>5</td>
<td>+96644143620</td>
<td>+515951182898</td>
<td>+72527965477</td>
<td>+167839458005</td>
</tr>
<tr>
<td>6</td>
<td>-3105064734</td>
<td>-1783762901</td>
<td>-2887658427</td>
<td>-8670547452</td>
</tr>
<tr>
<td>7</td>
<td>+106925302</td>
<td>+59157015</td>
<td>+110075215</td>
<td>+427595012</td>
</tr>
<tr>
<td>8</td>
<td>-3574353</td>
<td>-1902485</td>
<td>-4063750</td>
<td>-20387219</td>
</tr>
<tr>
<td>9</td>
<td>+116785</td>
<td>+50756</td>
<td>+146392</td>
<td>+947243</td>
</tr>
<tr>
<td>10</td>
<td>-3747</td>
<td>-1842</td>
<td>-5173</td>
<td>-43128</td>
</tr>
<tr>
<td>11</td>
<td>+118</td>
<td>+56</td>
<td>+180</td>
<td>+1932</td>
</tr>
<tr>
<td>12</td>
<td>-85</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$a$ +10.916991008220921 +10.226477662730623 +7.9897843924333070 +5.4216606462081334

$b$ +1.662396658721613 +1.851782370227759 +2.074275727048484 +2.3843327281057476
Table 3: Chebyshev polynomial coefficients of $m_E$: sub domains A through D. Same as Table 2 but for $m_E$. The zeroth coefficients must read $+0\cdots$, $+0.49\cdots$, $+0.70\cdots$, and $+0.83\cdots$. Also, the transformation from $E$ to $t$ is expressed as $t = a(E - b)$.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$+191.491596762105577$</td>
<td>$+4969412566025830$</td>
<td>$+7019787637353759$</td>
<td>$+83448854563061028$</td>
</tr>
<tr>
<td>1</td>
<td>$+1875.217509862407743$</td>
<td>$+1195168881417314$</td>
<td>$+8602065171825419$</td>
<td>$+4751257054900759$</td>
</tr>
<tr>
<td>2</td>
<td>$-3990799269547337$</td>
<td>$-244061880474704$</td>
<td>$-197779951710636$</td>
<td>$-98795130570153$</td>
</tr>
<tr>
<td>3</td>
<td>$-21715474908196$</td>
<td>$-63628928178$</td>
<td>$-5591458166$</td>
<td>$-3530719901$</td>
</tr>
<tr>
<td>4</td>
<td>$-172119411$</td>
<td>$-156070789$</td>
<td>$-14065$</td>
<td>$-897$</td>
</tr>
<tr>
<td>5</td>
<td>$-1376$</td>
<td>$-150$</td>
<td>$-106$</td>
<td>$-81$</td>
</tr>
</tbody>
</table>

\[a = -12.525637330469197, \quad b = +1.4909600469457057\]

are the reference function values. We obtained the approximate polynomials by issuing commands of Mathematica \[21\] like

\[
\text{InverseSeries}[\text{Series}[\text{Log}[\text{EllipticE}[1-\text{Exp}[-p]] - 1]], \{p, 1.8, 17\}]
\]

which gives the coefficients of $p_E$ for $p_0 = 1.8$.

After constructing the approximate polynomials, we again examined their errors by using \texttt{qcel}, and determined the separation values in terms of $p$ as 4.45, 7.95, and 14.63 for $K$ and 3.1, 6.0, and 11.2 for $E$. These are translated into the corresponding function values as $K_F = K (1 - e^{-4.45})$, $\cdots$, $K_H = K (1 - e^{-14.63})$, $E_F = E (1 - e^{-3.1})$, $\cdots$, $E_H = E (1 - e^{-11.2})$, and $r_E = -\log (E_E - 1)$, $\cdots$, and $r_H = -\log (E_H - 1)$. They are already listed in Table 1.

