Computer Physics Communications 180 (2009) 332–338

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

NumSBT: A subroutine for calculating spherical Bessel transforms numerically ☆

J.D. Talman

Department of Applied Mathematics, University of Western Ontario, London, Ontario, Canada N6A 5B7

ARTICLE INFO

Article history:
Received 13 May 2008
Received in revised form 26 August 2008
Accepted 5 October 2008
Available online 10 October 2008

PACS:
02.60.Gf
02.60.Jh

Keywords:
Fourier transforms
Spherical Bessel functions
Hankel transforms

ABSTRACT

A previous subroutine, LSFBTR, for computing numerical spherical Bessel (Hankel) transforms is updated with several improvements and modifications. The procedure is applicable if the input radial function and the output transform are defined on logarithmic meshes and if the input function satisfies reasonable smoothness conditions. Important aspects of the procedure are that it is simply implemented with two successive applications of the fast Fourier transform, and it yields accurate results at very large values of the transform variable. Applications to the evaluation of overlap integrals and the Coulomb potential of multipolar charge distributions are described.

Program summary

Program title: NumSBT
Catalogue identifier: AANZ_v2_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AANZ_v2_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: 476
No. of bytes in distributed program, including test data, etc.: 4451
Distribution format: tar.gz
Programming language: Fortran 90
Computer: Generic
Operating system: Linux
Classification: 4.6
Catalogue identifier of previous version: AANZ_v1_0
Journal reference of previous version: Comput. Phys. Comm. 30 (1983) 93
Does the new version supersede the previous version?: No
Nature of problem: This program is a subroutine which, for a function defined numerically on a logarithmic mesh in the radial coordinate, generates the spherical Bessel, or Hankel, transform on a logarithmic mesh in the transform variable. Accurate results for large values of the transform variable are obtained, that would not be otherwise obtainable.
Solution method: The program applies a procedure proposed by the author [1] that treats the problem as a convolution. The calculation then requires two applications of the fast Fourier transform method.
Reasons for new version: The method of computing the transform at small values of the transform variable has been substantially changed and the whole procedure simplified. In addition, the possibility of computing the transform for a single transform variable value has been incorporated. The code has also been converted to Fortran 90 from Fortran 77.
Restrictions: The procedure is most applicable to smooth functions defined on $(0, \infty)$ with a limited number of nodes.
Running time: The example provided with the distribution takes a few seconds to execute.
References:
[1] J.D. Talman, J. Comp. Phys. 29 (1978) 35.

☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

E-mail address: jtalman@uwo.ca.

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2008.10.003

1. Introduction

It is well known that momentum properties of a system described by a wave function $\psi(\mathbf{r})$ are given by the Fourier transform

$$\tilde{\psi}(\mathbf{k}) = \int e^{i\mathbf{k}\cdot\mathbf{r}}\psi(\mathbf{r})\,d\mathbf{r} \tag{1}$$

of the wave function. Thus, for example, the expectation value of any function $F(p)$ of the momentum will be given by (taking $\hbar = 1$)

$$\langle F\rangle = \int F(\mathbf{p})\left|\tilde{\psi}(\mathbf{p})\right|^2 d\mathbf{p}. \tag{2}$$

If ψ is an angular momentum eigenfunction, i.e. $\psi(\mathbf{r}) = f(r)Y_{lm}(\hat{\mathbf{r}})$,

$$\tilde{\psi}(\mathbf{k}) = 4\pi i^l Y_{lm}(\hat{\mathbf{k}})g(k), \tag{3}$$

where

$$g(k) = \int_0^\infty j_l(kr)f(r)r^2\,dr \tag{4}$$

and $j_l(kr)$ is a spherical Bessel function. This integral is a *spherical Hankel* or *spherical Bessel transform* (SBT). A closely related problem is the evaluation of integrals of the form

$$g(k) = \int_0^\infty J_n(kr)f(r)r\,dr \tag{5}$$

which frequently arises in the corresponding two-dimensional case.

In this article, we will be concerned with the numerical evaluation of the SBT for functions f that are defined numerically in terms of their values on a set of mesh points. This is a problem that arises frequently in atomic and molecular calculations.

The straightforward evaluation of the SBT by standard methods of numerical integration presents intractible difficulties if the value of the transform value k becomes large since the integrand is rapidly oscillatory. This precludes the application of methods such as Gauss–Laguerre integration. A variety of approaches have been developed to overcome this problem. For example, the integral can be partitioned into integrations over subintervals between nodal points of j_l. The result is a series that eventually becomes alternating and can be summed with an acceleration method. Evidently, such an approach must be applied separately for each k value.

An effective method for overcoming this problem has been developed by the author [1]. A computer program implementing the method has been subsequently provided [2]. The approach is based on the use of a logarithmic mesh for both the r and k variables. That is, the mesh points are uniformly spaced in $\rho = \ln r$ and $\kappa = \ln k$. The method essentially requires two successive applications of the fast Fourier transform (FFT) and is therefore computationally very efficient.

