Fast spherical Bessel transform via fast Fourier transform and recurrence formula

Masayuki Toyoda*, Taisuke Ozaki

Research Center for Integrated Science, Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Nomi, Ishikawa 932-1292, Japan.

DOI: 10.1016/j.cpc.2009.09.020.

Abstract

We propose a new method for the numerical evaluation of the spherical Bessel transform. A formula is derived for the transform by using an integral representation of the spherical Bessel function and by changing the integration variable. The resultant algorithm consists of a set of the Fourier transforms and numerical integrations over a linearly spaced grid of variable k in Fourier space. Because the k-dependence appears in the upper limit of the integration range, the integrations can be performed effectively in a recurrence formula. Several types of atomic orbital functions are transformed with the proposed method to illustrate its accuracy and efficiency, demonstrating its applicability for transforms of general order with high accuracy.

Key words: Hankel transforms, spherical Bessel functions, atomic orbitals
PACS: 02.30.Uu, 71.15.Ap, 31.15.-p

1. Introduction

Numerical evaluation of integrals containing the spherical Bessel function is of importance in many fields of computational science and engineering since the spherical Bessel function is often used as the eigenfunction for spherical

*Corresponding author:
Masayuki Toyoda
Postal Address: Research Center for Integrated Science, Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Nomi, Ishikawa 932-1292, Japan.
Telephone: +81-761-51-1987
E-mail address: m-toyoda@jaist.ac.jp

coordinate systems. The integrals are known as the spherical Bessel transform (SBT) which is classified into a more general family of the Hankel or Fourier-Bessel transforms. The SBT is involved in many physical problems such as the scattering in atomic or nuclear systems [1, 2], the simulation of the cosmic microwave background [3], and the interaction of electrons in molecules and crystals [4, 5].

Several computational methods have been developed for the Hankel transforms [6, 7, 8]. However, not many of them can be applied for SBT. There have been proposed three different approaches for SBT for general order: (a) recasting the transform integral as a convolution integral by changing the coordinate variables [9, 10, 11], (b) expansion in terms of a series of Fourier cosine and sine transforms by the trigonometric expansion of the spherical Bessel function [12], and (c) the discrete Bessel transform method which describes SBT as an orthogonal transform [13]. The approach (a) is quite fast since it utilizes the fast Fourier transform (FFT) algorithm. The computation time actually scales as $N \log_2 N$, where N is the number of quadrature points. It, however, has a major drawback caused by the logarithmic grid that almost all the grid points are located in close proximity of the origin. The computation time of the approach (b) which also uses FFT scales as $(\ell+1)N \log_2 N$, where ℓ is the order of transform. Therefore, it becomes slower for the higher order transform. In addition to that, since it requires the integrand to be multiplied by inverse powers of the radial coordinate, the high order transforms may become unstable. The computation time of the approach (c) scales as N^2. The quadrature points are located at each zero of the spherical Bessel function. The optimized selection of the quadrature points enables us to use a small number of N while keeping the accuracy of the computation. However, when consecutive transforms with different orders are required, it may become a minor trouble that the optimized quadrature points differ depending on the order of transform.

In this paper, we propose a new method for the numerical SBT which uses a linear coordinate grid. The transform is decomposed into the Fourier transforms and the numerical integrations which can be evaluated recursively. The computation time for the present method scales as $N \log_2 N$ with overhead for the numerical integration which scales as N. The linear coordinate grid prevents us from troubles caused by the non-uniform or order-dependent grid points. If the considered problem requires to transform a function with various orders, the present method has further the advantage that the results of the most time consuming calculations (*i.e.* the Fourier transforms and the integrations) for a transform with a certain order ℓ can also be used for transforms with any order less than ℓ.

2. Formulation

We are interested in the ℓ-th order SBT of a function $f(r)$ which is defined as follows:

$$\tilde{f}_\ell(k) = \int_0^\infty j_\ell(kr) f(r) r^2 \; dr, \tag{1}$$

where $j_\ell(z)$ is the spherical Bessel function of the first kind. The integral representation of the spherical Bessel function is given by

$$j_\ell(z) = \frac{1}{2i^\ell} \int_{-1}^{1} e^{izt} P_\ell(t) \; dt, \tag{2}$$

where $P_\ell(t)$ is the Legendre polynomials

$$P_{2n}(x) = \sum_{j=0}^{n} (-1)^{n-j} \frac{(2n+2j-1)!!}{(2j)!(2n-2j)!!} x^{2j}, \tag{3}$$

$$P_{2n+1}(x) = \sum_{j=0}^{n} (-1)^{n-j} \frac{(2n+2j+1)!!}{(2j+1)!(2n-2j)!!} x^{2j+1}. \tag{4}$$