For the sub domains $E$ and $F$, we again transformed the obtained polynomials into the form of the series of Chebyshev polynomials such as

\[
p_E = \sum_{n=0}^{N} c_n T_n(t),
\]
where the transformation of argument becomes

\[ t \equiv a(r - b), \quad (29) \]

with the transformation coefficients defined as

\[ a \equiv \frac{2}{r_U - r_L}, \quad b \equiv \frac{r_U + r_L}{2}, \quad (30) \]

where \( r_U \) and \( r_L \) are the upper and lower ends of the sub domain in terms of \( r \), which are already listed in Table 1. The Chebyshev polynomial coefficients and the transformation coefficients are explicitly given in Tables 4 and 5. Noting that \( \Delta m \approx e^{-p} \Delta p \), we may truncate the Chebyshev expansion in terms of \( p \) earlier than that in terms of \( m \).

3.4. Case near the singularity

For the sub domains G and H, we find that the Lanczos economization is not effective in reducing the computational labor. Of course, one may execute the transformation into Chebyshev polynomial series. For the sub domain G, which is semi-infinite, this is practically feasible by setting the upper end as \( K_I = +19.755 \) or \( r_I = +34.472 \), which roughly corresponds to the numerical value of \( K \) or \( E \) when \( m = 1 - \epsilon_{\text{double}} \) where \( \epsilon_{\text{double}} \equiv 2^{-53} \approx 1.11 \times 10^{-16} \) is the double precision machine epsilon.

However, we learn that the decreasing manner of the magnitude of the resulting coefficients of the Chebyshev polynomials is quite slow. This is because the minimax type approximation with respect to \( p \) does not guarantee a uniform error distribution in terms of \( m \) since \( \Delta m \approx e^{-p} \Delta p \). Therefore, we abandon the economization and present the original Taylor series themselves in Tables 6 and 7. We truncated them such that the numbers of terms are the necessary minimum to assure the maximum of \( |\Delta m| \) remain at the level of \( \epsilon_{\text{double}} \).

4. Numerical Experiments

Let us examine the computational cost and performance of the method described in the previous section. The absolute errors of \( m \) obtained by the method in the double precision environment are illustrated in Figs 4 and 5. They are measured by using \texttt{qce1}, the quadruple precision extension of Bulirsch’s procedure to evaluate the general complete elliptic integral [11].
Table 4: Chebyshev polynomial coefficients of \( p_K \): sub domains E and F. Same as Table 2 but for \( p_K \). The zeroth coefficients must read +3.37\cdots, and +6.20\cdots. The argument transformation becomes \( t = a(K - b) \) again. In truncating the series of Chebyshev polynomials, one should note that the final error in \( m \) is reduced by the factor \( e^{-p} \) as \(|\Delta m| \approx e^{-p}|\Delta p|\). The maximum reduction factor in each sub domain is also listed.

<table>
<thead>
<tr>
<th></th>
<th>( 10^{15}c_n )</th>
<th>( 10^{14}c_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>+3372842411096568</td>
<td>+620234816387263</td>
</tr>
<tr>
<td>1</td>
<td>+1082799019684612</td>
<td>+174947955583875</td>
</tr>
<tr>
<td>2</td>
<td>−6377916757929</td>
<td>−226305230130</td>
</tr>
<tr>
<td>3</td>
<td>+804553258630</td>
<td>+50978226844</td>
</tr>
<tr>
<td>4</td>
<td>−72934012596</td>
<td>−8407174303</td>
</tr>
<tr>
<td>5</td>
<td>+5172172149</td>
<td>+1058249590</td>
</tr>
<tr>
<td>6</td>
<td>−325770457</td>
<td>−103492197</td>
</tr>
<tr>
<td>7</td>
<td>+21898786</td>
<td>+7909234</td>
</tr>
<tr>
<td>8</td>
<td>−1696166</td>
<td>−487701</td>
</tr>
<tr>
<td>9</td>
<td>+136104</td>
<td>+30082</td>
</tr>
<tr>
<td>10</td>
<td>−10427</td>
<td>−2890</td>
</tr>
<tr>
<td>11</td>
<td>+780</td>
<td>+372</td>
</tr>
<tr>
<td>12</td>
<td>−60</td>
<td>−42</td>
</tr>
<tr>
<td>13</td>
<td>+5</td>
<td>+4</td>
</tr>
</tbody>
</table>

