BIT
2000, Vol., No., pp. 1–14

0006-3835/98/3804-0101 $12.00

COMPUTING THE KONTOROVICH–LEBEDEV INTEGRAL TRANSFORMS AND THEIR INVERSES [*]

WALTER GAUTSCHI[1]

[1] *Department of Computer Sciences, Purdue University, West Lafayette, IN 47907, USA. email: wxg@cs.purdue.edu*

Abstract.

The numerical evaluation of the transforms in the title, and their inverses, is considered, using a variety of decomposition, truncation, and quadrature methods. Extensive numerical testing is provided and an application given to the numerical evaluation of the kernel of a Fredholm integral equation of interest in mixed boundary value problems on wedge-shaped domains.

AMS subject classification: 44A15, 65D30, 65R10.

Key words: Kontorovich–Lebedev transform, inverse transform, Gaussian quadrature, integral equation

1 Introduction

The Kontorovich–Lebedev transforms are integral transforms whose kernels are modified Bessel functions of purely imaginary order, or complex order having real part 1/2. While the computation of these kernels has been the subject of some recent work, the computation of the transforms themselves, which presents peculiar difficulties, apart from work in [10], has received little attention. In §§2–4 of this paper, effective numerical procedures are developed based on a suitable decomposition of the interval of integration and appropriate Gaussian quadrature rules. Detailed results on testing these procedures are included. Sections 5 and 6 are devoted to computing the inverse transforms by Gauss quadrature evaluation of suitably truncated integrals, with some consideration being given to estimating the error of truncation. Here, too, numerical examples are provided to support the effectiveness of our methods. In §7, an application of the inverse transform is given to the evaluation of the kernel of a Fredholm integral equation of interest in some mixed boundary value problems on wedge-shaped domains.

All computations were done in Matlab. The relevant routines can be downloaded from the web site

`http://www.cs.purdue.edu/archives/2002/wxg/codes`

at the link KL-transform.

[*] Received … . Communicated by … .

2 The Kontorovich–Lebedev transforms

Let K_ν denote the modified Bessel function of order ν (also called the Macdonald function). Integral transforms with K_ν as kernel and complex $\nu = \mathrm{i}\beta$ resp. $\nu = 1/2+\mathrm{i}\beta$ are called *Kontorovich–Lebedev transform* (*KL-transform*) and *modified Kontorovich–Lebedev transform* (*modified KL-transform*), respectively. Specifically, we write

$$F(\beta) = \int_0^\infty K_{\mathrm{i}\beta}(x) f(x)\mathrm{d}x \tag{2.1}$$

for the former, and

$$F(\beta) = \int_0^\infty K_{1/2+\mathrm{i}\beta}(x) f(x)\mathrm{d}x \tag{2.2}$$

for the latter, where f is a real-valued, sufficiently regular function. Note that $F(\beta)$ is real-valued in (2.1), but complex-valued in (2.2). A pair of real-valued modified KL-transforms can be defined by (the definitions (3), (4) in [9] are misprinted, but given correctly in (19) and (24) of that reference)

$$F_+(\beta) = \int_0^\infty \mathrm{Re}\, K_{1/2+\mathrm{i}\beta}(x) f(x)\mathrm{d}x \tag{2.3}$$

and

$$F_-(\beta) = \int_0^\infty \mathrm{Im}\, K_{1/2+\mathrm{i}\beta}(x) f(x)\mathrm{d}x. \tag{2.4}$$

We shall use ν to denote any complex number, which may be assigned the special values $\nu = \mathrm{i}\beta$ or $\nu = 1/2 + \mathrm{i}\beta$, as the case may be, but could also be arbitrary complex, in which case we write

$$F(\nu) = \int_0^\infty K_\nu(x) f(x)\mathrm{d}x, \quad \nu \in \mathbb{C}, \tag{2.5}$$

in place of (2.1) and (2.2).

The kernels in (2.1)–(2.5) at infinity decay exponentially, since (see, e.g., [1, eqn 9.7.2])

$$K_\nu(x) \sim \sqrt{\frac{\pi}{2x}}\, e^{-x}, \quad x \to \infty, \quad \nu \in \mathbb{C}. \tag{2.6}$$

The behavior near zero is more intricate. Indeed (cf. [5, eqn (7.2)]),

$$K_\nu(x) \sim k_\nu(x), \quad k_\nu(x) = \frac{1}{2}\left[\left(\frac{2}{x}\right)^{\nu}\Gamma(\nu) + \left(\frac{2}{x}\right)^{-\nu}\Gamma(-\nu)\right], \; x \downarrow 0, \quad \nu \notin \mathbb{N}. \tag{2.7}$$

To see more precisely what this entails, consider the case $\nu = \mathrm{i}\beta$, $\beta > 0$. Letting

$$\gamma = \arg\Gamma(1+\mathrm{i}\beta) = \mathrm{Im}[\ln\Gamma(1+\mathrm{i}\beta)], \tag{2.8}$$

and recalling that $|\Gamma(1+\mathrm{i}\beta)|^2 = \pi\beta/\sinh(\pi\beta)$, one finds after a short calculation that

$$k_{\mathrm{i}\beta}(x) = \sqrt{\frac{\pi}{\beta\sinh(\pi\beta)}}\,\sin(\beta\ln(2/x)+\gamma), \quad x\downarrow 0 \tag{2.9}$$

(see also [3, eqn (2.14)]). Thus, $K_{\mathrm{i}\beta}(x)$ is densely oscillating near $x=0$, with amplitude of the order $\sqrt{2\pi/\beta e^{\pi\beta}}$ when β is large. Similar densely oscillating behavior holds also when $\nu = 1/2+\mathrm{i}\beta$, indeed when ν is arbitrary complex. For computational purposes, one does not really need to know the exact analytic form, since Matlab can handle expressions like (2.7) when ν is complex. Still, this densely oscillatory behavior near the origin presents a major challenge for computing KL-transforms.

2.1 Evaluation by standard Gauss quadratures

It seems appropriate, for computational purposes, to split the integral in (2.5) into two parts, say

$$F(\nu) = \left(\int_0^2 + \int_2^\infty\right) K_\nu(x)f(x). \tag{2.10}$$

For the second part, the behavior (2.6) at infinity suggests Gauss–Laguerre quadrature unless $f(x)$ has its own peculiar behavior at infinity, which would require additional weighting factors. In the absence of such complications, we write

$$\int_2^\infty K_\nu(x)f(x)\mathrm{d}x = \int_0^\infty K_\nu(2+t)f(2+t)\mathrm{d}t$$

and apply Gauss–Laguerre quadrature,

$$\begin{aligned}\int_2^\infty K_\nu(x)f(x)\mathrm{d}x &= \int_0^\infty [e^t K_\nu(2+t)f(2+t)]e^{-t}\mathrm{d}t \\ &\approx \sum_{k=1}^n \lambda_k^L e^{\tau_k^L} K_\nu(2+\tau_k^L)f(2+\tau_k^L),\end{aligned} \tag{2.11}$$

where τ_k^L, λ_k^L are the nodes and weights of the n-point Gauss–Laguerre quadrature rule. To avoid overflow of $e^{\tau_k^L}$ in (2.11), one must restrict n to, say, $n \le 185$.

