On the computation of Macdonald functions by numerical quadrature

WALTER GAUTSCHI
Purdue University

Abstract. The use of Gaussian quadrature formulae is explored for the computation of the Macdonald function

$$K_\nu(x) = \int_0^\infty e^{-x\cosh t}\cosh \nu t\,\mathrm{d}t$$

when $x > 0$ and ν is complex, $\nu = \alpha + \mathrm{i}\beta$. It is shown that Gaussian quadrature with weight function $w(t) = \exp(-e^t)$ on $[0,\infty]$ is a viable approach, unless x is small and/or β large, but in combination with Gauss–Legendre quadrature, even in these latter cases.

1. Introduction

The Macdonald function (or modified Bessel function) $K_\nu(x)$ with complex order $\nu = \alpha + \mathrm{i}\beta$ and $x > 0$ is given by

$$\begin{aligned} \operatorname{Re} K_{\alpha+\mathrm{i}\beta}(x) &= \int_0^\infty e^{-x\cosh t}\cosh \alpha t\cos \beta t\,\mathrm{d}t, \\ \operatorname{Im} K_{\alpha+\mathrm{i}\beta}(x) &= \int_0^\infty e^{-x\cosh t}\sinh \alpha t\sin \beta t\,\mathrm{d}t. \end{aligned} \tag{1}$$

Clearly, it suffices to consider $\alpha \geq 0$ and, in view of the recurrence relation for K_ν (which is stable in forward direction of ν), $0 \leq \alpha < 2$. Large values of β give rise to rapid oscillations and hence to numerical difficulties. We shall assume $0 \leq |\beta| \leq 10$.

In trying to use numerical integration in (1), one has to be cognizant of the extremely rapid decay of the first factor in the integrands as $t \to \infty$. We shall deal with this by employing Gaussian quadrature relative to the weight

function $w(t) = \exp(-e^t)$ on $[0, \infty]$. This turns out to be a viable approach when x is not too small, say $x \geq 1$, and β not too large. Naturally, these quadrature formulae are not classical and must be generated numerically. Suitably decomposing the interval of integration into two parts, $[0, \infty] = [0, c] \cup [c, \infty]$, and using Gauss–Legendre quadrature over the first, and the newly generated Gaussian quadratures over the second interval, allows us to deal also with considerably smaller values of x, say $x \geq .0001$.

Parameter values of particular interest are $\alpha = 0$ and $\alpha = \frac{1}{2}$, which yield the kernels in the ordinary and modified Kantorovich–Lebedev integral transforms, respectively [2], [5]. These in turn have been used for the solution of Dirichlet and other boundary value problems in wedge-shaped domains.

In §2 we discuss how the new Gaussian quadrature formulae can be generated. In §3 we develop the computational procedure for $x \geq 1$, and in §5 for $.0001 \leq x \leq 1$. Numerical results will be presented in §§4 and 6, plots of $K_{\mathrm{i}\beta}$ and $K_{1/2+\mathrm{i}\beta}$ in §7, and related software in the Appendix.

2. Gauss quadrature with weight function $w(t) = \exp(-e^t)$ on $[0, \infty]$

We need to construct the orthogonal polynomials with respect to the weight function w, that is, the coefficients α_k, β_k in the three-term recurrence relation

$$\begin{gathered} \pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k \pi_{k-1}(t), \quad k = 0, 1, 2, \ldots, \\ \pi_{-1}(t) = 0, \quad \pi_0(t) = 1, \end{gathered} \tag{2}$$

satisfied by the orthogonal polynomials $\pi_k(\,\cdot\,) = \pi_k(\,\cdot\,; w)$. Once these coefficients are known, the n-point Gauss formula for w,

$$\int_0^\infty f(t)w(t)\mathrm{d}t = \sum_{\nu=1}^{n} \lambda_\nu^G f(\tau_\nu^G) + R_n^G(f), \tag{3}$$

is readily obtained in terms of eigenvalues and eigenvectors of the $n \times n$ Jacobi matrix

$$J_n(w) = \begin{bmatrix} \alpha_0 & \beta_1 & & \\ \beta_1 & \alpha_1 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_{n-1} \end{bmatrix}. \tag{4}$$

Indeed (cf. [4, §3.1.1.1]), τ_ν^G are the eigenvalues of J_n and $\lambda_\nu^G = \beta_0 v_{\nu,1}^2$, where $\beta_0 = \int_0^\infty w(t)\mathrm{d}t$ and $v_{\nu,1}$ is the first component of the normalized eigenvector v_ν corresponding to the eigenvalue τ_ν^G.