\( a \) | 1.9044216389732810 | 1.1476357111552424 |
\( b \) | 3.0938718630113789 | 4.4903223079518519 |

\( \text{max } e^{-p} \) | 0.102 | \( \approx 0.0450 \)
Table 5: Chebyshev polynomial coefficients of $p_E$: sub domains E and F. Same as Table 4 but for $p_E$. The zeroth coefficients must read $+2.62 \cdots$ and $+4.56 \cdots$. The argument transformation is written as $t = a(r - b)$ where $r \equiv -\log(E - 1)$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$10^{15}c_n$ (E)</th>
<th>$10^{15}c_n$ (F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$+2621927976267172$</td>
<td>$+4563857924247435$</td>
</tr>
<tr>
<td>1</td>
<td>$+480434270533396$</td>
<td>$+1448978332531438$</td>
</tr>
<tr>
<td>2</td>
<td>$-2421516965161$</td>
<td>$-13787853000146$</td>
</tr>
<tr>
<td>3</td>
<td>$+60368511503$</td>
<td>$+1017879586053$</td>
</tr>
<tr>
<td>4</td>
<td>$-1103238537$</td>
<td>$-69972247182$</td>
</tr>
<tr>
<td>5</td>
<td>$+4267375$</td>
<td>$+3797035826$</td>
</tr>
<tr>
<td>6</td>
<td>$+654240$</td>
<td>$-100760933$</td>
</tr>
<tr>
<td>7</td>
<td>$-30728$</td>
<td>$-8981925$</td>
</tr>
<tr>
<td>8</td>
<td>$+745$</td>
<td>$+1749497$</td>
</tr>
<tr>
<td>9</td>
<td>$-5$</td>
<td>$-170996$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>$+11351$</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>$-400$</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>$-22$</td>
</tr>
</tbody>
</table>

\[ a = +2.62838060962536483 \quad \text{and} \quad b = +2.5090941878294072 \]

\[ \text{max } e^{-p} \approx 0.0450 \]
Table 6: Taylor series coefficients of $p_K$: sub domains G and H. Listed are the coefficients of the power series expression as $p_K = \sum_{n=0}^{N} a_n (K - K_0)^n$.

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>$a_n$</td>
<td>$n$</td>
</tr>
<tr>
<td>0</td>
<td>+10.500000000000</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>+2.0001414497655</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>-1.277026552E-4</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>+7.59811036E-5</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>-3.34231082E-5</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>+1.15490151E-5</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>-3.2468658E-6</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>+7.573936E-7</td>
<td>17</td>
</tr>
</tbody>
</table>

$K_0$ +6.6363331625866755 +10.186294413299100
Table 7: Taylor series coefficients of $p_E$; sub domains G and H. Same as Table 6 but for $p_E$ expressed as $p_E = \sum_{n=0}^{N} a_n (r - r_0)^n$ where $r \equiv -\log(E - 1)$.

