INTERNATIONAL JOURNAL OF QUANTUM CHEMISTRY, VOL. 40, 753–772 (1991)

The Prism Algorithm for Two-Electron Integrals

PETER M.W. GILL AND JOHN A. POPLE

Department of Chemistry, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

Abstract

A new approach to the evaluation of two-electron repulsion integrals over contracted Gaussian basis functions is developed. The new scheme encompasses 20 distinct, but interrelated, paths from simple shell-quartet parameters to the target integrals, and, for any given integral class, the path requiring the fewest floating-point operations (FLOPS) is that used. Both theoretical (FLOP counting) and practical (CPU timing) measures indicate that the method represents a substantial improvement over the HGP algorithm.

Introduction

> *Because of their large number, the evaluation and manipulation of two-electron integrals is the major difficulty in a Hartree–Fock calculation.*
>
> A. Szabo and N. S. Ostlund [1]

Over the decades, this realization has been the single most important driving force in formulating improvements to practical implementations of the Hartree–Fock self-consistent field (SCF) method. By 1980, several ingenious algorithms [2–4] for the evaluation of two-electron repulsion integrals (ERIS) were available, and since (in a conventional SCF calculation) the ERIS need be computed only once (after which they are stored and retrieved on each iteration of the SCF procedure), there appeared to be little to be gained by further improving the ERI evaluation algorithms.

The balance, however, was shifted substantially when, in 1982, Almlöf and coworkers introduced the "direct SCF" method [5] in which ERIS are *recomputed on each iteration of the SCF*. Recently, too, the "direct" approach has been extended to MP2 calculations [5, 6]. Direct methods allow very large calculations to be performed without prohibitively large disk requirements, but, naturally, they cost much more than do their conventional analogs. Indeed, the cost of a direct SCF or direct MP2 calculation is essentially some multiple of the cost of the associated ERI evaluation. Clearly, within such a framework, it is crucial that highly efficient methods for ERI computation be utilized.

After lying dormant for several years, the study of novel ERI algorithms was invigorated by the discovery of the Obara–Saika (OS) recurrence relation [7] (which had been implicit in earlier work [8] by Schlegel) in 1986. Two years later, Head-Gordon and Pople (HGP) suggested [9] that the OS methodology is improved

CCC 0020-7608/91/060753-20$04.00

if the OS recurrence relation is used in judicious combination with another that they called the horizontal recurrence relation. Recently, Hamilton and Schaefer showed that the addition of a third recurrence relation to the HGP scheme results in even greater efficiency [10], and we have argued that a hybrid of the McMurchie–Davidson (MD) scheme [4] and HGP methodology is superior to either [11].

Any ERI may be characterized by two quantities (to be defined later), its angular momentum $L_{tot} \geq 0$ and its degree of contraction $K_{tot} \geq 1$, and, given these, it is possible to make qualitative statements about the performance of each of the algorithms above in computing the ERI. For example, the Pople–Hehre (PH) axis-switching technique [2] is particularly effective when applied to ERIS with large K_{tot} but is otherwise very expensive. It is most commonly used for small-L_{tot} ERIS. Conversely, the Rys quadrature [3] is best suited to small-K_{tot} ERIS and is most commonly used in large-L_{tot} cases. The MD [4] and OS [7] formulations function well for all values of L_{tot} provided that K_{tot} is rather small, but become excessively expensive when this is not so. The more recent algorithms [9–11] behave better than does OS as K_{tot} increases, but are still inferior to the PH method for moderate to large K_{tot} values.

Is there any pattern to these algorithmic behaviors that might suggest how to construct a new methodology that performs well for all L_{tot} and K_{tot}? Fortunately, there is, and a very useful trichotomy is revealed:

(a) The algorithms that achieve their best results when K_{tot} is large but that become inefficient elsewhere are those in which the contraction step occurs very early in the methodology. The archetype of such algorithms is the PH axis-switch method [2].

(b) The defining characteristic of the algorithms that perform very well when K_{tot} is small but that lose their effectiveness elsewhere is that their contraction step occurs as the *last* stage in the methodology. The Rys [3] and OS [7] algorithms exemplify this category.

(c) Between these two extremes, we find the algorithms in which contraction is introduced at some intermediate stage. Although inferior to the PH method when K_{tot} is large and inferior to the OS method when K_{tot} is small, the algorithms in this category, which include the MD- and HGP-based [4, 9–11] methodologies, offer a generally useful performance compromise.

In recognition of this trichotomy, and in order to perform satisfactorily for all combinations of L_{tot} and K_{tot}, current ab initio programs have had to include several, quite distinct, ERI subprograms. For example, faced with the computation of a certain ERI, GAUSSIAN 90 [12] uses simple heuristic rules to select an algorithm from a repertoire consisting of PH, Rys, OS, and HGP codes. However, such code proliferation, and the inaccuracies associated with the automatic selection process, are clearly undesirable.

The PRISM algorithm, on the other hand, is explicitly designed to permit the contraction steps to occur precisely where it is most efficient for them to do so. It is hoped that this intrinsic flexibility will enable the PRISM algorithm to perform well for a wide range of classes of ERI.

Notation and Definitions

An *unnormalized primitive Cartesian Gaussian function,*

$$\varphi_{\mathbf{a}k}(\mathbf{r}) = (x - A_x)^{a_x}(y - A_y)^{a_y}(z - A_z)^{a_z}\exp[-\alpha_k(\mathbf{r} - \mathbf{A})^2], \tag{1}$$

is defined by its angular momentum vector $\mathbf{a} = (a_x, a_y, a_z)$, by its position vector $\mathbf{A}$, and by its exponent α_k. Its angular momentum $a = (a_x + a_y + a_z)$. We will refer to a set of primitive functions on a given center and with a given exponent as a *primitive shell.*

Primitive functions are often linearly combined to form a *contracted Cartesian Gaussian function*

$$\phi_{\mathbf{a}}(\mathbf{r}) = \sum_{k=1}^{K_A} D_{\mathbf{a}k}\varphi_{\mathbf{a}k}(\mathbf{r}), \tag{2}$$

where the $D_{\mathbf{a}k}$ are known as contraction coefficients and K_A is known as the degree of contraction of $\phi_{\mathbf{a}}$.

The *primitive four-center Gaussian* ERI is the integral

$$[\mathbf{a}_k\mathbf{b}_l \,|\, \mathbf{c}_m\mathbf{d}_n] = \int\int \varphi_{\mathbf{a}k}(\mathbf{r}_1)\varphi_{\mathbf{b}l}(\mathbf{r}_1)r_{12}^{-1}\varphi_{\mathbf{c}m}(\mathbf{r}_2)\varphi_{\mathbf{d}n}(\mathbf{r}_2)\,d\mathbf{r}_1\,d\mathbf{r}_2\,. \tag{3}$$

The left-hand subscripts are rarely of particular interest, and it is common to denote the integral (3) by $[\mathbf{ab}\,|\,\mathbf{cd}]$.

Combining (2) and (3) leads to a *contracted four-center Gaussian* ERI

$$(\mathbf{ab}\,|\,\mathbf{cd}) = \sum_{k_A}^{K_A}\sum_{k_B}^{K_B}\sum_{k_C}^{K_C}\sum_{k_D}^{K_D} D_{\mathbf{a}k_A}D_{\mathbf{b}k_B}D_{\mathbf{c}k_C}D_{\mathbf{d}k_D}[\mathbf{a}_{k_A}\mathbf{b}_{k_B}\,|\,\mathbf{c}_{k_C}\mathbf{d}_{k_D}], \tag{4}$$

which we distinguish from a primitive ERI by the use of parentheses instead of square brackets. A *class* of ERIs is defined as the set of all $(\mathbf{ab}\,|\,\mathbf{cd})$ associated with a given shell-quartet.

The *total angular momentum* of the ERI (4) is $L_{tot} = (a + b + c + d)$. Its *bra degree of contraction* and *ket degree of contraction* are $K_{bra} = K_AK_B$ and $K_{ket} = K_CK_D$, respectively. Its *total degree of contraction* is $K_{tot} = K_{bra}K_{ket}$.