To compute $J_n(w)$, we use a multiple-component discretization method ([4, §2.2.4]) based on the decomposition $[0,\infty] = [0,.75]\cup[.75,1.5]\cup[1.5,3]\cup[3,\infty]$ and the use of Fejér quadrature rules as a general-purpose means of discretization. This is implemented in the following Matlab routine using the package `OPQ`; the weight function w has to be properly identified in the routine `quadgp.m`.

```
global mc mp iq idelta irout DM uv AB
N=100; eps0=.5e-12;
mc=4; mp=0; iq=0; idelta=2; Mmax=900;
AB=[[0 .75];[.75 1.5];[1.5 3];[3 Inf]];
[ab,Mcap,kount]=mcdis(N,eps0,@quadgp,Mmax);
```

The choices made of `N` and `eps0` led to `Mcap` =801 and `kount` =6 iterations. The results are stored in the `N`×2 array `abmacdonald`. The routine

```
load -ascii abmacdonald;
xw=gauss(n,abmacdonald);
```

then produces the `n`-point Gauss formula (`n` ≤ `N`) with the nodes and weights stored in the first resp. second column of the `n`×2 array `xw`.

3. The function $K_\nu(x)$ for $x \geq 1$

The rapidly decaying factor $\exp(-x\cosh t)$ in (1), as already mentioned, ought to be treated as a weight function. To avoid its dependence on x, we write

$$e^{-x\cosh t} = e^{-\frac{1}{2}xe^t}\cdot e^{-\frac{1}{2}xe^{-t}} = e^{-\frac{1}{2}x}\cdot e^{-\frac{1}{2}x(e^t-1)}\cdot e^{-\frac{1}{2}xe^{-t}} \tag{5}$$

and define a new variable u by

$$\tfrac{1}{2}x(e^t-1) = e^u - 1, \tag{6}$$

so that $0 \leq u \leq \infty$ when $0 \leq t \leq \infty$. Letting

$$h(u) = 1 + (\tfrac{1}{2}x - 1)e^{-u}, \tag{7}$$

one computes

$$
\begin{aligned}
& e^{-x\cosh t} = e^{1-\frac{1}{2}x} e^{-e^u} e^{-\frac{1}{4}x^2 e^{-u}/h}, \\
& \cosh \alpha t = \tfrac{1}{2}(2h/x)^{\alpha} e^{\alpha u} \left(1 + (2h/x)^{-2\alpha} e^{-2\alpha u}\right), \\
& \cos \beta t = \cos \beta (u + \ln(2h/x)), \\
& \mathrm{d}t = \mathrm{d}u/h,
\end{aligned} \tag{8}
$$

with a similar formula for $\sinh \alpha t$ having a minus sign in place of the plus sign. Consequently, from (1),

$$
\begin{aligned}
& \operatorname{Re} K_{\alpha+\mathrm{i}\beta}(x) = \tfrac{1}{2}(2/x)^{\alpha} e^{1-\frac{1}{2}x} \int_0^{\infty} e^{-e^u} f(u) \mathrm{d}u, \\
& \operatorname{Im} K_{\alpha+\mathrm{i}\beta}(x) = \tfrac{1}{2}(2/x)^{\alpha} e^{1-\frac{1}{2}x} \int_0^{\infty} e^{-e^u} g(u) \mathrm{d}u,
\end{aligned} \tag{9}
$$

where

$$
\begin{aligned}
& f(u) = e^{\alpha u - \frac{1}{4}x^2 e^{-u}/h} h^{-(\alpha+1)} \left(h^{2\alpha} + (xe^{-u}/2)^{2\alpha}\right) \cos \beta (u + \ln(2h/x)), \\
& g(u) = e^{\alpha u - \frac{1}{4}x^2 e^{-u}/h} h^{-(\alpha+1)} \left(h^{2\alpha} - (xe^{-u}/2)^{2\alpha}\right) \sin \beta (u + \ln(2h/x)),
\end{aligned} \tag{10}
$$

with $h = h(u)$ defined in (7). Thus, both integrals in (9) can be evaluated by the Gauss formulae developed in §2.

4. Numerical results for $x \geq 1$

Given $x \geq 1$ and (without restriction of generality) $\beta \geq 0$, we need to determine the smallest order n of the Gauss quadrature rule that yields results to a prescribed relative accuracy ε_0 when applied to (9). For $\varepsilon_0 = \frac{1}{2}\times 10^{-9}$, numerical experiments were conducted with $\alpha = 0 : \frac{1}{2} : 2$, $\beta = 0 : 10$, and selected values of x between 1 and 100. They revealed that relatively low-order Gauss formulae suffice to achieve the desired accuracy. The number n of Gauss points required has consistently been ≤ 29 and usually much smaller. This is illustrated in Table 5.1 for a typical value of β and the two values $\alpha = 0$, $\alpha = \frac{1}{2}$. The integers n_r and n_i refer to real and imaginary part, respectively.

