A

Algorithm xxx: IIPBF, a MATLAB toolbox for infinite integral of products of two Bessel functions

J. TILAK RATNANATHER, Johns Hopkins University
JUNG H. KIM, Johns Hopkins University
SIRONG 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 eighteen test cases illustrate the robustness of the toolbox; five additional ones are used to compare IIPBF with the BESSELINT code for rational and exponential forms of $f(x)$ with $J_a(\rho x)J_b(\tau x)$. Reliability for a broad range of values of ρ and τ for the three different product types as well as different orders in one case is demonstrated. An appendix shows a novel derivation of formulae for five 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:
Ratnanather, J. T., Kim, J. H., Zhang, S., Davis, A. M. J., and Lucas, S. K. 2012. Algorithm xxx: IIPBF, a MATLAB toolbox for infinite integral of products of two Bessel functions. ACM Trans. Math. Softw. V, N, Article A (January YYYY), 16 pages.
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 two 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 ρ, τ, 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

---

Authors' addresses: J. T. Ratnanather, Center for Imaging Science and Institute for Computational Medicine, Whitaker Biomedical Engineering Institute, The Johns Hopkins University, Clark 301, 3400 N. Charles St, Baltimore, MD 21218-2826. email: tilak@cis.jhu.edu; 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.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

alone FORTRAN77 package containing functions from IMSL and QUADPACK libraries, the algorithm has been used in elasticity, electrodynamics, fluid dynamics, biophysics and geophysics to name but a few applications [Craster 1998; Davis and Stone 1998; Pan et al. 2007; Petrov and Schwille 2008; Robinson 2002; Salo et al. 2006; Sherwood 2005; Tartakovsky et al. 2000; Xu et al. 2003]. Briefly, the Lucas algorithm rewrites the product, $B_{a,b,\rho,\tau}$, as the sum of a high frequency function h_1 and a low frequency function h_2 whose oscillatory behaviour s are exploited in a three step procedure of (i) Integration, (ii) Summation and (iii) Extrapolation (ISE). ISE was first adopted for integrands involving $J_n(x)$ whose zeros form the subdivision of the range giving rise to a summation of integrals of alternating signs, convergence of which is accelerated via an extrapolation technique [Lucas and Stone 1995]. The decomposition of the product as the sum of two functions results in applying the ISE method twice. The algorithm used modifications of adaptive IMSL routines dqdage and dqdagie for finite and infinite adaptive integration with Gaussian quadrature respectively; further it used mWtrans as the mW transform [Sidi 1988] and dqelg in QUADPACK [Piessens et al. 1983] as the ε-algorithm to deal with h_1 and h_2 respectively 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 $J_a(\rho x)J_b(\tau x)$ 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 which recently arose in studies of a submerged viscous fluid jet impinging on a planar wall [Davis et al. 2012; Davis et al. 2013]. Ideally, $f(x)$ should be bounded i.e. not be dominated by $B_{a,b,\rho,\tau}(x)$ for $x \gg 1$ [Ikonomou et al. 1995; Sidi 2012].

In what follows, first the development of a MATLAB toolbox, IIPBF, is described for three types of $B_{a,b,\rho,\tau}$ i.e. $J_a(\rho x)J_b(\tau x)$, $J_a(\rho x)Y_b(\tau x)$ or $Y_a(\rho x)Y_b(\tau x)$. Next a modification of the general Lucas algorithm for dealing with zeros of h_1 for the mW transform and the use of MATLAB's own adaptive Gauss-Kronrod quadrature algorithm (quadgk) [Shampine 2008] are described. Results for several cases are presented. The discussion concludes the paper.

2. METHOD

This section describes the decomposition of $B_{a,b,\rho,\tau}$ as $h_1 + h_2$ whose asymptotic properties as trigonometric functions can be exploited. 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 ε-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

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

with the asymptotic behaviour for large x [Lucas 1995]

$$\left.\begin{aligned} h_1(x;a,b,\rho,\tau) &\sim \frac{1}{\pi\sqrt{\rho\tau}x}\cos\left((\rho+\tau)x-\frac{(a+b+1)\pi}{2}\right) \\ h_2(x;a,b,\rho,\tau) &\sim \frac{1}{\pi\sqrt{\rho\tau}x}\cos\left((\rho-\tau)x-\frac{(a-b)\pi}{2}\right) \end{aligned}\right\}. \tag{1}$$

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

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

whose asymptotic behavior is

$$\left.\begin{aligned} h_1(x;a,b,\rho,\tau) &\sim \frac{1}{\pi\sqrt{\rho\tau}x}\sin\left((\rho+\tau)x-\frac{(a+b+1)\pi}{2}\right) \\ h_2(x;a,b,\rho,\tau) &\sim -\frac{1}{\pi\sqrt{\rho\tau}x}\sin\left((\rho-\tau)x-\frac{(a-b)\pi}{2}\right) \end{aligned}\right\}. \tag{2}$$

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

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

whose asymptotic behaviour is

$$\left.\begin{aligned} h_1(x;a,b,\rho,\tau) &\sim -\frac{1}{\pi\sqrt{\rho\tau}x}\cos\left((\rho+\tau)x-\frac{(a+b+1)\pi}{2}\right) \\ h_2(x;a,b,\rho,\tau) &\sim \frac{1}{\pi\sqrt{\rho\tau}x}\cos\left((\rho-\tau)x-\frac{(a-b)\pi}{2}\right) \end{aligned}\right\}. \tag{3}$$

2.2. Calculating zeros of h_1 and 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. Since $Y_a(\rho x)$ and $Y_b(\tau x)$ are singular near 0, Lucas [1995] recommended that these zeros be approximated by the first zeros of Y_a and Y_b [Olver et al. 2010, Equation 10.21.40] beyond which Y_a and Y_b no longer dominate $h_1(x)$ and $h_2(x)$.

When $\rho=\tau$, the domain $[0,\infty)$ is split into $[0,ymax]$ and $[ymax,\infty)$ respectively for $f(x)B_{a,b,\rho,\tau}(x)$ and $f(x)h_1(x)+f(x)h_2(x)$. Since $h_2(x)$ is a non-oscillating monotonically decreasing function, its zeros are not needed. For $h_1(x)$, $ymax$ is determined by the first zero of Y_a and Y_b. Subsequent zeros for the mW transform [Sidi 1988] are the zeros from Equations 1-3.