Associated with two primitive Gaussian functions $\varphi_{\mathbf{a}}$ and $\varphi_{\mathbf{b}}$ (centered at $\mathbf{A}$ and $\mathbf{B}$, with angular momenta a and b, with exponents α and β, and with contraction coefficients D_A and D_B) is another center $\mathbf{P}$ and two paramters σ_P and U_P defined by

$$\sigma_P = 1/(2\alpha + 2\beta) \tag{5}$$

$$U_P = (8\pi^3)^{1/2}\sigma_P^{a+b+3/2}D_AD_B\exp[-2\alpha\beta\sigma_P(\mathbf{A} - \mathbf{B})^2] \tag{6}$$

$$\mathbf{P} = (2\alpha\mathbf{A} + 2\beta\mathbf{B})\sigma_P\,, \tag{7}$$

and, analogously associated with the primitive functions $\varphi_{\mathbf{c}}$ and $\varphi_{\mathbf{d}}$, we define another center $\mathbf{Q}$ and two parameters σ_Q and U_Q.

As we have previously shown [11, 13], to compute the class (4), we form

$$\mathbf{R} = \mathbf{Q} - \mathbf{P} \tag{8}$$

$$R^2 = R_x^2 + R_y^2 + R_z^2 \tag{9}$$

$$\vartheta^2 = 1/(2\sigma_P + 2\sigma_Q) \tag{10}$$

$$T = \vartheta^2 R^2 \tag{11}$$

$$U = U_P U_Q \tag{12}$$

$$[\mathbf{0}]^{(m)} = U(2\vartheta^2)^{m+1/2}\left(\frac{2}{\pi}\right)^{1/2} \int_0^1 t^{2m}\exp(-Tt^2)\,dt. \tag{13}$$

Given the set of $[\mathbf{0}]^{(m)}$ $(0 \leq m \leq L_{tot})$, as MD and we [4, 11] have shown, it is possible to use a recurrence relation to form the set of intermediate Hermite ERIS $[\mathbf{r}]$ $(0 \leq r \leq L_{tot})$. The $[\mathbf{r}]$ are equal, within a change of sign, to ERIS that we designate by $[\mathbf{p}\,|\,\mathbf{q}]$ and that represent the electrostatic interaction between one primitive (Hermite) function on center **P** and another on center **Q**. We refer to the function on center **P**, which we symbolize by $[\mathbf{p}\,|$, as a *p-bra*. Likewise, we refer to the function on center **Q**, which we symbolize by $|\,\mathbf{q}]$, as a *q-ket*. MD and we have shown [4, 11] that various recurrence relations may then be used to *bra*-transform the $[\mathbf{p}\,|\,\mathbf{q}]$ to $[\mathbf{ab}\,|\,\mathbf{q}]$ and to *ket*-transform the $[\mathbf{ab}\,|\,\mathbf{q}]$ to the desired $[\mathbf{ab}\,|\,\mathbf{cd}]$. Moreover, as we have argued [11], the *bra*- and *ket*-transformation steps can be generalized to produce not only ERIS like $[\mathbf{ab}\,|\,\mathbf{cd}]$ but, also, their derivatives with respect to **A**, **B**, **C**, and **D**. In this very general framework, we symbolize the results of the *bra*-transformation by $[\mathbf{bra}\,|\,\mathbf{q}]$ and those of the *ket*-transformation by $[\mathbf{bra}\,|\,\mathbf{ket}]$.

The issue of contraction, which is central to the PRISM algorithm, may now be discussed. Straightforward contraction is exemplified by

$$(\mathbf{bra}\,|\,\mathbf{ket}] = \sum^{K_{bra}} [\mathbf{bra}\,|\,\mathbf{ket}] \tag{14}$$

$$(\mathbf{bra}\,|\,\mathbf{ket}) = \sum^{K_{ket}} (\mathbf{bra}\,|\,\mathbf{ket}], \tag{15}$$

and we note that a contraction like (14) requires exactly $K_{bra} - 1$ additions. However, as we have previously suggested [11], it is frequently beneficial to generalize this concept to include simultaneous *scaling* of the uncontracted quantities by exponent ratios. Examples of this type of contraction are

$$_{a'b'p'}(\mathbf{r}] = \sum^{K_{bra}} \frac{(2\alpha)^{a'}(2\beta)^{b'}}{(2\alpha + 2\beta)^{p'}}[\mathbf{r}] \tag{16}$$

$$[\mathbf{0})^{(m)}_{c'd'q'} = \sum^{K_{ket}} \frac{(2\gamma)^{c'}(2\delta)^{d'}}{(2\gamma + 2\delta)^{q'}}[\mathbf{0}]^{(m)}. \tag{17}$$

Note that the contraction (16) requires $K_{bra} - 1$ additions *and* K_{bra} multiplies.

Throughout this paper, we will use the acronym FLOP for a floating-point operation, that is, a floating-point add, subtract, multiply, or divide. The number of

FLOPS required by a given algorithm to compute a class of ERIS (its FLOP-count) is usually a good theoretical indication of the practical performance of the algorithm when implemented on a computer, and this fact lies at the heart of the PRISM method.

A New Algorithm for Two-Electron Integrals

In Figure 1, we symbolically represent a number of interrelated paths from uncontracted $[\mathbf{0}]^{(m)}$ integrals (at the top left corner) to fully contracted $(\mathbf{bra}\,|\,\mathbf{ket})$'s (at the bottom right). The figure is structured so that an arrow to the right corresponds to a contraction step, while a downward pointing arrow indicates a transformation step, i.e., one in which recurrence relations are used to build or shift angular momentum. Since it is clear that any path from $[\mathbf{0}]^{(m)}$ to $(\mathbf{bra}\,|\,\mathbf{ket})$ must involve three transformation and two contraction steps in some order, it is possible to label each path uniquely by an acronym indicating the order in which the transformation and contraction steps are performed. For example, the path that passes through the $(\mathbf{0})^{(m)}$ would be termed CCTTT, while, at the other ex-

$$\begin{tikzcd}
{[\,\mathbf{0}\,]^{(m)}} \arrow[r, Rightarrow, "C_1"] \arrow[d, Rightarrow, "T_1"] & {{}_{a'b'p'}(\,\mathbf{0}\,]^{(m)}} \arrow[r, Rightarrow, "C_2"] \arrow[d, Rightarrow, "T_2"] & {{}_{a'b'p'}(\,\mathbf{0}\,)^{(m)}_{c'd'q'}} \arrow[d, Rightarrow, "T_3"] \\
{[\,\mathbf{r}\,]} \arrow[r, Rightarrow, "C_3"] \arrow[d, Rightarrow, "T_4"] & {{}_{a'b'p'}(\,\mathbf{r}\,]} \arrow[r, Rightarrow, "C_4"] \arrow[d, Rightarrow, "T_5"] & {{}_{a'b'p'}(\,\mathbf{r}\,)_{c'd'q'}} \arrow[d, Rightarrow, "T_6"] \\
{[\,\mathbf{bra}\mid\mathbf{q}\,]} \arrow[r, Rightarrow, "C_5"] \arrow[d, Rightarrow, "T_7"] & {(\,\mathbf{bra}\mid\mathbf{q}\,]} \arrow[r, Rightarrow, "C_6"] \arrow[d, Rightarrow, "T_8"] & {(\,\mathbf{bra}\mid\mathbf{q}\,)_{c'd'q'}} \arrow[d, Rightarrow, "T_9"] \\
{[\,\mathbf{bra}\mid\mathbf{ket}\,]} \arrow[r, Rightarrow, "C_7"] & {(\,\mathbf{bra}\mid\mathbf{ket}\,]} \arrow[r, Rightarrow, "C_8"] & {(\,\mathbf{bra}\mid\mathbf{ket}\,)}
\end{tikzcd}$$

Figure 1. Ten PRISM pathways from $[\mathbf{0}]^{(m)}$ to $(\mathbf{bra}\,|\,\mathbf{ket})$.

treme, the path that passes through [**bra**|**ket**] would be termed TTTCC. It should be apparent that there are exactly 10 paths through Figure 1.

We note that TTCTC is very similar to the standard MD algorithm, which, as MD indicated [4], can be shown to be always at least as efficient as TTTCC. We also note that, in the context of the BRAKET algorithm [11], we have previously termed TCTCT and TCCTT the "late contraction" and "early contraction" paths, respectively.