For the first part in (2.10), we write

$$\int_0^2 K_\nu(x)f(x)\mathrm{d}x = \int_0^2 [K_\nu(x)-k_\nu(x)]f(x)\mathrm{d}x + \int_0^2 k_\nu(x)f(x)\mathrm{d}x. \tag{2.12}$$

Here, the first integral on the right, unless f has peculiar behavior near zero, is amenable to Gauss–Legendre quadrature on $[0,2]$ with moderately many points. It is the second integral where the main difficulty resides. We can surmount it, with some effort, by making the change of variable $t=\ln(2/x)$ to obtain

$$\int_0^2 k_\nu(x)f(x)\mathrm{d}x = 2\int_0^\infty k_\nu(2e^{-t})f(2e^{-t})e^{-t}\mathrm{d}t. \tag{2.13}$$

Gauss–Laguerre quadrature, possibly of high degree (when $\operatorname{Im}\nu$ is large), is applicable to the last integral. Here, too, depending on the behavior of f, alternate ways of doing the integration may be called for.

All quadratures are performed to within a prescribed error tolerance of $\frac{1}{2}\,10^{-10}$, where error means "mollified" error (absolute error for quantities less than 1 in absolute value, and relative error otherwise).

The procedure described, for arbitrary complex ν, is implemented in the routine `KLT.m`, which calls upon the routine `macdonald.m` of [5] to evaluate $K_\nu(x)$. We recall that this latter routine relies on numerical techniques for $|\beta| \leq 10$, and on a symbolic Matlab routine when $|\beta| > 10$, where $\beta = \operatorname{Im}\nu$. Since the first two integrals involve K_ν in the integrand, each of the numerical quadratures described requires repeated calls to the routine `macdonald.m`. This makes the routine `KLT.m` very slow, when $|\beta| > 10$, even though, in programming it, some effort has gone into reducing the slowdown as much as possible.

2.2 Evaluation of the third integral by a special Gaussian quadrature

We have seen that in the third integral,

$$I_3 := \int_0^2 k_\nu(x) f(x)\mathrm{d}x, \tag{2.14}$$

the function $k_\nu(x)$ as $x \downarrow 0$ is densely oscillating, and therefore presents a challenge for any standard scheme of numerical integration. However, similarly as in [6], we may incorporate k_ν into a weight function and develop special Gaussian quadratures for this weight function. This is probably impractical if many values of ν must be dealt with, but seems feasible otherwise, provided f is smooth.

We first consider he case $\nu = \mathrm{i}\beta$. Here we have the behavior of $k_{\mathrm{i}\beta}$ indicated in (2.9). We write

$$I_3 = 2\int_0^1 k_{\mathrm{i}\beta}(2t) f(2t)\mathrm{d}t = 2\sqrt{\frac{\pi}{\beta\sinh\pi\beta}}\int_0^1 \sin(\beta\ln(1/t)+\gamma) f(2t)\mathrm{d}t$$

and introduce the nonnegative weight function on $[0,1]$,

$$w_\beta(t) = 1 + \sin(\beta\ln(1/t)+\gamma), \quad \gamma = \arg\Gamma(1+\mathrm{i}\beta). \tag{2.15}$$

Then

$$I_3 = 2\sqrt{\frac{\pi}{\beta\sinh\pi\beta}}\left[\int_0^1 w_\beta(t) f(2t)\mathrm{d}t - \int_0^1 f(2t)\mathrm{d}t\right]. \tag{2.16}$$

Here we compute the first integral by Gauss quadrature relative to the weight function w_β, and the second integral by a standard quadrature rule appropriate for f. In so doing, we expect to achieve good accuracy with relatively low-order quadrature rules.

To generate the Gauss formula for w_β, we start, as in [6], from the moments of w_β and apply the symbolic, variable-precision Chebyshev algorithm to compute the required recurrence coefficients for the respective orthogonal polynomials. By using sufficiently high precision, we are able to obtain these recurrence coefficients to Matlab machine precision. Once they are available, the desired Gauss formulae can be generated in the usual way via eigenvalues and -vectors of the Jacobi matrix for w_β.

The moments of w_β, on the other hand, are readily computed from

$$\begin{aligned} m_k &= \int_0^1 t^k w_\beta(t)\mathrm{d}t \\ &= \frac{1}{k+1} + \frac{1}{(k+1)^2+\beta^2}\,((k+1)\sin\gamma + \beta\cos\gamma), \quad k = 0,1,2,\dots\ . \end{aligned} \tag{2.17}$$

The generation of the recurrence coefficients from these moments is implemented in the routine `sr_RMKLTp.m` with input parameter `a` $= 0$.

In the more general case $\nu = \alpha + \mathrm{i}\beta$, $0 < \alpha < 1$, $\beta \geq 0$, we define

$$\Gamma_+ = |\Gamma(\nu)|,\ \gamma_+ = \arg\Gamma(\nu); \quad \Gamma_- = |\Gamma(-\nu)|,\ \gamma_- = \arg\Gamma(-\nu) \tag{2.18}$$

and

$$\psi_\pm = \psi_\pm(t) = \beta\ln(1/t) \pm \gamma_\pm. \tag{2.19}$$

From (2.7), we then obtain by an elementary computation

$$\begin{aligned} \operatorname{Re} I_3 &= 2\mathrm{Re}\int_0^1 k_\nu(2t)f(2t)\mathrm{d}t \\ &= \Gamma_+\int_0^1 t^{-\alpha}(1+\cos\psi_+)f(2t)\mathrm{d}t + \Gamma_-\int_0^1 t^{\alpha}(1+\cos\psi_-)f(2t)\mathrm{d}t \\ &\quad - \left(\Gamma_+\int_0^1 t^{-\alpha}f(2t)\mathrm{d}t + \Gamma_-\int_0^1 t^{\alpha}f(2t)\mathrm{d}t\right), \\ \operatorname{Im} I_3 &= 2\mathrm{Im}\int_0^1 k_\nu(2t)f(2t)\mathrm{d}t \\ &= \Gamma_+\int_0^1 t^{-\alpha}(1+\sin\psi_+)f(2t)\mathrm{d}t - \Gamma_-\int_0^1 t^{\alpha}(1+\sin\psi_-)f(2t)\mathrm{d}t \\ &\quad - \left(\Gamma_+\int_0^1 t^{-\alpha}f(2t)\mathrm{d}t - \Gamma_-\int_0^1 t^{\alpha}f(2t)\mathrm{d}t\right). \end{aligned} \tag{2.20}$$

Here we need special Gauss formulae on $[0,1]$ for the (nonnegative) weight functions

$$w_+^c(t;\alpha,\beta) = t^{-\alpha}(1+\cos\psi_+(t)), \quad w_-^c(t;\alpha,\beta) = t^{\alpha}(1+\cos\psi_-(t)) \tag{2.21}$$

and the analogous functions w_+^s, w_-^s with the cosine replaced by the sine. The last two integrals in each of (2.20) are agreeable to standard quadrature rules, e.g. the Gauss-Jacobi rule on $[0,1]$ with Jacobi parameters $\alpha^J = 0$, $\beta^J = \pm\alpha$.

We note that in the previous case $\alpha = 0$, one has $\Gamma_+ = \Gamma_- = |\Gamma(1+\mathrm{i}\beta)|/\beta$, and $\gamma_+ = -\gamma_-$, hence $\psi_+(t) = \psi_-(t)$. Since $\gamma := \arg\Gamma(1+\mathrm{i}\beta) = \arg(\mathrm{i}\beta\Gamma(\mathrm{i}\beta)) =$

$\gamma_+ + \pi/2$, one gets $w^c_+(t;0,\beta) = 1+\cos\psi_+ = 1+\cos(\beta\ln(1/t)+\gamma_+) = 1+\sin(\beta\ln(1/t)+\gamma)$, in agreement with (2.15).