The original motivation for the development of the approach was to obtain transforms of functions arising in atomic physics. For such problems, the logarithmic mesh is a very convenient choice [3] since the density of mesh points increases at the nucleus. Obviously, this will not be the case for all applications.

The subroutine discussed in [2] has been extensively applied, for example to momentum properties of atomic electrons and to the calculation of molecular multicenter integrals [4,5]. However, in the intervening period a number of modifications have been made to improve the efficiency, transparency and accuracy, and to upgrade the Fortran program to a more contemporary version. As well, time and memory considerations are now less significant. The purpose of this article is to describe these modifications and to provide an improved subroutine.

It should be mentioned that a number of articles have developed or refined methods for the two-dimensional case which is important in signal processing. The thrust of most of these has been to provide a 'fast' algorithm as originally suggested by Siegman [6]. Some of these methods have been reviewed by Secada [7]. It should be mentioned also that an alternative program, FFTLog, is available from A.J.S. Hamilton [8].

2. Theory

As stated in the Introduction, new variables $\rho = \ln r$ and $\kappa = \ln k$ are introduced in Eq. (4) which becomes

$$g(e^\kappa) = \int_{-\infty}^{\infty} j_l(e^{\kappa+\rho})f(e^\rho)e^{3\rho}\,d\rho. \tag{6}$$

The integration in this equation can be recognized as a convolution that can be obtained numerically by computing ordinary Fourier transforms. Explicitly, if

$$\tilde{f}(k) = \int_{\infty}^{\infty} e^{ikx}f(x)\,dx, \tag{7}$$

$$\tilde{g}(k) = \int_{-\infty}^{\infty} e^{ikx}g(x)\,dx \tag{8}$$

then

$$\int_{-\infty}^{\infty} f(x+y)g(y)\,dy = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ikx}\tilde{f}(k)\tilde{g}(-k)\,dk. \tag{9}$$

In the simplest implementation, Fourier transforms of the functions $j_l(e^x)$ and $e^{3x}f(e^x)$ can be evaluated, multiplied, and the inverse transform applied. However, Eq. (6) can be rewritten slightly as

$$g(e^\kappa) = e^{-\alpha\kappa}\int_{-\infty}^{\infty} e^{\alpha(\kappa+\rho)}j_l(e^{\kappa+\rho})e^{(3-\alpha)\rho}f(e^\rho)\,d\rho. \tag{10}$$

The parameter α can be chosen arbitrarily in any interval for which the transforms of the two factors exist. The Fourier transform of $e^{\alpha x}j_l(e^x)$ exists for $-l < \alpha < 2$. In particular, α may depend on the index l, and may depend on the range of k values. The choice of α values has been discussed in a note by van Veldhuizen et al. [9]. Although α is arbitrary, its choice is significant numerically. Any numerical inaccuracy in the implementation is scaled down at large k values and magnified for k near 0 if $\alpha > 0$, and conversely if $\alpha < 0$. In the previous version [2], α was chosen to be $3/2$ for large k values and $3/2 - l$ for k values close to 0. The transition point was chosen as the point of minimum difference of the two evaluations.

In this new version, $\alpha = 3/2$. This choice seems to be rather arbitrary, but is suggested by the observation that the SBT is the inner product of f and j_l on $(0,\infty)$ weighted by r^2 or the L^2 product of $xf(x)$ and $xj_l(x)$.

From the convolution theorem, Eq. (10) can also be written

$$e^{3\kappa/2}g(e^\kappa) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\kappa t}M_l(t)\zeta(t)\,dt, \tag{11}$$

where

$$M_l(t) = \int_{-\infty}^{\infty} e^{-ixt} e^{3x/2} j_l\big(e^x\big)\,dx = \int_0^{\infty} y^{1/2-it} j_l(y)\,dy, \tag{12}$$

$$\zeta(t) = \int_{-\infty}^{\infty} e^{ixt} e^{3x/2} f\big(e^x\big)\,dx. \tag{13}$$

In general, the functions f and g are real and Eq. (11) can be written

$$e^{3\kappa/2} g\big(e^{\kappa}\big) = \frac{1}{\pi} \int_0^{\infty} e^{i\kappa t} M_l(t) \zeta(t)\,dt. \tag{14}$$

The numerical method is then to obtain the transforms $M_l(t)$ and $\zeta(t)$, and compute and Fourier transform their product. The transforms $M_l(t)$ can be obtained analytically, and in a practical application, computed once and stored for future use. The function $\zeta(t)$ is obtained numerically and the FFT procedure used for efficiency.

For $l = 0$ and $l = 1$ Eq. (12) becomes

$$\begin{aligned} M_0(t) &= \int_0^{\infty} z^{1/2-it} j_0(z)\,dz \\ &= \int_0^{\infty} z^{-1/2-it} \sin z\,dz \\ &= \Gamma(1/2-it) \sin\left[\frac{\pi}{2}(1/2-it)\right], \end{aligned} \tag{15}$$

$$\begin{aligned} M_1(t) &= -\int_0^{\infty} z^{1/2-it} j_0'(z)\,dz \\ &= (1/2-it) \int_0^{\infty} z^{-3/2-it} \sin z\,dz \\ &= -(1/2-it) \sin\big[\pi(1/2+it)/2\big] \Gamma(-1/2-it) \\ &= \frac{(1/2-it)\sin[\pi(1/2+it)/2]}{(1/2+it)\sin[\pi(1/2-it)/2]} M_0(t) \\ &= e^{2i\phi} M_0(t), \end{aligned} \tag{16}$$

where $\phi = \tan^{-1}(\tanh(\pi t/2)) - \tan^{-1}(2t)$.