It thus appears that in the range $x \geq 1$, $0 \leq \alpha \leq 2$, $0 \leq \beta \leq 10$, a 30-point Gauss formula will be adequate throughout.

Table 5.1 *Gauss quadrature for the integrals in* (9)

β	α	x	n_r	n_i	$\operatorname{Re} K_{\alpha+\mathrm{i}\beta}(x)$	$\operatorname{Im} K_{\alpha+\mathrm{i}\beta}(x)$
5.00	0.00	1.00	19	2	3.80461828e–04	0.00000000e+00
		5.00	14	2	3.18591025e–04	0.00000000e+00
		10.00	14	2	5.27812177e–06	0.00000000e+00
		20.00	14	2	3.11005908e–10	0.00000000e+00
		50.00	16	2	2.66182488e–23	0.00000000e+00
		100.00	21	2	4.11189777e–45	0.00000000e+00
	0.50	1.00	19	18	6.75850406e–04	2.64552074e–04
		5.00	14	14	2.85418288e–04	1.66486655e–04
		10.00	13	14	5.18618578e–06	1.30924941e–06
		20.00	14	15	3.10593229e–10	3.84530876e–11
		50.00	16	18	2.66517386e–23	1.32270773e–24
		100.00	21	22	4.11574681e–45	1.02447087e–46

The same cannot be said for smaller values of x. Indeed, the required order n of Gauss quadrature increases rapidly with decreasing x. The difficulty is caused, in part, by the behavior of the function $1/h$ in (10), which for small x exhibits a steep boundary layer in the vicinity of $u = 0$.

In the next section we show how the difficulty can be resolved.

5. The function $K_\nu(x)$ for $x < 1$

The idea is to split the integral into two parts and use Gauss–Legendre quadrature on one, and our special Gauss formula on the other. Thus, with c a fixed constant (which will be determined presently), we write

$$\operatorname{Re} K_{\alpha+\mathrm{i}\beta}(x) = \left(\int_0^c + \int_c^\infty\right) e^{-x\cosh t}\cosh\alpha t\cos\beta t\,\mathrm{d}t, \tag{11}$$

and similarly for the imaginary part. In the second integral, we change variables, $t \mapsto \tau + c$, so that

$$\int_c^\infty e^{-x\cosh t}\cosh\alpha t\cos\beta t\,\mathrm{d}t = \int_0^\infty e^{-x\cosh(\tau+c)}\cosh\alpha(\tau+c)\cos\beta(\tau+c)\mathrm{d}\tau.$$

Similarly as in §3, we write the first factor on the right in the form

$$\begin{aligned} e^{-x\cosh(\tau+c)} &= e^{-\frac{1}{2}xe^{\tau}(\cosh c+\sinh c)-\frac{1}{2}xe^{-\tau}(\cosh c-\sinh c)} \\ &= e^{-\frac{1}{2}xe^{c}e^{\tau}}\cdot e^{-\frac{1}{2}xe^{-c}e^{-\tau}}, \end{aligned}$$

or, with

$$\xi = xe^c, \tag{12}$$

in the form

$$e^{-\frac{1}{2}\xi} \cdot e^{-\frac{1}{2}\xi(e^\tau - 1)} \cdot e^{-\frac{1}{2}\xi e^{-2c} e^{-\tau}}.$$

This looks exactly like (5) and suggests the new variable u defined by

$$\tfrac{1}{2}\xi(e^\tau - 1) = e^u - 1. \tag{13}$$

Letting $\eta(u) = 1 + (\frac{1}{2}\xi - 1)e^{-u}$ and carrying out a computation analogous to that in (8) yields

$$\int_c^\infty e^{-x\cosh t} \cosh \alpha t \cos \beta t \, \mathrm{d}t = \tfrac{1}{2}(2/\xi)^\alpha e^{1-\frac{1}{2}\xi + \alpha c} \int_0^\infty e^{-e^u} \varphi(u) \mathrm{d}u,$$

where

$$\varphi(u) = e^{\alpha u - \frac{1}{4}\xi^2 e^{-2c} e^{-u}/\eta} \, \eta^{-(\alpha+1)} \left(\eta^{2\alpha} + (\xi e^{-(u+c)}/2)^{2\alpha}\right) \cos \beta(u + \ln(2\eta/\xi) + c) \mathrm{d}u.$$

