New Algorithm for Electron Repulsion Integrals Oriented to the General Contraction Scheme

TAKESHI YANAI,[1] KAZUHIRO ISHIDA,[2] HARUYUKI NAKANO,[1] KIMIHIKO HIRAO[1]

[1]Department of Applied Chemistry, Graduate School of Engineering, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
[2]Department of Chemistry and Frontier Research Center for Computational Science, Faculty of Science, Science University of Tokyo, Kagurazaka 1-3, Shinjuku-ku, Tokyo 162-8601, Japan

Received 11 January 1999; revised 22 February 1999; accepted 9 March 1999

ABSTRACT: An algorithm for computing electron repulsion integrals (ERIs) oriented to the general contraction scheme is presented. The accompanying coordinate expansion (ACE) method of Ishida is utilized to derive an efficient algorithm. The performance estimated with the floating-point operation (FLOP) count is about N^2 times and more as efficient as the conventional algorithm for the segmented contraction scheme, where N indicates the number of contracted Gaussian-type orbitals (GTOs) contained in a set of generally contracted GTOs. The efficiency is also confirmed by using a realistic molecular system, the benzene molecule, with C:14s9p/3s2p, H:8s4p/2s1p, and C:14s9p/6s5p, H:8s4p/4s3p basis sets. The measured central processing unit (CPU) time is in good agreement with the FLOP count estimation. © 2000 John Wiley & Sons, Inc. Int J Quant Chem 76: 396–406, 2000

Key words: molecular integral; electron repulsion integral; general contraction; accompanying coordinate expansion method

Introduction

Ab initio calculations on large molecular systems containing thousands of atoms or tens of thousands of basis functions are a challenging problem in current quantum chemistry. Such systems can be seen in many important chemical models, such as surface–molecular interface systems in reactions on a surface or in solution and in biological reactions. Although computational techniques, such as the direct method [1] and parallel computation, have made great progress, computation of the huge number of electron repulsion

Correspondence to: K. Hirao.
Contract grant sponsor: Ministry of Education, Science, and Culture of Japan.

integrals (ERIs) is still a major problem. Two aspects are crucial in achieving fast computation of ERIs. One is the selection of an efficient basis set, and the other is the adoption of a sophisticated computational scheme. In the present work, we propose an elaborate algorithm for ERIs involving the advantages of the general contraction scheme for Gaussian-type orbitals (GTOs) [2] and the accompanying coordinate expansion (ACE) method developed by Ishida [3].

The general contraction scheme was first proposed by Raffenetti [2] in which the contracted basis is treated as a linear combination of the primitive basis, with each primitive function contributing to several (or all) of the contracted functions. Explorations of general contraction schemes and applications to molecular problems were carried out by Raffenetti, Ruedenberg, and Bardo [4]. General contraction was also utilized in the well-tempered basis sets of Huzinaga [5], but wider use did not take place until two recent developments of atomic natural orbital (ANO) basis sets by Almlöf and Taylor [6] and correlation-consistent basis sets by Dunning [7]. Unlike most earlier basis sets, which were determined on the basis of self-consistent field (SCF) calculations, these new types of generally contracted basis sets are designed specifically to be effective in the treatment of correlation. Performance comparisons with other segmented contraction basis sets have been very favorable and confirm that these general contractions fulfill their design objectives. The efficient aspect of the general contraction scheme is that the basis sets can describe the accurate atomic properties with small one-particle space and are simple enough to treat the wave functions.

In spite of the advantages of the general contraction scheme, the computation of ERIs over generally contracted basis functions is very expensive if we use the conventional routines for the segmented contraction scheme. Since the degree of contraction of generally contracted basis sets is usually much higher than that of segmentally contracted basis sets, the calculation of ERIs over them is very costly even if we use the best algorithms. Another cause of the cost of the calculation is that the conventional routines cannot utilize the advantage of efficiency of the general contraction scheme, since they treat the basis set for the general contraction scheme in the same way as that for the segmented contraction scheme.

In the present work, we present a refined algorithm for calculating ERIs for the general contraction scheme so as to reduce significantly the cost over that of the conventional algorithms. Our algorithm employs the ACE-b3k3 formula derived by Ishida, which is well-suited for the calculation of ERIs over highly contracted GTOs [3, 8, 9]. The performance is supported by a theoretical measure of the FLOP count [10, 11].

In the following section, we show theoretically our refined algorithm to adapt the ACE-b3k3 formula to the general contraction scheme, comparing it with the conventional method. The computational efficiency is discussed in the third section. Our conclusion is given in the fourth section.

Method

ACE-b3k3 FORMULA

Several types of ACE formulas have been derived by Ishida, such as ACE-b1k1, ACE-b2k1, ACE-b2k3, and ACE-b3k3 [3]. The most efficient type, namely the formula that gives the least FLOP count, varies depending on the degree of the contraction. In the case of high contraction, the ACE-b3k3 formula is most effective, and thus it is suitable for applying to the general contraction scheme. In the present work, we focus on the ACE-b3k3 formula case.

First, we give a brief review of the ACE-b3k3 formulas. In the ACE-b3k3 method, a primitive ERI over primitive GTOs is expressed as the summation of the products of an accompanying coordinate part C_4^{ABCD} and a core part $H_{4\,\lambda\mu\nu\xi}^{ABCD}$,

$$\left[\varphi_\lambda^A \varphi_\mu^B \mid \varphi_\nu^C \varphi_\xi^D\right] = S_{\lambda\mu}^{AB} S_{\nu\xi}^{CD} \sum_{\{\mathbf{N}_3\}} C_4^{ABCD}\{\mathbf{N}_3\} H_{4\,\lambda\mu\nu\xi}^{ABCD}\{\mathbf{N}_3\}. \quad (1)$$