To perform the transformation steps (T_1 to T_9) in Figure 1, appropriate recurrence relations must first be derived. Many years ago, MD developed ones appropriate for uncontracted *bra*- and *ket*-transformations (T_4, T_7, and T_8) and for the uncontracted *r*-transformation (T_1). More recently, we have presented a variety of recurrence relations [11] that are applicable to both uncontracted and contracted *bra*- and *ket*-transformations ($T_4 - T_9$). The recurrence relations for the remaining *r*-transformations (T_2 and T_3) can be deduced easily from that for the T_1 transformation:

$$[\mathbf{r}]^{(m)} = R_i[\mathbf{r} - \mathbf{1}_i]^{(m+1)} - (r_i - 1)[\mathbf{r} - \mathbf{2}_i]^{(m+1)}, \tag{18}$$

by expanding $R_i = Q_i - P_i$ in various ways. For the T_2 transformation, we use

$$R_i = \frac{2\alpha}{2\zeta}(B_i - A_i) + (Q_i - B_i), \tag{19}$$

when

$$\begin{aligned} {}_{a'b'p'}(\mathbf{r}]^{(m)} = {} & (B_i - A_i)_{(a'+1)b'(p'+1)}(\mathbf{r} - \mathbf{1}_i]^{(m+1)} + (Q_i - B_i)_{a'b'p'}(\mathbf{r} - \mathbf{1}_i]^{(m+1)} \\ & - (r_i - 1)_{a'b'p'}(\mathbf{r} - \mathbf{2}_i]^{(m+1)}, \end{aligned} \tag{20}$$

while, for the T_3 transformation, we use

$$R_i = \frac{2\alpha}{2\zeta}(B_i - A_i) + \frac{2\gamma}{2\eta}(C_i - D_i) + (D_i - B_i), \tag{21}$$

from which it follows that

$$\begin{aligned} {}_{a'b'p'}(\mathbf{r})^{(m)}_{c'd'q'} = {} & (B_i - A_i)_{(a'+1)b'(p'+1)}(\mathbf{r} - \mathbf{1}_i)^{(m+1)}_{c'd'q'} + (C_i - D_i)_{a'b'p'}(\mathbf{r} - \mathbf{1}_i)^{(m+1)}_{(c'+1)d'(q'+1)} \\ & + (D_i - B_i)_{a'b'p'}(\mathbf{r} - \mathbf{1}_i)^{(m+1)}_{c'd'q'} - (r_i - 1)_{a'b'p'}(\mathbf{r} - \mathbf{2}_i)^{(m+1)}_{c'd'q'}. \end{aligned} \tag{22}$$

As we have noted before [11], it is possible to improve the efficiency of the T_1, T_2, and T_3 transformations by solving the tree-search problem that is implicit in the T_1 transformation when $L_{tot} > 2$. We have examined this problem in detail and our results are presented elsewhere [14].

We now have at our disposal all of the recurrence relations necessary to form any desired braket from $[\mathbf{0}]^{(m)}$ integrals using any of the 10 paths in Figure 1. We note, however, that Figure 1 can be generalized further. The constraint that the *bra*-contraction precedes the *ket*-contraction *and* that the *bra*-transformation precedes the *ket*-transformation is unnecessary and, in Figure 2, it is removed. Figure 2 (which is, of course, a rectangular prism) contains 20 paths from $[\mathbf{0}]^{(m)}$ to (**bra**|**ket**) and, henceforth, we will refer to these as PRISM paths.

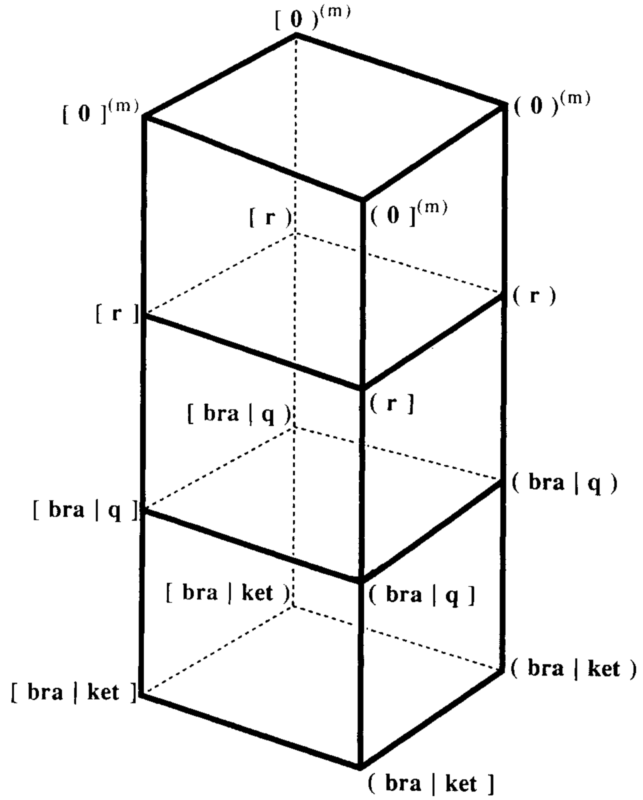

Figure 2. The general PRISM from $[\mathbf{0}]^{(m)}$ to (**bra** | **ket**).

At this point, it is instructive to present detailed examples of how a class of ERIS might be formed on various PRISM paths. Specifically, we will consider the problem of forming a class of $(ps\,|\,ps)$ ERIS and we will use the CCTTT and TCTTC paths as examples.

Example 1: Forming $(ps\,|\,ps)$ ERIS on the CCTTT Path

The discussion is facilitated if we work *backward* along the CCTTT path. Thus, we begin by considering the *ket*-transformation step, i.e., the formation of $(p_i s\,|\,p_j s)$ from $(p_i s\,|\,\mathbf{q})_{c'd'q'}$. From Eq. (45) in [11b], we have

$$(p_i s\,|\,p_j s) = (D_j - C_j)(p_i s\,|\,s)_{011} + (p_i s\,|\,p_j)_{001}, \tag{23}$$

and, as (23) costs two FLOPS for each of the nine $(p_i s\,|\,p_j s)$, the step-cost of the *ket*-transformation $T_9 = 18$ FLOPS.

Continuing backward, we next consider the *bra*-transformation step, i.e., the formation of $(p_i s\,|\,\mathbf{q})_{c'd'q'}$ from ${}_{a'b'p'}(\mathbf{p}\,|\,\mathbf{q})_{c'd'q'}$. As above, we have

$$(p_i s\,|\,\mathbf{q})_{c'd'q'} = (B_i - A_i)_{011}(s\,|\,\mathbf{q})_{c'd'q'} + {}_{001}(p_i\,|\,\mathbf{q})_{c'd'q'}, \tag{24}$$

and, as (24) costs two FLOPS per $(p_is\,|\,\mathbf{q})_{c'd'q'}$ and we have just found that there are 12 of these to form, the *bra*-transformation step-cost $\mathrm{T}_6 = 24$ FLOPS.

Knowing the four *p-bras* and four *q-kets* that are needed and the Hermite identity [4,11]

$$ {}_{a'b'p'}(\mathbf{p}\,|\,\mathbf{q})_{c'd'q'} = (-1)^q\,{}_{a'b'p'}(\mathbf{p}+\mathbf{q})_{c'd'q'} \tag{25}$$

enables us to list, as follows, the 13 ${}_{a'b'p'}(\mathbf{r})_{c'd'q'}$ that are needed:

$$ {}_{011}(\mathbf{0})_{011},\ {}_{011}(\mathbf{1}_i)_{001},\ {}_{001}(\mathbf{1}_i)_{011},\ {}_{001}(\mathbf{2}_{ij})_{001}\,. \tag{26}$$

