IIPBF: a MATLAB toolbox for infinite integrals of product of Bessel functions

J. TILAK RATNANATHER, Johns Hopkins University
JUNG H. KIM, Johns Hopkins University
SI RONG ZHANG, Beihang University
ANTHONY M. J. DAVIS, University of California San Diego
STEPHEN K. LUCAS, James Madison University

A MATLAB toolbox, IIPBF, for calculating infinite integrals involving a product of two Bessel functions $J_a(\rho x)J_b(\tau x)$, $J_a(\rho x)Y_b(\tau x)$ and $Y_a(\rho x)Y_b(\tau x)$, for non-negative integers $a, b$, and a well behaved function $f(x)$, is described. Based on the Lucas algorithm previously developed for $J_a(\rho x)J_b(\tau x)$ only, IIPBF recasts each product as the sum of two functions whose oscillatory behavior is exploited in the three step procedure of adaptive integration, summation and extrapolation. The toolbox uses customised QUADPACK and IMSL functions from a MATLAB conversion of the SLATEC library. In addition, MATLAB’s own quadgk function for adaptive Gauss-Kronrod quadrature results in a significant speed up compared with the original algorithm. Usage of IIPBF is described and thirteen test cases illustrate the robustness of the toolbox. The first five of these and five additional cases are used to compare IIPBF with the BESSELINT code for rational and exponential forms of $f(x)$ with $J_a(\rho x)J_b(\tau x)$. An appendix shows a novel derivation of formulae for three cases.

Categories and Subject Descriptors: G.1.4 [Numerical Analysis]: Quadrature and Numerical Differentiation; G.4 [Mathematics of Computing]: Mathematical Software

General Terms: Bessel Function, Quadrature

Additional Key Words and Phrases: extrapolation, adaptive quadrature, summation

ACM Reference Format:
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Lucas [1995] developed an algorithm for computing infinite integrals of products of Bessel functions of the first kind. Specifically, for given a well-behaved function $f(x)$, two non-negative integers $a, b$ and two positive real constants $\rho, \tau$, the algorithm computes

$$\int_0^\infty f(x)B_{a,b,\rho,\tau}(x)\,dx$$

where $B_{a,b,\rho,\tau}(x) = J_a(\rho x)J_b(\tau x)$. Available as a stand alone FORTRAN77 package containing functions from IMSL and QUADPACK libraries, the algorithm has been used in elasticity, electrodynamics, fluid dynamics, biophysics and geo-

Authors’ addresses: J. T. Ratnanather, Center for Imaging Science and Institute for Computational Medicine, Whitaker Biomedical Engineering Institute, The Johns Hopkins University, 3400 N. Charles St, Baltimore, MD 21218-2826; J. H. Kim, Dept. of Applied Mathematics and Statistics, The Johns Hopkins University, Baltimore, MD 21218; S. Zhang, School of Mathematical and System Sciences, Beihang University, Beijing, 100191, China; A. M. J. Davis, Dept. of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093-0411; S. K. Lucas, Dept. of Mathematics and Statistics, James Madison University, Harrisonburg, VA 22807.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© YYYY ACM 0098-3500/YYY/01-A RTA $10.00
DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

three types of endpoints with the $mW$ transform for the infinite integral involving frequency components respectively given by a modification of the Lucas algorithm in which the zeros of the recent availability of MATLAB be a valuable computational science tool. This can be achieved by taking advantage of the decomposition of the general Lucas algorithm for dealing with zeros of oscillatory components and acceleration of the summation of the integrals, and methods for finding or approximating the zeros of $h_1$ and $h_2$. Lucas [1995] obtained results for $J_n(x)J_0(x)$ approaching machine precision with less then 1000 function evaluations compared with a few digits of accuracy for up to 15000 evaluations with the IMSL routines.

The increasing usage of MATLAB suggests that a version of the Lucas algorithm would be a valuable computational science tool. This can be achieved by taking advantage of the recent availability of MATLAB versions of IMSL and QUADPACK functions via a conversion of the SLATEC library [Barrowes 2009]. Further, the algorithm has not previously been adapted for combinations of Bessel functions of the first and second kind.