When $\rho\neq\tau$, the domain $[0,\infty)$ is split into $[0,ymin]$, $[ymin,ymax]$ and $[ymax,\infty)$ with $ymin=min(h_{1,1},h_{2,1})$ and $ymax=max(h_{1,1},h_{2,1})$ where $h_{1,1}$ and $h_{2,1}$ are obtained from the first zeros of Y_a and Y_b. Subsequent zeros of $h_1(x)$ for the mW transform are the zeros from Equations 1-3, while zeros of $h_2(x)$ after $h_{2,1}$ for the ε-algorithm are obtained via stepwise increments of $\pi/|\rho-\tau|$ as recommended [Lucas 1995].

ALGORITHM 1: IIPBF Algorithm

Input: type, a, b, ρ, τ, abserr (absolute error tolerance), relerr (relative error tolerance)
Output: $I_{a,b,\rho,\tau}$, neval (number of evaluations of h_1 and h_2), esterr (estimated relative error)
if $|\rho/\tau - 1| < 10^{-10}$ **then**
  Calculate $h_{1,1}$ with abserr$/3$ and relerr$/3$;
  $I_1 = \int_0^{h_{1,1}} f(x)B_{a,b,\rho,\tau}(x)dx$ using dqagea;
  $I_2 = \int_{h_{1,1}}^{\infty} f(x)h_2(x;a,b,\rho,\tau)dx$ using dqagiea;
  $I_3 = \int_{h_{1,1}}^{\infty} f(x)h_1(x;a,b,\rho,\tau)dx$ using mWtrans;
  $I_4 = 0$
else if $|\rho/\tau| > 10^2$ *or* $|\tau/\rho| > 10^2$ **then**
  Calculate $h_{1,1}$ with abserr$/3$ and relerr$/3$;
  $I_1 = \int_0^{h_{1,1}} f(x)B_{a,b,\rho,\tau}(x)dx$ using dqagea;
  $I_2 = \int_{h_{1,1}}^{\infty} f(x)h_2(x;a,b,\rho,\tau)dx$ using epsalg;
  $I_3 = \int_{h_{1,1}}^{\infty} f(x)h_1(x;a,b,\rho,\tau)dx$ using epsalg;
  $I_4 = 0$
else
  Calculate $h_{1,1}$ and $h_{2,1}$ with abserr$/4$ and relerr$/4$;
  $I_2 = \int_{h_{1,1}}^{\infty} f(x)h_1(x;a,b,\rho,\tau)dx$ using mWtrans;
  $I_3 = \int_{h_{2,1}}^{\infty} f(x)h_2(x;a,b,\rho,\tau)dx$ using epsalg;
  if $h_{1,1} < h_{2,1}$ **then**
    $I_1 = \int_0^{h_1} f(x)B_{a,b,\rho,\tau}(x)dx$ using dqagea;
    $I_4 = \int_{h_{1,1}}^{h_{2,1}} f(x)h_2(x;a,b,\rho,\tau)dx$ using dqagea
  else
    $I_1 = \int_0^{h_{2,1}} f(x)B_{a,b,\rho,\tau}(x)dx$ using dqagea;
    $I_4 = \int_{h_{2,1}}^{h_{1,1}} f(x)h_1(x;a,b,\rho,\tau)dx$ using dqagea
  end
end
$I_{a,b,\rho,\tau} = I_1 + I_2 + I_3 + I_4$

2.3. The IIPBF Toolbox

Algorithm 1 describes the modified Lucas algorithm used in IIPBF for $0 \leq a, b \leq 1000$, $\rho, \tau \in \mathbb{R}$ and type = JJ,JY,YY which corresponds to one of the three forms of $B_{a,b,\rho,\tau}$; the adapted QUADPACK functions are indicated by the last letter a.

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. Additional modifications are needed to deal with extreme values of ρ and τ. First, when $|\rho/\tau - 1| < 10^{-10}$, h_2 is almost monotone and thus computed via dqagiea. Second, when $|\rho/\tau| > 10^2$ or $|\tau/\rho| > 10^2$, the asymptotic forms for h_1 and h_2 have almost the same frequency in which case both are computed via the ε-algorithm. Third, small values of ρ or τ can cause the first zeros of $Y_a(\rho x)$ and $Y_b(\tau x)$ to be large which suggests that the minimum of these zeros be used when the ratio of the larger zero to the smaller one is too large; otherwise the maximum is used as recommended by Lucas [1995]. The converted SLATEC routines [Barrowes 2009] together with the original mWtrans algorithm were customised to pass a second function to deal with the decomposition of $f(x)B_{a,b,\rho,\tau}(x)$. 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 eighteen cases that were used to test `IIPBF`. Cases 1-3 are from Lucas [1995], cases 4-16 are simplified forms of those calculated with the toolbox in a separate study [Davis et al. 2012] and cases 17-18 provide tests when $|a-b|$ is not zero or one. While evidently faster algorithms exist for the 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 decrements of 0.1. Table II shows actual errors for cases 11-13 with a wide range of values of ρ and τ based on the requested tolerance of 10^{-10}; note that the errors for cases 11 and 13 are a couple of orders of magnitude higher which may be attributed to `quadgk` having to deal with the left endpoint square root singularity in the integrands. For pure absolute error control, it was necessary to have `RelTol`$= 0$ for `quadgk` inside `dqagea`.

An alternative to `IIBPF` is `BESSELINT` [Van Deun and Cools 2006a; 2006b; 2008] for the integrand $f(x)\Pi_{i=1}^{k} J_{a_i}(\rho_i x)$ 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 `IIPBF` with `BESSELINT` for cases 1-5 in table I and five additional cases in table III. The tests were run with `MATLAB` version 7.14.0.739 (R2012a) on an Intel(R) Xeon(R) CPU E31290 at 3.60GHz which is a 64bit machine. Table IV lists runtime execution, absolute difference between computed values, and estimated relative errors for a requested relative error tolerance of 10^{-13}. Clearly while both `BESSELINT` and `IIPBF` give similar values with similar relative errors, the former is faster by a factor of 1.32-3.15. 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 and Cools 2006b], the bottleneck in the respective codes were `ira` and `mWtrans` which both deal with the infinite integration calculation. Finally, table V shows actual errors for case 23 with different values of (a,b) i.e. higher order based on the requested tolerance of 10^{-10}.