The recurrence coefficients for the orthogonal polynomials relative to the weight functions (2.21) (and the companion weight functions with the sine replacing the cosine) can again be generated by the symbolic, variable-precision Chebyshev algorithm. The respective moments are

(2.22)
$$\begin{aligned}
(m^c_k)_+ &:= \int_0^1 t^k\cdot t^{-\alpha}(1+\cos\psi_+)\mathrm{d}t = \frac{1}{k+1-\alpha} + \frac{(k+1-\alpha)\cos\gamma_+ - \beta\sin\gamma_+}{(k+1-\alpha)^2+\beta^2},\\
(m^c_k)_- &:= \int_0^1 t^k\cdot t^{\alpha}(1+\cos\psi_-)\mathrm{d}t = \frac{1}{k+1+\alpha} + \frac{(k+1+\alpha)\cos\gamma_- + \beta\sin\gamma_-}{(k+1+\alpha)^2+\beta^2},\\
(m^s_k)_+ &:= \int_0^1 t^k\cdot t^{-\alpha}(1+\sin\psi_+)\mathrm{d}t = \frac{1}{k+1-\alpha} + \frac{(k+1-\alpha)\sin\gamma_+ + \beta\cos\gamma_+}{(k+1-\alpha)^2+\beta^2},\\
(m^s_k)_- &:= \int_0^1 t^k\cdot t^{\alpha}(1+\sin\psi_-)\mathrm{d}t = \frac{1}{k+1+\alpha} - \frac{(k+1+\alpha)\sin\gamma_- - \beta\cos\gamma_-}{(k+1+\alpha)^2+\beta^2}
\end{aligned}$$

This is implemented, respectively, in the routines `sr_RMKLTp.m`, `sr_RMKLTm.m`, `sr_IMKLTp.m`, and `sr_IMKLTm.m`.

It may be worth observing that the same procedure can be applied to any of the weight functions (2.21) multiplied by a power of t, say t^λ, provided that $\lambda+1-\alpha>0$ for the former, and $\lambda+1+\alpha>0$ for the latter. (Recall that $\psi_\pm$ in (2.21) depend on α through the constants $\gamma_\pm$, so that we cannot simply change α by $\mp\lambda$.)

3 Numerical results for the KL-transform

The routine `KLT.m`, with $\nu = \mathrm{i}\beta$, was tested by the routine `testKLT.m` against most of the exact answers given in the table of [2, Ch. 11, §6]. (In the process we discovered two errors in this table: in #11.303 and #11.304, the factor 1/2 in the arguments of the cosine and sine should be removed; in #11.308, the answer should be multiplied by 1/2.) In most cases, our procedure worked routinely without the need of any intervention, but in some, the integration on the right of (2.13) requires special programming of the function

$$g_\nu(t) = k_\nu(2e^{-t})f(2e^{-t}) \tag{3.1}$$

before Gauss–Laguerre quadrature is applied to it; see, e.g., Examples 3.3 and 3.4.

Example 3.1 $f(x) = J_0(x\sinh\frac{a}{2})$, $a>0$.
Here, the KL-transform (2.1) is (*ibid.*, #11.311)

$$F(\beta) = \frac{\pi}{2\cosh\frac{\pi\beta}{2}}\,P_{-1/2+\mathrm{i}\beta/2}(\cosh a),$$

where $P_{-1/2+\mathrm{i}\tau}(x)$, $x>1$, is Mehler's conical function. (It can be computed by a backward recurrence algorithm based on the three-term recurrence relation for the sequence of conical functions $P^m_{-1/2+\mathrm{i}\tau}$, $m=0,1,2,\ldots$, where

$P^0_{-1/2+\mathrm{i}\tau} = P_{-1/2+\mathrm{i}\tau}$; see [4, pp. 57–59]. For a Matlab implementation, see the routine `mehler.m`.) Some numerical results for $a = 3$ are shown in Table 3.1, where n1, n2, n3 are the number of terms required in Gauss quadrature to obtain 10-decimal values of the integral (2.11), the first integral in (2.12), and (2.13), respectively. (None of these n are optimal, but rather obtained by the particular manner in which n was incremented in our implementation.) The second column contains the computed values of $F(\beta)$ and the third column their absolute errors. Note the disproportionately large expense incurred in computing the third integral (where, as was pointed out, the main difficulty resides). Naturally, as F becomes smaller, the *relative* error becomes larger, but the absolute error remains constant at the level 10^{-11}. In most applications, it is the *absolute* error that matters anyway.

Table 3.1 *Numerical results for Example 3.1*

β	F	err	n1	n2	n3
0.20	9.132165920e–01	2.12e–11	40	40	90
0.50	6.451224110e–01	3.10e–11	40	50	90
1.00	2.111726841e–01	9.55e–12	40	50	90
4.00	4.441944302e–04	3.64e–11	40	40	130
7.00	–6.873660349e–06	2.52e–12	40	40	250
10.00	–2.786930599e–09	1.80e–12	35	40	330
15.00	–1.661335602e–11	4.18e–13	35	45	335

Another interesting and relatively unproblematic case is #11.302 in [2, Ch. 11, §6], our next example.

Example 3.2 $f(x) = e^{-a^2x^2}$, $a > 0$.

The KL-transform here is

$$F(\beta) = \frac{\sqrt{\pi}}{4a}\, e^{1/(2a^2)}\, \frac{K_{\mathrm{i}\beta/2}(1/(8a^2))}{\cosh \pi\beta/2}.$$

Selected results for $a = 1/2, 1, 2, 4$ are shown in Table 3.2. The values of n1, n2, n3 (not shown) are similar to those in Table 3.1.

Example 3.3 $f(x) = x^\lambda e^{-x}$, $\lambda > -1$.

The KL-transform is expressible in terms of the gamma function,

$$F(\beta) = 2^\lambda\, \frac{\Gamma(\lambda+1)}{\Gamma(2\lambda+2)}\, |\Gamma(\lambda+1+\mathrm{i}\beta)|^2.$$

The third integral (2.13) here involves the function

$$f(2e^{-t}) = (2e^{-t})^\lambda e^{-2\exp(-t)},$$

which, evaluated in this way, may produce Not-a-Number (`NaN`) when e^{-t} underflows and λ is not an integer. To avoid this, one should use $(e^{-t})^\lambda = e^{-\lambda t}$ and,

Table 3.2 *Numerical results for Example 3.2*

a	β	F	err	a	β	F	err
0.5	0.8	6.444049245e–01	3.83e–11	2	0.8	2.638058636e–01	4.17e–11
	3.2	2.654701743e–03	8.61e–12		3.2	1.217781405e–04	8.33e–12
	6.4	–6.059005084e–07	1.06e–12		6.4	1.627404962e–07	1.09e–12
	10.0	–1.846755076e–10	2.10e–12		10.0	–1.259474686e–11	2.37e–12
	15.0	–2.532852648e–13	2.53e–13		15.0	3.305082397e–13	3.30e–13
1	0.8	4.595091276e–01	4.48e–11	4	0.8	1.173107607e–01	4.16e–11
	3.2	–9.793367187e–04	1.06e–11		3.2	1.456652714e–04	8.00e–12
	6.4	–2.651650649e–07	7.41e–13		6.4	1.550757649e–08	3.18e–13
	10.0	–5.820973531e–11	2.44e–12		10.0	4.205363155e–12	2.37e–12
	15.0	–2.535061005e–13	2.53e–13		15.0	3.304098625e–13	3.30e–13

in the interest of improved accuracy, combine this exponential with the weight function e^{-t} in (2.13) to obtain, after a simple change of variables,

$$\begin{aligned} &2\int_0^\infty k_{\mathrm{i}\beta}(2e^{-t})f(2e^{-t})e^{-t}\mathrm{d}t \\ &\quad= \sqrt{\frac{\pi}{\beta\sinh(\pi\beta)}}\,\frac{2^{\lambda+1}}{\lambda+1}\int_0^\infty e^{-2\exp(-t/(\lambda+1))}\sin(\beta t/(\lambda+1)+\gamma)e^{-t}\mathrm{d}t. \end{aligned}$$

The integral on the right is now ready for Gauss–Laguerre quadrature. High-order quadrature is required for negative values of λ and large β because of the increased frequency of the oscillations of the sine function.