It may be noted that

$$\big|\Gamma(1/2-it)\big| = \frac{\pi}{\cosh \pi t}, \tag{17}$$

$$\left|\sin\frac{\pi}{2}(1/2-it)\right|^2 = \frac{\cosh(\pi t)}{2}, \tag{18}$$

and therefore that

$$M_0(t) = \sqrt{\pi/2} e^{i(\Phi_1+\Phi_2)}, \tag{19}$$

where

$$\Phi_1 = \Im\big[\ln\big(\Gamma(1/2-it)\big)\big], \tag{20}$$

$$\Phi_2 = \Im\big[\ln\big(\sin(1/2-it)\big)\big] = -\tan^{-1}\big(\tanh(\pi t/2)\big). \tag{21}$$

The spherical Bessel functions satisfy

$$\begin{aligned} j_{l-1}(x) &= j_l'(x) + \frac{l+1}{x} j_l(x), \\ j_{l+1}(x) &= -j_l'(x) + \frac{l}{x} j_l(x). \end{aligned} \tag{22}$$

Multiplying by $x^{1/2-it}$, integrating and making an integration by parts gives, for $l > 0$,

$$M_{l-1}(t) = (l+1/2+it) \int_0^{\infty} x^{-1/2-it} j_l(x)\,dx,$$

$$M_{l+1}(t) = (l+1/2-it) \int_0^{\infty} x^{-1/2-it} j_l(x)\,dx,$$

leading to the recurrence relation

$$(l+1/2+it) M_{l+1}(t) = (l+1/2-it) M_{l-1}(t) \tag{23}$$

or

$$M_{l+1}(t) = e^{-2i\phi_l} M_{l-1}(t), \tag{24}$$

where $\phi_l = \tan^{-1}(2t/(2l+1))$.

The function $\Gamma(1/2-it)$ can be readily obtained using the identity

$$\Gamma(N+1/2-it) = \prod_{p=1}^{N} (p-1/2-it) \Gamma(1/2-it) \tag{25}$$

and Stirling's approximation

$$\begin{aligned} \ln\Gamma(z) = (z-1/2)\ln z - z + \ln(2\pi)/2 + \frac{1}{12z} \\ - \frac{1}{360z^3} + \frac{1}{1260z^5} - \frac{1}{1680z^7} + \frac{1}{1188z^9} \cdots. \end{aligned} \tag{26}$$

For $N = 10$, the error in $\Gamma(1/2) = \sqrt{\pi}$ is 1.5×10^{-14}.

If $z = N + 1/2 - it = re^{-i\phi}$, $r = [(N+1/2)+t^2]^{1/2}$, $\phi = \tan^{-1}(2t/(2N+1))$

$$\begin{aligned} &\Im\big[\ln\big(\Gamma(N+1/2-it)\big)\big] \\ &\quad = -t\ln(r) - N\phi + t + \frac{\sin\phi}{12r} - \frac{\sin 3\phi}{360r^3} \cdots. \end{aligned} \tag{27}$$

Then

$$\begin{aligned} \Phi_1 &= \Im\ln\big(\Gamma(1/2-it)\big) \\ &= \Im\ln\big(\Gamma(N+1/2-it)\big) + \sum_{p=1}^{N} \tan^{-1}\big(2t/(2p+1)\big). \end{aligned} \tag{28}$$

3. Implementation

The input to the subroutine is a vector of r values, r_m, a vector of function values, $f_m = f(r_m)$ and a vector of k values, k_p. The output is a vector of transform values $g_p = g(k_p)$. The r values are given by $r_m = e^{\rho_m}$ where $\rho_m = \rho_{\min} + (m-1)\Delta\rho$. The k values are $k_p = e^{\kappa_p}$ where $\kappa_p = \kappa_{\min} + (p-1)\Delta\kappa$. As mentioned in the Introduction, a very substantial improvement in the efficiency is achieved by applying the fast Fourier transform algorithm. However, this requires a constraint on the application that $\Delta\rho = \Delta\kappa$.