Likewise,

$$\int_c^\infty e^{-x\cosh t} \sinh \alpha t \sin \beta t \, \mathrm{d}t = \tfrac{1}{2}(2/\xi)^\alpha e^{1-\frac{1}{2}\xi + \alpha c} \int_0^\infty e^{-e^u} \gamma(u) \mathrm{d}u,$$

where

$$\gamma(u) = e^{\alpha u - \frac{1}{4}\xi^2 e^{-2c} e^{-u}/\eta} \, \eta^{-(\alpha+1)} \left(\eta^{2\alpha} - (\xi e^{-(u+c)}/2)^{2\alpha}\right) \sin \beta(u + \ln(2\eta/\xi) + c) \mathrm{d}u.$$

This is very much like (9) and (10). Since $x = 1$ was an admissible value before, we expect $\xi = 1$ to be admissible now. Therefore, we define c from (12) to be $c = \ln(1/x)$ and obtain

$$\begin{aligned}
\int_c^\infty e^{-x\cosh t} \cosh \alpha t \cos \beta t \, \mathrm{d}t &= \tfrac{1}{2}(2/x)^\alpha e^{\frac{1}{2}} \int_0^\infty e^{-e^u} \varphi_1(u) \mathrm{d}u, \\
\int_c^\infty e^{-x\cosh t} \sinh \alpha t \sin \beta t \, \mathrm{d}t &= \tfrac{1}{2}(2/x)^\alpha e^{\frac{1}{2}} \int_0^\infty e^{-e^u} \gamma_1(u) \mathrm{d}u,
\end{aligned}$$

with

$$\begin{aligned}
\varphi_1(u) &= e^{\alpha u - \frac{1}{4}x^2 e^{-u}/\eta_1} \, \eta_1^{-(\alpha+1)} \left(\eta_1^{2\alpha} + (xe^{-u}/2)^{2\alpha}\right) \cos \beta(u + \ln(2\eta_1/x)), \\
\gamma_1(u) &= e^{\alpha u - \frac{1}{4}x^2 e^{-u}/\eta_1} \, \eta_1^{-(\alpha+1)} \left(\eta_1^{2\alpha} - (xe^{-u}/2)^{2\alpha}\right) \sin \beta(u + \ln(2\eta_1/x)),
\end{aligned} \tag{14}$$

and

$$c = \ln(1/x), \quad \eta_1 = 1 - \tfrac{1}{2}e^{-u}. \tag{15}$$

Eqns (14) are the same as eqns (10), if in the latter h is replaced by η_1.

The first integral in (11) is written as

$$\int_0^c e^{-x\cosh t}\cosh\alpha t\cos\beta t\,\mathrm{d}t = c\int_0^1 e^{-x\cosh(\tau c)}\cosh(\alpha\tau c)\cos(\beta\tau c)\mathrm{d}\tau,$$

and similarly for the imaginary part. Therefore, altogether,

$$\begin{aligned}
\operatorname{Re} K_{\alpha+\mathrm{i}\beta}(x) &= c\int_0^1 e^{-x\cosh(\tau c)}\cosh(\alpha\tau c)\cos(\beta\tau c)\mathrm{d}\tau \\
&\qquad +\tfrac{1}{2}(2/x)^\alpha e^{\frac{1}{2}}\int_0^\infty e^{-e^u}\varphi_1(u)\mathrm{d}u, \\
\operatorname{Im} K_{\alpha+\mathrm{i}\beta}(x) &= c\int_0^1 e^{-x\cosh(\tau c)}\sinh(\alpha\tau c)\sin(\beta\tau c)\mathrm{d}\tau \\
&\qquad +\tfrac{1}{2}(2/x)^\alpha e^{\frac{1}{2}}\int_0^\infty e^{-e^u}\gamma_1(u)\mathrm{d}u,
\end{aligned} \tag{16}$$

with φ_1, γ_1 defined in (14) and c in (15). We now apply Gauss–Legendre quadrature on $[0,1]$ to the first integrals in (16), and our Gauss formula of §2 to the second.

6. Numerical results for $.0001 \le x \le 1$

Numerical experimentation for values of α and β as in §4 and selected values of x between .0001 and 1 revealed that 30-point Gauss rules of both types generally yield the same accuracy achieved earlier, that is, at least 9 significant decimal digits. This is not quite true when β is large. For example, when $9 \le \beta \le 10$ and $x = .001$, we may need as many as 34 Gauss points, and when $7 \le \beta \le 10$ and $x = .0001$ as many as 40, but never more. These higher precisions are required only for the first integrals in (16), i.e., when Gauss–Legendre quadrature is used.