Using the T_3 recurrence relation (22), these integrals can be reduced to ${}_{a'b'p'}(\mathbf{0})^{(m)}_{c'd'q'}$ as follows:

$$ {}_{011}(\mathbf{1}_i)_{001} = (B_i - A_i)_{112}(\mathbf{0})^{(1)}_{001} + (C_i - D_i)_{011}(\mathbf{0})^{(1)}_{102} + (D_i - B_i)_{011}(\mathbf{0})^{(1)}_{001} \tag{27}$$

$$ {}_{001}(\mathbf{1}_i)_{011} = (B_i - A_i)_{102}(\mathbf{0})^{(1)}_{011} + (C_i - D_i)_{001}(\mathbf{0})^{(1)}_{112} + (D_i - B_i)_{001}(\mathbf{0})^{(1)}_{011} \tag{28}$$

$$ \begin{aligned} {}_{001}(\mathbf{2}_{ij})_{001} &= (B_i - A_i)_{102}(\mathbf{1}_j)^{(1)}_{001} + (C_i - D_i)_{001}(\mathbf{1}_j)^{(1)}_{102} \\ &\quad + (D_i - B_i)_{001}(\mathbf{1}_j)^{(1)}_{001} - \delta_{ij}\,{}_{001}(\mathbf{0})^{(1)}_{001} \end{aligned} \tag{29}$$

$$ {}_{102}(\mathbf{1}_j)^{(1)}_{001} = (B_j - A_j)_{203}(\mathbf{0})^{(2)}_{001} + (C_j - D_j)_{102}(\mathbf{0})^{(2)}_{102} + (D_j - B_j)_{102}(\mathbf{0})^{(2)}_{001} \tag{30}$$

$$ {}_{001}(\mathbf{1}_j)^{(1)}_{102} = (B_j - A_j)_{102}(\mathbf{0})^{(2)}_{102} + (C_j - D_j)_{001}(\mathbf{0})^{(2)}_{203} + (D_j - B_j)_{001}(\mathbf{0})^{(2)}_{102} \tag{31}$$

$$ {}_{001}(\mathbf{1}_j)^{(1)}_{001} = (B_j - A_j)_{102}(\mathbf{0})^{(2)}_{001} + (C_j - D_j)_{001}(\mathbf{0})^{(2)}_{102} + (D_j - B_j)_{001}(\mathbf{0})^{(2)}_{001}\,, \tag{32}$$

and (27)–(32) imply that the r-transformation step-cost $\mathrm{T}_3 = 108$ FLOPS.

We have worked backward through the three transformation steps on the CCTTT path and, in this way, have reduced the nine target ERIS to the following 14 ${}_{a'b'p'}(\mathbf{0})^{(m)}_{c'd'q'}$ integrals:

$$ \begin{gathered} {}_{011}(\mathbf{0})^{(0)}_{011},\ {}_{001}(\mathbf{0})^{(1)}_{001},\ {}_{001}(\mathbf{0})^{(1)}_{011},\ {}_{001}(\mathbf{0})^{(1)}_{112},\ {}_{011}(\mathbf{0})^{(1)}_{001},\ {}_{011}(\mathbf{0})^{(1)}_{102},\ {}_{102}(\mathbf{0})^{(1)}_{011}, \\ {}_{112}(\mathbf{0})^{(1)}_{001},\ {}_{001}(\mathbf{0})^{(2)}_{001},\ {}_{001}(\mathbf{0})^{(2)}_{102},\ {}_{001}(\mathbf{0})^{(2)}_{203},\ {}_{102}(\mathbf{0})^{(2)}_{001},\ {}_{102}(\mathbf{0})^{(2)}_{102},\ {}_{203}(\mathbf{0})^{(2)}_{001}\,. \end{gathered} \tag{33}$$

In a *ket*-contraction costing $C_2 = (22K_{ket} - 14)$ FLOPS, these integrals can be formed from the following eight ${}_{a'b'p'}(\mathbf{0}]^{(m)}_{001}$ integrals:

$$ {}_{011}(\mathbf{0}]^{(0)}_{001},\ {}_{001}(\mathbf{0}]^{(1)}_{001},\ {}_{011}(\mathbf{0}]^{(1)}_{001},\ {}_{102}(\mathbf{0}]^{(1)}_{001},\ {}_{112}(\mathbf{0}]^{(1)}_{001},\ {}_{001}(\mathbf{0}]^{(2)}_{001},\ {}_{102}(\mathbf{0}]^{(2)}_{001},\ {}_{203}(\mathbf{0}]^{(2)}_{001}\,, \tag{34}$$

which, in turn, can be formed from the following three ${}_{001}[\mathbf{0}]^{(m)}_{001}$ integrals:

$$ {}_{001}[\mathbf{0}]^{(0)}_{001},\ {}_{001}[\mathbf{0}]^{(1)}_{001},\ {}_{001}[\mathbf{0}]^{(2)}_{001} \tag{35}$$

in a *bra*-contraction step costing $C_1 = (14K_{bra} - 8)K_{ket}$ FLOPS.

It turns out [13] that a set of ${}_{001}[\mathbf{0}]^{(m)}_{001}$ $(0 \le m \le 2)$ can be formed from the corresponding set of $F_m(T)$ integrals at a cost $X = 6K_{bra}K_{ket}$ FLOPS (and $K_{bra}K_{ket}$ square roots). Finally, to find the total cost of forming a class of $(ps\,|\,ps)$ ERIS from $F_m(T)$ integrals along CCTTT, it suffices simply to add together the six step-costs that we have just computed. Thus,

$$\begin{aligned}\text{CCTTT-cost} &= X + \mathrm{C}_1 + \mathrm{C}_2 + \mathrm{T}_3 + \mathrm{T}_6 + \mathrm{T}_9 \\ &= 6K_{bra}K_{ket} + (14K_{bra} - 8)K_{ket} + (22K_{ket} - 14) + 108 + 24 + 18 \\ &= 20K_{bra}K_{ket} + 14K_{ket} + 136. \qquad (36)\end{aligned}$$

Henceforth, we will often represent the FLOP-cost incurred along a given path when forming a given *braket* class by expressing it in this form, i.e.,

$$\text{Path-cost} = xK_{bra}K_{ket} + yK_{ket} + z, \qquad (37)$$

and we will refer to x, y, and z as the path-cost parameters for that path and braket class. Note that, for uniform contraction K, this becomes the familiar

$$\text{Path-cost} = xK^4 + yK^2 + z. \qquad (38)$$

Example 2: Forming $(ps\,|\,ps)$ ERIs on the TCTTC Path

As before, we will work *backward* along the TCTTC path. Thus, we begin by considering the *ket*-contraction step, i.e., the formation of $(ps\,|ps)$ from $(ps\,|\,ps]$, which is defined by the simple contraction formula

$$(p_is\,|\,p_js) = \sum (p_is\,|\,p_js]. \qquad (39)$$

The summation clearly involves $K_{ket} - 1$ additions and, since we are forming nine $(p_is\,|\,p_js)$, the *ket*-contraction cost $C_8 = 9K_{ket} - 9$ FLOPS.

Continuing up the PRISM, we next consider the *ket*-transformation step in which $(p_is\,|\,p_js]$ are formed from $(p_is\,|\,\mathbf{q}]$. From Eq. (45) in [11b], we have

$$(p_is\,|\,p_js] = (2\delta)(D_j - C_j)(p_is\,|\,s]_{001} + (p_is\,|\,p_j]_{001}, \qquad (40)$$

and, as this recurrence relation costs two FLOPS per $(p_is\,|\,p_js]$ and there are $9K_{ket}$ of these, the *ket*-transformation step-cost $\mathrm{T}_8 = 18K_{ket}$ FLOPS.

Proceeding to the *bra*-transformation step, in which $(p_is\,|\,q]_{001}$ are formed from ${}_{a'b'p'}(\mathbf{p}\,|\,\mathbf{q}]_{001}$, we have (as in the previous example)

$$(p_is\,|\,\mathbf{q}]_{001} = (B_i - A_i)_{011}(s\,|\mathbf{q}]_{001} + {}_{001}(p_i\,|\,\mathbf{q}]_{001}, \qquad (41)$$

and, as this recurrence relation costs two FLOPS per $(p_is\,|\,\mathbf{q}]_{001}$ and we have just discovered that there are $12\mathrm{K}_{\mathrm{ket}}$ of these to form, the bra-transformation step-cost $\mathrm{T}_5 = 24\mathrm{K}_{\mathrm{ket}}$ FLOPS.