In what follows, first the development of a MATLAB toolbox, IIPBF, is described for three types of $B_{a,b,\rho,\tau}$ i.e. $J_a(\rho x)J_b(\tau x)$. Next a modification of the general Lucas algorithm for dealing with zeros of $h_2$ for the $\varepsilon$-algorithm and the use of MATLAB's own adaptive Gauss-Kronrod quadrature algorithm (quadqk) [Shampine 2008] are described. Results for several test cases are presented. The discussion concludes the paper.

2. METHOD

This section describes the decomposition of $B_{a,b,\rho,\tau}$ as the sum of high and low frequency components respectively given by $h_1$ and $h_2$. This is followed by a summary of a modification of the Lucas algorithm in which the zeros of $h_1$ are used as integral endpoints with the $mW$ transform for the infinite integral involving $h_1$, and the zeros for $h_2$ estimated stepwise are used for the $\varepsilon$-algorithm.

2.1. Product decomposition

First $J_a(\rho x)J_b(\tau x)$ is expressed as $h_1(x;a, b, \rho, \tau) + h_2(x; a, b, \rho, \tau)$ where

$$h_1(x;a, b, \rho, \tau) = \frac{1}{2}(J_a(\rho x)J_b(\tau x) - Y_a(\rho x)Y_b(\tau x)),$$

$$h_2(x;a, b, \rho, \tau) = \frac{1}{2}(J_a(\rho x)J_b(\tau x) + Y_a(\rho x)Y_b(\tau x))$$

with the asymptotic behaviour for large $x$ [Lucas 1995]

$$h_1(x;a, b, \rho, \tau) \sim \frac{1}{\pi \sqrt{\rho \tau x}} \cos \left( (\rho + \tau)x - \frac{(a + b + 1)\pi}{2} \right),$$

$$h_2(x;a, b, \rho, \tau) \sim \frac{1}{\pi \sqrt{\rho \tau x}} \cos \left( (\rho - \tau)x - \frac{(a - b)\pi}{2} \right).$$

(1)
The ISE method is applied to infinite integrals of product of Bessel functions

\[ f(x) = J_a(\rho x)Y_b(\tau x) + J_b(\rho x)Y_a(\tau x) \]

whose asymptotic behavior is

\[
\begin{align*}
    h_1(x; a, b, \rho, \tau) &= \frac{1}{\pi} \sin \left( (\rho + \tau)x - \frac{(a + b + 1)\pi}{2} \right) \\
    h_2(x; a, b, \rho, \tau) &= \frac{1}{\pi} \sin \left( (\rho - \tau)x - \frac{(a - b)\pi}{2} \right)
\end{align*}
\]

\[ (2) \]

Third \( Y_a(\rho x)Y_b(\tau x) \) is expressed as \( h_1(x; a, b, \rho, \tau) + h_2(x; a, b, \rho, \tau) \) where

\[
\begin{align*}
    h_1(x; a, b, \rho, \tau) &= \frac{1}{2} (J_a(\rho x)J_b(\tau x) - Y_a(\rho x)Y_b(\tau x)), \\
    h_2(x; a, b, \rho, \tau) &= \frac{1}{2} (J_a(\rho x)J_b(\tau x) + Y_a(\rho x)Y_b(\tau x))
\end{align*}
\]

whose asymptotic behaviour is

\[
\begin{align*}
    h_1(x; a, b, \rho, \tau) &\sim -\frac{1}{\pi \sqrt{\rho^2 x}} \cos \left( (\rho + \tau)x - \frac{(a + b + 1)\pi}{2} \right) \\
    h_2(x; a, b, \rho, \tau) &\sim \frac{1}{\pi \sqrt{\rho^2 x}} \cos \left( (\rho - \tau)x - \frac{(a - b)\pi}{2} \right)
\end{align*}
\]

\[ (3) \]

2.2. Calculating zeros of \( h_1 \) an \( h_2 \)

The ISE method is applied to \( f(x)h_1(x) \) and \( f(x)h_2(x) \). While the zeros of \( h_1 \) are easy to find, those of \( h_2 \) can be difficult to compute. Let \( h_{1,1} \) and \( h_{2,1} \) denote the first zeros of \( h_1 \) \( x \) and \( h_2 \) \( x \) respectively.