A more serious problem is the fact that the two parts on the right of (16) have a tendency to cancel each other, much more so the larger β. We illustrate this in the typical case of $\beta = 5$ and $\alpha = 0$, $\alpha = \frac{1}{2}$. The degree *can* of cancellation between two numbers is measured by the logarithm (to base 10) of the ratio of the absolutely larger of the two numbers divided by the absolute value of their (algebraic) sum. This measure roughly indicates the

Table 5.2 *Gauss quadrature for the integrals in* (16)

β	α	x	$\mathrm{Re}\, K_{\alpha+\mathrm{i}\beta}(x)$	$\mathrm{Im}\, K_{\alpha+\mathrm{i}\beta}(x)$	*can*
5.00	0.00	1.0000	3.80461828e–04	0.00000000e+00	0.000
		0.5000	–4.24117148e–04	0.00000000e+00	1.804
		0.1000	–2.37141870e–05	0.00000000e+00	3.674
		0.0500	–1.15770402e–04	0.00000000e+00	2.895
		0.0100	–3.89483091e–04	0.00000000e+00	2.406
		0.0010	–3.61340609e–04	0.00000000e+00	1.615
		0.0001	3.20602062e–05	0.00000000e+00	3.549
	0.50	1.0000	6.75850406e–04	2.64552074e–04	0.000
		0.5000	–8.39993536e–04	–5.72771511e–04	1.587
		0.1000	1.47550860e–03	–1.57009337e–03	2.106
		0.0500	–2.70500618e–03	1.43843540e–03	1.862
		0.0100	–2.02652762e–03	–6.58012217e–03	2.424
		0.0010	–2.12376525e–02	–4.73676076e–03	0.368
		0.0001	–4.62274789e–02	5.09643228e–02	2.073

number of decimal digits lost, owing to cancellation. Results are shown in Table 5.2.

Apart from cancellation, the Gauss rules with 30 resp. 40 points yield 12–14 correct digits uniformly for $.0001 \le x \le 1$ and $0 \le \beta \le 9$, suffering a slight loss of accuracy when $\beta = 10$ and $x = .0001$. This higher accuracy of the Gauss quadratures more than compensates for the loss of accuracy caused by cancellation, at least when β is not too large (say, $\beta \le 7$).

A Matlab routine for computing $K_{\alpha+\mathrm{i}\beta}(x)$ is provided in the Appendix.

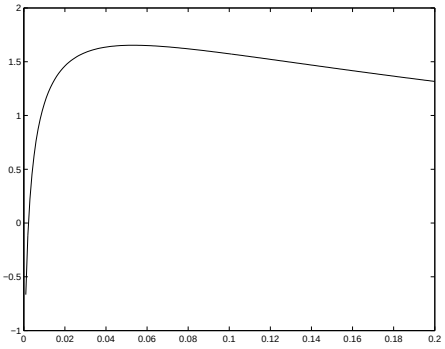
7. Plots of $K_{\mathrm{i}\beta}$ and $K_{1/2+\mathrm{i}\beta}$

While for large x the modified Bessel function $K_\nu(x)$, for all (real or complex) ν exhibits exponential decay [1, eqn 9.7.2],

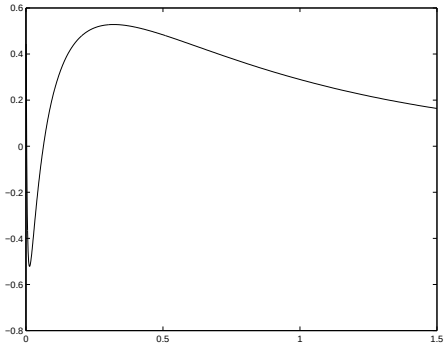
$$K_\nu(x) \sim \sqrt{\frac{\pi}{2x}}\, e^{-x}, \quad x \to \infty, \tag{17}$$

the behavior near $x = 0$ is more complicated.

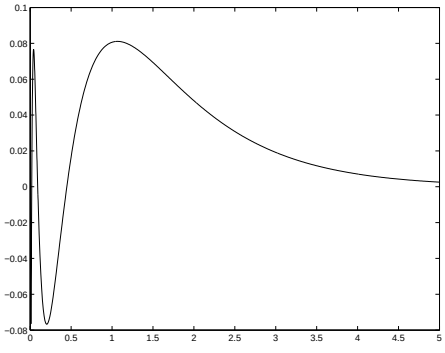
Figure 7.1 *The function* $K_{i\beta}(x)$