Here, $n!!$ is the double factorial

$$n!! = n(n-2)(n-4)\cdots \tag{5}$$

with an exceptional definition that $(-1)!! = (0)!! = 1$. By substituting Eq. (2) into Eq. (1), and using the parity property $P_\ell(-t) = (-1)^\ell P_\ell(t)$, the transform is rewritten as follows:

$$\begin{aligned}
\tilde{f}_\ell(k) &= \int_0^\infty j_\ell(kr) f(r) r^2 \; dr \\
&= \frac{1}{2i^\ell} \int_{-1}^{1} dt \; P_\ell(t) \int_0^\infty dr \; e^{ikrt} f(r) r^2 \\
&= \frac{1}{2i^\ell} \left[\int_0^1 dt \; P_\ell(t) \int_0^\infty dr \; e^{ikrt} f(r) r^2 \right. \\
&\qquad \left. + \int_0^1 dt \; P_\ell(-t) \int_0^\infty dr \; e^{-ikrt} f(r) r^2 \right] \\
&= \frac{1}{2i^\ell} \int_0^1 dt \; P_\ell(t) \int_0^\infty dr \; \left(e^{ikrt} + (-1)^\ell e^{-ikrt} \right) f(r) r^2.
\end{aligned} \tag{6}$$

Now, we change the variables as $t' = tk$, and it becomes

$$\tilde{f}_\ell(k) = \frac{(-1)^{\lfloor \ell/2 \rfloor}}{k} \int_0^k dt' P_\ell(t'/k) F_{(\ell)}(t'), \tag{7}$$

where $F_{(\ell)}(t)$ is the Fourier cosine/sine transform of $f(r)r^2$,

$$F_{(\ell)}(t) \equiv \begin{cases} \displaystyle\int_0^\infty dr\ \cos(tr) f(r) r^2, & \ell \text{ is even}, \\ \displaystyle\int_0^\infty dr\ \sin(tr) f(r) r^2, & \ell \text{ is odd}, \end{cases} \tag{8}$$

and $\lfloor x \rfloor$ is the largest integer that does not exceed x. Finally, by expanding the Legendre polynomials, the transform is decomposed into a sum of definite integrals $I_n(k)$

$$\tilde{f}_{2n}(k) = \sum_{j=0}^{n} (-1)^j \frac{(2n+2j-1)!!}{(2j)!(2n-2j)!!} I_{2j}(k), \tag{9}$$

$$\tilde{f}_{2n+1}(k) = \sum_{j=0}^{n} (-1)^j \frac{(2n+2j+1)!!}{(2j+1)!(2n-2j)!!} I_{2j+1}(k), \tag{10}$$

where $I_n(k)$ is given by

$$I_n(k) = \frac{1}{k^{n+1}} \int_0^k dt\ t^n F_{(n)}(t). \tag{11}$$

Since the integrals $I_n(k)$ appearing in Eqs. (9) and (10) have the same parity as the order of transform ℓ, either the Fourier cosine or sine transform is required to be performed, for given ℓ. In our implementation, as explained later, at most two more Fourier transforms are required to be performed to evaluate the derivatives of $F_{(n)}(t)$. Therefore, regardless of the order of transform, only three Fourier transforms are required. On the other hand, since the integrals $I_n(k)$ depend on the order of transform through t^n terms, a number of $\ell/2+1$ different integrals are required for the summation in Eqs. (9) and (10). Even so, however, this does not increase the computational cost because k does not appear in the integrand of Eq. (11) and thus the computation cost for $I_n(k)$ scales as N, rather than as N^2.

3. Implementation

In order to avoid problems arising from $1/k^{n+1}$ at the origin, we define the r- and k-grid points at half-interval shifted positions as follows:

$$r_j = (j+1/2)\Delta r \qquad (j = 0, 1, \ldots N-1), \tag{12}$$

$$k_m = (m+1/2)\Delta k \qquad (m = 0, 1, \ldots N-1). \tag{13}$$

At each point of k_m, the integral I_n is divided into its segments

$$I_n(k_m) = \frac{1}{k_m^{n+1}} \sum_{m'=0}^{m} T_{n,m'}, \tag{14}$$

where the segments are defined as

$$T_{n,0} \equiv \int_0^{k_0} dt\ t^n F_{(n)}(t), \tag{15}$$

$$T_{n,m'} \equiv \int_{k_{m'-1}}^{k_{m'}} dt\ t^n F_{(n)}(t) \qquad (0 < m' < N). \tag{16}$$

By defining the sum of the segments as $S_{n,m} \equiv \sum_{m'=0}^{m} T_{n,m'}$, the following simple recurrence formula is obtained:

$$S_{n,m} = S_{n,m-1} + T_{n,m}, \tag{17}$$

$$I_n(k_m) = \frac{S_{n,m}}{k_m^{n+1}}. \tag{18}$$

Therefore, the evaluation of the integral is accomplished for all the k-grid points through a summation of segments $T_{n,m'}$, where, at each step of the summation, the subtotal $S_{n,m}$ divided by k_m^{n+1} gives $I_n(k_m)$.