Knowing the four *p-bras* and four *q-kets* that are needed and the Hermite identity [4,11],

$${}_{a'b'p'}(\mathbf{p}\,|\,\mathbf{q}]_{c'd'q'} = (-1)^q{}_{a'b'p'}(\mathbf{p} + \mathbf{q}]_{c'd'q'}, \qquad (42)$$

enables us to list, as follows, the 13 ${}_{a'b'p'}(\mathbf{r}]$ that are needed:

$${}_{011}(\mathbf{0}]_{001}, {}_{011}(\mathbf{1}_i]_{001}, {}_{001}(\mathbf{1}_i]_{001}, {}_{001}(\mathbf{2}_{ij}]_{001}. \qquad (43)$$

These (r] can be formed, in the *bra*-contraction step, as follows:

$$ {}_{011}(\mathbf{0}]_{001} = \sum (2\beta)\, {}_{001}[\mathbf{0}]_{001} \tag{44}$$

$$ {}_{011}(\mathbf{1}_i]_{001} = \sum (2\beta)\, {}_{001}[\mathbf{1}_i]_{001} \tag{45}$$

$$ {}_{001}(\mathbf{1}_i]_{001} = \sum {}_{001}[\mathbf{1}_i]_{001} \tag{46}$$

$$ {}_{001}(\mathbf{2}_{ij}]_{001} = \sum {}_{001}[\mathbf{2}_{ij}]_{001} \,. \tag{47}$$

To form the four ${}_{001}(r]_{001}$ requires $4(K_{bra} - 1)K_{ket}$ additions and $4K_{bra}K_{ket}$ multiplies, while formation of the 9 ${}_{001}(r]_{001}$ requires $9(K_{bra} - 1)K_{ket}$ adds. Thus, the cost of the *bra*-contraction step is $C_3 = 17K_{bra}K_{ket} - 13K_{ket}$ FLOPS.

The required $[r]$ are formed in the T_1 step (18) as follows:

$$ {}_{001}[1_i]_{001} = R_i\, {}_{001}[0]^{(1)}_{001} \tag{48}$$

$$ {}_{001}[1_i]^{(1)}_{001} = R_i\, {}_{001}[0]^{(2)}_{001} \tag{49}$$

$$ {}_{001}[2_{ij}]_{001} = R_j\, {}_{001}[1_i]^{(1)}_{001} - \delta_{ij}\, {}_{001}[0]^{(1)}_{001} \,, \tag{50}$$

which, it is easily verified, costs $T_1 = 15K_{bra}K_{ket}$ FLOPS.

As before, the set of ${}_{001}[0]^{(m)}_{001}$ $(0 \le m \le 2)$ can be formed from the $F_m(T)$ integrals at a cost $X = 6K_{bra}K_{ket}$ FLOPS (and $K_{bra}K_{ket}$ square roots). Thus, the total cost of forming a class of $(ps\,|\,ps)$ ERIS from $F_m(T)$ integrals along TCTTC is

$$\begin{aligned} X + T_1 + C_3 + T_5 + T_8 + C_8 &= 6K_{bra}K_{ket} + 15K_{bra}K_{ket} + (17K_{bra} - 13)K_{ket} \\ &\quad + 24K_{ket} + 18K_{ket} + (9K_{ket} - 9) \\ &= 38K_{bra}K_{ket} + 38K_{ket} - 9\,. \end{aligned} \tag{51}$$

Thus, the x, y, and z path-cost parameters for $(ps\,|\,ps)$ formation on the TCTTC path are 38, 38, and −9, respectively.

The Theoretical Performance of the PRISM Algorithm

Throughout this paper, our principal interest is the efficient generation of $(\mathbf{bra}\,|\,\mathbf{ket})$ from $[\mathbf{0}]^{(m)}$, and we will not discuss how the $[\mathbf{0}]^{(m)}$ may be formed in the first place. Elsewhere [14], however, we have addressed this question in detail and have presented an optimized scheme for the formation of $[\mathbf{0}]^{(m)}$ from shell-pair data, and this scheme is easily modified to compute $[ss\,|\,ss]^{(m)}$ if these are required. In order to compare the FLOP-costs of methods that use $[\mathbf{0}]^{(m)}$ integrals with methods that use $[ss\,|\,ss]^{(m)}$ integrals, we must establish a "common denominator." We have shown [13] that a complete set of $[\mathbf{0}]^{(m)}$ $(0 \le m \le L_{tot})$ can be formed from the corresponding $F_m(T)$ integrals in $2L_{tot} + 2$ FLOPS and one square root, and it is straightforward to deduce that the analogous cost for forming a set of $[ss\,|\,ss]^{(m)}$ is $L_{tot} + 2$ FLOPS and one square root. Therefore, for FLOP-counting purposes in the present paper, *we will assume that all $F_m(T)$ have* *already been computed* and any path-cost parameters that we will discuss will pertain to the cost of forming ERIs from these F_m(T) integrals. We will simply ignore the square root since this is common to all algorithms that we will consider.

Analyses analogous to that performed in the foregoing section for $(ps\,|\,ps)$ along the CCTTT and TCTTC paths can be conducted for other ERI classes and other paths. Proceeding in this way for $(ps\,|\,ps)$, $(pp\,|\,pp)$, and $(dd\,|\,dd)$ classes leads to the associated PRISM step-costs (Figs. 3–5) and path-cost parameters (Table I).

Table I contains path-cost parameters for PRISM paths and for three other methods. The PH axis-switch technique and HGP algorithm have been described in detail elsewhere [2,9]. OS did not provide a prescription for the use of their recurrence relation [7], and, for this reason, it was necessary first for us to design one. We have adopted the following "left-to-right" approach: The recurrence relation is first used to reduce the angular momentum at A to zero, then it is simi-

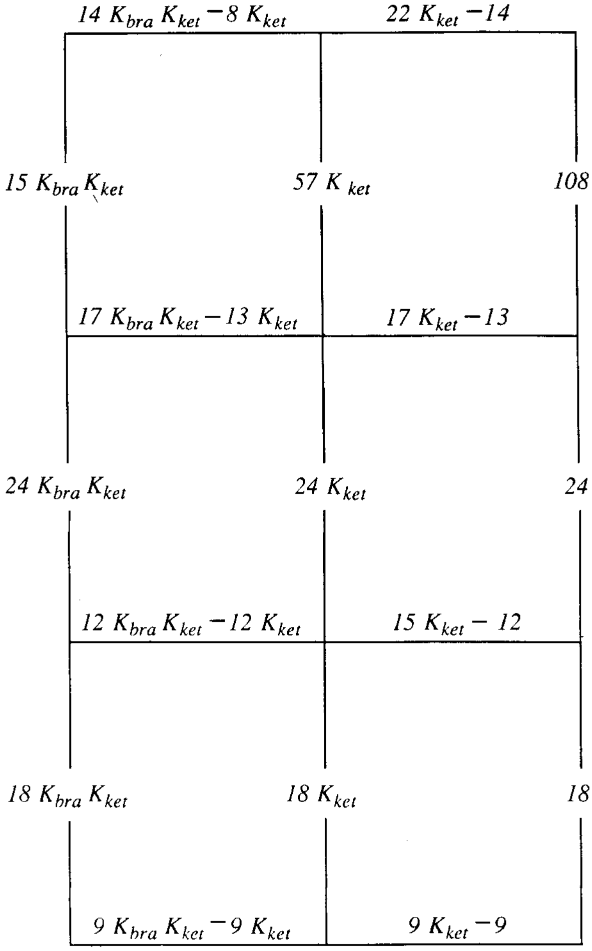

Figure 3. PRISM step-costs (in FLOPS) for $(ps\,|\,ps)$ formation.