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

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

Case	$f(x)B_{a,b,\rho,\tau}(x)$	Value
1	$J_0(x)J_1(3x/2)$	$2/3$
2	$J_0(x)J_5(2x)/x^4$	$27/4096$
3	$xJ_0(x)J_{20}(1.1x)/(1+x^2)$	$\approx -6.050747903049 \times 10^{-3}$
4	$J_0(x)J_1(x)/x$	$2/\pi$
5	$J_1(x)J_1(x)/x^2$	$4/3\pi$
6	$xK_0(xu)J_0(\rho x)J_0(\tau x)$	$\left[\left(u^2+\rho^2+\tau^2\right)^2 - 4\rho^2\tau^2\right]^{-1/2}$
7	$x^2K_1(xu)J_1(\rho x)J_1(\tau x)$	$4u\rho\tau\left[\left(u^2+\rho^2+\tau^2\right)^2 - 4\rho^2\tau^2\right]^{-3/2}$
8	$e^{-2ux}J_0(x)Y_0(x)$	$-(1/\pi)\int_0^{\pi/2}(1+u^2\cos^2 z)^{-1/2}dz$
9	$xe^{-x^2/u}J_2(x)Y_2(x)$	$\frac{4}{u\pi} - \frac{2}{\pi} - \frac{uK_2(u/2)}{2\pi e^{u/2}}$
10	$x^3e^{-x^2/u}J_2(x)Y_2(x)$	$-\frac{4}{\pi} + \frac{u^2(2+u)K_0(u/2)}{4\pi e^{u/2}} + \frac{u(8+4u+u^2)K_1(u/2)}{4\pi e^{u/2}}$
11	$e^{-ux}Y_0(\rho x)Y_0(\tau x)$	$-\frac{2}{\pi}\int_\tau^\infty \frac{G(u,\rho,y)dy}{\sqrt{y^2-\tau^2}}$ where $\rho<\tau$ and $G(u,\rho,y) = -\frac{2}{\pi}\left(\alpha^2+\frac{u^2y^2}{\alpha^2}\right)^{-1}\times$ $\left[\frac{uy}{\alpha}\ln\left(\frac{\alpha+y}{\rho}\sqrt{1+\frac{u^2}{\alpha^2}}\right)+\alpha\arctan\left(\frac{\alpha}{u}\right)\right]$ $\alpha^2 = \frac{1}{2}\left[\sqrt{(y^2-\rho^2-u^2)^2+4u^2y^2}+(y^2-\rho^2-u^2)\right] \quad (\alpha>0)$
12	$e^{-ux}J_0(\rho x)J_0(\tau x)$	$\frac{2}{\pi\sqrt{u^2+(\rho+\tau)^2}}\mathbf{K}\left[\frac{2\sqrt{\rho\tau}}{\sqrt{u^2+(\rho+\tau)^2}}\right]$
13	$e^{-ux}J_0(\rho x)Y_0(\tau x)$	$-\frac{2}{\pi}\int_\tau^\infty \frac{F(u,\rho,y)dy}{\sqrt{y^2-\tau^2}}$ where $F(u,\rho,y) = \frac{uy}{\alpha}\left(\alpha^2+\frac{u^2y^2}{\alpha^2}\right)^{-1} \quad (u,\rho,y>0)$
14	$Y_0(\rho x)Y_0(\tau x)$	$(2/\pi\tau)\mathbf{K}(\rho/\tau)$ where $\tau>\rho$
15	$J_0(\rho x)J_0(\tau x)$	$(2/\pi\tau)\mathbf{K}(\rho/\tau)$ where $\tau>\rho$
16	$J_0(\rho x)Y_0(\tau x)$	$-\frac{2}{\pi\rho}H(\rho-\tau)\mathbf{K}\left[\sqrt{1-\frac{\tau^2}{\rho^2}}\right]$
17	$x^{-1/2}e^{-ux}J_1(\rho x)J_{1/2}(\tau x)$	$\frac{\sqrt{2}}{\sqrt{\tau\pi}\rho}(\tau-\sqrt{\tau^2-l_1^2})$ where $\rho,\tau,u>0$ and $l_1 = \frac{1}{2}\left[\sqrt{(\rho+\tau)^2+u^2}-\sqrt{(\rho-\tau)^2+u^2}\right]$
18	$x^{-1/2}e^{-ux}J_2(\rho x)J_{3/2}(\tau x)$	$\frac{2\tau^{3/2}}{\sqrt{2\pi}\rho^2}\left(\frac{2}{3}-\frac{\sqrt{\tau^2-l_1^2}}{\tau}+\frac{(\tau^2-l_1^2)^{3/2}}{3\tau^3}\right)$

Formulae 4-8 and 17-18 were obtained from Gradshteyn & Ryzhik [2007]; those for cases 9-10 were obtained from Adamchik [1995]; for cases 11-16 (14-16 are special ones of 11-13 when $u = 0$) see Appendix A for derivation of formulae involving the elliptic function $\mathbf{K}$ that are simpler than those by Glasser [1974].