Here we omit the normalization factors. The GTOs, φ_λ^A, φ_μ^B, φ_ν^C, φ_ξ^D, are centered at coordinates **A**, **B**, **C**, **D** and have the exponents, α_λ^A, α_μ^B, α_ν^C, α_ξ^D, and the quantum numbers, $\mathbf{L}_A = (l_A m_A n_A)$, the sum of which, L_A, gives the angular momentum quantum number. The set $\{\mathbf{N}_3\}$ represents summation indices,

$$\{\mathbf{N}_3\} = \{M_A M_B M_C M_D i' j' k' h' i'' j'' k'' h''\}, \quad (2)$$

which obey the following restrictions:

$$
\begin{aligned}
&0 \le M_A \le L_A, \quad 0 \le i' \le L_A - M_A, \\
&\qquad\qquad 0 \le i'' \le i', \\
&0 \le M_B \le L_B, \quad 0 \le j' \le L_B - M_B, \\
&0 \le j'' \le j', \\
&0 \le M_C \le L_C, \quad 0 \le k' \le L_C - M_C, \\
&\qquad\qquad 0 \le k'' \le k', \\
&0 \le M_D \le L_D, \quad 0 \le h' \le L_D - M_D, \\
&\qquad\qquad 0 \le h'' \le h'.
\end{aligned}
\tag{3}
$$

The symbols $S_{\lambda\mu}^{AB}$ and $S_{\nu\xi}^{CD}$ are the remaining parts that do not depend on $\{\mathbf{N}_3\}$,

$$
S_{\lambda\mu}^{AB} = (2\pi)^{-1/4}(\pi/\gamma_1)^{3/2} \exp\left(-\alpha_\lambda^A \alpha_\mu^B \overline{AB}^2/\gamma_1\right), \tag{4}
$$

$$
S_{\nu\xi}^{CD} = (2\pi)^{-1/4}(\pi/\gamma_2)^{3/2} \exp\left(-\alpha_\nu^C \alpha_\xi^D \overline{CD}^2/\gamma_2\right), \tag{5}
$$

with

$$
\gamma_1 = \alpha_\lambda^A + \alpha_\mu^B, \qquad \gamma_2 = \alpha_\nu^C + \alpha_\xi^D. \tag{6}
$$

The accompanying coordinate part C_4^{ABCD}, which is independent of the exponents $(\alpha_\lambda^A, \alpha_\mu^B, \alpha_\nu^C, \alpha_\xi^D)$, is given by

$$
C_4^{ABCD}\{\mathbf{N}_3\} = \sum_{\{\mathbf{M}\}} D_{i'i''}^{A3} D_{j'j''}^{B3} D_{k'k''}^{C3} D_{h'h''}^{D3}, \tag{7}
$$

with

$$
D_{i'i''}^{A3} = \sum_{\{\mathrm{i}'\}} \sum_{\{\mathrm{i}''\}} D_{i'_x i''_x}^{A3x} D_{i'_y i''_y}^{A3y} D_{i'_z i''_z}^{A3z} \tag{8}
$$

$$
D_{i'_x i''_x}^{A3x} = \binom{l_A}{M_{Ax}} \binom{l_A - M_{Ax}}{i'_x} \binom{i'_x}{i''_x} \times AB_x^{l_A - M_{Ax} - i'_x} CD_x^{i'_x - i''_x} AC_x^{i''_x}. \tag{9}
$$

The sets $\{\mathbf{M}\}$, $\{\mathbf{i}'\}$, and $\{\mathbf{i}''\}$ are again summation indices,

$$
\begin{aligned}
\{\mathbf{M}\} &= \{\mathbf{M}_A \mathbf{M}_B \mathbf{M}_C \mathbf{M}_D\}, \\
\mathbf{M}_A &= (M_{Ax} M_{Ay} M_{Az}),
\end{aligned}
\tag{10}
$$

$$
\{\mathbf{i}'\} = \{i'_x i'_y i'_z\}, \qquad \{\mathbf{i}''\} = \{i''_x i''_y i''_z\}, \tag{11}
$$

which obey the restrictions,

$$
\begin{aligned}
&0 \le M_{Ax} \le l_A, \quad 0 \le i'_x \le l_A - M_{Ax}, \\
&\qquad 0 \le i''_x \le i'_x, \\
&0 \le M_{Ay} \le m_A, \quad 0 \le i'_y \le m_A - M_{Ay}, \\
&\qquad 0 \le i''_y \le i'_y, \\
&0 \le M_{Az} \le n_A, \quad 0 \le i'_z \le n_A - M_{Az}, \\
&\qquad 0 \le i''_z \le i'_z, \\
&\qquad \cdots
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
&M_{Ax} + M_{Ay} + M_{Az} = M_A, \quad i'_x + i'_y + i'_z = i', \\
&\qquad i''_x + i''_y + i''_z = i'', \\
&\qquad \cdots
\end{aligned}
\tag{13}
$$

$$
M_{Ax} + M_{Bx} + M_{Cx} + M_{Dx} = \text{even}, \ldots. \tag{14}
$$

Note that C_4^{ABCD} are determined only by the quantum numbers $\mathbf{L}_A$, $\mathbf{L}_B$, $\mathbf{L}_C$, $\mathbf{L}_D$, and the function centers $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$.