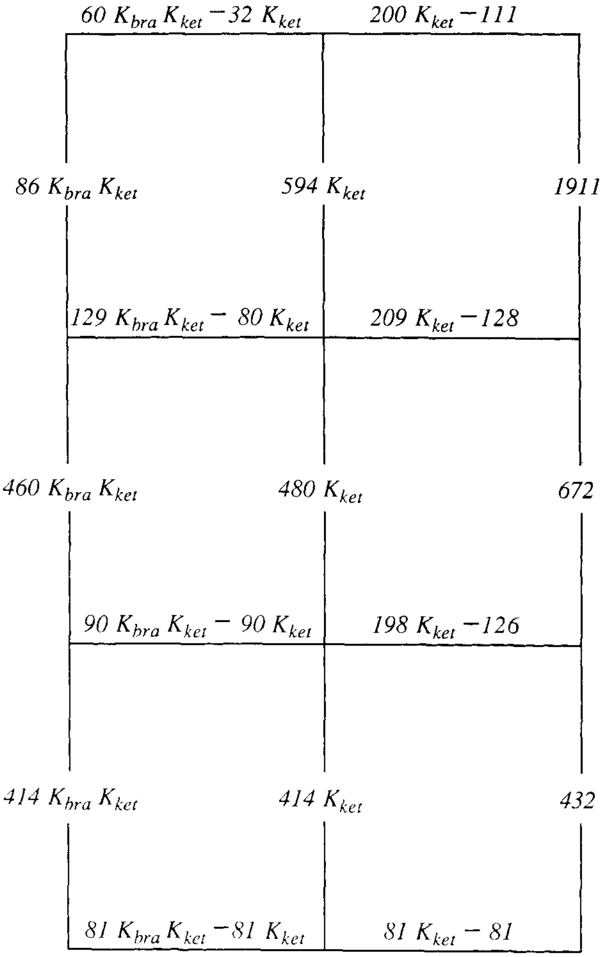

Figure 4. PRISM step-costs (in FLOPS) for $(pp\,|\,pp)$ formation.

larly applied to **B**, then to **C** and, finally, to **D**. When a function can be reduced in more than one way (e.g., d_{xy}), we choose the reduction direction from the optimal T_1 transformation that we have examined elsewhere [14]. Specifically, we reduce d_{xy}, d_{xz}, and d_{yz} in the x, z, and y directions, respectively.

Similarly, as we have discussed previously [11], there is often more than one way in which the various one-electron recurrence relations that we have constructed [11] can be used to form a desired class of *bras* (or *kets*) in the PRISM approach. The $(ps\,|\,$*bras*) that we needed in the foregoing section can be formed in only one way, namely,

$$(p_i s| = (B_i - A_i)_{011}(s| + {}_{001}(p_i| . \tag{52}$$

In general, our program reduces more complicated *bras* (e.g., $(dd|$ etc.) by "left-to-right" application of the (**p** → **a**) recurrence relation (Eq. (45) in [11b]) and its analogs, (**p** → **b**), (**q** → **c**), and (**q** → **d**).

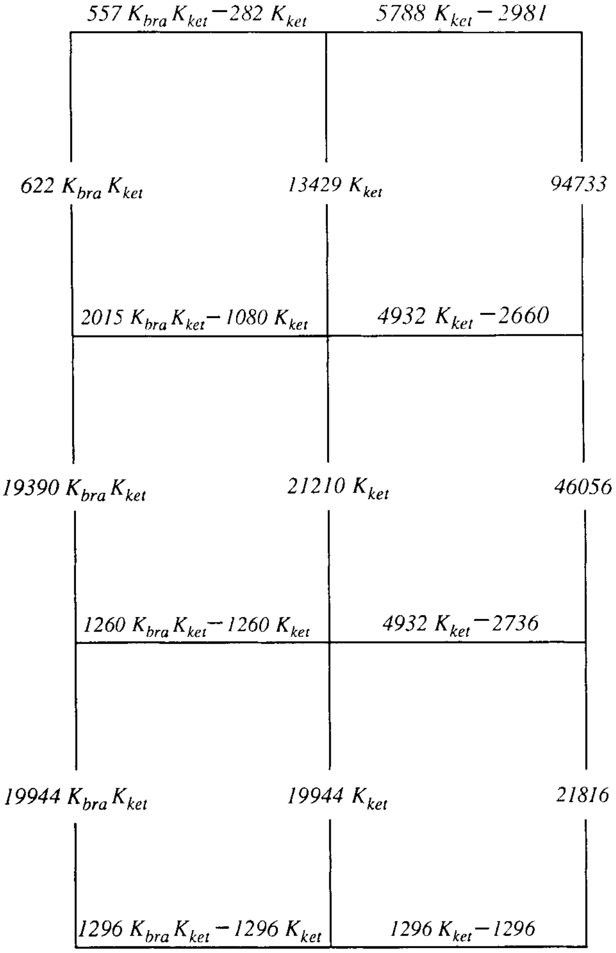

Figure 5. PRISM step-costs (in FLOPS) for $(dd|dd)$ formation.

Although we do not expect the “left-to-right” recipe to lead to *optimal* reduction schemes for either OS or PRISM, its performance appears to be adequate. In one important special case, however, that of $(pp|$bras), the PRISM program uses the following, more carefully optimized, reduction scheme:

$$ {}_{012}(p_j| = {}_{001}(p_j| - {}_{102}(p_j| \tag{53}$$

$$ {}_{101}(p_is| = (B_i - A_i)_{112}(s| + {}_{102}(p_i| \tag{54}$$

$$ (p_ip_j| = (A_j - B_j)_{101}(p_is| + (B_i - A_i)_{012}(p_j| + {}_{002}(d_{ij}| + \delta_{ij}\,{}_{001}(s|\,. \tag{55}$$

It is clear from Table I that the most efficient path to any given ERI class will be very dependent on the K_{bra} and K_{ket} values of the class. For a particular class and particular values of K_{bra} and K_{ket}, the path-costs can be evaluated using the formulae in Table I and the optimal PRISM path thereby selected. The results of such analyses are displayed in Figures 6–8 that we term “Choice Diagrams” for

TABLE I. Path-cost parameters for ERI formation.[a]

	$(ps\,\vert\,ps)$			$(pp\,\vert\,pp)$			$(dd\,\vert\,dd)$		
Path	x	y	z	x	y	z	x	y	z
CCTTT	20	14	136	70	168	2904	575	5,506	159,624
CTCTT	20	66	29	70	771	976	575	18,079	65,212
CTTCT	20	88	6	70	1240	306	575	39,289	19,080
CTTTC	20	100	−9	70	1537	−81	575	55,597	−1,296
TCCTT	38	4	29	225	129	976	2,655	3,852	65,212
TCTCT	38	26	6	225	598	306	2,655	25,062	19,080
TCTTC	38	38	−9	225	895	−81	2,655	41,370	−1,296
TTCCT	57	3	6	646	108	306	21,290	3,672	19,080
TTCTC	57	15	−9	646	405	−81	21,290	19,980	−1,296
TTTCC	72	0	−9	1051	0	−81	41,270	0	−1,296
PH[b]	19	74	90						
HGP[c]	64	0	−9	750	0	243	13,466	0	10,295
OS[d]	64	0	−9	1017	0	−81	45,823	0	−1,296

[a] It is assumed that all necessary $F_m(T)$ integrals are available (see text).
[b] The Pople–Hehre axis-switch algorithm [2].
[c] The Head-Gordon–Pople algorithm [9].
[d] The Obara–Saika algorithm [7] using a "left-to-right" reduction scheme (see text).

the PRISM paths. On the basis of the three ERI classes analyzed, we can make a number of useful observations:

(a) Large values of $K_{tot} = K_{bra}K_{ket}$ favor paths in which contraction is introduced early, whereas small values favor paths with late contraction.

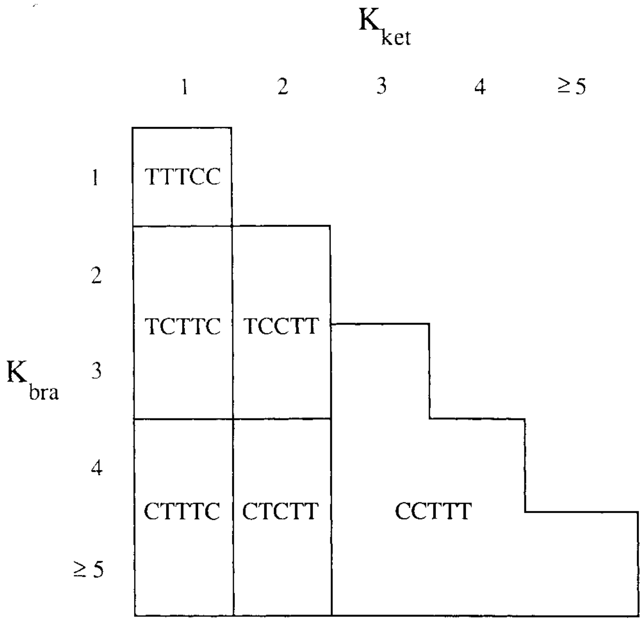

Figure 6. Choice diagram for $(ps\,|\,ps)$ formation.