One might expect, in this example, that the first integral on the right of (2.12) could be computed more efficiently by a Gauss–Jacobi quadrature with Jacobi parameters $\alpha^J = 0$, $\beta^J = \lambda$, but experience has shown that this is not the case.

Example 3.4 $f(x) = \exp(-x - a^2/(2x))/(2x)$, $a > 0$.
Here we have (*ibid.*, #11.301)

$$F(\beta) = K_{\mathrm{i}\beta}^2(a).$$

This is another example where the function $f(2e^{-t})$ in the integral (2.13), for t large, may result in division by zero if not properly programmed. We write

$$f(2e^{-t}) = \tfrac{1}{4}\,e^{-2e^{-t}-(a^2/4)e^t+t}$$

and let the function on the right underflow when t is large. In this way, for $a = 1$, the results in Table 3.3 are obtained.

Among all the other examples we tried, the only one that caused some problem was $f(t) = K_\lambda(t)$ ([2, #11.313]): n-point Gauss–Laguerre quadature of the third integral (2.13) may result in overflow if n is too large. This was dealt with by increasing n in steps of 1 and lowering the accuracy requirement if necessary. For another resolution of the problem, see Example 3.6.

Table 3.3 *Numerical results for Example 3.4*

β	F	err	n1	n2	n3
0.20	1.721480307e–01	1.14e–12	30	45	330
0.50	1.474890388e–01	4.57e–11	30	40	250
1.00	8.376858862e–02	1.61e–12	30	40	290
4.00	4.668685743e–06	2.49e–12	30	40	250
7.00	2.048630917e–11	3.96e–13	30	40	250
10.00	4.383100309e–12	4.37e–12	25	40	210
15.00	1.114846318e–13	1.11e–13	25	45	215

Example 3.5 Evaluation of I_3 by (2.16) for $\beta = 5$ and $f(x) = e^{-x^2}$.

In order to generate the first 40 recurrence coefficients for the weight function $w_\beta = w_+^{\mathrm{c}}(\,\cdot\,;0,\beta)$ by means of the routine `sr_RMKLTp.m`, spotchecking in the range $.01 \le \beta \le 15$ revealed that symbolic computation with 75 digits yields results to Matlab machine precision (cf. the routine `testsr_RIMKLTp.m`). The required Gauss formulae are then easily generated and the difference of the two integrals in (2.16) readily calculated as explained there. The results for I_3, using n-point quadratures, are shown in Table 3.3. Convergence is seen to be quite fast, in contrast to the evaluation by (2.13), which requires some 150 Gauss-Laguerre points for 10-digit accuracy.

Table 3.3 *Gauss quadrature approximations for the integral* I_3

n	I_3
3	1.08... e–05
6	1.032758... e–05
9	1.0327566311... e–05
12	1.0327566314453... e–05
15	1.0327566314453... e–05

Example 3.6 Same as Example 3.5, but for $f(x) = \mathrm{erfc}\sqrt{x}$ and $f(x) = K_\lambda(x)$.

The same approach as in Example 3.5 can be used, provided the singularities in f are properly accounted for. In the case of the complementary error function, there is a square root singularity at zero, since

$$\mathrm{erfc}\sqrt{x} = 1 - \sqrt{x}\,g(x), \quad g(x) = \frac{\mathrm{erf}\sqrt{x}}{\sqrt{x}}\,,$$

with g a smooth (entire) function. Therefore,

$$\int_0^1 w_\beta(t)\mathrm{erfc}\sqrt{2t}\,\mathrm{d}t = \beta_0(w_\beta) - \sqrt{2}\int_0^1 \sqrt{t}\,w_\beta(t)g(2t)\mathrm{d}t,$$

and

$$\int_0^1 \mathrm{erfc}\sqrt{2t}\,\mathrm{d}t = 1 - \sqrt{2}\int_0^1 \sqrt{t}\,g(2t)\mathrm{d}t.$$

Thus, the first integral calls for Gauss quadrature rules relative to the weight function $\sqrt{t}\,w_\beta(t)$, which can be obtained in a manner indicated at the end of §2. The second integral is amenable to Gauss-Jacobi quadrature on $[0,1]$ with Jacobi parameters $\alpha^J = 0$, $\beta^J = 1/2$. The results are very similar in quality to those in Table 3.3.

In the case of the modified Bessel function $f(x) = K_\lambda(x)$, one has $K_\lambda(2t) \sim \frac{1}{2}\Gamma(\lambda)t^{-\lambda}$ as $t \downarrow 0$, so that the first integral in (2.16),

$$\int_0^1 t^{-\lambda} w_\beta(t)[t^\lambda K_\lambda(2t)]\mathrm{d}t,$$

can be computed by Gauss quadrature relative to the weight function $t^{-\lambda}w_\beta(t)$, and the second integral,

$$\int_0^1 t^{-\lambda}[t^\lambda K_\lambda(2t)]\mathrm{d}t,$$

by Gauss–Jacobi quadrature on $[0,1]$, with parameters $\alpha^J = 0$, $\beta^J = -\lambda$. Numerical results for $\lambda = 3/4$ are shown in Table 3.4.

Table 3.4 *Gauss quadrature evaluation of* I_3 *for* $f(x) = K_\lambda(x)$

n	I_3
10	–1.235. . . e–05
20	–1.23388. . . e–05
30	–1.233801. . . e–05
40	–1.2337912. . . e–05

4 Numerical results for the modified KL-transform

Fewer analytic results are known for the modified KL-transform than for the ordinary KL-transform. Some, however, are provided in [9, Tables 1 and 2] for the real transforms F_+ and F_- of (2.3), (2.4). We tested our routine `KLT.m` with $\nu = 1/2 + \mathrm{i}\beta$ against a few of them; see `testMKLT.m`.

One of the difficulties encountered, which is more prominent here than before, is the need for very high-order Gauss–Laguerre quadratures to evaluate the integral on the right of (2.13). For some of the large Gauss nodes τ_k^L, the exponential e^{-t} then may underflow for $t = \tau_k^L$, which causes the computation of $k_\nu(2e^{-t})$ to fail and may also cause problems in the evaluation of $f(2e^{-t})$. Generally, this imposes a limit $n \leq 350$ on the order n of the Gauss–Laguerre formula, and possibly a more severe limitation, depending on the function f.

Example 4.1 $f(x) = \mathrm{erfc}\sqrt{x}$.
In this case ([9, Table 1, #4]),

$$F_+(\beta) = \frac{\pi}{2\sqrt{2}}\,\frac{1}{\cosh(\pi\beta/2)\cosh(\pi\beta)}\,.$$

Our routine yields the results in Table 4.1. It can be seen that the limit $n = 350$

Table 4.1 *Numerical results for Example 4.1*

β	F_+	err	n1	n2	n3
0.20	8.788209728e–01	1.10e–11	30	75	90
0.50	3.341839551e–01	6.06e–11	30	75	90
1.00	3.818710224e–02	5.52e–11	30	80	90
4.00	2.895850464e–08	2.47e–11	30	70	250
7.00	1.506884314e–10	1.51e–10	30	45	350
10.00	2.204738588e–09	2.20e–09	30	40	350
15.00	1.059987616e–11	1.06e–11	35	45	355

on the order becomes active when $\beta \geq 7$. While the absolute error (in column "err") still remains acceptably small, the relative error, as before, is much larger. Indeed, the absolute error is seen to essentially coincide with the computed value of F_+ when $\beta \geq 7$.