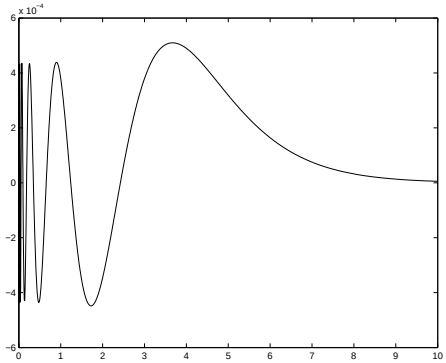
$\beta = 1/2$



$\beta = 1$

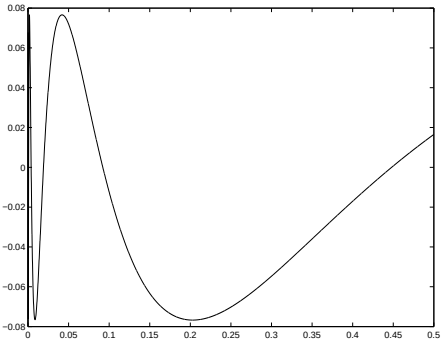


$\beta = 2$



$\beta = 5$

Closeups near $x = 0$ in the cases $\beta = 2$ and $\beta = 5$ are shown below.



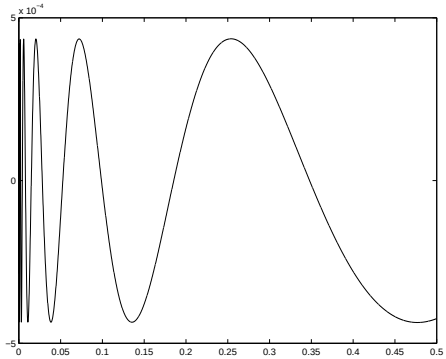
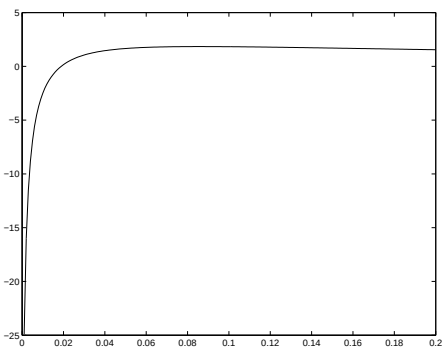
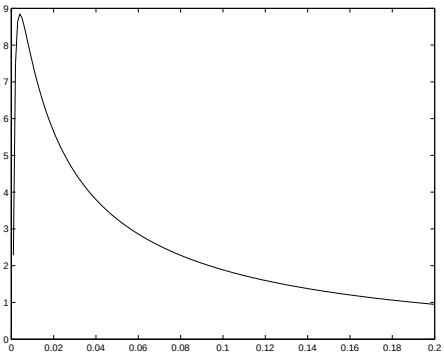
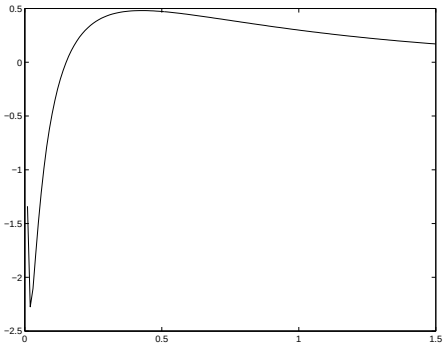


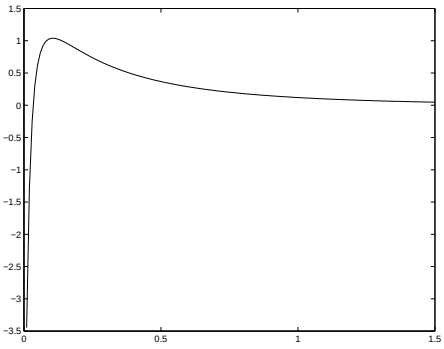
Figure 7.2 *The function* $K_{1/2+\mathrm{i}\beta}(x)$



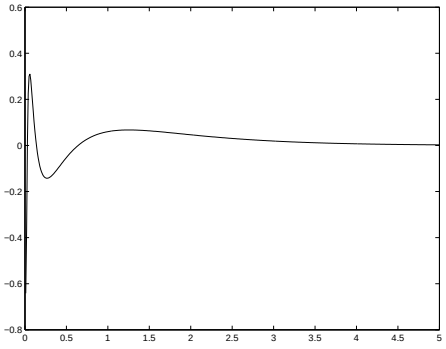


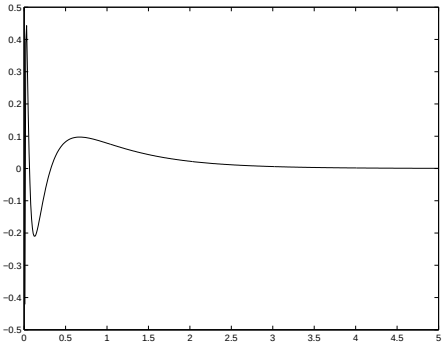
$\mathrm{Re}\, K_{1/2+\mathrm{i}\beta}$ $\beta = 1/2$ $\mathrm{Im}\, K_{1/2+\mathrm{i}\beta}$