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>$a_n$</td>
<td>$a_n$</td>
</tr>
<tr>
<td>0</td>
<td>+8.000000000000000000</td>
<td>+15.000000000000000000</td>
</tr>
<tr>
<td>1</td>
<td>+1.113855845808138764</td>
<td>+1.0634010131365226997</td>
</tr>
<tr>
<td>2</td>
<td>-7.16073700925904412E-3</td>
<td>-2.137214443579717823E-3</td>
</tr>
<tr>
<td>3</td>
<td>+6.14816637761468272E-4</td>
<td>+9.89052796853563E-5</td>
</tr>
<tr>
<td>4</td>
<td>-5.7949395860221721E-5</td>
<td>-5.240426861207633E-6</td>
</tr>
<tr>
<td>5</td>
<td>+5.451526070327390E-6</td>
<td>+2.99271455626484E-7</td>
</tr>
<tr>
<td>6</td>
<td>-4.80075731204772E-7</td>
<td>-1.7869634729734E-8</td>
</tr>
<tr>
<td>7</td>
<td>+3.7093529426652E-8</td>
<td>+1.092804281815E-9</td>
</tr>
<tr>
<td>8</td>
<td>-2.222894130707E-9</td>
<td>-6.7254295377E-11</td>
</tr>
<tr>
<td>9</td>
<td>+5.377967260E-11</td>
<td>+4.096019951E-12</td>
</tr>
<tr>
<td>10</td>
<td>+1.071061734E-11</td>
<td>-2.42723757E-13</td>
</tr>
<tr>
<td>11</td>
<td>-2.33974846E-12</td>
<td>+1.3740808E-14</td>
</tr>
<tr>
<td>12</td>
<td>+3.0858084E-13</td>
<td>-7.2627E-16</td>
</tr>
<tr>
<td>13</td>
<td>-3.2399999E-14</td>
<td>+3.455E-17</td>
</tr>
<tr>
<td>14</td>
<td>+2.84089E-15</td>
<td>-1.36E-18</td>
</tr>
<tr>
<td>15</td>
<td>-2.0071E-16</td>
<td>+3.E-20</td>
</tr>
<tr>
<td>16</td>
<td>+9.0E-18</td>
<td>-1.4E-21</td>
</tr>
<tr>
<td>17</td>
<td>+3.E-19</td>
<td>+7.E-23</td>
</tr>
<tr>
<td>18</td>
<td>-2.E-19</td>
<td></td>
</tr>
<tr>
<td>$r_0$</td>
<td>+7.1066021667717055</td>
<td>+13.5665483242003067</td>
</tr>
</tbody>
</table>
Figure 4: Parameter dependence of the errors of the inverse complete elliptic integrals: case of small parameters. Shown are the absolute errors of the double precision computation of \( m_K(K) \) and \( m_E(E) \) as functions of their values. The unit of the errors is the double precision machine epsilon, \( \epsilon_{\text{double}} \approx 1.11 \times 10^{-16} \).

Figure 5: Parameter dependence of the errors of the inverse complete elliptic integrals: case of large parameters. Same as Fig. 4 but as functions of \( p \equiv -\log(1 - m) \). The maximum value of \( p \) shown here is that corresponding to the extreme case, \( m = 1 - \epsilon_{\text{double}} \).

More specifically speaking, we first prepare \( m \) in the quadruple precision environment. Next, we compute \( K(m) \) and \( E(m) \) in the quadruple precision by \texttt{qcel} as

\[
K(m) = \texttt{qcel}(\sqrt{1.0 - m}, 1.0, 1.0, 1.0),
\]

\[
E(m) = \texttt{qcel}(\sqrt{1.0 - m}, 1.0, 1.0, 1.0 - m).
\]

Third, we compute the double precision values of \( m_K \) and \( m_E \) from the double precision value of \( K(m) \) and \( E(m) \) using the procedures described in the previous section, namely the finite series of the Chebyshev polynomials in the sub domains A through F and the inverted Taylor series in the sub domains G and H. Finally, we take the difference of \( m_K \) and \( m_E \) from the original \( m \) in the quadruple precision environment.

Figs 4 and 5 show that, in the double precision computation, the magnitude of the absolute errors of \( m_K \) and \( m_E \) are less than 3 and 5 machine epsilons, respectively.

On the other hand, Table 8 compares the CPU time of the fastest procedures to compute \( K(m) \) or \( E(m) \) [6, 7] as well as the procedures to obtain \( m_K(K) \) and \( m_E(E) \) using Horner’s method to evaluate the economized or original power series. The coefficients of the economized power series are obtained from the tables in the previous section by the conversion procedure described in Appendix A.2. The CPU times are averaged for \( 2^{26} \) grid points evenly distributed in the parameter domain, \( 0 \leq m < 1 \). The measurements are conducted at a PC with an Intel Core i7-2675QM run at 2.20 GHz clock under Windows 7.