When \( \rho = \tau \), the integration domain is split with \([0, \gamma_{\text{max}}]\) for \( f(x)B_{a,b,\rho,\tau}(x) \) and \([\gamma_{\text{max}}, \infty]\) for \( f(x)h_1(x) + f(x)h_2(x) \). From Eqs. 1-3, \( h_2(x) \) is a non-oscillating monotonically decreasing function, extrapolation is used for \( h_1(x) \) and thus the zeros of \( h_2(x) \) are not needed. \( \gamma_{\text{max}} \) is calculated as the first zero of \( h_1(x) \) beyond which point \( Y_a(\rho x) \) no longer dominates \( h_1(x) \) [Olver 1960]. Subsequent zeros for the mW transform [Sidi 1988] are computed using the asymptotic approximations from Equations 1-3.

When \( \rho \neq \tau \), the estimate for the zeros of \( Y_a(x) \) [Olver 1960] are used initially to establish \( h_{1,1} \) and \( h_{2,1} \). The domain \([0, \infty]\) is split into \([0, \gamma_{\text{min}}]\), \([\gamma_{\text{min}}, \gamma_{\text{max}}]\) and \([\gamma_{\text{max}}, \infty]\) with \( \gamma_{\text{min}} = \min(h_{1,1}, h_{2,1}) \) and \( \gamma_{\text{max}} = \max(h_{1,1}, h_{2,1}) \). Subsequent zeros of \( h_1(x) \) for the mW transform are computed using the asymptotic approximations from Equations 1-3, while zeros of \( h_2(x) \) after \( h_{2,1} \) for the z-algorithm are obtained via step-wise increments of \( \pi/|\rho - \tau| \) [Lucas 1995].

2.3. The IIPBF Toolbox

By adding (a) to QUADPACK function names, IIPBF algorithm for \( 0 < a, b < 1000 \), \( \rho, \tau \in \mathbb{R} \) and \( \text{type} \in \{1, 2, 3\} \) for one of the three forms of \( B_{a,b,\rho,\tau} \) is described in algorithm 1.

The functions dqagiea, dqagea and dqelga were adapted from the MATLAB version of the SLATEC library [Barrowes 2009]. In particular, dqagea and dqagiea use the recently developed adaptive Gauss-Kronrod quadrature algorithm (quadgk) [Shampine 2008] now available in MATLAB instead of dqk15 and dqk15i respectively; dqelga is called via a wrapper epsalg. The converted SLATEC routines [Barrowes 2009] together with the original mWtrans algorithm were customised to pass a second function to deal with...
ALGORITHM 1: IIPBF Algorithm

Input: type, a, b, ρ, τ, abserr (absolute error tolerance), relerr (relative error tolerance)
Output: I_{a,b,ρ,τ}, neval (number of evaluations of h_1 and h_2), esterr (estimated relative error)

if ρ = τ then
   Calculate h_1,1 with abserr/3 and relerr/3;
   \[ I_1 = \int_{h_{1,1}} f(x)B_{a,b,ρ,τ}(x)dx \text{ using dqagea}; \]
   \[ I_2 = \int_{h_{1,1}} f(x)h_2(x;a,b,ρ,τ)dx \text{ using dqagiea}; \]
   \[ I_3 = \int_{h_{1,1}} f(x)h_1(x;a,b,ρ,τ)dx \text{ using mWtrans}; \]
   \[ I_4 = 0 \]
else
   Calculate h_{1,1} and h_{2,1} with abserr/4 and relerr/4;
   \[ I_2 = \int_{h_{2,1}} f(x)h_2(x;a,b,ρ,τ)dx \text{ using mWtrans}; \]
   \[ I_3 = \int_{h_{2,1}} f(x)h_2(x;a,b,ρ,τ)dx \text{ using epsalg}; \]
   if h_{1,1} < h_{2,1} then
      \[ I_1 = \int_{h_{1,1}} f(x)B_{a,b,ρ,τ}(x)dx \text{ using dqagea}; \]
      \[ I_4 = \int_{h_{1,1}} f(x)h_2(x;a,b,ρ,τ)dx \text{ using dqagea}; \]
   else
      \[ I_1 = \int_{h_{2,1}} f(x)B_{a,b,ρ,τ}(x)dx \text{ using dqagea}; \]
      \[ I_4 = \int_{h_{2,1}} f(x)h_1(x;a,b,ρ,τ)dx \text{ using dqagea}; \]
end
end