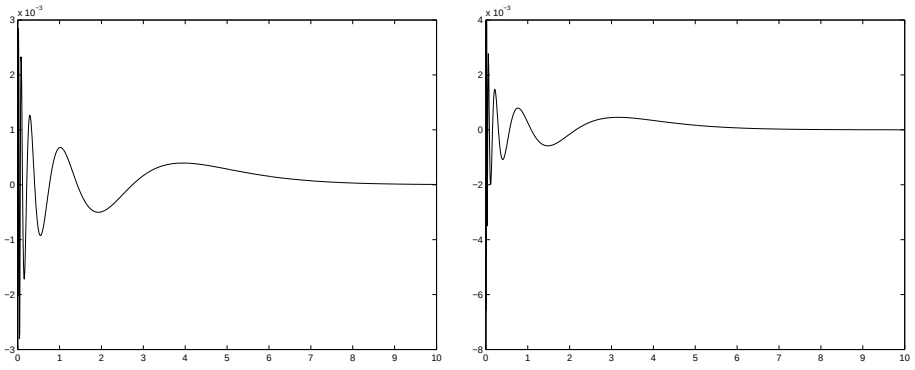


$\mathrm{Re}\, K_{1/2+\mathrm{i}\beta}$ $\beta = 1$ $\mathrm{Im}\, K_{1/2+\mathrm{i}\beta}$





$\mathrm{Re}\, K_{1/2+\mathrm{i}\beta}$ $\beta = 2$ $\mathrm{IM}\, K_{1/2+\mathrm{i}\beta}$



$\operatorname{Re} K_{1/2+\mathrm{i}\beta}$ $\beta = 5$ $\operatorname{Im} K_{1/2+\mathrm{i}\beta}$

Indeed, by [3, eqn (2.14)][1], as $x \downarrow 0$,

$$K_{\mathrm{i}\beta}(x) \sim \sqrt{\frac{\pi}{\beta \sinh \beta\pi}}\, \sin(\beta \ln(2/x) + \arg \Gamma(1+\mathrm{i}\beta)), \quad \beta > 0, \tag{18}$$

while for $\alpha > 0$ ([1, eqn 9.6.9])

$$\begin{aligned} K_{\alpha+\mathrm{i}\beta}(x) \sim \tfrac{1}{2}\, |\Gamma(\alpha+\mathrm{i}\beta)| \left(\tfrac{1}{2}\, x\right)^{-\alpha} [\cos(\beta \ln(2/x) + \arg \Gamma(\alpha+\mathrm{i}\beta)) \\ +\mathrm{i} \sin(\beta \ln(2/x) + \arg \Gamma(\alpha+\mathrm{i}\beta))]. \end{aligned} \tag{19}$$

The behavior in (18) is illustrated in Fig. 7.1 for selected values of β, and the behavior in (19) in Fig. 7.2 for $\alpha = 1/2$ and the same values of β.

The dense oscillations near $x = 0$, typical in all cases, pose interesting questions as to good methods of calculating the Kantorovich–Lebedev integral transforms. We hope to address this problem in future work.

[1] There is a misprint in eqn (2.8) of [3], which leads to eqn (2.14), in that the exponent of $x^2/4$ should be s, not 2.

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. *Integrak transforms and operational calculus*, Pergamon Press, Oxford.

[3] Dunster, T. M. 1990. Bessel functions of purely imaginary order, with an application to secon-order linear differential equations having a large parameter, *SIAM J. Math. Anal. 21*, 995–1018.

[4] Gautschi, Walter 2004. *Orthogonal polynomials: computation and approximation*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford.

[5] Lebedev, N. N. and Skal'skaya, I. P. 1976. Some integral transforms related to the Kantorovich–Lebedev transform in the problems of mathematical physics, *Nauka, Leningr. otd.*, Leningrad.