Each segment is evaluated by locally interpolating $F_{(n)}(t)$ with a polynomial curve. Since no grid point is available in between the both ends of the integration range, the derivatives of $F_{(n)}(t)$ are also required to interpolate with higher order polynomials. By noting that only the trigonometric functions depend on t in the integrand of (8), the derivatives and second derivatives are obtained analytically as follows:

$$F'_{(n)}(t) = \begin{cases} -\displaystyle\int_0^\infty dr\ \sin(tr) f(r) r^3, & n \text{ is even}, \\ \displaystyle\int_0^\infty dr\ \cos(tr) f(r) r^3, & n \text{ is odd}, \end{cases} \tag{19}$$

$$F''_{(n)}(t) = \begin{cases} -\displaystyle\int_0^\infty dr\ \cos(tr) f(r) r^4, & n \text{ is even}, \\ -\displaystyle\int_0^\infty dr\ \sin(tr) f(r) r^4, & n \text{ is odd}. \end{cases} \tag{20}$$

By interpolating $F_{(n)}(t)$ with a P-th order polynomial, the integral segments are given by

$$\begin{aligned} T_{n,m'} &\approx \int_{k_{m'-1}}^{k_{m'}} dt\ t^n \sum_{\xi=0}^{P} c_\xi t^\xi \\ &= \sum_{\xi=0}^{P} \frac{c_\xi}{n+\xi+1} \left(k_{m'}^{n+\xi+1} - k_{m'-1}^{n+\xi+1} \right), \end{aligned} \tag{21}$$

where the coefficients c_ξ are determined from the values and derivatives of $F_{(n)}(t)$ at $t = k_{m'}$ and $t = k_{m'-1}$ (see Appendix A). We have performed the interpolation with linear ($P = 1$), cubic ($P = 3$), and quintic ($P = 5$) polynomi-

als. Since a higher order interpolation suffers more severely from the arithmetic errors, interpolations with a polynomial of order $P > 5$ have not been tested.

To summarize, the present method computes the transform of a function $f(r)$ with the order ℓ via the following steps:

- Fourier cosine/sine transforms of $f(r)$ multiplied by a power of r (Eqs. (8), (19), and (20)).
- Evaluation of the integral segments $T_{n,m'}$ by the piecewise polynomial interpolation. (Eq. (21)).
- Evaluation of I_n through the summation of $T_{n,m'}$ (Eqs. (17) and (18)).
- Weighted summation of I_n to give the transformed function $\tilde{f}_\ell(k)$ (Eqs. (9) and (10)).

The order of transform explicitly appears only in the final step. Therefore, once a transform with an order ℓ is performed, then one can also perform another transform with another order $\ell' < \ell$ by repeating only the final step and skipping the others. The computation cost for the first step is $p \times N \log_2 N$, where p is the required number of the Fourier transforms. If the quintic interpolation is used and a transform with one specific order is required, $p = 3$, as mentioned before, while $p = 6$ if transforms with various orders are required. The computation cost for the second and third steps scales as N because of the use of the recurrence formula. It is, however, multiplied by the overhead due to the polynomial interpolation which depends on the order of interpolation P. The computation cost for the final step is $(\ell/2 + 1)N$.

4. Computation results

The present method has been applied in transforming the atomic orbital wave functions which are used in quantum chemistry and condensed matter physics. Three types of atomic orbitals have been examined: the Gaussian-type orbital (GTO), the Slater-type orbital (STO), and the numerically defined pseudo-atomic orbital (PAO) functions. The first two functions are good examples to investigate in detail the numerical error accompanied by the present method since the exact form of the transformed function is available, while the PAO function, being strictly localized within a certain radius, is another example to check the applicability of the method for non-analytic functions.

The normalized GTO function in spherical coordinate is defined as follows:

$$\chi_\ell^{\mathrm{GTO}}(r) = N_\ell \; r^\ell e^{-\zeta r^2}, \tag{22}$$

where the normalization factor is given by

$$N_\ell^{\mathrm{GTO}} = \left(\frac{1}{2\pi\zeta}\right)^{1/4} \sqrt{\frac{(4\zeta)^{\ell+2}}{(2\ell+1)!!}}. \tag{23}$$

The corresponding transformed function is

$$\tilde{\chi}_\ell^{\mathrm{GTO}}(r) = \int_0^\infty j_\ell(kr)\chi_\ell^{\mathrm{GTO}}(r)r^2\ dr$$
$$= N_\ell^{\mathrm{GTO}}\sqrt{\frac{\pi}{4\zeta}}\left(\frac{1}{2\zeta}\right)^{\ell+1} k^\ell e^{-k^2/4\zeta}. \tag{24}$$

Figure 1 (a) shows the the numerical error in the present method in the 0th-order transform of the GTO function, where the integral segments $T_{n,m'}$ have been evaluated by interpolating $F_{(n)}(t)$ with the linear (solid line), cubic (dashed line), and quintic (dash-dotted line) polynomials. The parameters used in the calculation are as follows: the exponent of GTO $\zeta = 1$, the number of grid points $N = 128$, and the maximum value of r-grid $r_{\max} = 20$. It is clearly observed that the error is reduced quickly as the order of interpolation polynomial increases, and that the quintic polynomial gives a sufficiently accurate result. In Fig. 1 (b), the numerical error for the 15th-order transform of GTO is shown, where the same parameters as the previous calculation and the quintic polynomial interpolation is used. Even in such a high order transform, the error remains small. This implies that the numerical error comes mainly from the integration of the segments while the possible rounding-off error in Eqs. (9), (10), and (11) is actually negligible unlike in the approach (b) referred in the introduction of this paper [12].