Example 4.2 $f(x) = x^\lambda e^{-x}$, $\lambda > -1/2$.
The modified KL-transforms are [9]

$$F_+(\beta) = \frac{\sqrt{\pi}\,|\Gamma(\lambda + \frac{1}{2} + \mathrm{i}\beta)|^2}{2^{\lambda+1}\Gamma(\lambda + \frac{1}{2})}\,, \quad F_-(\beta) = \frac{\sqrt{\pi}\,|\Gamma(\lambda + \frac{1}{2} + \mathrm{i}\beta)|^2}{2^{\lambda+1}\Gamma(\lambda + \frac{3}{2})}\,.$$

Similarly as in Example 3.3, the integral on the right of (2.13), with $\nu = \alpha + \mathrm{i}\beta$, requires reformulation as

$$\begin{aligned} &2\int_0^\infty k_\nu(2e^{-t}) f(2e^{-t}) e^{-t}\mathrm{d}t \\ &= \frac{2^\lambda}{\lambda+1-\alpha}\int_0^\infty [e^{\mathrm{i}\beta t/(\lambda+1-\alpha)}\Gamma(\nu) + e^{-(2\alpha+\mathrm{i}\beta)t/(\lambda+1-\alpha)}\Gamma(-\nu)] \\ &\qquad\qquad \cdot \exp(-2e^{-t/(\lambda+1-\alpha)})e^{-t}\mathrm{d}t \end{aligned}$$

prior to evaluation by Gauss–Laguerre quadrature. Numerical results for $\lambda = \frac{1}{2}$ are shown in Table 4.2. The values of n1, n2, n3 (not shown) are similar to those in Table 4.1, but n3 ≤ 335. As λ is decreased, the restriction on the quadrature

Table 4.2 *Numerical results for Example 4.2*

β	F_+	F_-	err_+	err_-
0.20	5.872477937e–01	1.174495587e–01	3.30e–11	5.73e–12
0.50	4.277369709e–01	2.138684855e–01	1.36e–11	2.68e–11
1.00	1.704689302e–01	1.704689302e–01	2.88e–11	1.88e–11
4.00	5.492426030e–05	2.196971484e–04	2.16e–11	2.09e–11
7.00	7.751168645e–09	5.429512218e–08	5.47e–12	1.33e–12
10.00	3.480986546e–12	1.299855249e–11	2.59e–12	4.06e–12
15.00	2.426814450e–13	–1.216683892e–12	2.43e–13	1.22e–12

order again becomes an issue.

Example 4.3 Evaluation of $\mathrm{Re}\, I_3$ and $\mathrm{Im}\, I_3$ by (2.20) for $\alpha = 1/2$, $\beta = 1$ and $f(x) = e^{-x}$.

This will engage all four weight functions w_+^c, w_-^c, w_+^s, and w_-^s. For the first, we already found in Example 3.5 that symbolic computation with 75 digits is sufficient to obtain the first 40 recurrence coefficients for the respective orthogonal polynomials to Matlab machine precision. We determined that the same is true for the other three weight functions. All quadrature rules required for computing the integrals in (2.20) thus become available up to order 40. Numerical results for n-point quadrature are generated by the routine `testsr_RIMKLTpm.m` and shown in Table 4.3.

Table 4.3 *Gauss quadrature approximations to* $\mathit{Re}\, I_3$ *and* $\mathit{Im}\, I_3$

n	$\mathrm{Re}\, I_3$	$\mathrm{Im}\, I_3$
3	1.00674. . . e–01	2.20061. . . e–01
6	1.006788892637. . . e–01	2.2005787295314. . . e–01
9	1.0067888926385. . . e–01	2.2005787295307. . . e–01
12	1.0067888926385. . . e–01	2.2005787295307. . . e–01

5 The inverse KL- and modified KL-transform

Both transforms, (2.1) and (2.2), admit inversion formulae, namely

$$f(x) = \frac{2}{\pi^2 x}\int_0^\infty K_{\mathrm{i}\beta}(x)\beta\sinh(\pi\beta)F(\beta)\mathrm{d}\beta \tag{5.1}$$

for (2.1), and

$$\begin{aligned} f(x) &= \frac{4}{\pi^2}\int_0^\infty \mathrm{Re}\, K_{1/2+\mathrm{i}\beta}(x)\cosh(\pi\beta)\mathrm{Re}\, F(\beta)\mathrm{d}\beta \\ &= \frac{4}{\pi^2}\int_0^\infty \mathrm{Im}\, K_{1/2+\mathrm{i}\beta}(x)\cosh(\pi\beta)\mathrm{Im}\, F(\beta)\mathrm{d}\beta \end{aligned} \tag{5.2}$$

for (2.2). Equation (5.1) suggests to call

$$\Phi(x) = \int_0^\infty K_{\mathrm{i}t}(x)\varphi(t)\mathrm{d}t \tag{5.3}$$

the *inverse Kontorovich–Lebedev transform* (*inverse KL-transform*). Applied to $\varphi(t) = (2/\pi^2 x)t\sinh(\pi t)F(t)$, where F is given by (2.1), it yields $f(x)$ in (5.1). Likewise,

$$\Phi(x) = \int_0^\infty K_{1/2+\mathrm{i}t}(x)\varphi(t)\mathrm{d}t \tag{5.4}$$

may be called the *inverse modified Kontorovich–Lebedev transform* (*inverse modified KL-transform*). Its real [imaginary] part, applied to $\varphi(t) = (4/\pi^2)\cosh(\pi t)$

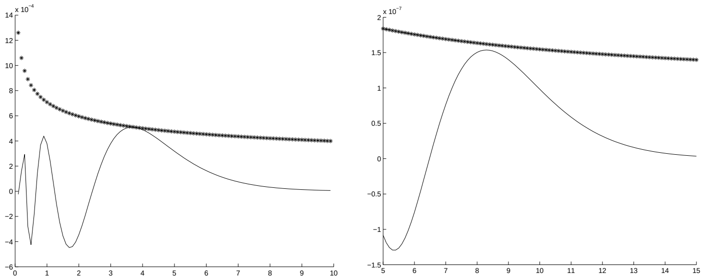
$\operatorname{Re} F(t)$ [$\varphi(t) = (4/\pi^2)\cosh(\pi t)\operatorname{Im} F(t)$], where F is given by (2.2), yields $f(x)$ in (5.2).

Both inverse transforms, (5.3) and (5.4), will be computed by truncating the infinite interval at $t = B$ in such a way that the tail integral from B to ∞ is sufficiently small. For estimating this tail integral, it is important to have available estimates for the respective kernels.