Table 8 shows that the procedures to compute the inverse functions of \( K(m) \) and \( E(m) \) are 30 and 40 % faster than those to compute themselves [6, 7], respectively. This is mainly due to the faster convergence of the inverted Taylor series than the Taylor series and partly due to the Lanczos economization of the polynomial expressions.
Table 8: CPU time comparison. Listed are the averaged CPU time of some procedures to compute the complete elliptic integrals and their inverse in the double precision environment. The unit of CPU time is nano second at a latest consumer PC with an Intel Core i7-2675QM run at 2.20 GHz clock.

<table>
<thead>
<tr>
<th>Target</th>
<th>Procedure</th>
<th>Reference</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K(m)$</td>
<td>celk</td>
<td>[6]</td>
<td>71</td>
</tr>
<tr>
<td>$m_K(K)$</td>
<td>icelk</td>
<td>This article</td>
<td>55</td>
</tr>
<tr>
<td>$E(m)$</td>
<td>celbd</td>
<td>[7]</td>
<td>83</td>
</tr>
<tr>
<td>$m_E(E)$</td>
<td>icele</td>
<td>This article</td>
<td>58</td>
</tr>
</tbody>
</table>

5. Conclusion

We developed the numerical procedures to compute $m_K(K)$ and $m_E(E)$, the inverse complete elliptic integrals of the first and the second kind, $K \equiv K(m)$ and $E \equiv E(m)$, in the double precision environment, respectively. The key idea of the procedures is splitting the integral value domain into two regions: that corresponding to not-so-large $m$, say smaller than 0.9 or so, and that near the logarithmic singularity, $m = 1$. In the former region, we obtained a piecewise power series approximation of the inverse functions by using the inverted Taylor series expansion with respect to $m$. In the latter region, we (1) changed the main variable from $m$ to $p \equiv -\log(1-m)$, (2) regarded $K$ or $r \equiv -\log(E-1)$ as a function of $p$, and (3) inverted the Taylor series of $K$ and $r$ with respect to $p$. The resulting piecewise polynomials consist of four polynomials for each region. Their selection rules are determined from the error measurement of the obtained polynomials by means of the quadruple precision computation of the integral values. Except for the two sub domains near the singularity point, $m = 1$, we rearranged the polynomial expressions into the finite series of Chebyshev polynomials such that one may truncate them at an arbitrary level of precision. This makes easy to obtain the single precision procedures. Thanks to the effectiveness of the policy of divide-and-rule, the developed procedures are so precise that, in the double precision environment, the maximum absolute error of $m_K$ and $m_E$ is less than 3 and 5 machine epsilons, respectively. Also, the procedures run significantly faster than the fastest procedures of the integral value computation [6].
The double precision Fortran functions of the developed procedures, icelk and icele, are available from the author’s personal WEB page at ResearchGate:

https://www.researchgate.net/profile/Toshio_Fukushima/

Acknowledgments

The author thanks the anonymous referee for many valuable advices to improve the quality of the present article including the application of the Lanczos economization.

Appendix A. Transformation between power series and Chebyshev polynomial series

Any polynomial can be expressed in two ways, a power series and a series of Chebyshev polynomials [26]. Namely, it is written as

\[
\sum_{n=0}^{N} a_n x^n = \sum_{n=0}^{N} c_n T_n(x),
\]

where \(N\) is the degree of the polynomial, \(T_n(x)\) is the Chebyshev polynomial of the first kind of degree \(n\) [1, Table 18.3.1], and we assumed that the argument \(x\) is in the standard interval, \([-1, +1]\). Below, we summarize the transformation between the coefficients of the power series, \(\{a_n\}\), and those of the Chebyshev polynomials, \(\{c_n\}\).