Since the numerical error in the integration of a segment remains in the summation in Eq. (14), the numerical error of the segments in the small-k region also contributes to the error in the large-k region. This explains why the damping of the error at large-k region is so slow in Fig. 1 (a). A downward summation is effective to reduce this error. In Fig. 2, the numerical error in the 0th-order transform of GTO is plotted, where the summation of the segments are performed upward (solid line) and downward (dashed line). In the downward summation, the numerical error in the large-k region becomes much smaller, while, in the small-k region, the error becomes larger because of the accumulation of the error from the large-k region. Therefore, by connecting the results of the upward and downward summations at a certain k point (for example, $k = 5$), accurate results can be obtained in both small- and large-k regions.

So far, the transforms have been performed where the order of transform ℓ and the order of the GTO function ℓ' are equivalent. In Fig. 3, the numerical error in the transform with the order $\ell = 1$ for the GTO function whose order is $\ell' = 0$ is plotted with a variety of grid spacings in real space. It is found that the error (solid line) is larger than that of the case $\ell = \ell' = 0$ calculated with the same condition (dash-dotted line in Fig. 1 (a)). The decrease of the error by smaller grid spacing implies that fine grid spacing is necessary for the accurate sine transform in Eq. (8) since the GTO function of the order $\ell' = 0$ has a finite value at $r = 0$ while the spherical Bessel function of the order $\ell = 1$ vanishes proportionally to r. The different behaviors near the origin results in

another undesirable fact that the transformed function in Fourier space has a very long tail. In the case considered here, the analytic form of the SBT of the GTO function is given as

$$\int_0^\infty j_1(kr)e^{-r^2}r^2dr = \left(\frac{1}{2}+\frac{1}{k^2}\right)D(k/2)-\frac{1}{2k}, \tag{25}$$

where $D(x)$ is the Dawson's integral [14] which is defined as

$$D(x) \equiv e^{-x^2}\int_0^x e^{t^2}dt. \tag{26}$$

From the asymptotic form of the Dawson's integral, the SBT of the GTO function is known to behave as

$$\int_0^\infty j_1(kr)e^{-r^2}r^2dr \approx \frac{1}{k^3}, \tag{27}$$

when k is large. Therefore, in general, a wide range of k-grid has to be employed where the orders of transform and the function are not equivalent.

In quantum chemistry, the STO wave functions are often used as the basis functions for the molecular orbitals. The radial part of the STO functions is given by

$$\chi_\ell^{\rm STO}(r) = N_\ell^{\rm STO}\ r^\ell e^{-\zeta r}, \tag{28}$$

where the normalization factor is

$$N_\ell^{\rm STO} = (2\zeta)^{\ell+1}\sqrt{\frac{2\zeta}{(2\ell+2)!!}}. \tag{29}$$

The corresponding transformed function is given as

$$\tilde{\chi}_\ell^{\rm STO}(k) = N_\ell^{\rm STO}\frac{2\zeta}{(\zeta^2+k^2)^2}\left(\frac{2k}{\zeta^2+k^2}\right)^\ell. \tag{30}$$

In Fig. 4, plotted is the numerical error for the 0th- and 15th-order transforms of STO. The parameters used in the calculations are as follows: the number of grid points $N = 256$ and the maximum value of the r-grid $r_{\rm max} = 20$ (solid line in Fig. 4 (a)); $N = 2048$ and $r_{\rm max} = 30$ (dashed line in Fig. 4 (a)); $N = 256$ and $r_{\rm max} = 80$ (solid line in Fig. 4 (b)); $N = 1024$ and $r_{\rm max} = 600$ (dashed line in Fig. 4 (b)). The exponent of STO is $\zeta = 1$ for all the calculations. There are two kinds of sources for the numerical error in the SBT for STO: the cusp at the origin and the long tail of STO, and they are illustrated in Fig. 4. The use of the fine grid by increasing N reduces the error as shown in Fig. 4 (a), which implies the requirement of a fine grid near the cusp for the accurate integration in Eq. (8). On the other hand, the error is reduced by using the wide range of r-grid as shown in Fig. 4 (b) because of the long tail of STO ($\ell = 15$).

In the electronic structure calculations based on the density functional theory (DFT), the non-analytic atomic orbitals are often used as well as the analytic functions such as the GTO and STO functions. The PAO function is one of those non-analytic basis functions and used in many $O(N)$ DFT calculation methods. A PAO function is calculated by solving the atomic Kohn-Sham equation with confinement pseudopotentials [15]. Figure 5 (a) shows the PAO functions of the $1s$ and $3d$ states of an oxygen atom, where the confinement radius is $r_c = 4.5$ a.u. The corresponding transformed functions are plotted in Fig. 5 (b). Since the mathematical formula for the forward and backward SBT is equivalent except for an additional factor of $2/\pi$, the function should recover the original shape by two consecutive transforms and thus we can illustrate the numerical error by comparing the input function and the back-and-forth transformed function. In Fig. 6, the numerical error accompanied by the present method in two consecutive transforms of the PAO functions of the oxygen $1s$ and $3d$ states is plotted. The parameters used for the calculations are: $N = 512$ and $r_{\max} = 24$. The error is less than 10^{-5} which may not be sufficiently small but is acceptable in practice depending on the particular problems.