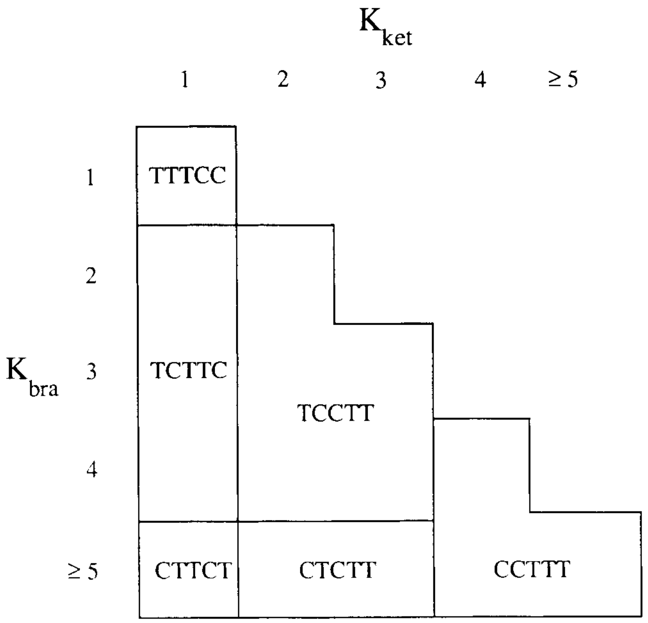

Figure 7. Choice diagram for $(pp|pp)$ formation.

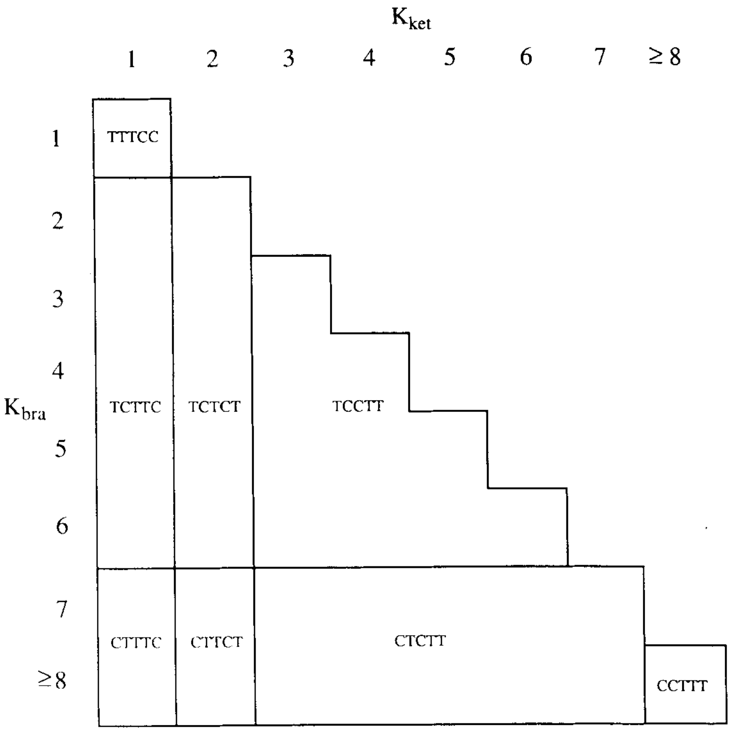

Figure 8. Choice diagram for $(dd|dd)$ formation.

(b) Generally speaking, paths in which contraction is introduced early become less favorable as L_{tot} increases. For example, it is clear that the CCTTT path will always become optimal when $K_{bra} \geq K_{crit}$ and $K_{ket} \geq K_{crit}$ for some class-dependent critical value K_{crit}. However, while $K_{crit} = 1$ (trivially) for $(ss|ss)$, we find that $K_{crit} = 3$ for $(ps|ps)$, $K_{crit} = 4$ for $(pp|pp)$ and $K_{crit} = 8$ for $(dd|dd)$.

(c) The path-cost parameters for HGP (which could be viewed as a TCCT method) most closely resemble those of the TTCCT PRISM path.

(d) HGP is much cheaper than is OS for *uncontracted* $(dd|dd)$. This important fact has not previously been recognized in the literature.

(e) The path-cost parameters of the PRISM paths range from being comparable to those of the PH axis-switch technique to being similar to those of the OS method. This suggests that PRISM has the flexibility to perform well for a wide range of ERI classes but that it will be poorest for uncontracted, high-angular momentum classes.

The Practical Performance of the PRISM Algorithm

To compare the *practical* performance of our implementation of PRISM with implementations of other methods, we have measured CPU timings on MicroVax-II and VAXStation-3100 computers. Although our PRISM program is vectorizable (it is modeled on our BRAKET program [11] and its loop structure is shown in Fig. 9), these machines are both scalar computers. We plan to publish CPU timing comparisons on vector computers (Alliant FX-8 and Cray Y-MP) in the near future.

Head-Gordon and Pople have previously discussed the practical performance of their implementation of the HGP algorithm [9] relative to the GAUSSIAN 86 implementations of the PH [2] and Rys [3] algorithms. They found that HGP was uniformly faster than was Rys for all ERI classes and faster than PH except for highly contracted classes. These results were consistent with FLOP counts that they had also computed. It appears, therefore, that the best standards against which to compare our PRISM program are the HGP and PH programs.

Following HGP's example [9], we first examine the relative performances of the ERI programs on "pure" systems, i.e., ones that contain only one type of shell. Each case consists of 12 shells in a bicubic arrangement. In the $(ss|ss)$ case, the shells are those from the STO-4G basis set for hydrogen [15] and the edge length of the bicube is 0.8 Å. In the $(pp|pp)$ and $(sp,sp|sp,sp)$ cases, the shells are those from the STO-2G basis set for carbon [15] and the edge length is 1.4 Å. In the $(dd|dd)$ case, the shells are uncontracted with exponent 0.8 and, again, the edge length is 1.4 Å. In Table II, we compare the times taken by the HGP, PH, and PRISM codes to compute all of the ERIs in each of the four systems. No point-group symmetry was used, and the requested accuracy was 10^{-10} for HGP and PRISM and about 10^{-8} for PH. The computer used was a MicroVAX-II running VMS.

The $(ss|ss)$ results reflect the different approaches of the three codes to the task of computing the incomplete gamma function: HGP uses 6th-order Taylor in-

```
Call CalcS2 and CalcS3 to compute and order shell-pair information
For L = 0 to MaxL
    Call TabGmT to set up table for interpolating $G_L(T)$
    Call MakMD2 to form driver for $[\mathbf{0}]^{(m)} \rightarrow [\mathbf{r}]$ transformation
    For each bra angular momentum type
        Call MakleT to form drivers for bra transformation
        For each ket angular momentum type
            Call MakleT to form drivers for ket transformation
            Call various routines to form contraction drivers etc.
            For each bra degree of contraction
                For each ket degree of contraction
                    Call Choose to select the optimal PRISM Path
                    Call CalcSF to compute two-centre scaling factors
                    Call CalcBS to find maximum possible batch size
                    For each batch of shell-quartets, until finished
                        Call PickS4 to select the batch
                        Call CalcS4 to form basic shell-quartet parameters
                        Call Calc0m to form $[\mathbf{0}]^{(m)}$
                        Call DoCont to do the bra-contraction
                        Call DoCont to do the ket-contraction
                        Call DoMD4 to do the $(\mathbf{0})^{(m)} \rightarrow (\mathbf{r})$ transformation
                        Call DoShuf to form $(\mathbf{p}|\mathbf{q}) = (-1)^q(\mathbf{p}+\mathbf{q})$
                        Call DoTran to transform $(\mathbf{p}| \rightarrow (\mathbf{bra}|$
                        Call DoShuf to transpose $(\mathbf{bra}|\mathbf{q})$
                        Call DoTran to transform $|\mathbf{q}) \rightarrow |\mathbf{ket})$
                    Next batch of shell-quartets
                Next ket degree of contraction
            Next bra degree of contraction
        Next ket angular momentum type
    Next bra angular momentum type
Next L
```

Figure 9. Program loop structure for braket formation using the CCTTT path. (Other paths result from permuting the calls to DoCont, DoMD4, and DoTran.)

terpolation, as recommended by MD [4] and Harris [16]; PH uses Everett's Formula with Throwback [17]; PRISM employs 3rd-order Chebyshev interpolation [13]. The high efficiency of Chebyshev interpolation is well known.