In the case of (5.3), we have an inequality due to Lebedev [7],

$$|K_{\mathrm{i}t}(x)| \le A x^{-1/4} e^{-\pi t/2}, \tag{5.5}$$

where A is a sufficiently large, but unknown, positive constant. In trying to determine this constant for t relatively large, say $t \ge 5$, we noticed that the most critical values of t are those near $t = x$, the turning point of the differential equation satisfied by $K_{\mathrm{i}t}(x)$. This is illustrated for $x = 5$ and $x = 10$ in Fig. 5.1 for the value $A = 1.825$ that we determined. Similar tests, with the same value of A, were conducted for $x = 6 : 9$, $x = 12 : 2 : 18$, and $x = 20 : 10 : 50$. The function $|K_{\mathrm{i}t}(x)|$ for $t \ge x$ then becomes quite small, of the order $10^{-14} - 10^{-35}$ when $20 \le x \le 50$, and will be negligible in most applications. Finally, (5.5) with $A = 1.825$ was verified for $t = 5 : 9$ and $t = 10, 12, 15, 20, 50$, each time with $x = [.001\ .01\ .1\ .5\ 1\ 2\ 5\ 10\ 20\ 50]$.



$x = 5$ $x = 10$

Figure 5.1: $K_{\mathrm{i}t}(x)$ (solid line) and upper bound (5.5) with $A = 1.825$ (starred curve) as functions of t near $t = x$

Inequality (5.5) allows us to estimate the tail integral in (5.3), at least for $B \ge 5$, by

$$\left| \int_B^\infty K_{\mathrm{i}t}(x)\varphi(t)\mathrm{d}t \right| \le A x^{-1/4} \varphi_B, \quad A = 1.825, \tag{5.6}$$

where

$$\varphi_B = \int_B^\infty |\varphi(t)| e^{-\pi t/2} \mathrm{d}t. \tag{5.7}$$

This will enable us to estimate the absolute error incurred if the infinite interval in (5.3) is truncated at B, and also the relative error, if an estimate of $\Phi(x)$ is at hand. The integral from 0 to B in (5.3) is computed by Gauss-Legendre quadrature on $[0, B]$ with relative error tolerance at $\frac{1}{2}\,10^{-8}$, unless stated otherwise.

In the case (5.4), the relevant inequalities are

$$\begin{aligned} &|\mathrm{Re}\, K_{1/2+\mathrm{i}t}(x)| \leq cte^{-\pi t/2}x^{-3/4} + \sqrt{2\pi/x}\, e^{-x}e^{-\pi t}, \\ &|\mathrm{Im}\, K_{1/2+\mathrm{i}t}(x)| \leq c_0 te^{-\pi t/2}x^{-3/4}, \end{aligned} \tag{5.8}$$

which have been derived in [10] with unspecified positive constants c, c_0. Similarly as for (5.5), we found numerical values for these constants, when $t \geq 5$, to be

$$c = .535, \quad c_0 = .482. \tag{5.9}$$

Thus, from (5.4), when $B \geq 5$,

$$\begin{aligned} \left|\int_B^\infty \mathrm{Re}\, K_{1/2+\mathrm{i}t}(x)\varphi(t)\mathrm{d}t\right| &\leq cx^{-1/4}\varphi_{B,0} + \sqrt{2\pi/x}\, e^{-x}\varphi_{B,1}, \\ \left|\int_B^\infty \mathrm{Im}\, K_{1/2+\mathrm{i}t}(x)\varphi(t)\mathrm{d}t\right| &\leq c_0 x^{-3/4}\varphi_{B,0}, \end{aligned} \tag{5.10}$$

where

$$\varphi_{B,0} = \int_B^\infty te^{-\pi t/2}|\varphi(t)|\mathrm{d}t, \quad \varphi_{B,1} = \int_B^\infty e^{-\pi t}|\varphi(t)|\mathrm{d}t. \tag{5.11}$$

This will be useful for estimating the tail integrals of (5.4) for the real and imaginary part. The integrals from 0 to B are computed, as before, by Gauss–Legendre quadrature on $[0, B]$.

The procedure described is implemented in the routines `KLinvT.m`, `RMKLinvT.m`, and `IMKLinvT.m` for, respectively, the inverse KL-transform and the real and imaginary part of the inverse modified KL-transform.

6 Numerical results for the inverse KL-transform

We have tested the procedure outlined in §5 against some of the known transforms in [2, Ch. 11, §5]; see `testKLinv.m`. Performance was as expected and will be illustrated by a few examples.

Example 6.1 $\varphi(t) = \cos(at)$, $a \geq 0$.

The inverse transform is $\Phi(x) = \frac{1}{2}\,\pi e^{-x\cosh a}$ (*ibid.*, #11.273), whereas from (5.7) we have

$$\varphi_B \leq \int_B^\infty e^{-\pi t/2}\mathrm{d}t = \tfrac{2}{\pi}\, e^{-\pi B/2}.$$

With $B = 10$, results for selected values of x and for $a = 1$ and $a = 2$ are shown in Table 6.1. Here, n is the number of Gauss points required for a relative accuracy of $\frac{1}{2}\,10^{-8}$, and err0 is an estimate of the relative error, obtained from (5.6) and Φ, the computed value of $\Phi(x)$. The last three columns show the exact

Table 6.1 *Numerical results for Example 6.1*

x	n	err0	Φ	exact	abserr	relerr
$a = 1$						
0.5	19	2.87e–07	7.26179809e–01	7.26179818e–01	8.65e–09	1.19e–08
3.0	15	8.68e–06	1.53343301e–02	1.53343603e–02	3.01e–08	1.97e–06
5.5	14	3.53e–04	3.23769574e–04	3.23807684e–04	3.81e–08	1.18e–04
8.0	14	1.51e–02	6.87791157e–06	6.83767789e–06	4.02e–08	5.88e–03
10.0	14	2.94e–01	3.34872372e–07	3.12324338e–07	2.25e–08	7.22e–02
$a = 2$						
0.5	23	8.70e–07	2.39424971e–01	2.39424982e–01	1.06e–08	4.44e–08
3.0	21	6.76e–03	1.96791489e–05	1.96977849e–05	1.86e–08	9.46e–04
5.5	21	7.58e+00	−1.50757200e–08	1.62056077e–09	1.67e–08	1.03e+01
8.0	20	5.31e+00	1.96237461e–08	1.33325509e–13	1.96e–08	1.47e+05
10.0	18	4.06e+00	2.42442519e–08	7.19633865e–17	2.42e–08	3.37e+08

answer and the actual absolute and relative errors achieved. Evidently, as the exact answer becomes smaller, the relative accuracy deteriorates, whereas the absolute accuracy stays roughly the same at the level 10^{-8}. It can be seen, however, that our estimate err0 of the relative error is fairly realistic as long as the true relative error is less than 100%. Once it exceeeds that amount (i.e., the exact answer is much smaller), then the estimate err0 is unable to correctly predict it, as is evident in the second half of the table. The estimate (5.6) of the absolute error is still meaningful, however.

Example 6.2 $\varphi(t) = t \cdot \tanh(\pi t) P_{-1/2+\mathrm{i}t}(a)$, $a > 1$; $\Phi(x) = (\pi x/2)^{1/2} e^{-ax}$.

Here, $P_{-1/2+\mathrm{i}t}$ is Mehler's conical function (cf. Example 3.1). From [11, eqn (6)] it is known that for large t,

$$|P_{-1/2+\mathrm{i}t}(a)| \approx \sqrt{\frac{2}{\pi t \sqrt{a^2-1}}}\,.$$

Numerical evidence suggests that the right-hand side is in fact an upper bound, even for t as small as 1. Since $|\tanh(\pi t)| \leq 1$, we can thus estimate

$$\begin{aligned}\varphi_B &= \int_B^\infty |t \cdot \tanh(\pi t) P_{-1/2+\mathrm{i}t}(a)| e^{-\pi t/2} \mathrm{d}t \\ &\leq \sqrt{\frac{2}{\pi\sqrt{a^2-1}}} \int_B^\infty t^{1/2} e^{-\pi t/2} \mathrm{d}t \\ &= \left(\frac{2}{\pi}\right)^2 (a^2-1)^{-1/4}\, \Gamma(3/2, \pi B/2),\end{aligned}$$

where $\Gamma(a, x)$ is the incomplete gamma function. For large B, therefore, using [1, eqn 6.5.32], one finds, at least approximately,

$$\varphi_B \leq \left(\frac{2}{\pi}\right)^{3/2} (a^2-1)^{-1/4} \sqrt{B} e^{-\pi B/2}.$$

Numerical results for $a = 1.5, 2$, and 4 behave very similarly to the ones in Table 6.1.