Appendix A.1. From power series to Chebyshev polynomial series

It is well known that \(T_n(x)\) is a polynomial of degree \(n\) with integral coefficients [27, Eq.(3.32)] as

\[
T_n(x) = \sum_{\ell=0}^{[n/2]} T_{n\ell} x^{n-2\ell},
\]

where \(T_{n\ell}\) is an integer and \([x]\) is the floor function. Note that the range of the second index is \(0 \leq \ell \leq [n/2] \leq [N/2]\). Its length is roughly the half of that of the first one since \(0 \leq n \leq N\).
The explicit expression of $T_{n\ell}$ is known [27, Eq.(3.33)] as
\[ T_{00} = T_{10} = 1, \tag{A.3} \]
\[ T_{n\ell} = (-1)^\ell 2^{n-2\ell-1} \left( \frac{n}{n-\ell} \right)^\ell. \tag{2 \leq n, \ 0 \leq \ell \leq \lfloor n/2 \rfloor} \tag{A.4} \]
The latter is simplified when $n = 2\ell$ [27, Eq.(3.34)] as
\[ T_{2\ell,\ell} = (-1)^\ell. \tag{A.5} \]
Instead of evaluating these expressions, we compute $T_{n\ell}$ by recursion as
\[ T_{n+1,0} = 2T_{n0}, \ (1 \leq n \leq N - 1) \tag{A.6} \]
\[ T_{n+1,\ell} = 2T_{n\ell} - T_{n-1,\ell-1}, \ (2 \leq n \leq N - 1, \ 1 \leq \ell \leq \lfloor n/2 \rfloor - 1) \tag{A.7} \]
\[ T_{2\ell,\ell} = -T_{2(\ell-1),\ell-1}, \ (1 \leq \ell \leq \lfloor N/2 \rfloor) \tag{A.8} \]
starting from the initial conditions, Eq.(A.3). The above recurrence formulas are nothing but the translation of the three term recurrence relation of $T_n(x)$ [27, Eq.(3.20)],
\[ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \ (1 \leq n) \tag{A.9} \]
in terms of the coefficients. The recursion consists of integer operations only, and therefore it suffers no round-off errors when $n$ is sufficiently small, say less than 30 when using the 4 byte integers. Also, its execution is fairly fast.

Using $T_{n\ell}$, we obtain a transformation formula from \{c_n\} to \{a_n\} as
\[ a_n = \sum_{\ell=0}^{\lfloor (N-n)/2 \rfloor} T_{n+2\ell,\ell} \ c_{n+2\ell}. \ (0 \leq n \leq N) \tag{A.10} \]
This is simplified when $n = 0$ as
\[ a_0 = \sum_{\ell=0}^{\lfloor N/2 \rfloor} (-1)^\ell \ c_{n+2\ell}. \tag{A.11} \]

Usually the magnitude of $c_n$ is monotonically decreasing with the index, $n$. Then, we execute the summation in the above transformation formula in the reverse order, i.e. the order decreasing $\ell$. This trick significantly suppresses the accumulation of round-off errors in the transformation procedure. This is the reason why we prepare the coefficient matrix, $T_{n\ell}$, at the cost of additional memory. For the readers’ convenience, we present a double precision Fortran subroutine to do the transformation in Table A.9.
Table A.9: Double precision Fortran subroutine to transform the Chebyshev polynomial coefficients, \{c_n\}, into the power series coefficients, \{a_n\}.

```fortran
subroutine cheb2poly(n,c,a)
integer JMAX,NMAX
parameter (JMAX=15,NMAX=JMAX*2)
integer n,T(0:NMAX,0:JMAX),m,j
real*8 c(0:n),a(0:n),am
logical first/.TRUE./
save first,T
if(first) then
    first=.FALSE.;T(0,0)=1;T(1,0)=1
    do j=1,JMAX
        T(2*j,j)=-T(2*(j-1),j-1)
    enddo
    do m=2,NMAX
        T(m,0)=2*T(m-1,0)
        do j=1,(m-1)/2
            T(m,j)=2*T(m-1,j)-T(m-2,j-1)
        enddo
    enddo
endif
    do m=0,n
        am=0.d0
        do j=(n-m)/2,0,-1
            am=am+dble(T(m+2*j,j))*c(m+2*j)
        enddo
        a(m)=am
    enddo
return;end
```