I_{a,b,ρ,τ} = I_1 + I_2 + I_3 + I_4

the decomposition of f(x)B_{a,b,ρ,τ}(x). Available at http://www.cis.jhu.edu/software/iipbf, the toolbox is run as follows:

[result,est_error,nevals]=IIPBF(f,rho,tau,a,b,abserr,relerr,type);

with input parameters:

- f = user-defined function: f(x)
- rho = ρ
- tau = τ
- a, b = non-negative integers
- abserr = absolute error tolerance
- relerr = relative error tolerance
- type = product type

and output:

- result = computed integral
- esterr = estimated relative error
- neval = number of function evaluations for esterr to fall below tolerance

3. RESULTS

Table I lists the cases that were used to test IIPBF. The first three cases are from Lucas [1995] and the remaining ones are simplified forms of those being calculated with the toolbox in a separate study. While evidently faster algorithms exist for simpler integrals, it is important to be able to demonstrate feasibility of IIPBF. Fig. 1 and 2 plots actual error and estimated relative error with respect to the number of evaluations of h_1 and h_2 for requested relative error tolerances from 10^{-4} to 10^{-14} in 10 decrements of 0.1.
Infinite integrals of product of Bessel functions

Table I. Test cases for $f(x)B_{\nu,\rho,\tau}(x)$ with parameters $u, \rho, \tau \in \mathbb{Z}^+$.  

<table>
<thead>
<tr>
<th>Case</th>
<th>$f(x)B_{\nu,\rho,\tau}(x)$</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$J_0(x)J_1(3x/2)$</td>
<td>$2/3$</td>
</tr>
<tr>
<td>2</td>
<td>$J_0(x)J_0(2x)/x^2$</td>
<td>$27/4096$</td>
</tr>
<tr>
<td>3</td>
<td>$xJ_0(x)J_1(x)J_0(1.1x)/(1 + x^2)$</td>
<td>$\approx -6.650747903049 \times 10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>$J_0(x)J_1(x)/x$</td>
<td>$2\pi$</td>
</tr>
<tr>
<td>5</td>
<td>$J_1(x)J_1(x)/x^2$</td>
<td>$4/3\pi$</td>
</tr>
<tr>
<td>6</td>
<td>$xK_0(x)J_0(x)J_0(\tau x)$</td>
<td>$1/\left(\left(u^2 + \rho^2 + \tau^2\right)^2 - 4\rho^2\tau^2\right)^{1/2}$</td>
</tr>
<tr>
<td>7</td>
<td>$x^2K_1(x)J_0(x)J_1(\tau x)$</td>
<td>$4u\rho/\left(\left(u^2 + \rho^2 + \tau^2\right)^2 - 4\rho^2\tau^2\right)^{3/2}$</td>
</tr>
<tr>
<td>8</td>
<td>$e^{-2ux}J_0(x)J_0(x)$</td>
<td>$(1/\pi) \int_0^{\pi/2} (1 + u^2 \cos^2 z)^{-1/2} dz$</td>
</tr>
<tr>
<td>9</td>
<td>$xe^{-x^2/u}J_2(x)$</td>
<td>$\frac{4}{\pi} - 2 + \frac{uK_2(u/2)}{2\pi e^{u^2/4}} + \frac{2\pi e^{u^2/4}}{u(8 + 4u + u^2)K_1(u/2)}$</td>
</tr>
<tr>
<td>10</td>
<td>$x^3e^{-x^2/u}J_2(x)Y_2(x)$</td>
<td>$-\frac{4}{\pi} + \frac{u^2(2 + u)K_0(u/2)}{4\pi e^{u^2/4}}$</td>
</tr>
<tr>
<td>11</td>
<td>$e^{-ux}Y_0(\rho x)Y_0(\tau x)$</td>
<td>$\frac{2}{\pi} \int_0^{\rho} \frac{G(y)dy}{\sqrt{y^2 - \rho^2}}$ where $\rho &gt; \tau$ and $\sqrt{\rho^2 - y^2} = \sqrt{u^2 + \rho^2 - y^2}$, $\tau &lt; \rho$</td>
</tr>
<tr>
<td>12</td>
<td>$J_0(\rho x)J_0(\tau x)$</td>
<td>$\frac{(2\pi \rho)K(\tau/\rho)}{\sqrt{\rho^2 + u^2}}$</td>
</tr>
<tr>
<td>13</td>
<td>$Y_0(\rho x)Y_0(\tau x)$</td>
<td>$\frac{(2\pi \rho)K(\tau/\rho)}{\sqrt{\rho^2 + u^2}}$</td>
</tr>
</tbody>
</table>