Example 6.3 $\varphi(t) = t \cdot \tanh(\pi t) K_{\mathrm{i}t}(a)$, $a > 0$.
The inverse transform, according to [2, #11.275], is

$$\Phi(x) = \frac{\pi}{2} \frac{\sqrt{ax}}{a+x} \, e^{-(a+x)}.$$

One computes, using (5.5), that

$$\varphi_B \leq \frac{1.825}{\pi} \, a^{-1/4} e^{-\pi B} \left(B + \frac{1}{\pi} \right).$$

This is much smaller than the φ_B in the previous examples. We have two options: either leave the accuracy requirement as is and lower B; or keep B at 10 and ask for more accuracy. We do the latter and set the relative error tolerance at $\frac{1}{2}\,10^{-12}$. Taking $a = 2$ and $a = 10$, we obtain the results of Table 6.2. As can be seen, the level of relative accuracy is indeed higher, unless x and/or a is large, and thus Φ small.

Table 6.2 *Numerical results for Example 6.3*

x	n	err0	approx/exact	abserr	relerr
			$a = 2$		
0.5	46	4.82e–12	5.157552572933e–02 5.157552572935e-02	2.25e–14	4.37e–13
3.0	45	3.06e–11	5.185051667617e–03 5.185051667590e-03	2.68e–14	5.17e–12
5.5	45	3.55e–10	3.841902525371e–04 3.841902524995e-04	3.77e–14	9.80e–11
8.0	45	4.35e–09	2.852561712095e–05 2.852561716306e-05	4.21e–14	1.48e–09
10.0	45	3.27e–08	3.596829424365e–06 3.596829454192e-06	2.98e–14	8.29e–09
			$a = 10$		
0.5	46	1.80e–08	9.211355009897e–06 9.211355001555e–06	8.34e–15	9.06e–10
3.0	45	7.10e–08	1.495921819728e–06 1.495921786783e-06	3.29e–14	2.20e–08
5.5	42	6.54e–07	1.394457184586e–07 1.394456766782e-07	4.18e–14	3.00e–07
8.0	41	6.99e–06	1.188748391160e–08 1.188753178490e-08	4.79e–14	4.03e–06
10.0	41	4.85e–05	1.618785708968e–09 1.618826269543e-09	4.06e–14	2.51e–05

The next example is a bit more challenging.

Example 6.4 $\varphi(t) = t\sinh(\pi t/2)K_{\mathrm{i}t/2}(a)$, $a > 0$.

This is [2, #11.277] corrected (the sine should be a hyperbolic sine; also, in the transform, $2^{\frac{1}{2}}$ should read $2^{\frac{3}{2}}$; similarly, in the transform of #11.276, $2^{\frac{5}{2}}$ should read $2^{\frac{7}{2}}$). Here, Lebedev's inequality and $|\sinh(\pi t/2)| \le \frac{1}{2}e^{\pi t/2}$ imply

$$|\varphi(t)| \le \tfrac{1}{2}\,Aa^{-1/4}te^{\pi t/4}, \quad t \ge B, \quad A = 1.825,$$

and therefore, by (5.7),

$$\varphi_B \le \frac{2}{\pi}\,Aa^{-1/4}e^{-\pi B/4}\left(B + \frac{4}{\pi}\right).$$

To achieve accuracies comparable to those in the earlier examples, one needs to increase B from the comfortable $B = 10$ to the more laborious $B = 24$. The increased effort required comes from the fact that the evaluation of the integrand in (5.3) involves a good deal of symbolic computation (for the values of $K_{\mathrm{i}t}(x)$ at the Gaussian nodes $t = \tau_k > 10$); moreover, the Gauss formula itself is not as rapidly convergent as it was before. Numerical results are shown for $a = 1$ in Table 6.3.

Table 6.3 *Numerical results for Example 6.4*

x	n	err0	Φ	exact	abserr	relerr
0.5	44	1.18e–06	3.50981025e–01	3.50981023e–01	2.15e–09	6.12e–09
3.0	32	3.38e–08	7.05383495e–01	7.05383499e–01	4.38e–09	6.21e–09
5.5	29	2.26e–07	9.07971240e–02	9.07971075e–02	1.65e–08	1.81e–07
8.0	28	9.60e–06	1.94363824e–03	1.94365628e–03	1.80e–08	9.28e–06
10.0	29	6.54e–04	2.69855461e–05	2.69900886e–05	4.54e–09	1.68e–04

7 A numerical example for the inverse modified KL-transform

For the analytic solution of mixed boundary value problems involving a wedge-shaped domain $D_\alpha = \{(r,\vartheta) :\ 0 \le r < \infty,\ |\vartheta| \le \alpha\}$ (in polar coordinates), $0 < \alpha \le \pi$, integral equations have been proposed whose kernels are given by [8, eqs (3.6) and (3.8)]

$$K(x,y) = \frac{4}{\pi}\int_0^\infty \frac{\sinh((\pi-\alpha)t)}{\sinh(\alpha t)}\,\mathrm{Re}\,K_{1/2+\mathrm{i}t}(x)\mathrm{Re}\,K_{1/2+\mathrm{i}t}(y)\mathrm{d}t \tag{7.1}$$

and a similar expression with cosh in place of sinh. Thus, $K(x,y)$ in (7.1) is the real part of the inverse modified KL-transform (5.4) applied to the function

$$\varphi(t) = \frac{4}{\pi}\,\frac{\sinh((\pi-\alpha)t)}{\sinh(\alpha t)}\,\mathrm{Re}\,K_{1/2+\mathrm{i}t}(y). \tag{7.2}$$

For the special values $\alpha = \pi/n$, $n = 1, 2, \ldots$, of the angle α, the kernel is known explicitly; for example, $K(x,y) = 0$ when $\alpha = \pi$,

$$K(x,y) = K_0(x+y) + K_1(x+y) \quad \text{when } \alpha = \pi/2, \tag{7.3}$$

and

(7.4)
$$K(x,y)=\sqrt{3}K_0(\sqrt{x^2+y^2+xy})+\frac{\sqrt{3}(x+y)}{\sqrt{x^2+y^2+xy}}\,K_1(\sqrt{x^2+y^2+xy}),$$
$$\text{when } \alpha=\pi/3.$$

In order to estimate the quantities $\varphi_{B,0}$, $\varphi_{B,1}$ in (5.11) when applied to the function φ of (7.2), we use

$$\frac{\sinh((\pi-\alpha)t)}{\sinh(\alpha t)}\le\frac{e^{-(2\alpha-\pi)t}}{1-e^{-2\alpha B}}\,,\quad t\ge B,$$

and the first inequality in (5.8) (with x replaced by y) to obtain

$$|\varphi(t)|\le\frac{4}{\pi}\,\frac{e^{-(2\alpha-\pi/2)t}}{1-e^{-2\alpha B}}\left(cty^{-3/4}+\sqrt{\frac{2\pi}{y}}\,e^{-y}e^{-\pi t/2}\right),\quad c=.535. \tag{7.5}$$

The tail integral in (7.1), therefore, can be estimated by the first inequality in (5.10), where

$$\begin{aligned}\varphi_{B,0}\le\frac{4}{\pi}\,\frac{e^{-2\alpha B}}{1-e^{-2\alpha B}}&\left\{\frac{c}{2\alpha}\,y^{-3/4}\left[\left(B+\frac{1}{2\alpha}\right)^2+\frac{1}{4\alpha^2}\right]\right.\\&\left.+\frac{1}{2\alpha+\pi/2}\sqrt{\frac{2\pi}{y}}\,e^{-y}e^{-\pi B/2}\left(B+\frac{1}{2\alpha+\pi/2}\right)\right\}\end{aligned} \tag{7.6}$$

and

$$\begin{aligned}\varphi_{B,1}\le\frac{4}{\pi}\,\frac{e^{-(2\alpha+\pi/2)B}}{1-e^{-\alpha B}}&\left\{\frac{c}{2\alpha+\pi/2}\,y^{-3/4}\left(B+\frac{1}{2\alpha+\pi/2}\right)\right.\\&\left.+\frac{1}{2\alpha+\pi}\sqrt{\frac{2\pi}{y}}\,e^{-y}e^{-\pi B/2}\right\}.\end{aligned} \tag{7.7}$$

In the context of mixed boundary value problems, the integral equation in question lives on the interval $[r_0,\infty)$, and the kernel K therefore on $[r_0,\infty)\times[r_0,\infty)$, where $r=r_0$ is the point on the boundary of D_α at which the type of boundary conditions changes.

Example 7.1 The kernel $K(x,y)$ on $[1,\infty)\times[1,\infty)$ for the domain D_α with $\alpha=\pi/2$.

The routine `RMKLinvT.m` was run in `testRMKLinv.m` to compute $K(x,y)$ to 9 significant digits for $x,y=[1:.2:2,2.5:.5:5,6:10]$, $y\le x$, with the computed answers checked against the exact ones from (7.3). The number of quadrature points never exceeded $n=19$, and the relative [absolute] error observed was generally $10^{-10}-10^{-11}$ $[10^{-12}-10^{-15}]$, but never larger than 3.00×10^{-6} $[1.64\times10^{-10}]$.

Example 7.2 Same as Example 7.1, but for D_α with $\alpha = \pi/3$.

The same computations were performed as in Example 7.1 and the results compared with exact ones from (7.4). The absolute errors observed are similar to those in Example 7.1, the largest being 9.15×10^{-10}. The relative errors are somewhat larger, more in the range $10^{-7} - 10^{-9}$, and as large as 5.71×10^{-3} near $x = y = 10$.

The numerical solution of the integral equation with kernel $K(x, y)$ on $[r_0, \infty)$ will involve quadrature on a truncated interval $[r_0, R]$, $r_0 < R < \infty$. If τ_k, $k = 1, 2, \ldots, n$, are the quadrature nodes, $r_0 \le \tau_1 < \tau_2 < \cdots < \tau_n \le R$, and $\boldsymbol{K}_n = [K(\tau_k, \tau_\ell)]$ is the $(n \times n)$-matrix of the system of linear algebraic equations to be solved, it is useful to know the condition number, $\mathrm{cond}\, \boldsymbol{K}_n$, of the matrix $\boldsymbol{K}_n$.

Example 7.3 The condition of the matrix $\boldsymbol{K}_n$ for Gauss–Legendre quadrature on $[r_0, R]$.

The desired condition number is computed in the routine `condK.m`, which combines our routine `RMKLinvT.m` with the Matlab routine `cond.m`. To our dismay we discovered that $\boldsymbol{K}_n$ is rather ill-conditioned, even for small values of n. This is illustrated in Table 7.1 for $r_0 = 1$ and a few values of α and R. It can be seen that smaller angles α yield better conditioned systems, and larger values of R worse conditioned systems. The condition numbers for equally spaced quadrature nodes are about the same, give or take one order of magnitude. It may well be that for better conditioning, the quadrature nodes should be distributed more densely near the beginning of the interval $[r_0, R]$ and less so near the end. Determining a good, or even best, choice of quadrature points is an opem problem worthy of further study.

Table 7.1 *The condition of the matrix* $\boldsymbol{K}_n$

	$R = 5$			$R = 10$	
n	α	$\mathrm{cond}\, \boldsymbol{K}_n$	n	α	$\mathrm{cond}\, \boldsymbol{K}_n$
2	$\pi/2$	1.89e+03	2	$\pi/2$	4.45e+05
	$\pi/3$	2.61e+02		$\pi/3$	2.31e+04
	$\pi/5$	4.03e+01		$\pi/5$	9.15e+02
5	$\pi/2$	7.72e+08	5	$\pi/2$	1.22e+12
	$\pi/3$	3.70e+06		$\pi/3$	1.14e+09
	$\pi/5$	3.04e+04		$\pi/5$	3.09e+06
8	$\pi/2$	1.08e+13	8	$\pi/2$	8.24e+16
	$\pi/3$	7.21e+10		$\pi/3$	7.85e+14
	$\pi/5$	1.24e+08		$\pi/5$	2.84e+12

Acknowledgments The author thanks Dr. Juri Rappoport for helpful comments, and for providing the author with a copy of Reference [9]. He also is grateful to Oscar Chinellato for assistance in the matter of using the Matlab/Maple interface.

REFERENCES

1. Abramowitz, Milton and Stegun, Irene A. 1964. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, National Bureau of Standards, Applied Mathematics Series 55, U.S. Government Printing Office, Washington, D.C.
2. Ditkin, V. A. and Prudnikov, A. P. 1965. *Integral transforms and operational calculus*, Pergamon Press, Oxford.
3. Dunster, T. M. 1990. Bessel functions of purely imaginary order, with an application to second-order linear differential equations having a large parameter, *SIAM J. Math. Anal. 21*, 995–1018.
4. Gautschi, Walter 1967. Computational aspects of three-term recurrence relation, *SIAM Rev. 9*, 24–82.
5. Gautschi, Walter 2005. Numerical quadrature computation of the Macdonald function for complex orders, *BIT*, to appear.
6. Gautschi, Walter 2005. Computing polynomials orthogonal with respect to densely oscillating and exponentially decaying weight functions and related integrals, *J. Comput. Appl. Math.*, to appear.
7. Lebedev, N. N. 1950. Some integral transforms of mathematical physics (Russian), Dissertation, Leningrad State University, Leningrad. (Unpublished)
8. Lebedev, N. N. and Skal'skaya, I. P. 1974. Dual integral equations connected with the Kontorovich–Lebedev transform (Russian), *Prikl. Mat. Mekh. 38*, 1090–1097.
9. Lebedev, N. N. and Skal'skaya, I. P. 1976. Some integral transforms related to the Kontorovich–Lebedev transform (Russian), in: *Problems of Mathematical Physics* (Russian), Akad. Nauk SSSR, Izdat. Naula, Leningrad, 68–79.
10. Rappoport, Ju. M. 1979. On modified Kontorovich–Lebedev integral transforms and some of their applications (Russian), Akad. Nauk SSSR, Vychisl. Centr, Moscow (42 pp.).
11. Žurina, M. I. and L. N. Karmazina 1962. *Tables of the Legendre function* $P_{\frac{1}{2}+\mathrm{i}\tau}(x)$ (Russian), Vol. II, Izdat. Akad. Nauk SSSR, Moscow.