The integrals in Eqs. (13) and (14) are computed numerically using the trapezoidal rule on intervals $(\rho_{\min}, \rho_{\max})$ and $(0, t_{\max})$, respectively. The mesh points for the intermediate variable t are then $t_n = n\Delta t$, $0 \leqslant n \leqslant N-1$. The integral of Eq. (13) is to be approximated as

$$\begin{aligned} \zeta_n &= \int_{-\infty}^{\infty} e^{ixt_n} e^{3x/2} f\big(e^x\big)\,dx \approx \int_{\rho_{\min}}^{\rho_{\max}} e^{ixt_n} e^{3x/2} f\big(e^x\big)\,dx \\ &\approx \Delta\rho \sum_{m=1}^{N} e^{i(\rho_{\min}+(m-1)\Delta\rho)n\Delta t} r_m^{3/2} f_m \\ &= \Delta\rho e^{in\rho_{\min}\Delta t} \sum_{m=1}^{N} e^{i(m-1)n\Delta t\Delta\rho} r_m^{3/2} f_m. \end{aligned} \tag{29}$$

The FFT procedure applied in this program evaluates

$$y_m = \sum_{n=0}^{N-1} e^{2\pi imn/N} x_n, \tag{30}$$

where $N = 2^p$ in $\sim pN$ operations. Comparison of the last two equations shows that Δt is constrained by $\Delta\rho\Delta t = 2\pi/N$.

Likewise, the integral in Eq. (14) is approximated as

$$\int_0^\infty e^{i\kappa t} M_l(t)\zeta(t)\,dt \approx \int_0^{T/2} e^{i\kappa t} M_l(t)\zeta(t)\,dt \approx \Delta t \sum_{n=0}^{N/2-1} e^{i\kappa t_n} M_l(t_n)\zeta(t_n). \tag{31}$$

This leads to the evaluation of Eq. (14) as

$$\begin{aligned} g_p &= \frac{1}{k_p^{3/2}\pi}\Delta t \sum_{n=0}^{N/2-1} e^{i(\kappa_{\min}+(p-1)\Delta\kappa)n\Delta t} M_l(t_n)\zeta_n \\ &= \frac{\Delta t\Delta\rho}{k_p^{3/2}\pi} \sum_{n=0}^{N/2-1} e^{i(\kappa_{\min}+(p-1)\Delta\kappa)n\Delta t} M_l(t_n) e^{in\rho_{\min}\Delta t} \\ &\quad \times \sum_{m=0}^{N-1} e^{i(m-1)n\Delta t\Delta\rho} r_m^{3/2} f_m \\ &= \frac{2}{Nk_p^{3/2}} \sum_{n=0}^{N/2-1} e^{2\pi i(p-1)n/N} \left[e^{i(\kappa_{\min}+\rho_{\min})n\Delta t} M_l(t_n) \right. \\ &\quad \left. \times \sum_{m=1}^{N} e^{2\pi i(m-1)n/N} r_m^{3/2} f_m \right]. \end{aligned} \tag{32}$$

Therefore the algorithm proceeds as follows:

- Multiply f_m by $r_m^{3/2}$.
- Obtain the FFT of the resulting array to obtain an array y_n.
- Multiply y_n by $e^{i(\rho_{\min}+\kappa_{\min})n\Delta t} M_l(t_n)$ and set $y_n = 0$ for $n > N/2$.
- Obtain the FFT to obtain an array z_k.
- Multiply by z_k by $2/Nk^{3/2}$ to obtain the final result.

Values of $e^{i(\rho_{\min}+\kappa_{\min})p\Delta t} M_l(t_n)$ are computed once on the initial call to the subroutine and saved for subsequent calls.

An important point is that the intermediate factor $e^{i(\rho_{\min}+\kappa_{\min})n\Delta t} M_l(t_n)$, which is computed on the initial call to the subroutine and stored, depends only on $\rho_{\min}+\kappa_{\min}$. Therefore, the same set of intermediate factors can be used for both the direct and inverse transforms.

4. Numerical considerations

It is probably not feasible to give a rigorous derivation of error bounds on the procedure for a general input function, but it is possible to make some general observations. The Euler–Maclaurin summation formula [10]

$$\begin{aligned} \int_0^{x_n} f(x)\,dx &= h\left[\frac{1}{2}f_0 + f_1 + \cdots + f_{n-1} + \frac{1}{2}f_n\right] \\ &\quad - \frac{B_2h^2}{2!}(f_n' - f_0') \cdots \\ &\quad - \frac{B_{2k}h^{2k}}{(2k)!}\left(f_0^{(2k-1)} - f_n^{(2k-1)}\right) + R_{2k}, \end{aligned} \tag{33}$$

where $R_{2k} = O(h^{2k+2}\max|f^{(2k-1)}|)$ and B_{2k} are Bernoulli numbers, suggests that the error is largely determined by the values of the integrand at the end points and the numerical evidence confirms this. In general, $f(r)$ behaves as r^l at small r and the $l = 0$ or s-wave case is the most difficult. In the present implementation, $f(r)$ is extrapolated to N values of r in the interval $(\rho_{\min} - N\Delta\rho, \rho_{\min})$, i.e. the number of mesh points is doubled. The extrapolation is of the form Cr^{l+n}, where n is an additional input parameter. Normally, $n = 0$, but can be chosen differently in special cases, such as, e.g., a Yukawa potential. This doubling of the radial mesh has the additional effect of reducing Δt by $1/2$. The additional N transform values at larger k values generated in this way are probably outside the range of physical interest and are not returned from the subroutine. This doubling procedure is akin to the padding process often used in applications of the FFT to convolutions.