Table II. Actual errors in cases 11, 12 and 13 for ρ and τ

					Case 11			
					ρ			
		0.001	0.01	0.1	1	10	100	1000
	0.0011	$3.74\text{x}10^{-12}$	-	-	-	-	-	-
	0.011	$2.10\text{x}10^{-13}$	$3.91\text{x}10^{-13}$	-	-	-	-	-
	0.11	$1.92\text{x}10^{-12}$	$5.24\text{x}10^{-13}$	$2.25\text{x}10^{-13}$	-	-	-	-
τ	1.1	$7.57\text{x}10^{-13}$	$5.04\text{x}10^{-13}$	$3.75\text{x}10^{-13}$	$2.14\text{x}10^{-13}$	-	-	-
	11	$2.74\text{x}10^{-13}$	$3.01\text{x}10^{-13}$	$4.37\text{x}10^{-13}$	$2.48\text{x}10^{-13}$	$1.47\text{x}10^{-13}$	-	-
	101	$6.98\text{x}10^{-13}$	$3.46\text{x}10^{-13}$	$4.51\text{x}10^{-13}$	$5.03\text{x}10^{-13}$	$2.53\text{x}10^{-12}$	$2.52\text{x}10^{-13}$	-
	1001	$5.89\text{x}10^{-13}$	$5.16\text{x}10^{-13}$	$5.32\text{x}10^{-13}$	$1.74\text{x}10^{-11}$	$3.69\text{x}10^{-13}$	$5.27\text{x}10^{-13}$	$1.42\text{x}10^{-13}$
					Case 12			
					ρ			
		0.001	0.01	0.1	1	10	100	1000
	0.0011	$3.00\text{x}10^{-14}$	$1.54\text{x}10^{-14}$	$5.55\text{x}10^{-16}$	$1.11\text{x}10^{-16}$	$7.70\text{x}10^{-15}$	$2.32\text{x}10^{-15}$	$2.51\text{x}10^{-15}$
	0.011	$2.21\text{x}10^{-14}$	$2.33\text{x}10^{-15}$	$6.66\text{x}10^{-16}$	$2.22\text{x}10^{-16}$	$8.05\text{x}10^{-16}$	$1.63\text{x}10^{-14}$	$2.41\text{x}10^{-15}$
	0.11	0.00	$4.44\text{x}10^{-16}$	$1.11\text{x}10^{-16}$	$3.89\text{x}10^{-15}$	$1.13\text{x}10^{-12}$	$4.25\text{x}10^{-15}$	$9.26\text{x}10^{-14}$
τ	1.1	$2.22\text{x}10^{-16}$	0.00	$8.33\text{x}10^{-15}$	$1.94\text{x}10^{-14}$	$3.71\text{x}10^{-13}$	$2.04\text{x}10^{-12}$	$6.86\text{x}10^{-16}$
	11	$9.81\text{x}10^{-15}$	$3.39\text{x}10^{-15}$	$5.17\text{x}10^{-14}$	$2.20\text{x}10^{-14}$	$1.26\text{x}10^{-13}$	$2.62\text{x}10^{-13}$	$1.88\text{x}10^{-12}$
	101	$2.32\text{x}10^{-15}$	$1.63\text{x}10^{-14}$	$4.37\text{x}10^{-16}$	$1.02\text{x}10^{-13}$	$6.06\text{x}10^{-14}$	$2.13\text{x}10^{-13}$	$6.69\text{x}10^{-13}$
	1001	$2.51\text{x}10^{-15}$	$2.42\text{x}10^{-15}$	$9.36\text{x}10^{-14}$	$2.14\text{x}10^{-15}$	$4.72\text{x}10^{-14}$	$2.18\text{x}10^{-13}$	$1.75\text{x}10^{-13}$
					Case 13			
					ρ			
		0.001	0.01	0.1	1	10	100	1000
	0.0011	$4.98\text{x}10^{-13}$	$4.52\text{x}10^{-13}$	$3.85\text{x}10^{-13}$	$4.70\text{x}10^{-13}$	$3.34\text{x}10^{-13}$	$4.82\text{x}10^{-13}$	$5.88\text{x}10^{-13}$
	0.011	$2.36\text{x}10^{-13}$	$3.56\text{x}10^{-13}$	$3.20\text{x}10^{-13}$	$2.67\text{x}10^{-13}$	$5.11\text{x}10^{-13}$	$4.99\text{x}10^{-13}$	$5.49\text{x}10^{-13}$
	0.11	$5.32\text{x}10^{-13}$	$4.29\text{x}10^{-13}$	$3.26\text{x}10^{-13}$	$2.73\text{x}10^{-13}$	$1.73\text{x}10^{-12}$	$5.22\text{x}10^{-13}$	$5.28\text{x}10^{-13}$
τ	1.1	$2.37\text{x}10^{-13}$	$2.36\text{x}10^{-13}$	$1.49\text{x}10^{-13}$	$2.24\text{x}10^{-14}$	$2.01\text{x}10^{-13}$	$2.30\text{x}10^{-12}$	$5.02\text{x}10^{-13}$
	11	$3.13\text{x}10^{-13}$	$3.15\text{x}10^{-13}$	$4.15\text{x}10^{-13}$	$2.35\text{x}10^{-13}$	$3.03\text{x}10^{-13}$	$2.36\text{x}10^{-13}$	$2.05\text{x}10^{-12}$
	101	$5.54\text{x}10^{-13}$	$5.54\text{x}10^{-13}$	$5.58\text{x}10^{-13}$	$5.70\text{x}10^{-13}$	$3.76\text{x}10^{-13}$	$2.62\text{x}10^{-13}$	$9.84\text{x}10^{-14}$
	1001	$4.56\text{x}10^{-13}$	$4.56\text{x}10^{-13}$	$4.56\text{x}10^{-13}$	$4.51\text{x}10^{-13}$	$2.81\text{x}10^{-14}$	$1.24\text{x}10^{-13}$	$1.99\text{x}10^{-13}$

In case 11, there was a warning that the maximum number of intervals in `quadgk` had been exceeded for $\rho = 0.001, \tau = 0.0011$ as either the spacing between the first two zeros or the first zero or both were too large.

Table III. Additional cases for comparison of `BESSELINT` and `IIPBF`.

Case	$f(x)B_{a,b,\rho,\tau}(x)$	Value
19	$e^{-2ux}xJ_0(\rho x)J_1(\rho x)$	$\dfrac{\mathbf{K}-\mathbf{E}}{2\pi\rho\sqrt{\rho^2+u^2}}$
20	$e^{-2ux}[J_0(\rho x)]^2$	$\dfrac{\mathbf{K}}{\pi\sqrt{\rho^2+u^2}}$
21	$e^{-2ux}[J_1(\rho x)]^2$	$\dfrac{(2u^2+\rho^2)\mathbf{K}-2(u^2+\rho^2)\mathbf{E}}{\pi\rho^2\sqrt{\rho^2+u^2}}$
22	$e^{-2ux}[xJ_1(x)]^2$	$\dfrac{3}{4\pi}\displaystyle\int_0^{\pi/2}\dfrac{\cos^2 z dz}{[u^2+\cos^2 z]^{5/2}}$
23	$\dfrac{x^{b-a+1}}{x^2+u^2}J_a(\rho x)J_b(\tau x)$	$u^{b-a}I_a(\rho u)K_b(\tau u)$ where $2+a>b>-1$ for $\tau>\rho$

MATLAB functions for $\mathbf{K}$ and $\mathbf{E}$ 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 Gradshteyn & Ryzhik [2007].