Appendix Software

```
% MACDONALD The Macdonald function K_{a+ib}(x) for vector-valued x
%     with components >=.0001 and scalar-valued a with 0<=a<=2
%     and b with |b|<=10
%
function [Kr,Ki,can]=macdonald(x,a,b)
s=find(x<0.0001); if isempty(s)==0, error('some x too small'), end
if (a<0|a>2), error('a not in range'), end
if abs(b)>10, error('b not in range'), end
Kr=zeros(size(x)); Ki=zeros(size(x)); can=ones(size(x));
high=find(x>=1); low=find(x<1);
xh=x(high); xl=x(low);
xw=[4.564663380664168e--03 4.287538074680644e--03;
    2.399650066269523e--02 9.741486852990847e--03;
    5.873468399800575e--02 1.464206832501757e--02;
    1.084117044901780e--01 1.869440592845371e--02;
    1.724948111309838e--01 2.165868763097852e--02;
    2.503013975044684e--01 2.336464161681504e--02;
    3.410118509736825e--01 2.373320814115209e--02;
    4.436849907527657e--01 2.279621340294418e--02;
    5.572764794087276e--01 2.070815700299486e--02;
    6.806601527589202e--01 1.774277972108011e--02;
    8.126520036779143e--01 1.426748222579914e--02;
    9.520363367322534e--01 1.069283935734008e--02;
    1.097593374210178e+00 7.403088436218105e--03;
    1.248127372136165e+00 4.684407727030422e--03;
    1.402494146231245e+00 2.675177034817990e--03;
    1.559626865928630e+00 1.358952116388364e--03;
    1.718559093728858e+00 6.039890296625708e--04;
    1.878444350531441e+00 2.305151488151106e--04;
    2.038571973447098e+00 7.396539554555196e--05;
    2.198379685262503e+00 1.948011918039385e--05;
    2.357464114420453e+00 4.096784782359267e--06;
    2.515591563139814e+00 6.663045390029950e--07;
    2.672712867084893e+00 8.066819183114999e--08;
    2.828988875430204e+00 6.936693254682742e--09;
    2.984838567853008e+00 3.989746605007782e--10;
    3.141034650587455e+00 1.415589123973778e--11;
    3.298905708346789e+00 2.756825757932287e--13;
    3.460812119872501e+00 2.447812182923222e--15;
    3.631502065566629e+00 7.035283084505492e--18;
    3.823752912290281e+00 2.818173138652719e--21];
xg=xw(:,1); wg=xw(:,2);
```

```
if isempty(high)==0
   const=.5*(2./xh).^a.*exp(1-.5*xh);
   yr=zeros(size(xh)); yi=zeros(size(xh));
   for nu=1:30
     h=1+(.5*xh-1)*exp(-xg(nu));
     yr=yr+wg(nu)*exp(a*xg(nu)-.25*xh.^2*exp(-xg(nu))./h) ...
       .*h.^(-a-1).*(h.^(2*a)+(.5*xh*exp(-xg(nu))).^(2*a)) ...
       .*cos(b*(xg(nu)+log(2*h./xh)));
     yi=yi+wg(nu)*exp(a*xg(nu)-.25*xh.^2*exp(-xg(nu))./h) ...
       .*h.^(-a-1).*(h.^(2*a)-(.5*xh*exp(-xg(nu))).^(2*a)) ...
       .*sin(b*(xg(nu)+log(2*h./xh)));
   end
   Kr(high)=const.*yr; Ki(high)=const.*yi;
end
if isempty(low)==0
   abl=r_jacobi01(40);
   xwl=gauss(40,abl);
%
% To increase efficiency, the two preceding statements could
% be replaced by the (40x2)-array xwl
%
   const=.5*(2./xl).^a*exp(.5);
   xgl=xwl(:,1); wgl=xwl(:,2);
   yr=zeros(size(xl)); yi=zeros(size(xl));
   for nu=1:30
     h1=1-.5*exp(-xg(nu));
     yr=yr+wg(nu)*exp(a*xg(nu)-.25*xl.^2*exp(-xg(nu))/h1) ...
       *h1^(-a-1).*(h1^(2*a)+(.5*xl*exp(-xg(nu))).^(2*a)) ...
       .*cos(b*(xg(nu)+log(2*h1./xl)));
     yi=yi+wg(nu)*exp(a*xg(nu)-.25*xl.^2*exp(-xg(nu))/h1) ...
       *h1^(-a-1).*(h1^(2*a)-(.5*xl*exp(-xg(nu))).^(2*a)) ...
       .*sin(b*(xg(nu)+log(2*h1./xl)));
   end
   yr=const.*yr; yi=const.*yi;
   c=log(1./xl);
   zr=zeros(size(xl)); zi=zeros(size(xl));
   for nu=1:40
     zr=zr+wgl(nu)*exp(-xl.*cosh(xgl(nu).*c)) ...
       .*cosh(a*xgl(nu).*c).*cos(b*xgl(nu).*c);
     zi=zi+wgl(nu)*exp(-xl.*cosh(xgl(nu).*c)) ...
       .*sinh(a*xgl(nu).*c).*sin(b*xgl(nu).*c);
   end
   zr=c.*zr; zi=c.*zi;
   Kr(low)=yr+zr; Ki(low)=yi+zi;
   can(low)=log10(max(abs(yr),abs(zr))./abs(yr+zr));
end
```