Formulae 4-8 were obtained from Gradshteyn & Ryzhik [2007]; those for cases 9-10 were obtained from Adamchik [1995]; for cases 11-13 see Appendix A for derivation of formulae involving the elliptic function $K$ that are simpler than those by Glasser [1974].

Table II. Additional test cases for comparison of BESSELINT and I1PBF.  

<table>
<thead>
<tr>
<th>Case</th>
<th>$f(x)B_{\nu,\rho,\tau}(x)$</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>$e^{-2ux}xJ_0(x)J_1(x)$</td>
<td>$\frac{2\pi \rho \sqrt{\rho^2 + u^2}}{\sqrt{\rho^2 + u^2}}$</td>
</tr>
<tr>
<td>15</td>
<td>$e^{-2ux}[J_0(x)]^2$</td>
<td>$\pi \sqrt{\rho^2 + u^2}$</td>
</tr>
<tr>
<td>16</td>
<td>$e^{-2ux}[J_1(x)]^2$</td>
<td>$(2\rho^2 + \rho^2)K - 2(u^2 + \rho^2)E$</td>
</tr>
<tr>
<td>17</td>
<td>$e^{-2ux}[J_0(x)]^2$</td>
<td>$\frac{\pi \rho^2 \sqrt{\rho^2 + u^2}}{\cos^2 \theta}$</td>
</tr>
<tr>
<td>18</td>
<td>$\frac{x}{u^2 + x^2}J_0(\rho x)J_0(\tau x)$</td>
<td>$\frac{1}{\pi} \int_0^{\pi/2} \sqrt{u^2 + \cos^2 z} dz$</td>
</tr>
</tbody>
</table>

MATLAB functions for $K$ and $E$ which are the complete elliptic integrals of the first and second kind, respectively, of modulus $\rho/\sqrt{\rho^2 + u^2}$ where $u \in \mathbb{Z}^+$ were used; formulae were taken from Gradshtein & Ryzhik [2007].

Fig. 1. Comparing actual error to number of function evaluations of $h_1$ and $h_2$ with requested relative error tolerances from $10^{-4}$ to $10^{-14}$ in 10 decrements of 0.1 for the test cases in table I. In cases 6 and 7, parameters $\{u, \rho, \tau\} = \{1, 2, 1\}$. For cases 8, 9 and 10 $u = 1.5$, 0.2 and 2 respectively. For case 11, $\{u, \rho, \tau\} = \{0.1, 2, 1\}$. For cases 12 and 13, $\{\rho, \tau\} = \{3, 1\}$.