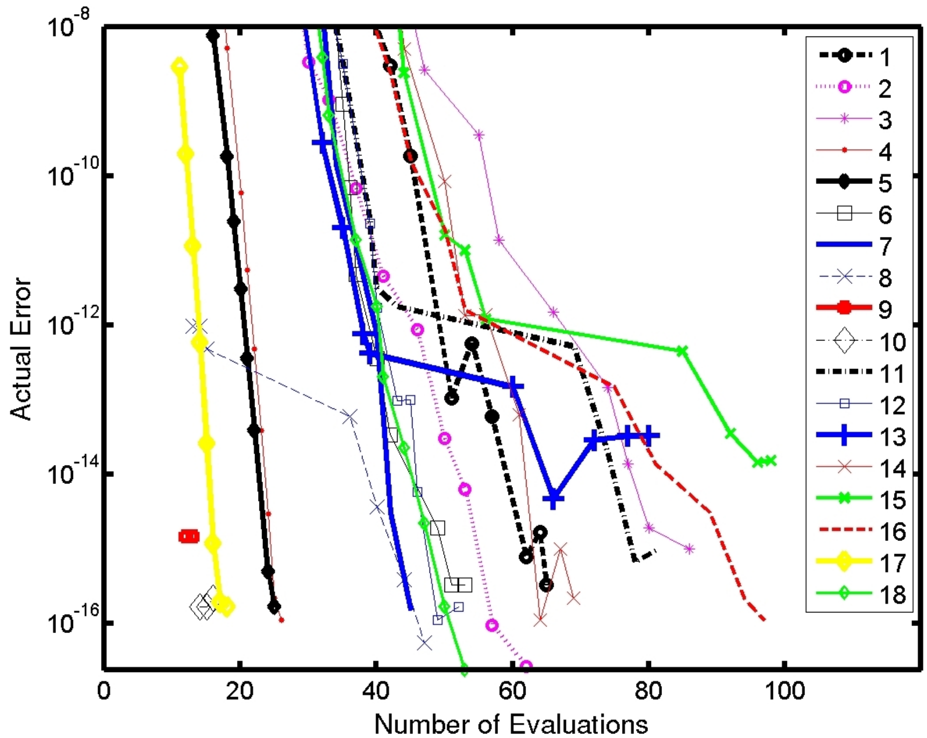

Fig. 1. Comparing actual error to number of function evaluations of h_1 and h_2 with requested relative error tolerances from 10^{-4} to 10^{-14} in 10 decrements of 0.1 for the cases in table I. In cases 6 and 7, parameters $\{u, \rho, \tau\} = \{1, 2, 1\}$. For cases 8, 9 and 10 $u = 1.5$, 0.2 and 2 respectively. For cases 11-13,17-18, $\{u, \rho, \tau\} = \{0.1, 2, 1\}$. For cases 14-16, $\{\rho, \tau\} = \{3, 1\}$.

Table IV. Comparison of `BESSELINT` and `IIPBF`

Case	Time (secs)		Estimated Error		Absolute
	`BESSELINT`	`IIPBF`	`BESSELINT`	`IIPBF`	Difference
1	0.0905	0.1744	1.87x10^{-16}	1.80x10^{-15}	1.11x10^{-16}
2	0.0863	0.1705	5.26x10^{-16}	1.48x10^{-15}	2.78x10^{-17}
3	0.1351	0.2390	1.32x10^{-14}	1.17x10^{-15}	7.11x10^{-17}
4	0.0823	0.1434	3.49x10^{-16}	2.99x10^{-15}	1.11x10^{-16}
5	0.0833	0.1474	5.64x10^{-22}	4.00x10^{-16}	1.11x10^{-16}
19	0.0914	0.1253	5.55x10^{-15}	7.50x10^{-16}	6.94x10^{-18}
20	0.0415	0.1307	1.17x10^{-14}	2.06x10^{-15}	2.78x10^{-16}
21	0.0897	0.1214	1.66x10^{-14}	9.78x10^{-17}	1.91x10^{-17}
22	0.0931	0.1229	7.27x10^{-15}	1.16x10^{-16}	1.30x10^{-18}
23	0.0990	0.1782	8.80x10^{-16}	1.16x10^{-15}	6.94x10^{-18}

Parameters used in cases 19-23 are $u = 2, \rho = 1, a = 1, \tau = 2$. The last column is the absolute difference between the computed values.

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]. Generally, `quadgk` performs the adaptive subdivision locally and conservatively, is able to process subintervals simultaneously, samples many points within the subintervals

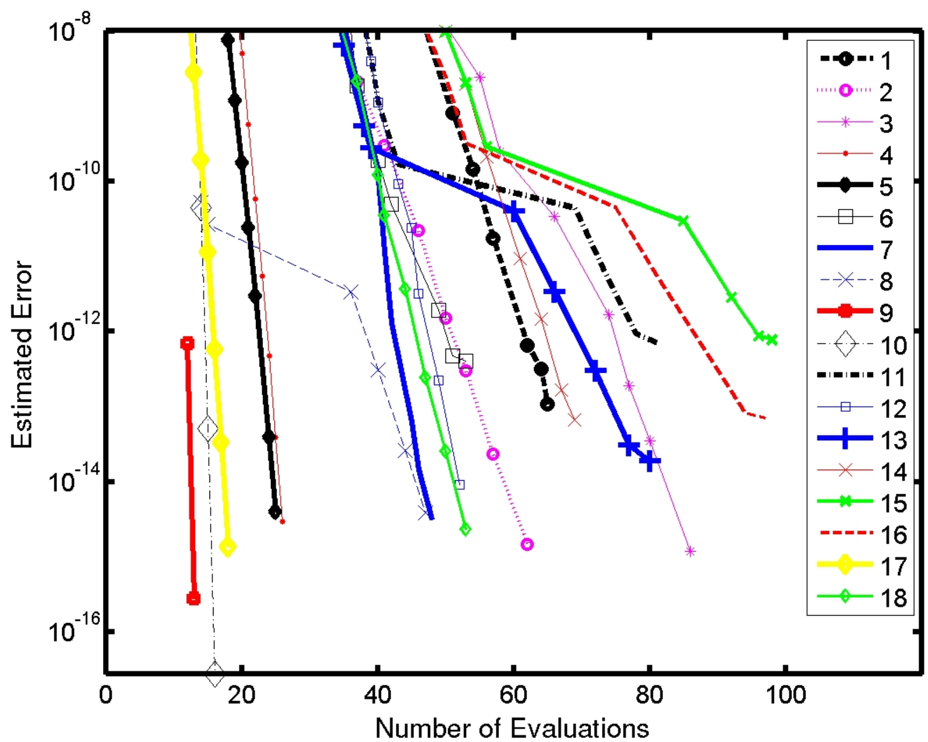

Fig. 2. Comparing estimated error to number of function evaluations of h_1 and h_2 with requested relative error tolerances from 10^{-4} to 10^{-14} in 10 decrements of 0.1 for the cases in table I. In cases 6 and 7, parameters $\{u, \rho, \tau\} = \{1, 2, 1\}$. For cases 8, 9 and 10 $u = 1.5$, 0.2 and 2 respectively. For cases 11-13,17-18, $\{u, \rho, \tau\} = \{0.1, 2, 1\}$. For cases 14-16, $\{\rho, \tau\} = \{3, 1\}$.

and is therefore faster and more reliable than its predecessor [Shampine 2008; Gonnet 2012]. But the structure of `quadgk` makes vectorizing `mWtrans` on the lines of Bailey and Borwein [2012] difficult.