For the purpose of comparison, we have performed the back-and-forth transform of the PAO function of the oxygen $3d$ state with the method proposed by Siegman and Talman [9, 10]. In the calculations, we have used $\rho_{\min} = -7.0$ and $\Delta t = 0.45$. As shown in Fig. 7, to suppress the error less than 10^{-5} in the region $r < r_c$, a very large number of quadrature points ($N \geq 2048$) are required. As in this case, due to the use of the linear coordinate grid, a small number of the grid points is enough for the present method to achieve the same degree of accuracy. The computation time has also been measured by taking the average of the CPU time used for 1000 repetitions of a set of transforms, which was carried out on a machine with an Intel Core i7 processor at 2.67 GHz. The program code of the Siegman-Talman method is also implemented by the authors according to the article [10]. The Siegman-Talman method costs 0.64 msec per transform with $N = 2048$, and the present method costs 1.32 msec per transform with $N = 512$. In this measurement, each transform is performed independently so that any speed-up technique is not used such as skipping redundant calculations and reusing resources. This is just a rough estimation, but looks reasonable considering the theoretical computation cost of the both methods (*i.e.* the computation cost for the Siegman-Talman method is basically coming from that of two Fourier transforms, whereas the present method requires three times of the Fourier transforms as well as the additional numerical integrations). In the present method, as mentioned before, once a transform with an order ℓ is performed, another transform with another order $\ell' < \ell$ can also be performed by repeating only the final step. In the set of transform with 15 different orders for a single input function, a substantial amount of redundant calculations are included. In fact, similar redundancy arises also in the Siegman-Talman method. By skipping those redundant calculations, average time per transform reduces to 0.45 msec for the Siegman-Talman method and to 0.22 msec for the present method, in the same measurement as previous one except for the skipping. Therefore, the present method can be quite

effective for a particular type of applications where a single input function is required to be transformed several times with different orders. An example of such applications is the calculation of two-electron integrals in molecules and solids [4, 5]. The authors believe that the present method is an alternative to the Siegman-Talman method with comparable efficiency. The most significant difference of the present method from the Siegman-Talman method is the use of a uniform coordinate grid. For many physical systems which are better described by the uniform coordinate grid, a small number of the quadrature points are required, which gives the advantages not only in the computation speed but also in, for example, the reduction in the memory size and communication traffic in parallelized computations.

5. Summary

In conclusion, we have proposed and demonstrated a new method for the numerical evaluation of SBT. Sufficiently accurate results are obtained in transforming analytic atomic orbital functions. Even the non-analytic PAO functions can be transformed by the present method with a practically acceptable accuracy. Application of the present method to evaluate the electron-electron repulsion integrals is currently in progress by the authors. A similar framework could also be developed for the Hankel transform of integer order, by using the integral expression of the Bessel function in terms of the Gegenbauer polynomials.

6. Acknowledgments

This work was partly supported by CREST-JST and the Next Generation Super Computing Project, Nanoscience Program, MEXT, Japan.

A. Piecewise polynomial interpolation

For given values and derivatives of $F(t)$ at two points, a polynomial which interpolates locally between the points can be obtained by simply solving simultaneous equations. For example, a cubic ($P = 3$) polynomial which interpolates locally between $t = t_0$ and $t = t_1$ is given by

$$F(t) \simeq c_0 + c_1(t - t_0) + c_2(t - t_0)^2 + c_3(t - t_0)^3, \tag{A1}$$

where the coefficients are

$$c_0 = F(t_0), \tag{A2}$$

$$c_1 = F'(t_0), \tag{A3}$$

$$c_2 = \frac{3(F(t_1) - F(k_0))}{(t_1 - t_0)^2} - \frac{F'(t_1) - F'(k_0)}{t_1 - t_0}, \tag{A4}$$

$$c_3 = \frac{-2(F(t_1) - F(t_0))}{(t_1 - t_0)^3} + \frac{F'(t_1) + F'(t_0)}{(t_1 - t_0)^2}. \tag{A5}$$

A quintic ($P = 5$) polynomial is obtained by using the second derivatives of $F(t)$,

$$F(t) \simeq c_0 + c_1(t - t_0) + c_2(t - t_t)^2 + c_3(t - t_0)^3 + c_4(t - t_0)^4 + c_5(t - t_0)^5, \tag{A6}$$

where the coefficients are

$$c_0 = F(t_0), \tag{A7}$$
$$c_1 = F'(t_0), \tag{A8}$$
$$c_2 = F''(t_0), \tag{A9}$$
$$c_3 = \frac{10(F(t_1) - F(t_0))}{(t_1 - t_0)^3} - \frac{4F'(t_1) + 6F'(t_0)}{(t_1 - t_0)^2} + \frac{F''(t_1) - 3F''(t_0)}{2(t_1 - t_0)}, \tag{A10}$$
$$c_4 = -\frac{15(F(t_1) - F(t_0))}{(t_1 - t_0)^4} + \frac{7F'(t_1) + 8F'(t_0)}{(t_1 - t_0)^3} - \frac{2F'(t_1) - 3F'(t_0)}{2(t_1 - t_0)^2}, \tag{A11}$$
$$c_5 = \frac{6(F(t_1) - F(t_0))}{(t_1 - t_0)^5} - \frac{3(F'(t_1) + F'(t_0))}{(t_1 - t_0)^4} + \frac{F'(t_1) - F'(t_0)}{2(t_1 - t_0)^3}. \tag{A12}$$