The core part is given by

$$
\begin{aligned}
&H_{4\,\lambda\mu\nu\xi}^{ABCD}\{\mathbf{N}_3\} \\
&= \sum_{i_1}^{\left[\frac{M_A+M_B}{2}\right]} \sum_{i_2}^{\left[\frac{M_C+M_D}{2}\right]} G_{i_1 i_2} \sigma_A^{b_A+d_A} \sigma_B^{a_B+c_B} \sigma_C^{d_C+b_C} \\
&\quad \times \sigma_D^{c_D+a_D} \sigma_1^{M_A+M_B-i_1} \\
&\qquad \times \sigma_2^{M_C+M_D-i_1}/(\sigma_1+\sigma_2)^{M-i_1-i_2} \\
&\quad \times \sum_{s_1}^{a_B+b_A} \sum_{s_2}^{c_D+d_C} (-1)^{s_1+s_2} \binom{a_B+b_A}{s_1} \binom{c_D+d_C}{s_2} \\
&\quad \times \left(\frac{\sigma_1}{\sigma_1+\sigma_2}\right)^{i'+j'+s_1} \left(\frac{\sigma_2}{\sigma_1+\sigma_2}\right)^{k'+h'+s_2} \\
&\quad \times F_{\zeta+s_1+s_2}(z),
\end{aligned}
\tag{15}
$$

where

$$
\sigma_1 = 1/(2\gamma_1), \qquad \sigma_2 = 1/(2\gamma_2), \tag{16}
$$

$$
\begin{aligned}
&\sigma_A = \alpha_\lambda^A/\gamma_1, \quad a_B = L_A - M_A - i', \\
&\qquad\qquad a_D = i' - i'', \\
&\sigma_B = \alpha_\mu^B/\gamma_1, \quad b_A = L_B - M_B - j', \\
&\qquad\qquad b_C = j' - j'', \\
&\sigma_C = \alpha_\nu^C/\gamma_2, \quad c_D = L_C - M_C - k', \\
&\qquad\qquad c_B = k' - k'', \\
&\sigma_D = \alpha_\xi^D/\gamma_2, \quad d_C = L_D - M_D - h', \\
&\qquad\qquad d_A = h' - h'',
\end{aligned}
\tag{17}
$$

$$M = (M_A + M_B + M_C + M_D)/2, \tag{18}$$

$$\zeta = i' + j' + k' + h' + M - i_1 - i_2, \tag{19}$$

and

$$G_{i_1 i_2} = (-1)^{M_A + M_B + M - (i_1 + i_2)} \sum_{\{\mathbf{I}\}} g_x g_y g_z, \tag{20}$$

$$g_x = \binom{M_{Ax} + M_{Bx}}{2i_{1x}} \binom{M_{Cx} + M_{Dx}}{2i_{2x}} (2i_{1x} - 1)!! \\ \times (2i_{2x} - 1)!!(2M_x - 2i_{1x} - 2i_{2x} - 1)!!. \tag{21}$$

The summation indices

$$\{\mathbf{I}\} = \{i_{1x} i_{1y} i_{1z} i_{2x} i_{2y} i_{2z}\} \tag{22}$$

obey the restrictions,

$$\begin{gathered} 0 \le i_{1x} \le [(M_{Ax} + M_{Bx})/2], \\ \cdots \\ 0 \le i_{2x} \le [(M_{Cx} + M_{Dx})/2], \\ \cdots \end{gathered} \tag{23}$$

$$i_1 = i_{1x} + i_{1y} + i_{1z}, \qquad i_2 = i_{2x} + i_{2y} + i_{2z}, \tag{24}$$

$$2M_x = M_{Ax} + M_{Bx} + M_{Cx} + M_{Dx}. \tag{25}$$

The function $F_m(z)$ in Eq. (15) is the molecular incomplete gamma function defined by

$$F_m(z) = \int_0^1 t^{2m} \exp(-zt^2)\, dt \tag{26}$$

with $z = \overline{PQ}^2/4\delta$, $\delta = 1/(4\gamma_1) + 1/(4\gamma_2)$, $\mathbf{P} = \sigma_A \mathbf{A} + \sigma_B \mathbf{B}, \mathbf{Q} = \sigma_C \mathbf{C} + \sigma_D \mathbf{D}$.

One important advantage of Eq. (1) is that the core part depending on the exponents $(\alpha_\lambda^A, \alpha_\mu^B, \alpha_\nu^C, \alpha_\xi^D)$ is identical for all integrals for the shell-quadruplet, having a common set of angular momentum quantum numbers, (L_A, L_B, L_C, L_D). For example, 81 $(= 3^4)$ individual integrals, $(p_x p_y \mid p_z p_x)$, $(p_y p_z \mid p_y p_x)$, and so on, have a common set of angular momentum quantum numbers. Once we calculate a core part for a set of angular momentum quantum numbers, (L_A, L_B, L_C, L_D), it can be used for all integrals for the shell-quadruplet having the same set of angular momentum quantum numbers. Utilizing this observation, we can reduce dramatically the cost of the calculation in the contraction step.

FLOP COUNT ESTIMATION FOR CALCULATION OF GENERALLY CONTRACTED ERIS

In this subsection, we give the FLOP count expression for generally contracted ERIs. First, let us consider the FLOP count expression for a usual contracted ERI (CERI).

A contracted GTO (CGTO) is a linear combination of primitive GTOs with all primitives having the same angular momentum and function center but different exponents,

$$\phi_p^A(\mathbf{r}) = \sum_\lambda^{K_A} d_{p\lambda}^A \varphi_\lambda^A(\mathbf{r}), \tag{27}$$

where K_A is the degree of contraction, and $d_{p\lambda}^A$ are the contraction coefficients. Combining Eqs. (1) and (27), we obtain an expression for a contracted ERI:

$$\left(\phi_p^A \phi_q^B \mid \phi_r^C \phi_s^D\right) = \sum_{\mathbf{N}_3} C_4^{ABCD}\{\mathbf{N}_3\} \ddot{H}_{4\,pqrs}^{ABCD}\{\mathbf{N}_3\}, \tag{28}$$

where $\ddot{H}_{4\,pqrs}^{ABCD}$ is the contracted core part including the factors $S_{\lambda\mu}^{AB}$ and $S_{\nu\xi}^{CD}$,

$$\ddot{H}_{4\,pqrs}^{ABCD}\{\mathbf{N}_3\} = \sum_\lambda^{K_A} \sum_\mu^{K_B} d_{p\lambda}^A d_{q\mu}^B S_{\lambda\mu}^{AB} \\ \times \sum_\nu^{K_C} \sum_\xi^{K_D} d_{r\nu}^C d_{s\xi}^D S_{\nu\xi}^{CD} H_{4\,\lambda\mu\nu\xi}^{ABCD}\{\mathbf{N}_3\}. \tag{29}$$