The modification of the general method for finding zeros of h_1 is necessary to avoid using the Newton method in regions where h_1 behaves like a double 'pole' as was the case in an application of the toolbox in a separate study [Davis et al. 2012]. This use of approximate zeros of h_1 and h_2 reduces the number of function evaluations further.

An alternative to product decomposition in the Lucas algorithm is the strategy by Sidi [2012] who modified the mW transform, mW$^{(s)}$, to avoid loss of precision when the original mW transform was applied to integrands of even multiples of $J_a(\rho x)$ [Bailey and Borwein 2012]. In `IIPBF`, the product decomposition of two Bessel functions with the same argument (i.e. when $\rho = \tau$) asymptotically results in just one oscillatory function rather than two oscillatory functions. Though Sidi [2012, see Example 7.7] shows how mW$^{(2)}$ could work, both figs. 1-2 and table IV show there is no need for mW$^{(2)}$ as predicted [Sidi 2012, Example 7.8].

Codes for computing infinite integrals of products of Bessel functions of the first kind of order 0 or 1 have emerged in recent years. These use digital filters of Hankel transforms [Key 2012], trapezoid-like integration of the asymptotic form beyond a user-supplied breakpoint [Li et al. 2010] or linear kernel [Singh and Mogi 2005]. In particular, Key [2012] showed that adaptive quadrature with feedback based on er-

Table V. Actual errors in case 23 and $u = 2$

Case 23, $a = 1, b = 1$

	τ \ ρ	0.001	0.01	0.1	1	10	100	1000
	0.0011	$3.70\text{x}10^{-14}$	-	-	-	-	-	-
	0.011	$6.88\text{x}10^{-14}$	$3.66\text{x}10^{-14}$	-	-	-	-	-
	0.11	$3.49\text{x}10^{-14}$	$6.89\text{x}10^{-14}$	$3.71\text{x}10^{-14}$	-	-	-	-
τ	1.1	$2.57\text{x}10^{-16}$	$1.65\text{x}10^{-13}$	$7.37\text{x}10^{-14}$	$4.30\text{x}10^{-14}$	-	-	-
	11	$7.39\text{x}10^{-16}$	$4.23\text{x}10^{-17}$	$7.62\text{x}10^{-15}$	$1.15\text{x}10^{-13}$	$3.29\text{x}10^{-13}$	-	-
	101	$1.65\text{x}10^{-92}$	$3.47\text{x}10^{-18}$	$5.82\text{x}10^{-16}$	$1.12\text{x}10^{-13}$	$1.59\text{x}10^{-12}$	$4.56\text{x}10^{-13}$	-
	1001	0.00	$3.47\text{x}10^{-18}$	$5.64\text{x}10^{-18}$	$1.01\text{x}10^{-16}$	$6.38\text{x}10^{-14}$	$2.25\text{x}10^{-13}$	$1.66\text{x}10^{-12}$

Case 23, $a = 3, b = 5$

	τ \ ρ	0.001	0.01	0.1	1	10	100	1000
	0.0011	$1.04\text{x}10^{-7}$	-	-	-	-	-	-
	0.011	$6.67\text{x}10^{-10}$	$4.22\text{x}10^{-10}$	-	-	-	-	-
	0.11	$3.16\text{x}10^{-11}$	$2.50\text{x}10^{-12}$	$5.68\text{x}10^{-13}$	-	-	-	-
τ	1.1	$2.79\text{x}10^{-9}$	$5.10\text{x}10^{-13}$	$1.10\text{x}10^{-13}$	$3.28\text{x}10^{-13}$	-	-	-
	11	$8.60\text{x}10^{-20}$	$3.64\text{x}10^{-12}$	$1.17\text{x}10^{-14}$	$4.54\text{x}10^{-13}$	$1.95\text{x}10^{-13}$	-	-
	101	$1.17\text{x}10^{-98}$	$2.27\text{x}10^{-13}$	$1.33\text{x}10^{-15}$	$2.38\text{x}10^{-14}$	$8.92\text{x}10^{-13}$	$2.42\text{x}10^{-12}$	-
	1001	0.00	0.00	0.00	$2.17\text{x}10^{-19}$	$2.64\text{x}10^{-15}$	$3.70\text{x}10^{-12}$	$3.59\text{x}10^{-12}$

For $a = 3, b = 5$, there were warnings that the maximum number of intervals in `quadgk` had been exceeded for $\rho = 0.001, \tau = 0.0011$ and $\rho = 0.001, \tau = 0.11$ as either the spacing between the first two zeros or the first zero or both were too large. It was also necessary to use `Mathematica` to calculate the actual value for $\rho = 1000, \tau = 1001$.

ror together with a `MATLAB` implementation of the ε-algorithm such as that used here should be preferred to digital transforms. With regards to `BESSELINT` [Van Deun and Cools 2006a; 2008; 2006b], the rational exponential form of $f(x)$ is used to facilitate stable quadrature over a converging series in the asymptotic part whereas `IIPBF` can be used for any well-behaved $f(x)$. Quadrature in both cases are adaptive however `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 `quadgk` function for weak logarithmic and algebraic forms via a simple transform whereas extrapolation is used in `BESSELINT`. The breakpoint beyond which asymptotic properties are used is determined by the first zero of h_1 and h_2 in `IIPBF` whereas in `BESSELINT` it is determined via the incomplete gamma function.

Tables II-V show that `IIPBF` should work for a wide range of ρ and τ together with higher orders. But for extreme values of ρ and τ which can arise in imaging problems [Li et al. 2010] the results should be checked via asymptotic analysis. Although $|a-b| > 5$ should work [Lucas 1995], problems may arise when the Bessel functions become very small [Van Deun and Cools 2008]. Indeed, warnings occured when the maximum number of intervals used in `quadgk` was exceeded as either the first zero or the spacing between the first two zeros or both were too large. So as in `BESSELINT`, such messages should prompt a case by case inspection. Also, it should be pointed out that case 23 is a special case of one of the tests used by van Deun and Cools [2008, see Eq. 23]. While it would be more laborious to derive asymptotic forms of more general f, in future it might be possible to combine the best features of both `IIPBF` and `BESSELINT` in which case Hankel functions may be useful [Huybrechs and Vandewalle 2006] in developing the procedure for Bessel functions of the second kind.