An alternative to IIBPF is BESSELINT [Van Deun and Cools 2006a; 2008; 2006b] for the integrand $f(x)\Pi_{i=1}^{n}J_{a_i}(\rho_i x)$ where $f(x) = x^{s}e^{-ux}/(t^2 + x^2)$. Briefly, the algorithm uses asymptotic expansions for $J_{a_i}$ and the incomplete Gamma function [Van Deun and Cools 2006c] to approximate the infinite part of the integral. Therefore it is appropriate to compare the performance and reliability of IIBPF with BESSELINT for cases 1-5 in table I and five additional cases in table II. The tests were run with MATLAB version 7.9.0.529 (R2009b) on Intel Core2 Duo CPU L6700 with 2.66GHz. Table III lists runtime execution, absolute difference between computed values, and estimated relative errors for a requested relative error tolerance of $50\varepsilon$. Clearly while both BESSELINT and IIBPF give similar values with similar relative errors, the former is faster by a factor of 4-50. Further, MATLAB’s profiler tool was used to identify which parts of the code consumed most of the CPU processing. Other than calling besselj [Van Deun...
Fig. 2. Comparing estimated error to number of function evaluations of $h_1$ and $h_2$ with requested relative error tolerances from $10^{-4}$ to $10^{-14}$ in 10 decrements of 0.1 for the test cases in table I. In cases 6 and 7, parameters $(u, \rho, \tau) = (1, 2, 1)$. For cases 8, 9 and 10 $u = 1.5, 0.2$ and 2 respectively. For case 11, $(u, \rho, \tau) = (0.1, 2, 1)$. For cases 12 and 13, $(\rho, \tau) = (3, 1)$.

and Cools 2006b], the bottleneck in the respective codes were `ira` and `mtrans` which both deal with the infinite range approximation. However the latter consumed more due to the nature of its implementation.

### 4. DISCUSSION

A MATLAB toolbox, IIPBF, for infinite integration of products of Bessel functions of the first and second kind has been developed and tested.

A key component of IIPBF is the use of adaptive quadrature algorithms. Specifically using `quadgk` in `dqagea` and `dqagiea` enabled error estimates to be derived robustly with significantly less function evaluations than in the original Lucas algorithm by nearly two orders of magnitude for cases 1-3; indeed, the number of function evaluations in an early version of IIPBF were about the same as those by Lucas [1995].
Table III. Comparison of \textsc{Besselint} and \textsc{Iipbf}

<table>
<thead>
<tr>
<th>Case</th>
<th>BESSELINT Time (secs)</th>
<th>IIPBF Estimated Error</th>
<th>Absolute Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1189</td>
<td>0.6542 3.95x10^{-16}</td>
<td>1.11x10^{-16}</td>
</tr>
<tr>
<td>2</td>
<td>0.0351</td>
<td>0.6454 8.38x10^{-16}</td>
<td>1.73x10^{-17}</td>
</tr>
<tr>
<td>3</td>
<td>0.1830</td>
<td>0.7743 1.51x10^{-15}</td>
<td>5.47x10^{-17}</td>
</tr>
<tr>
<td>4</td>
<td>0.0163</td>
<td>0.5725 2.95x10^{-15}</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.0140</td>
<td>0.5838 1.80x10^{-15}</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0.0992</td>
<td>0.5347 7.51x10^{-16}</td>
<td>6.94x10^{-18}</td>
</tr>
<tr>
<td>15</td>
<td>0.0088</td>
<td>0.5283 2.06x10^{-15}</td>
<td>2.78x10^{-16}</td>
</tr>
<tr>
<td>16</td>
<td>0.0176</td>
<td>0.5407 8.32x10^{-16}</td>
<td>2.17x10^{-17}</td>
</tr>
<tr>
<td>17</td>
<td>0.0106</td>
<td>0.5447 1.59x10^{-16}</td>
<td>8.67x10^{-19}</td>
</tr>
<tr>
<td>18</td>
<td>0.0385</td>
<td>0.6564 4.13x10^{-13}</td>
<td>2.40x10^{-15}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.54x10^{-16}</td>
</tr>
</tbody>
</table>

The last column is the absolute difference between the computed values.

Generally, \textsc{quadgk} performs the adaptive subdivision locally and conservatively, is able to process subintervals simultaneously, samples many points within the subintervals and is therefore faster and more reliable than its predecessor [Shampine 2008; Gander and Gautschi 2001; Gonnet 2009]. In addition, \textsc{dqagiea} exploits the ability of \textsc{quadgk} to handle infinite integrals but if convergence is slow, the $\varepsilon$-algorithm is used instead.