21
Appendix A.2. From Chebyshev polynomial series to power series

A monomial of degree \( n \) is expressed as a finite linear sum of the Chebyshev polynomial of the same or lower degree [27, Eq.(3.35)] as

\[
x^n = \sum_{\ell=0}^{\lfloor n/2 \rfloor} B_{n\ell} T_{n-2\ell}(x), \tag{A.12}
\]

where the coefficients \( B_{n\ell} \) are explicitly written as

\[
B_{00} = 1, \tag{A.13}
\]

\[
B_{n\ell} = \frac{1}{2^{n-1}} \binom{n}{\ell}, \quad (1 \leq n, \quad 0 \leq \ell \leq \lfloor n/2 \rfloor) \tag{A.14}
\]

These coefficients are finite fractions in the binary machines, and therefore can be expressed without round-off errors when \( n \) is sufficiently small, say less than 50 in the double precision environment. Again, we prefer computing them by recursion as

\[
B_{n+1,0} = 2^{-1}B_{n0}, \quad (1 \leq n \leq N - 1) \tag{A.15}
\]

\[
B_{n+1,\ell} = 2^{-1}(B_{n,\ell-1} + B_{n\ell}), \quad (2 \leq n \leq N - 1, \quad 1 \leq \ell \leq \lfloor (n-1)/2 \rfloor) \tag{A.16}
\]

\[
B_{2\ell,\ell} = B_{2\ell-1,\ell-1}, \quad (1 \leq \ell \leq \lfloor N/2 \rfloor) \tag{A.17}
\]

starting from the initial condition, Eq.(A.13). At any rate, the transformation from \( \{a_n\} \) to \( \{c_n\} \) is of the same form as Eq.(A.10) as

\[
c_n = \sum_{\ell=0}^{\lfloor (N-n)/2 \rfloor} B_{n+2\ell,\ell} a_{n+2\ell}, \quad (1 \leq n \leq N) \tag{A.18}
\]

We anticipate that the magnitude of \( a_n \) is also monotonically decreasing with the index, \( n \). Then, we again execute the summation in the transformation formula in the reverse order.

As for the most important coefficient, \( c_0 \), we adopt a different approach. We solve the expression of \( a_0 \), Eq.(A.11), with respect to \( c_0 \). Then, using thus-computed coefficients \( c_n \) for \( n \neq 0 \) and \( a_0 \), we evaluate the solution form of \( c_0 \) by Horner’s method as

\[
c_0 = a_0 + (c_2 - (c_4 - (\cdots (c_{M-2} - c_M) \cdots))), \tag{A.19}
\]

where \( M \equiv 2\lfloor N/2 \rfloor \). This technique further reduces the accumulation of round-off errors. Table A.10 lists a double precision Fortran subroutine to do the transformation. We used its quadruple precision extension in deriving the tables in the main text.
Table A.10: Double precision Fortran subroutine to transform the power series coefficients, \( \{a_n\} \), into the Chebyshev polynomial coefficients, \( \{c_n\} \).

```fortran
subroutine poly2cheb(n,a,c)
  integer JMAX,NMAX
  parameter (JMAX=15,NMAX=JMAX*2)
  integer n,m,j
  real*8 a(0:n),c(0:n),B(0:NMAX,0:JMAX),cm
  logical first/.TRUE./
  save first,B
  if(first) then
    first=.FALSE.;B(0,0)=1.d0;B(1,0)=1.d0
    do m=2,NMAX
      B(m,0)=0.5d0*B(m-1,0)
      do j=1,(m-1)/2
        B(m,j)=0.5d0*(B(m-1,j-1)+B(m-1,j))
      enddo
      if(m.eq.m/2*2) then
        B(m,m/2)=B(m-1,m/2-1)
      endif
    enddo
  endif
  do m=1,n
    cm=0.d0
    do j=(n-m)/2,0,-1
      cm=cm+B(m+2*j,j)*a(m+2*j)
    enddo
    c(m)=c(m)
  enddo
  cm=c(n/2*2)
  do j=n/2-1,1,-1
    cm=cm-c(2*j)-cm
  enddo
  c(0)=a(0)+cm
  return;end
```

23
References