This approach is found to yield accurate results except in the case that k is very near the origin, where, as observed above, the final division by $k_n^{3/2}$ magnifies the errors. This problem is readily avoided, at the expense of a less efficient procedure by applying Eq. (10) with $\alpha = 0$. In this case, the Fourier transform of $j_0(e^\rho)$ does not exist. However, Eq. (10) can be discretized directly as

$$g\left(e^{\kappa_m}\right) \approx \Delta\rho \sum_{n=1}^{N} j_l\left(e^{\kappa_{\min}+\rho_{\min}+(m+n-2)\Delta\rho}\right) e^{3\rho_n} f\left(e^{\rho_n}\right), \tag{34}$$

where $1 \leqslant m \leqslant N$. Eq. (34) is a finite convolution and can be evaluated as the discrete analog of Eq. (9). Explicitly, the sum is of the form

$$s_m = \sum_{n=1}^{N} a_{m+n} b_n, \tag{35}$$

where $a_n = j_l(e^{\kappa_{\min}+\rho_{\min}+(n-2)\Delta\rho})$, $1 \leqslant n \leqslant 2N$, $b_n = e^{3\rho_n} f(e^{\rho_n})$, $1 \leqslant n \leqslant N$. This can be expressed as

$$s_m = \frac{1}{2N} \sum_{k=0}^{2N-1} e^{2\pi imk/2N} \hat{a}_k^* \hat{b}_k, \tag{36}$$

where

$$\hat{a}_k = \sum_{n=0}^{2N-1} e^{2\pi ikn/2N} a_{n+1}, \tag{37}$$

$$\hat{b}_k = \sum_{n=0}^{N-1} e^{2\pi ikn/2N} b_{n+1}. \tag{38}$$

Again, the summations can be carried out using the FFT of dimension $2N$. In fact, since the inputs are real, the dimension can be halved. However, this has not been implemented in the procedure. This direct integration approach is that proposed by Siegman [6] for cylindrical Bessel functions.

The transition point between the two methods is chosen, as in the previous version, to be the point at which their absolute difference is minimum.

Values of $j_l(x)$ are obtained using the relation

$$j_{l-1}(x) - \frac{2l+1}{x} j_l(x) + j_{l+1}(x) = 0 \tag{39}$$

with the starting values $j_0(x) = \sin x/x$, $j_1(x) = [j_0(x) - \cos x]/x$. For small x this relation is unstable. It can be estimated that the onset of this instability is at $x \approx 2l_{\max}/e$. Therefore, for $x < 0.75l_{\max}$ the ratio $j_{l_{\max}-1}(x)/j_{l_{\max}}(x)$ is obtained using a continued fraction approach based on Eq. (39) as described by Press et al. [11] and the recursion is made downwards. The sequence is finally adjusted using the explicit value of $j_0(x)$.

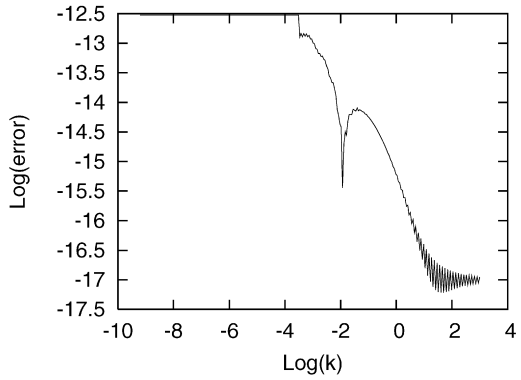

Fig. 1. The logarithm to the base 10 of the error in the transform of the Slater function defined in Eq. (40) with $\lambda = 1$.

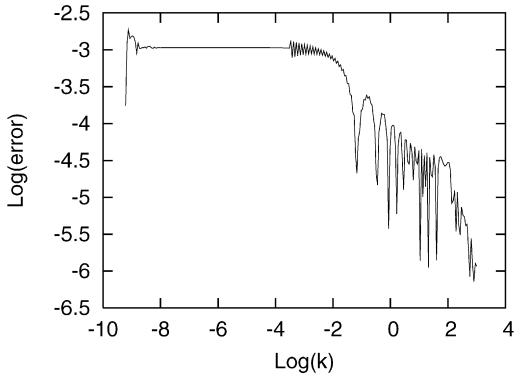

Fig. 2. The logarithm to the base 10 of the error in the transform of the function defined in Eq. (42) with $a = 1$.

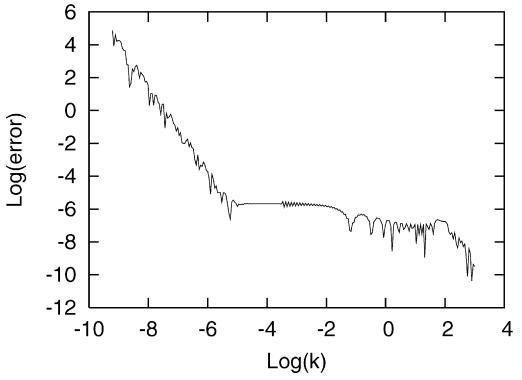

Fig. 3. The logarithm to the base 10 of the error in the transform of the function defined in Eq. (44) with $a = 1$. The values shown for $k < 10^{-6}$ are spurious.