The quartet summation in Eq. (29) can be regarded as the double summation for (ν, ξ) in the double summation for (λ, μ). These double summations are named the K_{BRA} and K_{KET} summations, where $K_{\mathrm{BRA}} = K_A K_B$ and $K_{\mathrm{KET}} = K_C K_D$. Then the K_{BRA} and K_{KET} summations can be split into two steps:

$$\ddot{H}_{4\,pqrs}^{ABCD}\{\mathbf{N}_3\} = \sum_\lambda^{K_A} \sum_\mu^{K_B} d_{p\lambda}^A d_{q\mu}^B S_{\lambda\mu}^{AB} \dot{H}_{4\,\lambda\mu,rs}^{ABCD}\{\mathbf{N}_3\}, \tag{30}$$

$$\dot{H}_{4\,\lambda\mu,rs}^{ABCD}\{\mathbf{N}_3\} = \sum_\nu^{K_C} \sum_\xi^{K_D} d_{r\nu}^C d_{s\xi}^D S_{\nu\xi}^{CD} H_{4\,\lambda\mu\nu\xi}^{ABCD}\{\mathbf{N}_3\}. \tag{31}$$

Considering the FLOP count in Eqs. (28), (30), and (31), the total FLOP count for calculating a CERI can be expressed as a function of K_{BRA} and K_{KET},

$$xK_{\mathrm{BRA}} K_{\mathrm{KET}} + yK_{\mathrm{BRA}} + z. \tag{32}$$

The coefficients of x, y, and z are called FLOP count parameters [11]. The term, $xK_{\mathrm{BRA}} K_{\mathrm{KET}}$, is the FLOP count for K_{KET} summations for all indices in $\{\mathbf{N}_3\}$, λ $(1 \le \lambda \le K_A)$ and μ $(1 \le \mu \le K_B)$. The term yK_{BRA} is the FLOP count for K_{BRA} summations for all indices in $\{\mathbf{N}_3\}$ after K_{KET} summations are finished. The constant value z, depending on neither K_{BRA} nor K_{KET}, is the FLOP count for calculating a CERI after K_{BRA} and K_{KET} summations for all indices in $\{\mathbf{N}_3\}$ are finished. Suppose that all contraction lengths are equal, i.e., $K_A = K_B = K_C = K_D = K$, the FLOP count expression can then be given with K by

$$xK^4 + yK^2 + z, \tag{33}$$

which gives a plain representation of the order for the FLOP count expression.

Again, note that the ACE-b3k3 formula gives the best FLOP count parameters concerned with the contraction length, that is, the values of the x and y parameters in Eq. (33) are small, and thus the total FLOP count for large K is reduced.

Now, let us consider the FLOP count expression for generally contracted ERIs (GCERIs). We here introduce FLOP count expressions for the conventional algorithm and for our refined algorithm.

A set of generally contracted GTOs (GCGTOs),

$$\left\{\phi_p^A(\mathbf{r}) \mid 1 \le p \le N_A\right\} \tag{34}$$

is composed of N_A's GCGTOs. Each is a linear combination of the same set of primitive GTOs with a different set of contraction coefficients. A set of GCERIs is given by

$$\left\{\left(\phi_p^A\phi_q^B \mid \phi_r^C\phi_s^D\right) \mid 1 \le p \le N_A, 1 \le q \le N_B, \right.$$
$$\left. 1 \le r \le N_C, 1 \le s \le N_D\right\}, \tag{35}$$

which contains $(N_{\mathrm{BRA}} N_{\mathrm{KET}})$'s GCERIs. $(N_{\mathrm{BRA}} = N_A N_B$ and $N_{\mathrm{KET}} = N_C N_D$.)

In the conventional algorithm, to obtain a set of GCERIs, all the GCERIs are calculated one by one, with a routine for computing a single CERI. Thus the FLOP count for calculating all the GCERIs is given by

$$N_{\mathrm{BRA}} N_{\mathrm{KET}}(xK_{\mathrm{BRA}} K_{\mathrm{KET}} + yK_{\mathrm{BRA}} + z) \tag{36}$$
$$= N^4(xK^4 + yK^2 + z). \tag{37}$$

However, if we use the ACE-b3k3 formulas, we can utilize the fact that $S_{\nu\xi}^{CD} H_{4\,\lambda\mu\nu\xi}^{ABCD}\{\mathbf{N}_3\}$ in Eq. (31) is identical for each GCERI, since it includes no parameters depending on the contraction coefficients. Therefore the above FLOP count expression in Eqs. (36) and (37) can be reduced to

$$(x_2 N_{\mathrm{KET}} + x_0) K_{\mathrm{BRA}} K_{\mathrm{KET}} + (y_4 N_{\mathrm{BRA}} N_{\mathrm{KET}}$$
$$+ y_2 N_{\mathrm{BRA}} + y_0) K_{\mathrm{BRA}} + z_4 N_{\mathrm{BRA}} N_{\mathrm{KET}} \tag{38}$$
$$= (x_2 N^2 + x_0) K^4 + (y_4 N^4 + y_2 N^2 + y_0)$$
$$\times K^2 + z_4 N^4. \tag{39}$$

Note that the FLOP count for the K_{KET} summation has been effectively reduced by $1/N^2$ order. The loop structures being used in conventional algorithms and our algorithm are illustrated in Figures 1 and 2. It can be easily seen that the conventional algorithm shown in Figure 1 wastes time by performing the same K_{KET} summations N_{BRA} times. On the other hand, in our refined algorithm, shown in Figure 2, the necessary K_{KET} summations are performed just once outside the N_{BRA} loop and stored in the computer memory.