As alluded above, `IIPBF` was used in a model of a submerged viscous fluid jet impinging on a planar wall wherein the computations whose kernels involved modified Bessel functions of the first kind agreed with asymptotic behaviour [Davis et al. 2012, see Fig. 4]. Further, `IIPBF` includes additional test cases not shown here such as a couple with complex-valued kernels and several with real-valued order which arise in imaging and physics problems [McPhedran et al. 1992; Gebremariam et al. 2010; Li et al. 2010; Hosseinbor et al. 2013; Davis et al. 2013]. So IIPBF has the potential to be used in a wider range of problems than the original algorithm.

ACKNOWLEDGMENTS

We thank Zachary McGraw and Geoffrey Gunter for technical assistance.

A. FORMULAE FOR CASES 11-16

This appendix describes the derivation of expressions for integrals in cases 11-16 (14-16 are special ones of 11-13 when $u = 0$) in table I that are found to be simplificatons and extensions of those obtained by Glasser [1974].

The elliptic integral $\mathbf{K}(k)$ is defined by

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

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

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

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

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

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

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

and

$$\int_0^\infty e^{-ux}J_0(\rho x)Y_0(\tau x)\, dx = \frac{2}{\pi}\int_0^\rho \frac{G(u,\tau,y)dy}{\sqrt{\rho^2-y^2}} = \frac{2}{\pi}\int_0^{\pi/2} G(u,\tau,\rho\sin\eta)d\eta$$

where

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

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 $G(u,\tau,y)$ to be evaluated by replacing u by $u + \imath y$ and taking the real part of the result. If $\sqrt{(u+\imath y)^2+\tau^2} = uy/\alpha + \imath\alpha$, then $u, y, \tau > 0$ implies $\alpha > 0$ where

$$\alpha^2 = \frac{1}{2}\left[\sqrt{(y^2-\tau^2-u^2)^2+4u^2y^2} + (y^2-\tau^2-u^2)\right].$$

After some algebra,

$$G(u,\tau,y) = -\frac{2}{\pi}\left(\alpha^2+\frac{u^2y^2}{\alpha^2}\right)^{-1}\left[\frac{uy}{\alpha}\ln\left(\frac{\alpha+y}{\tau}\sqrt{1+\frac{u^2}{\alpha^2}}\right) + \alpha\arctan\left(\frac{\alpha}{u}\right)\right].$$

and hence $G(u,\tau,y) \sim -y^{-1}$ as $y \to \infty$. The known expressions for G when $u = 0, y > \tau$ are recovered. We also have

$$\int_0^\infty e^{-ux} J_0(\rho x) Y_0(\tau x) dx = -\frac{2}{\pi} \int_\tau^\infty \frac{F(u,\rho,y)dy}{\sqrt{y^2 - \tau^2}}$$

where

$$F(u,\tau,y) = \int_0^\infty e^{-ux} J_0(\tau x) \cos xy dx = \Re[\tau^2 + (u + \imath y)^2]^{-1/2}$$

$$= \frac{uy}{\alpha} \left(\alpha^2 + \frac{u^2 y^2}{\alpha^2} \right)^{-1} \qquad (u,\tau,y > 0).$$

In the limit $u = 0$, $F(u,\tau,y) = H(\tau - y)(\tau^2 - y^2)^{-1/2}$ and thus

$$\int_0^\infty J_0(\rho x) Y_0(\tau x) dx = -\frac{2}{\pi \rho} H(\rho - \tau) \mathbf{K} \left[\sqrt{1 - \frac{\tau^2}{\rho^2}} \right].$$

Likewise a simpler result is

$$\int_0^\infty e^{-ux} J_0(\tau x) J_0(\rho x) dx = \frac{2}{\pi \sqrt{u^2 + (\rho + \tau)^2}} \mathbf{K} \left[\frac{2\sqrt{\rho\tau}}{\sqrt{u^2 + (\rho + \tau)^2}} \right].$$

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

$$\int_0^\infty K_0(\rho x) Y_0(\tau x) \, dx = -\int_\tau^\infty \frac{dy}{\sqrt{(y^2 + \rho^2)(y^2 - \tau^2)}} = -\frac{1}{\sqrt{\rho^2 + \tau^2}} \mathbf{K} \left(\frac{\rho}{\sqrt{\rho^2 + \tau^2}} \right),$$

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 $\mathbf{K}$. Similarly, with $\rho < \tau$,

$$\int_0^\infty K_0(\rho x) K_0(\tau x) \, dx = \frac{\pi}{2} \int_0^\infty \frac{dy}{\sqrt{(y^2 + \rho^2)(y^2 + \tau^2)}} = \frac{\pi}{2\tau} \mathbf{K} \left(\sqrt{1 - \frac{\rho^2}{\tau^2}} \right),$$

after setting $y^2 = \tau^2 (X^2 - 1)$. This appears to contradict the second of Glasser's Eq.(8) but the hypergeometric function identity,

$$F(\alpha, \alpha - \beta + 1/2; \beta + 1/2, z^2) = (1 + z)^{-2\alpha} F \left[\alpha, \beta; 2\beta, \frac{4z}{(1 + z)^2} \right],$$

yields, with $\alpha = \beta = 1/2$ and $k < 1$,

$$\mathbf{K}(k) = \frac{\pi}{2} F(1/2, 1/2; 1, k^2) = \frac{\pi}{2(1 + k)} F \left(1/2, 1/2; 1, \frac{4k}{(1 + k)^2} \right) = \frac{1}{1 + k} \mathbf{K} \left(\frac{2\sqrt{k}}{1 + k} \right).$$

In particular,

$$\mathbf{K} \left(\frac{\tau - \rho}{\tau + \rho} \right) = \frac{\tau + \rho}{2\tau} \mathbf{K} \left(\sqrt{1 - \frac{\rho^2}{\tau^2}} \right) \qquad (\rho < \tau),$$

$$\mathbf{K}(\rho/\tau) = \frac{\tau}{\tau+\rho}\mathbf{K}\left(\frac{2\sqrt{\tau\rho}}{\tau+\rho}\right) \qquad (\rho < \tau),$$

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

ADAMCHIK, V. 1995. The evaluation of integrals of Bessel functions via G-function identities. *J. Comp. Appl. Math. 64*, 283–290.

BAILEY, D. H. AND BORWEIN, J. H. 2012. Hand-to-hand combat with thousand-digit integrals. *J. Comp. Sci. 3*, 7786. http://dx.doi.org/10.1016/j.jocs.2010.12.004.

BARROWES, B. 2009. *The SLATEC library converted into MATLAB functions*. http://www.mathworks.com/matlabcentral/fileexchange/14535.