5. Single *k* result

In the usual situation, the transform results will be obtained on the mesh of k values. However, in some instances, the result may be required only for a fixed single k value. This can obviously be obtained from results on the mesh by interpolation. In the present implementation, the possibility of evaluating the integral representation of Eq. (14) for an explicit k value has been included. This is not, however, substantially more efficient than evaluating the entire array.

It should be noted that Eq. (36) is valid only for k values at the mesh points and fails badly at intermediate values. Therefore, for the purpose of generality, the full calculation is carried out to determine the transition point. For a k value less than the transition point, the numerical integration is carried out directly.

This is clearly not an efficient procedure in the case of small k values, and is included only for completeness.

6. Applicability

As with all numerical procedures, the validity of the method is strongly dependent on the analytic properties of the functions being approximated. The ideal input function for the present approach is one analytic on $[0, \infty)$, with exponential decrease for $r \to \infty$, r^l behavior at $r = 0$ and a small number of nodes. These conditions obviously describe bound state solutions of the radial Schrödinger equation, or products of such solutions.

The behavior of the error for such a function is illustrated in Fig. 1 which shows the logarithm to the base 10 of the absolute value of the error in the case that

$$f(r) = \frac{\lambda^3}{2} e^{-\lambda r} \tag{40}$$

for which

$$g(k) = \frac{\lambda^4}{(k^2 + \lambda^2)^2}. \tag{41}$$

In this, and the following examples, the functions are normalized so that $g(0) = 1$. The results indicate that for such functions, the procedure should provide adequate accuracy for practically all applications.

Exponential decrease is not an essential condition. The procedure can clearly be applied to functions that decrease as r^{-p} provided that $r_{\max}^{3/2-p}$ is sufficiently small. A situation where this may not be the case is that of the inverse transform an s-type orbital function with a cusp at the origin for which the transform behaves as k^{-4}. However, the example in Section 8 of the potential calculation for such a function in which the integrand decreases as k^{-6} demonstrates a satisfactory application.

The procedure can obviously be applied to functions not meeting the above conditions, but in general there may be a considerable degradation in accuracy. For example, certain applications use functions with finite support which usually have a discontinuous derivative of some order. An extreme case of such a function, with a discontinuous derivative, is

$$f(r) = \begin{cases} 12(a-r)/a^4, & r \leqslant a, \\ 0, & r > a. \end{cases} \tag{42}$$

for which

$$g(k) = 24\left(1 - \cos(ak) - \frac{ak}{2}\sin(ak)\right)\Big/(ak)^4. \tag{43}$$

The error in this case is shown in Fig. 2. In this case, the procedure provides only a rough estimate of the transform.

A less extreme example is the function with a discontinuous third derivative given by

$$f(r) = \begin{cases} 60(a-r)^3/a^6, & r \leqslant a, \\ 0, & r > a. \end{cases} \tag{44}$$

for which

$$g(k) = 360\big[(ka)^2 + 4\big(\cos(ka) - 1\big) + (ka)\sin(ka)\big]/k^6. \tag{45}$$

The corresponding error is shown in Fig. 3. (The error for k close to the origin is spurious and arises from cancellation errors in the evaluation of the analytic expression in Eq. (45).) The results indicate that for many purposes the procedure can produce satisfactory accuracy for functions with discontinuities in the third derivative.

These calculations were made with $\rho_{\min} = -9.0$, $\rho_{\max} = \ln(40.0)$, $\delta\rho \approx 0.05$. The error in the third calculation (Eq. (44)) scales as $\delta\rho^4$, while the error in the Slater orbital calculation does not appear to follow any simple prescription.

7. Program description

The subroutine is called NumSBT. It can be called as

CALL NumSBT(FF,GG,RR,KK,LL,NP,NEXP,NR,FIXEDK,SGG,SINGLEK)

with the following meanings.

- FF—REAL. The array of the input function values
- GG—REAL. The array of the transform values
- RR—REAL. The array of the r values
- KK—REAL. The array of the k values
- LL—INTEGER. The l value
- NP—INTEGER. Normally NP = 0. If $g(r) \propto r^{l+n}$ for $r \to 0$, NP = n
- NEXP—INTEGER. The array dimensions are 2**NEXP
- NR—INTEGER. NR=2**NEXP
- FIXEDK—REAL. The single input k value if SINGLEK
- SGG—REAL. The result in the case of a single k
- SINGLEK—LOGICAL

If SINGLEK is .TRUE. only the transform value at the single k value FIXEDK is returned as SGG.

The subroutine calls a generic subroutine NLOGN to compute the FFTs. This is included for completeness. However, in the case of a very large scale application, it may be useful to use a more refined library FFT algorithm. The program also contains a subroutine XJL that computes a table of values of $j_l(x)$ for a fixed x.

On the first call to the subroutine for a particular set of mesh parameters, values of $M_l(t_n), r_n^{3/2}, k_n^{-3/2}$, etc. are computed and stored for $0 \leqslant l \leqslant l_{\max}$ for subsequent application. If the mesh parameters change, these stored values are recomputed. However, as noted above, if $\rho_{\min} + \kappa_{\min}$ is unchanged, this is unnecessary. The value of $l_{\max}$ is set at 20 and the maximum mesh size is set at 512. However, these values can readily be changed before compilation. The procedure stops with an error message if these values are exceeded. These initializations are performed in a subroutine INITIALIZE. Real and complex variables are defined in 8-byte arithmetic (kind = 8).

The package includes an illustrative program to compute an overlap integral as described in the next section. This program computes the overlap integral of two s-type Slater orbitals and compares the result with the exact result.

This illustrative program can be run interactively, calling for the input of parameters a and b, and a separation R.

8. Sample applications

If $f_{l_1 m_1}(\mathbf{r})$ and $g_{l_2 m_2}(\mathbf{r})$ are angular momentum eigenfunctions of the form $f_l(r)Y_{lm}(\hat{\mathbf{r}})$, the overlap integral

$$I(\mathbf{r}) = \int g_{l_2 m_2}(\mathbf{r} - \mathbf{R})^* f_{l_1 m_1}(\mathbf{r})\, d\mathbf{r} \tag{46}$$

is given in terms of $3-j$ coefficients by

$$\begin{aligned} I(\mathbf{r}) = {} & 8 \sum_{LM} i^{l_2+L-l_1} \left[\frac{(2l_1+1)(2l_2+1)(2L+1)}{4\pi}\right]^{1/2} \\ & \times (-1)^{m_1} \begin{pmatrix} l_1 & l_2 & L \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l_1 & l_2 & L \\ -m_1 & m_2 & M \end{pmatrix} Y_{LM}(\hat{\mathbf{R}}) \\ & \times \int_0^\infty j_L(kR)\tilde{g}_{l_2}(k)\tilde{f}_{l_1}(k)k^2\, dk, \end{aligned} \tag{47}$$

where $\tilde{g}_{l_2}(k)$ and $\tilde{f}_{l_1}(k)$ are the transforms of $g_{l_2}(r)$ and $f_{l_1}(r)$.

In the case of s functions, $l_1 = l_2 = 0$, this reduces to

$$I(R) = \frac{2}{\pi} \int_0^\infty j_0(kR)\tilde{g}(k)\tilde{f}(k)k^2\, dk. \tag{48}$$

If f and g are Slater functions, $f(r) = e^{-ar}$, $g(r) = e^{-br}$, the integral is given by

$$\begin{aligned} I(R) = {} & \frac{8\pi}{(b^2-a^2)^2}\left(ae^{-bR} + be^{-aR}\right) \\ & - \frac{32\pi ab}{R(b^2-a^2)^3}\left(e^{-aR} - e^{-bR}\right). \end{aligned} \tag{49}$$

In the case $a = b$, the integral is given by

$$I(R) = \pi\left(1 + aR + (aR)^2/3\right)e^{-aR}. \tag{50}$$

A second example is provided by the evaluation of the potential produced by a multipole charge distribution $\rho_{lm}(\mathbf{r}) = \rho_l(r)Y_{lm}(\hat{\mathbf{r}})$ which is required in large scale molecular structure calculations, in particular DFT calculations. The potential is given by

$$\begin{aligned} V(\mathbf{r}) = {} & \int \frac{1}{|\mathbf{r}-\mathbf{r}'|}\rho_{lm}(\mathbf{r}')\, d\mathbf{r}' \\ = {} & \frac{1}{2l+1}\left[\frac{1}{r^{l+1}}\int_0^r r'^{l+2}\rho_l(r')\, dr' + r^l \int_r^\infty r'^{-l+1}\rho_l(r')\, dr'\right] Y_{lm}(\hat{\mathbf{r}}) \\ = {} & V_l(r)Y_{lm}(\hat{\mathbf{r}}). \end{aligned} \tag{51}$$

The integration is a convolution, and therefore can also be obtained in momentum space. Explicitly,

$$V_l(r) = \frac{2}{\pi}\int_0^\infty j_l(kr)\tilde{\rho}_l(k)\, dk, \tag{52}$$

where

$$\tilde{\rho}_l(k) = \int_0^\infty j_l(kr)\rho_l(r)r^2\, dr. \tag{53}$$

It appears that the potential can be readily obtained by transforming $\rho_l(r)$ to momentum space, dividing by k^2 and making a second transformation. For small l values, in particular $l = 0$, this will not succeed since the integrand does not vanish at $k = 0$. This problem can be remedied by subtracting from $\tilde{\rho}_l(k)$ a function with known transform, and the same behavior at $k = 0$.

A possible choice for such a function is

$$\rho_0(r) = r^l e^{-\alpha r}. \tag{54}$$

The corresponding transform is given by

$$\tilde{\rho}_0(k) = \frac{2^{l+1}(l+1)!\alpha k^l}{(k^2+\alpha^2)^{l+2}} \tag{55}$$

and the potential is given by

$$\begin{aligned} V_0(r,\alpha) = {} & \frac{1}{(2l+1)r^{l+1}}\frac{(2l+2)!}{\alpha^{2l+3}}\left[1 - I_{2l+2}(\alpha r)\right] \\ & + \frac{r^l}{(2l+1)\alpha^2} I_1(\alpha r), \end{aligned} \tag{56}$$

where

$$\begin{aligned} I_n(\alpha r) = {} & \frac{\alpha^{n+1}}{n!}\int_r^\infty r'^n e^{-\alpha r'}\, dr' = \sum_{k=0}^n \frac{(\alpha r)^k}{k!}e^{-\alpha r} \\ = {} & 1 - \sum_{k=n+1}^\infty \frac{(\alpha r)^k}{k!}e^{-\alpha r}. \end{aligned} \tag{57}$$

The second expression may be more suitable for $n > \alpha r$.

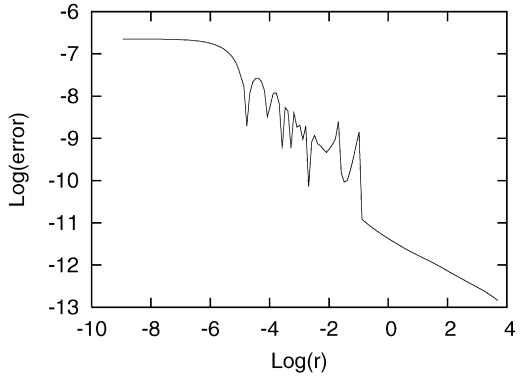

Fig. 4. The logarithm to the base 10 of the error in the potential produced by the charge density of Eq. (61) with $l = 0$, $\beta = 2$.

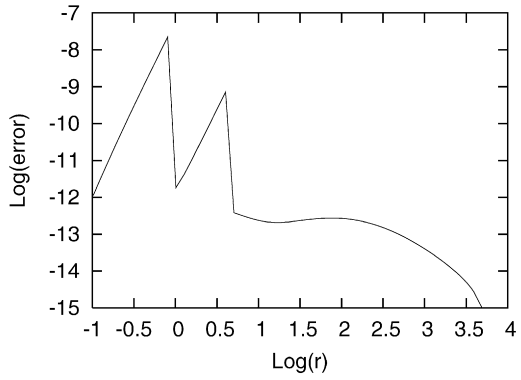

Fig. 5. The logarithm to the base 10 of the error in the potential produced by the charge density of Eq. (61) with $l = 4$, $\beta = 2$.

This suggests a subtraction of

$$\psi(k) = C\frac{k^l}{(k^2+\alpha^2)^{l+2}} \tag{58}$$

with

$$C = \frac{\alpha^{2l+4}}{(2l+1)!!}\int_0^\infty r^{l+2}\rho_l(r)\,dr \tag{59}$$

from $\tilde{\rho}(k)$. The corresponding potential to be added back is then given by

$$\frac{C}{2^{l+1}(l+1)!\alpha}V_0(r,\alpha). \tag{60}$$

The value of α in the counter-term can be chosen arbitrarily, but should conform to the distance scale of the particular system. Fig. 4 shows the error resulting if this method is applied to a unit exponential charge density

$$\rho(\mathbf{r}) = \frac{\beta^{l+3}}{l!}r^l e^{-\beta r} \tag{61}$$

in the case $\beta = 2$ and $l = 0$ and $\alpha = 1$ in the counter-term.

The corresponding error in the potential in the case $l = 4$ is shown in Fig. 5. It is evident that for most purposes this approach is applicable to the evaluation of Coulomb potentials for multipole charge distributions.

Acknowledgements

This work was initiated at the MPIPKS, Dresden. I am indebted to Professor Peter Fulde for hospitality at the Institute and to Professor Dietrich Foerster of the University of Bordeaux for many discussions. This research was supported by the Natural Sciences and Engineering Research Council of Canada and the facilities of the Shared Hierarchical Academic Research Computing Network.

References

[1] J.D. Talman, J. Comput. Phys. 29 (1978) 35.
[2] J.D. Talman, Comput. Phys. Comm. 30 (1983) 93.
[3] C. Froese Fischer, The Hartree–Fock Method for Atoms: A Numerical Approach, Wiley-Interscience, New York, 1977.
[4] T. Koga, Ajit J. Thakkar, J. Phys. B: At. Mol. Opt. Phys. 29 (1996) 2973.
[5] J.D. Talman, Int. J. Quantum Chem. 93 (2003) 72.
[6] A.E. Siegman, Opt. Lett. 1 (1977) 13.
[7] J.D. Secada, J. Comput. Phys. 116 (1999) 278.
[8] A.J.S. Hamilton, Mon. Not. R. Astron. Soc. 312 (2000) 257.
[9] M. van Veldhuizen, R. Nieuwenhuizen, W. Zul, J. Comp. Phys. 110 (1994) 196.
[10] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions, Dover, New York, 1965 (Eq. (25.4.7)).
[11] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes, Cambridge University Press, Cambridge, 1986.