To show that the ACE-b3k3 method is sufficiently adaptable to obtain GCERIs more efficiently, we present some numerical estimations of the FLOP count. Tables I and II list the FLOP count parameters of all classes of four-centered GCERIs over s and p GCGTOs for our algorithm and the conventional algorithm with the ACE-b3k3 formula. The FLOP count for the molecular incomplete gamma function [Eq. (26)] is not included.

```
loop over $N_{KET}$
    loop over $N_{BRA}$
        $K_{KET}$ summations for $N_{KET}$
        $K_{BRA}$ summations for $N_{BRA}$ and $N_{KET}$
    end loop $N_{BRA}$
end loop $N_{KET}$
```

FIGURE 1. Loop structure of the conventional algorithm to obtain GCERIs.

```
loop over $N_{KET}$
    $K_{KET}$ summations for $N_{KET}$
    loop over $N_{BRA}$
        $K_{BRA}$ summations for $N_{BRA}$ and $N_{KET}$
    end loop $N_{BRA}$
end loop $N_{KET}$
```

FIGURE 2. Loop structure of our algorithm to obtain GCERIs.

TABLE I

FLOP count parameters of algorithm with ACE-b3k3 formula.[a,b]

ERI class	x_2	x_0	y_4	y_2	y_0	z_4
$(ss \mid ss)$	2	0	2	0	0	0
$(ps \mid ss)$	4	4	6	1	0	15
$(ss \mid ps)$	6	3	6	2	0	15
$(pp \mid ss)$	10	12	18	4	0	93
$(ss \mid pp)$	12	7	18	10	1	93
$(ps \mid ps)$	12	12	20	8	0	93
$(pp \mid ps)$	24	26	54	27	0	439
$(ps \mid pp)$	24	22	54	29	3	439
$(pp \mid pp)$	44	41	134	80	3	1908

[a] Ignoring the calculation of $F_m(z)$.
[b] Total FLOP count is given by Eq. (38).

TABLE II

FLOP count parameters of conventional algorithm with ACE-b3k3 formula.[a,b,c]

ERI class	x	y	z
$(ss \mid ss)$	3	0	0
$(ps \mid ss)$	8	3	15
$(ss \mid ps)$	7	7	15
$(pp \mid ss)$	19	11	93
$(ss \mid pp)$	14	26	93
$(ps \mid ps)$	19	22	93
$(pp \mid ps)$	38	61	439
$(ps \mid pp)$	34	69	439
$(pp \mid pp)$	63	167	1908

[a] Ignoring the calculation of $F_m(z)$.
[b] Total FLOP count is given by Eq. (36).
[c] Two FLOPs for $(d_{p\lambda}^{A} d_{q\mu}^{B})(d_{r\nu}^{C} d_{s\xi}^{D})F_m(z)$ are included into x, while these are counted for making $F_m(z)$ in a previous work (Ref. [3]).

Table III shows the FLOP counts estimated by Eq. (39) for our algorithm in the case $K = 10$ and $3 \leq N \leq 6$. The rates of speed-up over the conventional algorithm are given in parentheses in the table.

Our algorithm gives small values of the x and y parameters throughout all classes of GCERIs. This result is advantageous for GCGTOs, since they are contracted to a high degree in many cases. In the case of low-angular-momentum GCGTOs, the calculation of the GCERIs speeds up by almost N^2 order. For the $(ss \mid ss)$ class, the calculation of which is fastest, the FLOP count expression by Eq. (39) for the refined algorithm is $(2N^2K^4 + 2N^4K^2)$, while the FLOP count expression by Eq. (37) for the conventional algorithm is $(3N^4K^4)$. As seen from these FLOP count expressions, most of the calculation of $(ss \mid ss)$ is spent on K_{KET} summations. Therefore the performance by N^2 order works more effectively when the values of the y and z parameters are small. On the other hand, when the angular momentum becomes higher, the y and z parameters increase so sharply that the term $(yN^4K^2 + zN^4)$, which is included in both of the FLOP count expressions by Eqs. (39) and (37), becomes too large to be ignored.

Tables IV and V show the FLOP count parameters of $(ps \mid ps)$ and $(pp \mid pp)$ for the case that some of the four function centers, **A**, **B**, **C**, **D**, have the same coordinates, namely overlap. There are 14 cases: the one-center case is $(AA \mid AA)$; two-center cases are $(AA \mid CC)$, $(AB \mid AB)$, $(AB \mid BA)$, $(AA \mid AD)$, $(AA \mid CA)$, $(AB \mid AA)$, and $(AB \mid BB)$; three-center cases are $(AA \mid CD)$, $(AB \mid AD)$, $(AB \mid CA)$, $(AB \mid BD)$, $(AB \mid CB)$, and $(AB \mid CC)$. For $(AA \mid CD)$, the relation $\mathbf{AB} = 0$ can be substituted for $\mathbf{AB}$ in the accompanying coordinate part in Eq. (7).

TABLE III

Total FLOP count of algorithm in case *K* = 10 and 3 ≤ *N* ≤ 6.[a,b]

ERI class	$N = 3$	$N = 4$	$N = 5$	$N = 6$
$(ss \mid ss)$	196000 (12.4)	371000 (20.7)	625000 (30.0)	979000 (39.7)
$(ps \mid ss)$	451000 (12.7)	839000 (21.5)	1427000 (31.0)	2281000 (40.2)
$(pp \mid ss)$	1177000 (9.8)	2211000 (16.5)	3813000 (23.4)	6188000 (29.9)
$(ps \mid ps)$	1377000 (11.3)	2589000 (19.0)	4448000 (27.0)	7181000 (34.7)
$(pp \mid ps)$	2880000 (9.8)	5601000 (15.9)	9942000 (21.8)	16532000 (27.2)
$(pp \mid pp)$	5682000 (9.2)	11498000 (14.4)	21181000 (19.1)	36384000 (23.1)

[a] Ignoring the calculation of $F_m(z)$.
[b] The rate of speed-up over the conventional algorithm is given in parenthesis.