Two codes for computing infinite integrals of products of Bessel functions of the first kind have emerged in recent years. One is a \textsc{fortran} code for products of Bessel functions of order 0 or 1 and $f(x) = 1$ or $x$ [Singh and Mogi 2005]. The other is \textsc{Besselint} [Van Deun and Cools 2006a; 2006b] with modifications for rational and exponential $f(x)$ [Van Deun and Cools 2008]. When comparing \textsc{Besselint} and \textsc{IIPBF}, the latter can be used for any well-behaved $f(x)$ while the rational exponential form of $f(x)$ in the former is used to facilitate stable quadrature over a converging series in the asymptotic part. Quadrature in both cases are adaptive however \textsc{Besselint} uses hard-coded Gauss-Legendre with 15 points followed 19 points over sub-intervals determined approximately by the zeros of the integrand. The singularity near $x = 0$ is dealt with internally in the \textsc{quadgk} function for weak logarithmic and algebraic forms via a simple transform whereas extrapolation is used in \textsc{Besselint}. The breakpoint beyond which asymptotic properties are used is determined by the first zero of $h_1$ and $h_2$ in \textsc{IIPBF} whereas in \textsc{Besselint} it is automatically determined via the incomplete gamma function.

Since \textsc{IIPBF} is as accurate as \textsc{Besselint}, it will have wider applicability as it would be more laborious to derive asymptotic forms of more general $f(x)$. Eventually, it is possible that \textsc{IIPBF} will be superseded by combining the best features of both codes in which case Hankel functions may be useful [Huybrechs and Vandewalle 2006] in developing the procedure for $Y_{\alpha_i}(\rho_i x)$.
A. FORMULAE FOR CASES 11-13

This appendix describes the derivation of expressions for integrals in cases 11-13 in table I that are found to be simplifications of those obtained by Glasser [1974].

The elliptic integral $K(k)$ is defined by

$$K(k) = \int_0^1 \frac{dx}{\sqrt{(1-x^2)(1-k^2x^2)}} = \int_1^\infty \frac{dX}{\sqrt{(X^2-1)(X^2-k^2)}} \quad (k^2 < 1).$$

An alternative form, whose integrand has a strictly positive denominator, is

$$K(k) = \int_0^\infty \frac{dt}{\sqrt{2(1-k^2)\cosh t + 2(1+k^2)}}.$$

Since

$$\int_0^\infty Y_0(\tau x) \cos xy \, dx = -\frac{H(y-\tau)}{\sqrt{y^2-\tau^2}} = -\int_0^\infty J_0(\tau x) \sin xy \, dx,$$

where $H(u)$ denotes the Heaviside unit function, we may use the Parseval equations for cosine and sine transforms to show that

$$\int_0^\infty Y_0(\tau x)Y_0(\rho x) \, dx = \int_0^\infty J_0(\tau x)J_0(\rho x) \, dx = \frac{2}{\pi\rho}K(\tau/\rho) \quad (\rho > \tau),$$

which is demonstrated below to be a simpler form of Eq. (10) in Glasser [1974].

The above use of cosine transforms can be extended to show that

$$\int_0^\infty e^{-ux}Y_0(\tau x)Y_0(\rho x) \, dx = -\frac{2}{\pi} \int_\rho^\infty \frac{G(y) \, dy}{\sqrt{y^2-\rho^2}} \quad (\rho > \tau),$$

where

$$G(y) = \int_0^\infty e^{-ux}Y_0(\tau x) \cos xy \, dx. \quad (u, \tau, y > 0).$$

The formula,

$$\int_0^\infty e^{-ux}Y_0(\tau x) \, dx = -\frac{2}{\pi\sqrt{u^2 + \tau^2}} \ln \left[ \frac{\sqrt{u^2 + \tau^2} + u}{\tau} \right],$$

enables $G(y)$ to be evaluated by replacing $u$ by $u - iy$ and taking the real part of the result. If $\sqrt{(u+iy)^2 + \tau^2} = \alpha_1 + i\alpha_2$, then $u, y, \tau > 0$ implies $\alpha_1, \alpha_2 > 0$ where

$$\alpha_1, \alpha_2 = \frac{1}{\sqrt{2}} \left[ \sqrt{(u^2 + \tau^2 - y^2)^2 + 4u^2y^2} \pm (u^2 + \tau^2 - y^2) \right]^{1/2}.$$

After some algebra, it is found that

$$G(y) = -\frac{2}{\pi(\alpha_1^2 + \alpha_2^2)} \left[ \alpha_1 \ln \left( \frac{(\alpha_1 + u)^2 + (\alpha_2 + y)^2}{b} \right) + \alpha_2 \arctan \left( \frac{\alpha_2 + y}{\alpha_1 + u} \right) \right].$$

Hence $G(y) = O(y^{-2} \ln y)$ as $y \to \infty$. The known expressions for $G$ when $u = 0$ or $y = 0$ are recovered.

Consistency with most of Glasser’s formulae is demonstrated as follows. Since

$$\int_0^\infty K_0(\rho x) \cos xy \, dx = \frac{\pi}{2\sqrt{\rho^2 + y^2}},$$

the Parseval formula gives
\[
\int_0^\infty Y_0(\tau x)K_0(\rho x)\,dx = -\int_\tau^\infty \frac{dy}{\sqrt{(y^2 + \rho^2)(y^2 - \tau^2)}} = -\frac{1}{\sqrt{\rho^2 + \tau^2}}K\left(\frac{\rho}{\sqrt{\rho^2 + \tau^2}}\right),
\]
after setting \(y^2 = (\rho^2 + \tau^2)X^2 - \rho^2\). This agrees with the first of Glasser's Eqs (8) and confirms that his \(K\) denotes the elliptic integral \(K\). Similarly, with \(\rho > \tau\),
\[
\int_0^\infty K_0(\tau x)K_0(\rho x)\,dx = \frac{\pi}{2} \int_\tau^\infty \frac{dy}{\sqrt{(y^2 + \rho^2)(y^2 + \tau^2)}} = \frac{\pi}{2\rho} K\left(\sqrt{\frac{1 - \rho^2}{\rho^2}}\right),
\]
after setting \(y^2 = \rho^2X^2 - \rho^2\). This appears to contradict the second of Glasser's Eq.(8) but the hypergeometric function identity,
\[
\begin{align*}
F(\alpha, \alpha - \beta + 1/2; \beta + 1/2, z^2) &= (1 + z)^{-2\alpha} F\left[\alpha, \beta; \frac{4z}{(1 + z)^2}\right], \\
K(k) &= \frac{\pi}{2} F\left[1/2, 1/2; 1, \frac{4k}{(1 + k)^2}\right] = \frac{1}{1 + k} K\left(\frac{2\sqrt{k}}{1 + k}\right).
\end{align*}
\]
yields, with \(\alpha = \beta = 1/2\) and \(k < 1\),
\[
K(k) = \frac{\pi}{2} F\left[1/2, 1/2; 1, k^2\right] = \frac{\pi}{2(1 + k)} F\left[1/2, 1/2; 1, \frac{4k}{(1 + k)^2}\right] = \frac{1}{1 + k} K\left(\frac{2\sqrt{k}}{1 + k}\right).
\]
In particular,
\[
K\left(\frac{\rho - \tau}{\rho + \tau}\right) = \frac{\rho + \tau}{2\rho} K\left(\sqrt{1 - \frac{\tau^2}{\rho^2}}\right) \quad (\rho > \tau),
\]
\[
K\left(\frac{\tau}{\rho}\right) = \frac{\rho - \tau}{\rho + \tau} K\left(\frac{2\sqrt{\rho\tau}}{\rho + \tau}\right) \quad (\rho > \tau),
\]
which shows the equivalence of Glasser’s second Eq. (8) and Eq. (10) and the above concise formulae. Since Glasser’s Eq. (9) is used only to obtain Eq. (10), there is no need to be concerned with its derivation.
REFERENCES