In the $(pp\,|\,pp)$ example, the PRISM program selected the CCTTT path. (Note that this could have been anticipated from Fig. 7 with $K_{bra} = K_{ket} = 4$). It is easy to deduce, from the path-cost parameters in Table I, that CCTTT needs less than half as many FLOPs as does HGP for $K = 2$ $(pp\,|\,pp)$ and, accordingly, the ratio of the HGP and PRISM timings is greater than 2. The performance of the PH program is intermediate between HGP and PRISM, but the comparison here is not quite fair: PH begins by assuming that each of the p shells is an sp shell and, therefore, computes many more ERIs than are required. Only afterward does the PH program discard the unwanted ERIs.

TABLE II. Comparative timings for ERI evaluation.[a]

	$(ss\|ss)$ $K = 4$	$(pp\|pp)$ $K = 2$	$(sp,sp\|sp,sp)$ $K = 2$	$(dd\|dd)$ $K = 1$
HGP[b]	346	550	788	919
PH[c]	333	409	437	[d]
PRISM[e]	258	215	408	1592

[a] In CPU seconds on a MicroVAX-II computer (see text for details).
[b] The Head-Gordon-Pople method [9] as implemented in GAUSSIAN 90 [12].
[c] The Pople–Hehre method [2]: Link 311 of GAUSSIAN 90 [12].
[d] The PH method is not implemented for d functions in GAUSSIAN 90.
[e] The present algorithm.

A fairer comparison between PRISM and PH is afforded by the $(sp, sp|sp, sp)$ example. Here, the TCCTT path was chosen, and as can be seen, there is little difference between the performances of the PRISM and axis-switch codes. The HGP program, on the other hand, is clearly the slowest of the three. It is interesting to note that the HGP/PRISM ratio is smaller for $(sp, sp|sp, sp)$ than for $(pp|pp)$.

The axis-switch technique is not implemented in GAUSSIAN 90 for d shells, but it would be expected to perform poorly in the uncontracted example that we have selected: Axis-switching is worthwhile only when the degree of contraction is sufficiently large. It is particularly interesting to note, however, that HGP is considerably faster than is PRISM (which selected the TTTCC path) in the $(dd|dd)$ example in Table II. Again, reference to Table I reveals that this could have been predicted on the basis of their relative FLOP counts. In the uncontracted case, the total FLOP cost is simply the sum of the x, y, and z parameters, which is 23,761 FLOPs for HGP and 39,974 FLOPs for TTTCC. This reveals the major deficiency of the PRISM algorithm: Even those paths that are best suited to uncontracted ERI classes (e.g., TTTCC) become increasingly less and less competitive with HGP as the ERI angular momentum increases. Thus, although TTTCC, OS, and HGP are comparable for uncontracted $(pp|pp)$, TTTCC and OS are roughly 70% more expensive than is HGP for uncontracted $(dd|dd)$. We project that the difference is even larger for uncontracted $(ff|ff)$ classes.

We are now in a position to examine some timing results on naphthalene—a "real" molecule. All C—C and C—H bond lengths, respectively, were 1.4 and 1.1 Å. All angles were 120 degrees. Because it cannot be used in direct SCF calculations in GAUSSIAN 90, the axis-switch program could not be fairly compared with the HGP and PRISM programs. In Table III, we present timings for a single direct SCF iteration, i.e., the timings are composed of a contribution from *computing* the ERIs and a contribution from *assimilating* them into Fock matrices. The relative importance of the assimilation component is greatest (25%–30%) when the basis set used is least contracted (3-21G) and is negligible for strongly contracted basis sets (STO-3G). The calculations were performed on a VAXStation-3100 running VMS.

TABLE III. Comparative timings for first direct SCF cycle on naphthalene.[a]

	STO-3G	3-21G	6-31G	6-31G*
HGP[b]	323	445	1104	3246
PRISM[c]	142	349	593	2618

[a] In CPU seconds on a VAXStation-3100 computer (see text for details).
[b] The Head-Gordon–Pople method [9] as implemented in GAUSSIAN 90 [12].
[c] The present algorithm.

With its ability to match the performance of the axis-switch method in highly contracted ERI classes, PRISM fares best in the STO-3G calculation where it required only 44% of the time needed by HGP. This fraction rises to 54% for 6-31G, to 81% for 6-31G*, and to 78% for 3-21G. The explanation for this trend is twofold: First, weakly contracted basis sets offer PRISM little opportunity to utilize paths that are substantially more efficient than is HGP, and, second, the assimilation phase (which is essentially a constant added to both the PRISM and HGP ERI-evaluation times) is most expensive for these same basis sets.

Conclusions

The *overall* performance of the PRISM algorithm is superior to that of the PH axis-switch technique (which is poor for weakly contracted ERI classes) and to that of the HGP methodology (which is poor for strongly contracted ERI classes). PRISM is substantially inferior to HGP; however, for weakly contracted classes of high angular momentum, a modified version of PRISM that does not suffer from this defect needs to be developed.

Acknowledgments

We thank Elizabeth Gill for assistance with the figures in this paper and Benny Johnson for useful comments on its content. This work was supported by the National Science Foundation under grant CHEM-8918623 and was presented at the Argonne National Laboratory's Methodology of the Evaluation of Integrals in LCAO Calculations workshop in August 1990.

Bibliography

[1] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry* (McGraw-Hill, New York, 1989).
[2] J. A. Pople and W. J. Hehre, J. Comput. Phys. **27**, 161 (1978).
[3] (a) M. Dupuis, J. Rys, and H. F. King, J. Chem. Phys. **65**, 111 (1976). (b) H. F. King and M. Dupuis, J. Comput. Phys. **21**, 44 (1976). (c) J. Rys, M. Dupuis, and H. F. King, J. Comput. Chem. **4**, 154 (1983).
[4] L. E. McMurchie and E. R. Davidson, J. Comput. Phys. **26**, 218 (1978).
[5] (a) J. Almlöf, K. Faegri, and K. Korsell, J. Comput. Chem. **3**, 385 (1982). (b) S. Saebø and J. Almlöf, Chem. Phys. Lett. **154**, 83 (1989).
[6] (a) M. Head-Gordon, J. A. Pople, and M. J. Frisch, Chem. Phys. Lett. **153**, 503 (1988). (b) M. J. Frisch, M. Head-Gordon, and J. A. Pople, Chem. Phys. Lett. **166**, 275 (1990). (c) M. J. Frisch, M. Head-Gordon, and J. A. Pople, Chem. Phys. Lett. **166**, 281 (1990).

[7] S. Obara and A. Saika, J. Chem. Phys. **84**, 3963 (1986).
[8] H. B. Schlegel, J. Chem. Phys. **77**, 3676 (1982).
[9] M. Head-Gordon and J. A. Pople, J. Chem. Phys. **89**, 5777 (1988).
[10] T. P. Hamilton and H. F. Schaefer III, Chem. Phys. **150**, 163 (1991).
[11] (a) P. M.W. Gill, M. Head-Gordon, and J. A. Pople, Int. J. Quantum Chem. Symp. **23**, 269 (1989). (b) P. M.W. Gill, M. Head-Gordon, and J. A. Pople, J. Phys. Chem. **94**, 5564 (1990).
[12] Frisch et al., *GAUSSIAN 90* (Carnegie Mellon Quantum Chemistry Publishing Unit, Carnegie Mellon University, Pittsburgh, PA 15213).
[13] P. M.W. Gill, B. G. Johnson, and J. A. Pople, Int. J. Quantum Chem.
[14] B.G. Johnson, P. M.W. Gill, and J. A. Pople, Int. J. Quantum Chem.
[15] W. J. Hehre, R. F. Stewart, and J. A. Pople, J. Chem. Phys. **51**, 2657 (1969).
[16] M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions* (Dover, New York, 1965).
[17] F. E. Harris, Int. J. Quantum Chem. **23**, 1469 (1983).

Received November 20, 1990
Accepted for publication December 11, 1990