TABLE IV
FLOP count parameters of our algorithm for $(ps \mid ps)$ considering the overlaps of function centers.[a,b]

Overlap type		x_2	x_0	y_4	y_2	y_0	z_4
4 centers	$(AB \mid CD)$	12	12	20	8	0	93
3 centers	$(AA \mid CD)$	4	4	10	6	0	48
	$(AB \mid AD)$	10	13	10	5	1	49
	$(AB \mid CA)$	12	12	10	9	0	48
	$(AB \mid BD)$	10	13	10	7	1	48
	$(AB \mid CB)$	12	12	10	12	0	48
	$(AB \mid CC)$	10	10	10	1	0	48
2 centers	$(AA \mid CC)$	4	4	4	2	0	12
	$(AB \mid AB)$	8	13	4	5	0	12
	$(AB \mid BA)$	10	13	4	7	0	13
	$(AA \mid AD)$	4	4	4	4	1	12
	$(AA \mid CA)$	4	4	4	6	0	12
	$(AB \mid AA)$	4	8	4	2	0	12
	$(AB \mid BB)$	4	9	4	2	0	12

[a] Ignoring the calculation of $F_m(z)$.
[b] Total FLOP count is given by Eq. (38).

This simplifies the formula and, particularly, makes the values of the y and z parameters much smaller for high angular momentum. The case of $(AA \mid CC)$ is most effective as shown in Tables VI and VII. For $(pp \mid pp)$, the FLOP count of $(AA \mid CC)$ is about six times smaller than that of the four-center case. For $(ps \mid ps)$ the FLOP count is about three times smaller than that of the four-center case. To calculate GCERIs faster, it should be effective, especially for high-order GCGTOs, to provide individual routines for each case of overlapping.

TABLE V
FLOP count parameters of algorithm for $(pp \mid pp)$ considering the overlaps of function centers.[a,b]

Overlap type		x_2	x_0	y_4	y_2	y_0	z_4
4 centers	$(AB \mid CD)$	44	41	134	80	3	1908
3 centers	$(AA \mid CD)$	12	11	42	33	3	570
	$(AB \mid AD)$	44	42	66	91	3	885
	$(AB \mid CC)$	40	40	22	13	0	576
2 centers	$(AA \mid CC)$	12	11	12	8	1	87
	$(AB \mid AB)$	38	46	18	79	0	142
	$(AA \mid AD)$	12	11	12	25	5	101
	$(AB \mid AA)$	14	35	12	7	1	103

[a] Ignoring the calculation of $F_m(z)$.
[b] Total FLOP count is given by Eq. (38).

Computational Results and Discussion

A program code based on our algorithm was implemented and tested using some practical molecular calculations. The loop structure of our program is shown in Figure 3. The routines listed in Figures 3 and 2 give the same total FLOP count. However, considering memory caching, the amount of memory required in what we call the K^4 step has been reduced in the routine of Figure 3. In our program we use Ishida's algorithm shown in Ref. [3] to compute the molecular incomplete gamma function.

The routines for calculating GCERIs for $(ss \mid ss)$, $(ps \mid ss)$, $(pp \mid ss)$, $(ps \mid ps)$, $(pp \mid ps)$, and $(pp \mid pp)$ have been implemented in the present case.

TABLE VI
Total FLOP count of algorithm for $(ps \mid ps)$ considering the overlaps of function centers in case $K = 10$ and $3 \leq N \leq 6$.[a]

Overlap type		$N = 3$	$N = 4$	$N = 5$	$N = 6$
4 centers	$(AB \mid CD)$	1377000	2589000	4448000	7181000
3 centers	$(AA \mid CD)$	490000	958000	1710000	2860000
	$(AB \mid AD)$	1119000	2007000	3298000	5108000
	$(AB \mid CA)$	1121000	2009000	3303000	5113000
2 centers	$(AA \mid CC)$	435000	788000	1302000	2021000
	$(AB \mid AB)$	887000	1523000	2400000	3561000
	$(AB \mid BA)$	1070000	1847000	2906000	4290000
	$(AA \mid CA)$	439000	795000	1312000	2035000
	$(AA \mid AD)$	436000	790000	1305000	2025000

[a] Ignoring the calculation of $F_m(z)$.

TABLE VII
Total FLOP count of algorithm for $(pp \mid pp)$ considering the overlaps of function centers in case $K = 10$ and $3 \leq N \leq 6$[a]

Overlap type		$N = 3$	$N = 4$	$N = 5$	$N = 6$
4 centers	$(AB \mid CD)$	5682000	11497000	21177000	36377000
3 centers	$(AA \mid CD)$	1606000	3304000	6174000	10731000
	$(AB \mid AD)$	5068000	9522000	16325000	26286000
2 centers	$(AA \mid CC)$	1302000	2372000	3934000	6127000
	$(AB \mid AB)$	4108000	7163000	11371000	16941000
	$(AA \mid AD)$	1318000	2404000	3986000	6207000

[a] Ignoring the calculation of $F_m(z)$.

```
sum over K_BRA                                              $K_{BRA}$ summations
    sum over K_KET                                          $K_{KET}$ summations
        calculate [ $S^{CD}_{\nu\xi} H^{ABCD}_{4\,\lambda\mu\nu\xi}$ ]                         : $K^4$ step
        loop over N_KET
            sum up [ $d^C_{r\nu} d^D_{s\xi}\ S^{CD}_{\nu\xi} H^{ABCD}_{4\,\lambda\mu\nu\xi}$ ]          : $K^4N^2$ step
        end loop N_KET
    end sum K_KET
    loop over N_KET
        calculate [ $S^{AB}_{\lambda\mu} \dot{H}^{ABCD}_{4\,\lambda\mu\nu\xi}$ ]                         : $K^2N^2$ step
        loop over N_BRA
            sum up [ $d^A_{p\lambda} d^B_{q\mu}\ S^{AB}_{\lambda\mu} \dot{H}^{ABCD}_{4\,\lambda\mu\nu\xi}$ ]          : $K^2N^4$ step
        end loop N_BRA
    end loop N_KET
end sum K_BRA
loop over N_KET
    loop over N_BRA
        obtain GC-ERI                                       : $N^4$ step
    end loop N_BRA
end loop N_KET
```

FIGURE 3. Loop structure of our program to obtain GCERIs.