B. Asymptotic expansion for $T_{n,0}$

The segment of integral $T_{n,0}$ is evaluated in another way. Since k_0 is usually very small, the Fourier cosine/sine transform can be expanded in a Taylor's series. If n is even, the Fourier cosine transform becomes

$$F_{(n)}(t) = \int_0^\infty \cos(tr) f(r) r^2 \, dr \tag{B1}$$
$$= \int_0^\infty \left(1 - \frac{(tr)^2}{2} + \frac{(tr)^4}{12} + \cdots\right) \tag{B2}$$
$$= \int_0^\infty f(r) r^2 \, dr - \frac{t^2}{2} \int_0^\infty f(r) r^4 \, dr + \frac{t^4}{12} \int_0^\infty f(r) r^6 \, dr + O(t^6) \tag{B3}$$
$$= \gamma_2 - \frac{\gamma_4}{2} t^2 + \frac{\gamma_6}{24} t^4 + O(t^6), \tag{B4}$$

where

$$\gamma_n \equiv \int_0^\infty f(r) r^n \, dr. \tag{B5}$$

Therefore, the segment for an even number of n is given as

$$T_{n,0} \approx \gamma_2 \int_0^{k_0} t^n \, dt - \frac{\gamma_4}{2} \int_0^{k_0} t^{n+2} \, dt + \frac{\gamma_6}{24} \int_0^{k_0} t^{n+4} \, dt \tag{B6}$$
$$= \frac{\gamma_2}{n+1} k_0^{n+1} - \frac{\gamma_4}{2(n+3)} k_0^{n+3} + \frac{\gamma_6}{24(n+5)} k_0^{n+5}. \tag{B7}$$

Similarly, for an odd number of n, it is

$$T_{n,0} \approx \frac{\gamma_3}{n+2} k_0^{n+2} - \frac{\gamma_5}{6(n+4)} k_0^{n+4} + \frac{\gamma_7}{120(n+6)} k_0^{n+6}. \tag{B8}$$

References

[1] L. W. Person, P. Benioff, Calculations for quasi-elastic scattering ^{12}C, ^{16}O and ^{40}Ca at 460 MeV, Nucl. Phys. A187 (1972) 401.

[2] A. E. Glassgold, G. Ialongo, Angular distribution of the outgoing electrons in electronic ionization, Phys. Rev. 175 (1968) 151.

[3] M. Liguori, A. Yadav, F. K. Hansen, E. Komatsu, S. Matarrese, B. Wandelt, Temperature and polarization CMB maps from primordial non-Gaussiantities of the local type, Phys. Rev. D 76 (2007) 105016.

[4] J. D. Talman, Numerical methods for multicenter integrals for numerically defined basis functions applied in molecular calculations, Int. J. Quantum Chem. 93 (2003) 72.

[5] M. Toyoda, T. Ozaki, Numerical evaluation of electron repulsion integrals for pseudoatomic orbitals and their derivatives, J. Chem. Phys. 130 (2009) 124114.

[6] S. M. Candel, An algorithms for the Fourier Bessel transform, Comput. Phys. Comm. 23 (1981) 343.

[7] J. D. Secada, Numerical evaluation of the Hankel transform, Comput. Phys. Comm. 116 (1999) 278.

[8] V. K. Singh, Om P. Singh, R. K. Pandey, Numerical evaluation of the Hankel transform by using linear Legendre multi-wavelets, Comput. Phys. Comm. 179 (2008) 424.

[9] A. E. Siegman, Quasi fast Hankel transform, Opt. Lett. 1 (1977) 13.

[10] J. D. Talman, Numerical Fourier and Bessel transforms in logarithmic variables, J. Comput. Phys. 29 (1978) 35.

[11] J. D. Talman, NumSBT: A subroutine for calculating spherical Bessel transforms numericaly, Computer Phys. Comm. 180 (2009) 332.

[12] O. A. Sharafeddin, H. F. Bowen, D. J. Kouri, D. K. Hoffman, Numerical evaluation of spherical Bessel transforms via fast Fourier transforms, J. Comput. Phys. 100 (1992) 294.

[13] D. Lemoine, The discrete Bessel transform algorithm, J. Chem. Phys. 101 (1994) 3936.

[14] M. Abramowitz, I. A. Stegun (Eds.), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Dover publications, 1972.

[15] T. Ozaki, Variationally optimized atomic orbitals for large-scale electronic structures, Phys. Rev. B 67 (2003) 155108.

List of Figure Captions

Figure 1:
(a) Numerical error accompanied by the 0th-order transform of the GTO function ($\ell = 0$), where the linear (solid line), cubic (dashed line), and quintic (dash-dotted line) polynomial interpolations are used to evaluate integrals. (b) Numerical error accompanied by the 15th-order transform of the GTO function ($\ell = 15$), where the quintic polynomial interpolation is used to evaluate integrals.

Figure 2:
Numerical error accompanied by the 0th-order transform of the GTO function ($\ell = 0$), where the upward (solid line) and downward (dashed line) summations are used.