CRASTER, R. V. 1998. Scattering by cracks beneath fluid-solid interfaces. *J. Sound Vib. 209*, 343–372.

DAVIS, A. M. J., KIM, J. H., CERITOGLU, C., AND RATNANATHER, J. T. 2012. A Stokesian analysis of a submerged viscous jet impinging on a planar wall. *J. Fluid Mech. 712*, 531–551.

DAVIS, A. M. J., KIM, J. H., GUNTER, G. M., AND RATNANATHER, J. T. 2013. The Stokesian flow field of an oscillatory submerged viscous jet impinging on a planar wall. *Proc. Roy. Soc. Lond. A*.

DAVIS, A. M. J. AND STONE, H. A. 1998. Slow translation, rotation or oscillation of a disk in a rotating fluid: Effect of a plane wall or another disk. *Q. J. Mech. App. Math. 51*, 495–513.

GEBREMARIAM, B., DUGUET, T., AND BOGNER, S. K. 2010. Symbolic integration of a product of two spherical bessel functions with an additional exponential and polynomial factor. *Computer Physics Communications 181*, 1136–1143.

GLASSER, M. L. 1974. Some definite integrals of the product of two Bessel functions of the second kind: (order zero). *Math. Comp. 28*, 613–615.

GONNET, P. 2012. A review of error estimation in adaptive quadrature. *ACM Comput. Surv. 44,* 4, 22:1–22:36.

GRADSHTEYN, I. S. AND RYZHIK, I. M. 2007. *Gradshteyn and Ryzhik's Table of Integrals, Series, and Products* 7th Ed. Academic Press, San Diego. A. Jeffrey and D. Zwillinger (Eds.).

HOSSEINBOR, A. P., CHUNG, M. K., WU, Y.-C., AND ALEXANDER, A. L. 2013. Bessel Fourier Orientation Reconstruction (BFOR): an analytical diffusion propagator reconstruction for hybrid diffusion imaging and computation of q-space indices. *Neuroimage 64*, 650–670.

HUYBRECHS, D. AND VANDEWALLE, S. 2006. On the evaluation of highly oscillatory integrals by analytic continuation. *SIAM J. Num. Anal. 44*, 1026–1048.

IKONOMOU, M., KÖHLER, P., AND JACOB, A. F. 1995. Computation of integrals over the half-line involving products of Bessel functions, with application to microwave transmission lines. *Z. angew Math. Mech. 75*, 917–926.

KEY, K. 2012. Is the fast Hankel transform faster than quadrature? *Geophysics 77*, F21–F30.

LI, M., HE, W., LI, Q., XU, Z., AND LUO, C. 2010. The numerical integration algorithm of dual Bessel function and its application. *Int. J. Appl. Electromagnetics and Mechanics 33*, 727–734.

LUCAS, S. K. 1995. Evaluating infinite integrals involving products of Bessel functions of arbitrary order. *J. Comp. App. Math. 64*, 269–282.

LUCAS, S. K. AND STONE, H. A. 1995. Evaluating infinite integrals involving Bessel functions of arbitrary order. *J. Comp. App. Math. 64*, 217–231.

MCPHEDRAN, R. C., DAWES, D. H., AND SCOTT, T. C. 1992. On a Bessel function integral. *Applicable Algebra in Engineering, Communication and Computing 2*, 207–216.

OLVER, F. W. J., LOZIER, D. W., BOISVERT, R. F., AND CLARK, C. W. 2010. *NIST Handbook of Mathematical Functions: Companion to the Digital Library of Mathematical Functions*. Cambridge University Press, New York.

PAN, E., BEVIS, M., HAN, F., ZHOU, H., AND ZHU, R. 2007. Surface deformation due to loading of a layered elastic half-space: a rapid numerical kernel based on a circular loading element. *Geophys. J. Int. 171*, 11–24.

PETROV, E. P. AND SCHWILLE, P. 2008. Translational diffusion in lipid membranes beyond the Saffman-Delbruck approximation. *Biophys. J. 94*, L41–L43.

PIESSENS, R., DONCKER-KAPENGA, E. D., UBERHUBER, C. W., AND KAHAUER, D. K. 1983. *QUADPACK, a sub-routine package for automatic integration*. Springer-Verlag, Berlin.

ROBINSON, N. I. 2002. An isotropic elastic medium containing a cylindrical borehole with a rigid plug. *Int. J. Solids Struct. 39*, 4889–4904.

SALO, J., WEI-SALLABI, H., AND VAINIKAINEN, P. 2006. Statistical analysis of the multiple scattering radio channel. *IEEE Trans. Ant. Prop. 54*, 3114–3124.

SHAMPINE, L. F. 2008. Vectorized adaptive quadrature in MATLAB. *J. Comp. App. Math. 211*, 131–140.

SHERWOOD, J. D. 2005. Optimal probes for withdrawal of uncontaminated fluid samples. *Phys. Fluids 17*, 083102.

SIDI, A. 1988. A user-friendly extrapolation method for oscillatory infinite integrals. *Math. Comp. 51*, 249–266.

SIDI, A. 2012. A user-friendly extrapolation method for computing infinite range integrals of products of oscillatory functions. *IMA J. Num. Anal. 32*, 602–631.

SINGH, N. P. AND MOGI, T. 2005. Electromagnetic response of a large circular loop source on a layered earth: A new computation method. *Pure and Applied Geophysics 162*, 181–200.

TARTAKOVSKY, D. M., MOULTON, J. D., AND ZLOTNIK, V. A. 2000. Kinematic structure of minipermeameter flow. *Water Resources Research 36*, 2433–2442.

VAN DEUN, J. AND COOLS, R. 2006a. Algorithm 858: Computing infinite range integrals of an arbitrary product of Bessel functions. *ACM Trans. Math. Soft. 32*, 580–596.

VAN DEUN, J. AND COOLS, R. 2006b. A Matlab implementation of an algorithm for computing integrals of products of Bessel functions. *LNCS 4151*, 284–295.

VAN DEUN, J. AND COOLS, R. 2006c. A stable recurrence for the incomplete gamma function with imaginary second argument. *Numer. Math. 104*, 445–456.

VAN DEUN, J. AND COOLS, R. 2008. Integrating products of Bessel functions with an additional exponential or rational factor. *Comp. Phys. Comm. 178*, 578–590.

XU, Y. S., LIN, J., LI, H. M., AND WU, F. M. 2003. Analysis for the potential function of the digital microstructure image of porous media. *Comm. Theor. Phys. 40*, 393–394.