We measured the central processing unit (CPU) time for calculating a set of GCERIs over GCGTOs to exhibit the computational performance of our algorithm. The size of the GCGTOs used in the calculations is the same as that of the GCGTOs used in the FLOP count estimation. By doing so, the theoretical and realistic computational efficiency could be compared directly. All the calculations were carried out on a single node of IBM SP2(133 MHz) clusters. The CPU time does not include input/output (I/O) operation time.

The four-center GCERIs are evaluated with the GCGTOs $K = 10$ and $3 \leq N \leq 6$. Table VIII lists the CPU times of our algorithm and the rates of speed-up (in parentheses) over the conventional algorithm. It can be seen that the efficiency measured by CPU time far exceeds N^2 order, which is the suggestion from theoretical assessments in the previous section. Especially for $(ss \mid ss)$, the rates of speed-up obey the order of almost N^4. A similar behavior is seen in the other results of CPU time. This bonus improvement of the computational efficiency originates from the cost of computing the molecular incomplete gamma function, which is ignored in the FLOP count estimation in the previous section. In our refined algorithm, the

TABLE VIII
CPU times of algorithm in case $K = 10$ and $3 \leq N \leq 6$ (in millisecond).[a]

ERI class	$N = 3$	$N = 4$	$N = 5$	$N = 6$
$(ss \mid ss)$	22 (70.3)	26 (188)	31 (385)	40 (618)
$(ps \mid ps)$	49 (54.6)	66 (128)	116 (177)	189 (226)
$(pp \mid pp)$	139 (27.3)	300 (40.0)	595 (49.3)	1090 (55.8)

[a] The rate of speed-up over the conventional algorithm is given in parenthesis.

necessary molecular incomplete gamma function is computed just once in the K^4 step in Figure 3. On the other hand, the conventional algorithm carries out the same computation of the molecular incomplete gamma function N^4 times. If we use the FLOP count for the molecular incomplete gamma function, it can be added into the parameters x_0 and y_0 for Eq. (39) and the parameters x and y for Eq. (37). In Ishida's algorithm the FLOP count for the molecular incomplete gamma function is given to be $(16 + S)K^4 + 5K^2$ for $(ss \mid ss)$, $(29 + S)K^4 + 5K^2$ for $(ps \mid ps)$, and $(35 + S)K^4 + 5K^2$ for $(pp \mid pp)$, where the parameter S is the FLOP count for one square-root calculation and dependent on computer environments. We measured a CPU time for one square-root calculation and for one floating-point multiplication, and then estimated a condition $S = 65$ on our computer environment. To show the numerical effect on the FLOP count, including the calculation of the molecular incomplete gamma function, Table IX lists the rate of speed-up for the total FLOP count, assuming the condition $S = 65$ compared with that in the measured CPU time. It can be seen that the rate of speed-up in FLOP count estimation show good agreement with that in the computational measurement. As this result shows, the computation of the incomplete gamma function is a vital part of evaluating the ERIs for all segmented contraction methods. However, it is a trivial part in the present method. This feature confers a very important advantage, which can give an efficiency of almost N^4 order for $(ss \mid ss)$, when the GCGTOs have low angular momentum, i.e., the values of the y and z parameters are small.

Tables X and XI show the computation times of GCERIs $(ps \mid ps)$ and $(pp \mid pp)$ considering the overlaps of the function centers. Similarly to the FLOP count estimation, we examine the case of $(AA \mid CC)$. The CPU time shows a similar efficiency as the FLOP count estimation. The case of $(AA \mid CC)$ is the fastest. For $(pp \mid pp)$, the CPU time of $(AA \mid CC)$ is about six times as fast as that of the four-center case. For $(ps \mid ps)$ the CPU time is about three times as fast as that of the four-center case. These results also show that the programs

TABLE X

CPU times of our algorithm for $(ps \mid ps)$ considering the overlaps of the function centers in case $K = 10$ and $3 \le N \le 6$ (in milliseconds).

Overlap type		$N = 3$	$N = 4$	$N = 5$	$N = 6$
4 centers	$(AB \mid CD)$	49	66	116	189
3 centers	$(AA \mid CD)$	38	45	64	103
	$(AB \mid AD)$	46	58	80	130
	$(AB \mid CA)$	45	58	80	123
2 centers	$(AA \mid CC)$	38	43	52	70
	$(AB \mid AB)$	42	50	64	86
	$(AB \mid BA)$	44	55	72	94
	$(AA \mid CA)$	37	41	53	66
	$(AA \mid AD)$	36	43	52	68

TABLE XI

CPU times of algorithm for $(pp \mid pp)$ considering the overlaps of function centers in case $K = 10$ and $3 \le N \le 6$ (in milliseconds).

Overlap type		$N = 3$	$N = 4$	$N = 5$	$N = 6$
4 centers	$(AB \mid CD)$	139	300	595	1090
3 centers	$(AA \mid CD)$	62	104	204	359
	$(AB \mid AD)$	122	226	418	697
2 centers	$(AA \mid CC)$	56	76	113	188
	$(AB \mid AB)$	101	154	255	400
	$(AA \mid AD)$	57	74	113	174

TABLE IX

Comparision with theoretical efficiency, including the molecular incomplete gamma function calculation and computational efficiency in case $K = 10$ and $3 \le N \le 6$.

ERI class		$N = 3$	$N = 4$	$N = 5$	$N = 6$
$(ss \mid ss)$	FLOP rate	67.6	182	365	608
	CPU time rate	70.3	188	385	618
$(ps \mid ps)$	FLOP rate	39.6	82	131	180
	CPU time rate	54.6	128	177	226
$(pp \mid pp)$	FLOP rate	20.0	33.8	46.4	57.1
	CPU time rate	27.3	40.0	49.3	55.8

specific to each case may work with the good performance predicted from the FLOP count.

Finally, we measured the CPU time for computing all the ERIs in a benzene molecule, as an example of a realistic molecular orbital (MO) calculation. Table XII shows the results given for two kinds of basis sets for carbon and hydrogen atoms: (A) C:$14s9p/3s2p$, H:$8s4p/2s1p$, and (B) C:$14s9p/6s5p$, H:$8s4p/4s3p$. Here the basis sets specified with the format, $(K_s)s(K_p)p/(N_s)s(N_p)p$ have (N_s)'s s-type GCGTOs containing (K_s)'s primitive GTOs and (N_p)'s p-type GCGTOs containing (K_p)'s primitive GTOs. The number of functions in the basis sets is (A) 84 and (B) 204. The calculations were also performed with the ERI routine of Gaussian94 [12] (using the PRISM [11] method) and HONDO [13] (using the PH method [10]). Since these routines can treat only a segmented contraction scheme, we give these basis sets as segmentally CGTO to the program. In Table XII, the ratio of CPU time of the present routine to that of the routine in Gaussian94 is (A) 2.5 and (B) 28. The performance of our program is very good compared with conventional programs, especially when the numbers of GCGTOs, namely N_s and N_p, are large. The ratio of the CPU time for the two basis sets, (A) and (B), is about 2 in the present program, about 23 in Gaussian94 and HONDO. With our routine, fast calculation of ERIs over high-quality basis sets can be performed.

Summary

We proposed a new algorithm and developed programs for the efficient computation of a set of GCERIs over s and p GCGTOs. We utilized the ACE-b3k3 method developed by Ishida, which can rapidly calculate a highly contracted ERI. In our routine, the necessary K_{KET} summation, or K^4 summations, are performed just once outside the N_{BRA} loop, and stored in computer memory. The conventional calculation of GCERIs used in many ab initio packages unnecessarily calculates the identical K^4 terms N^2 times. By the described refinement, we obtain efficiency by N^2 order for the FLOP count estimation with s and p GCGTOs. The FLOP count estimation shows that the calculation of GCERIs is effective when the GCGTOs have low angular momentum.

The performance measured by CPU time exceeds that obtained by the theoretical estimation and shows an efficiency of almost N^4 order in $(ss \mid ss)$. Our program can work more effectively than the theoretical estimation does, since it is a trivial cost to compute the molecular incomplete gamma functions in our algorithm.

When the function centers of GCGTOs overlap each other, the relations of coordinates can simplify the formulas and make the values of the y and z parameters small. When the GCGTOs have high angular momentum, this simplification of the y and z parameters is very effective.

Very recently, Ishida has shown that the ACE-b3k3 method becomes more effective with a high order of GTO [8, 9], while the present work demonstrates that the effectiveness with the general contraction scheme works well even for a low order of GTO.

We expect that our new algorithm and routine for GCERIs will open a new world in the ab initio molecular orbital calculation for large-scale systems.

TABLE XII
CPU times of algorithm for computing all ERIs in benzene molecule (in seconds).

	Basis sets	Present	Gaussian[a,b]	HONDO[a,c]
(A)	C:$14s9p/3s2p$ H:$8s4p/2s1p$	600	2450	5740
(B)	C:$14s9p/6s5p$ H:$8s4p/4s3p$	1180	57100	137800

[a] Including prescreening, indexation, packing, and sorting.
[b] Using the PRISM method [11].
[c] Using the PH method [10].

ACKNOWLEDGMENTS

One of us (T.Y.) thanks Dr. Roland Lindh for valuable discussions. The present research is supported in part by a grant-in-aid for scientific research from the Ministry of Education, Science and Culture of Japan.

References

1. Almlöf, J.; Faegri, Jr., K.; Korsell, K. J Comput Chem 1982, 3, 385.
2. Rafenetti, R. C. J Chem Phys 1973, 58, 4452.
3. Ishida, K. Int J Quant Chem 1996, 59, 209.
4. Rafenetti, R. C.; Ruedenberg, K. J Chem Phys 1973, 59, 5978. Bardo, R. D.; Ruedenberg, K. J Chem Phys 1973, 59, 5966. Bardo, R. D.; Ruedenberg, K. J Chem Phys 1974, 60, 918, 932.
5. Huzinaga, S.; Klobukowski, M.; Tatewaki, H. Can J Chem 1985, 63, 1812.
6. Almlöf, J.; Taylor, P. R. J Chem Phys 1987, 86, 4070.
7. Dunning, Jr., T. H. J Chem Phys 1989, 90, 1007.
8. Ishida, K. J Comput Chem 1998, 19, 923.
9. Ishida, K. J Chem Phys 1998, 109, 881.
10. Pople, J. A.; Hehre, W. J. J Comput Phys 1978, 27, 161.
11. Gill, P. M. W.; Pople, J. A. Int J Quant Chem 1991, 40, 753.
12. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Gill, P. M. W.; Johnson, B. G.; Robb, M. A.; Cheeseman, J. R.; Keith, T.; Petersson, G. A.; Montgomery, J. A.; Raghavachari, K.; Al-Laham, M. A.; Zakrzewski, V. G.; Ortiz, J. V.; Foresman, J. B.; Cioslowski, J.; Stefanov, B. B.; Nanayakkara, A.; Challacombe, M.; Peng, C. Y.; Ayala, P. Y.; Chen, W.; Wong, M. W.; Andres, J. L.; Replogle, E. S.; Gomperts, R.; Martin, R. L.; Fox, D. J.; Binkley, J. S.; Defrees, D. J.; Baker, J.; Stewart, J. P.; Head-Gordon, M.; Gonzalez, C.; Pople, J. A. Gaussian 94, Revision D.1, Gaussian, Pittsburgh, 1995.
13. Dupuis, M.; Chin, S.; Marquez, A. In Relativistic and Electron Correlation Effects in Molecules And Clusters, Malli, G. L., Ed.; NATO ASI Series, Plenum: New York, 1992.