Figure 3:
Numerical error accompanied by the 1st-order transform of the GTO function ($\ell = 0$), where the quintic polynomial interpolation and the upward summation are used. The calculations are performed with various N, namely, 128 (solid line), 256 (dashed line), 512 (dash-dotted line), and 1024 (dotted line), while keeping $r_{\rm max} = 10$.

Figure 4:
(a) Numerical error accompanied by the 0th-order transform of the STO function ($\ell = 0$), where the following two configurations are used: $N = 256$, $r_{\rm max} = 20$, and $k_{\rm max} = 40.2$ (solid line); $N = 2048$, $r_{\rm max} = 30$, and $k_{\rm max} = 214.5$ (dashed line). The quintic interpolation and the upward summation are used. (b) Numerical error accompanied by the 15th-order transform of the STO function ($\ell = 15$), where the following two configurations are used: $N = 256$, $r_{\rm max} = 80$, and $k_{\rm max} = 10.0$ (solid line); $N = 1024$, $r_{\rm max} = 600$, and $k_{\rm max} = 5.4$ (dashed line). The quintic interpolation and the upward summation are used.

Figure 5:
(a) PAO functions of the oxygen $1s$ (solid line) and $3d$ (dashed line) states, where the confinement length is $r_c = 4.5$ a.u. (indicated by the vertical dashed line). (b) Transformed PAO functions of the oxygen $1s$ (solid line) and $3d$ (dashed line) states.

Figure 6:
Numerical error accompanied by the back-and-forth transform of the PAO functions of the oxygen $1s$ (solid line) and $3d$ (dashed line) states.

Figure 7:
Numerical error accompanied by the Siegman-Talman method in performing the back-and-forth transform of the PAO function of the oxygen $3d$ state. The calculations are performed with various N, namely, 128 (solid line), 1024 (dashed

line), and 2048 (dash-dotted line), while keeping $\rho_{\min} = -7.0$ and $\Delta t = 0.45$.

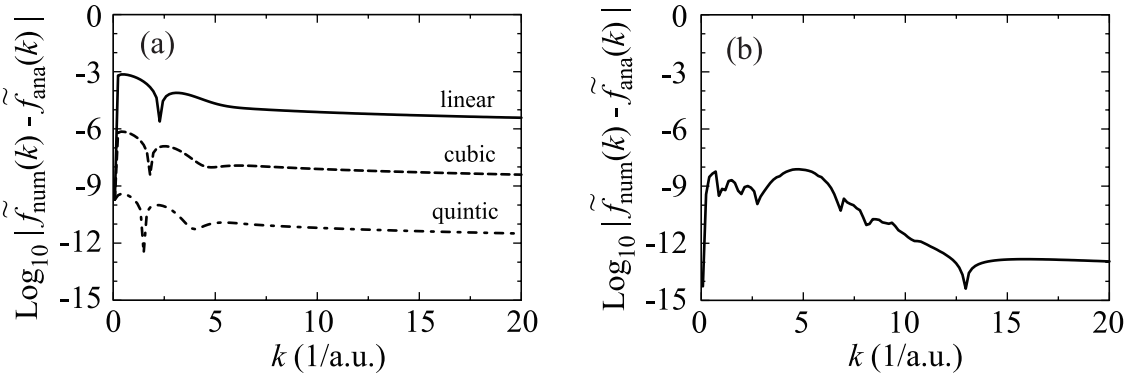
(a)
(b)
linear
cubic
quintic
$\mathrm{Log}_{10}|\tilde{f}_{\mathrm{num}}(k) - \tilde{f}_{\mathrm{ana}}(k)|$
k (1/a.u.)
0
-3
-6
-9
-12
-15
0
5
10
15
20

Figure 1:

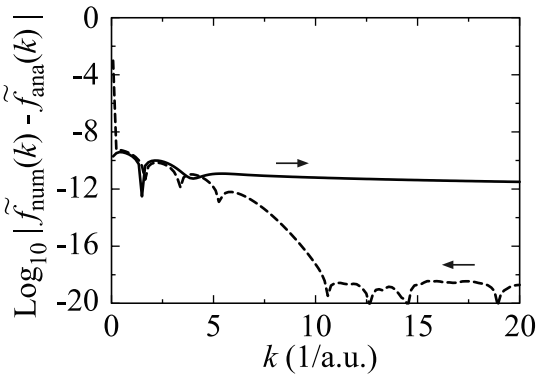
0
-4
-8
-12
-16
-20
0
5
10
15
20
k (1/a.u.)
$\mathrm{Log}_{10}\,|\,\tilde{f}_{\mathrm{num}}(k) - \tilde{f}_{\mathrm{ana}}(k)\,|$

Figure 2:

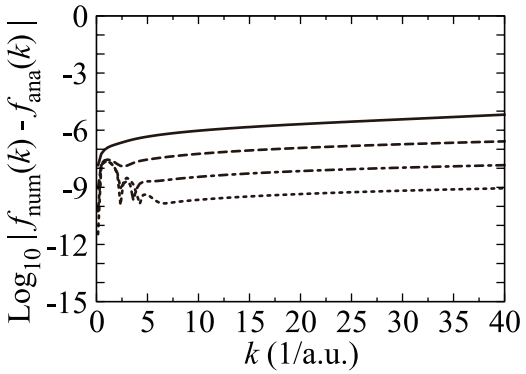
0
-3
-6
-9
-12
-15
0
5
10
15
20
25
30
35
40
k (1/a.u.)
$\mathrm{Log}_{10}|\tilde{f}_{\mathrm{num}}(k) - \tilde{f}_{\mathrm{ana}}(k)|$

Figure 3:

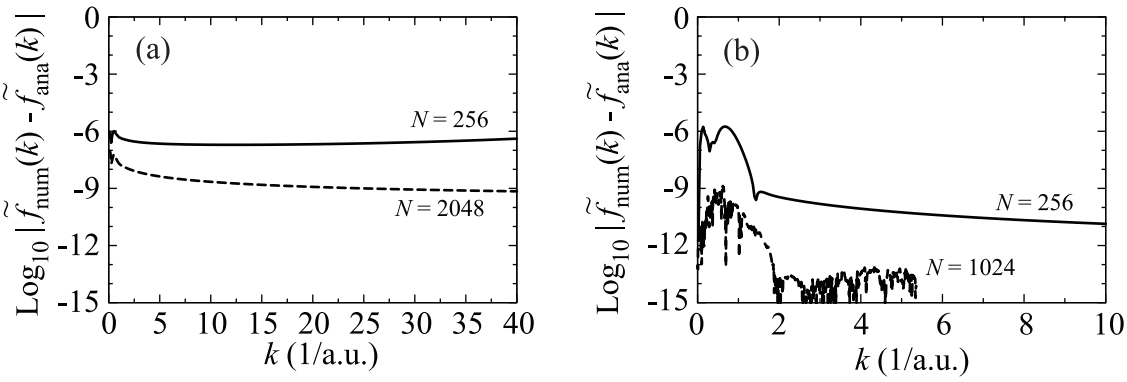

Figure 4:

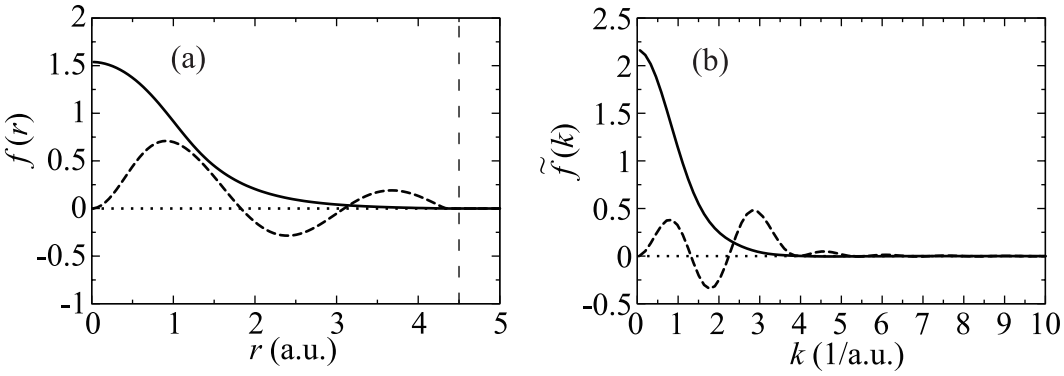

Figure 5:

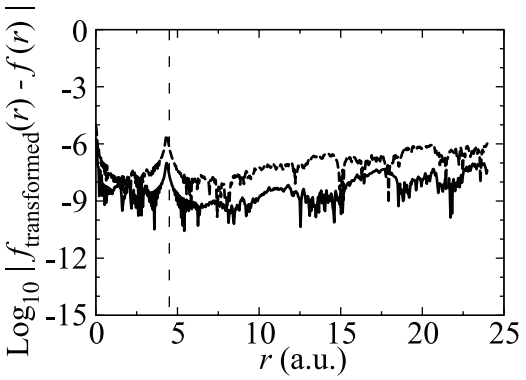
0
-3
-6
-9
-12
-15
0
5
10
15
20
25
$\mathrm{Log}_{10}\,|f_{\mathrm{transformed}}(r) - f(r)\,|$
r (a.u.)

Figure 6:

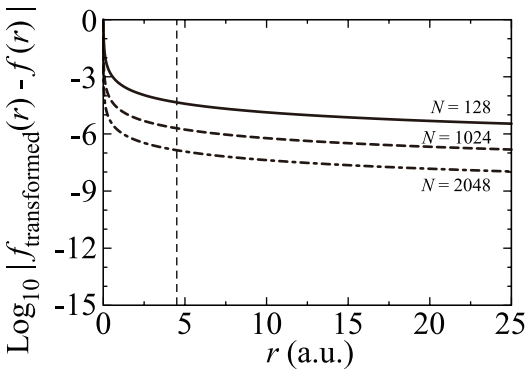
0
-3
-6
-9
-12
-15
0
5
10
15
20
25
$\mathrm{Log}_{10}\,|f_{\mathrm{transformed}}(r) - f(r)\,|$
r (a.u.)
$N = 128$
$N = 1024$
$N = 2048$

Figure 7: