

The Rys quadrature revisited: A novel formulation for the efficient computation of electron repulsion integrals over Gaussian functions

Michel Dupuis

*Pacific Northwest National Laboratory, Environmental Molecular Sciences Laboratory, K8-91,
P.O. Box 999, Battelle Boulevard, Richland, Washington 99352*

Antonio Marquez

Department of Chemistry, University of Sevilla, Sevilla, Spain

(Received 7 August 2000; accepted 7 November 2000)

A novel formulation of the Rys quadrature algorithm for the calculation of the electron repulsion integrals over Gaussian basis functions is presented. The new algorithm is specifically designed for high contractions. As for the original Rys quadrature algorithm, the new algorithm is very efficient for high angular momentum functions. In addition it is also equally efficient for low angular momentum functions. The new algorithm takes unique advantage of (1) the numerical Rys quadrature methodology in (2) dealing with charge distributions *a la McMurchie–Davidson* and in (3) scaling integral blocks as a means of transferring angular momentum *a la Gill–Head–Gordon–Pople*. An analysis of the algorithm suggests very favorable floating-point operation counts.

© 2001 American Institute of Physics. [DOI: 10.1063/1.1336541]

I. INTRODUCTION

The evaluation of the electron repulsion integrals,

$$(\chi_i \chi_j | \chi_k \chi_l) = \int \int \chi_i(1) \chi_j(1) \frac{1}{r_{12}} \chi_k(2) \chi_l(2) dr_1 dr_2, \quad (1)$$

where χ_i , χ_j , χ_k , and χ_l are one-electron Gaussian basis functions, and of the derivatives of these integrals with respect to their centers, continues to be one of the major computational tasks in *ab initio* molecular orbital theory (MO) and density functional theory (DFT) methods using atom-centered Gaussian bases. In traditional algorithms where the electron repulsion integrals are computed and stored on disk, their computation and handling is most often the time-limiting step of the calculation. This is even more so with the “direct” algorithms where the integrals are computed several times, whenever they are needed. The new generation of basis sets, with their large degree of contraction of primitive Gaussian functions and very high angular momentum functions, bring also a heavy computational demand. Thus advances in integral computation technology will continue to help push the limits of molecular calculations on large molecules and on molecules with many heavy elements.

In atom-centered bases the necessary angular dependence of a basis function is achieved explicitly through the use of spherical harmonics or equivalently through the use of integer powers of the Cartesian coordinates. When the latter is combined with a Gaussian radial factor, as suggested by Boys,¹ a primitive basis function takes the form

$$\eta_i(1, a_i, I) = (x_1 - x_i)^{n_x i} (y_1 - y_i)^{n_y i} (z_1 - z_i)^{n_z i} \times \exp(-a_i r_{1i}^2), \quad (2)$$

where “1” denotes the electron with coordinates (x_1, y_1, z_1) , (x_i, y_i, z_i) represent the coordinates of the basis

function center I , r_{1i} is the distance between electron “1” and the center I , and a_i is the Gaussian exponent.

The sum of the powers,

$$\lambda_i = n_x i + n_y i + n_z i \quad (3)$$

is closely related to the total angular momentum quantum number. In practice the basis functions χ 's are linear combinations of primitive basis functions,

$$\chi_i(1, I) = \sum_{ai}^K c_{ai} \eta_i(1, a_i, I), \quad (4)$$

where K (not to be confused with center K of χ_k and η_k) is called the degree or level of contraction. The attractive feature of Gaussian functions resides in the fact that there exist practical closed-form expressions for the electron repulsion integrals.^{1,2}

A major breakthrough in electron repulsion integral technology was the method introduced by Pople and Hehre (PH),³ a method that proved to be very efficient for low-angular momentum highly contracted basis functions. The successive rotations involving local axes, albeit extremely efficacious for s and p functions, proved however difficult to extend to higher angular momentum functions. McMurchie and Davidson (MD)⁴ proposed a method highly efficient for contracted functions and readily extendable to high angular momentum functions. Dupuis, Rys, and King (DRK)⁵ developed a radically different method based on an exact numerical quadrature using roots and weights generated from a set of orthogonal polynomials, the Rys polynomials. The implementation of the DRK method proved efficient for basis functions with a low degree of contraction. In the DRK approach the six-dimensional primitive integrals take the form of sums of products of three two-dimensional integrals,

each associated with a Cartesian direction. Rys *et al.*⁶ developed also several recursion formulas for the elementary two-dimensional integrals.

Over 10 years after the work of MD and DRK, Obara and Saika (OS)⁷ proposed some new recursion formulas for the six-dimensional integrals of Eq. (1). These recursion formulas turned out to be closely related to those of DRK, except that they apply to the six-dimensional primitive integrals directly. OS's work generated a new wave of research in electron repulsion evaluation methodologies with significant contributions from Head-Gordon and Pople (HGP),⁸ Gill, Head-Gordon, and Pople (GHGP),⁹ Lindh, Ryu, and Liu (LRL),¹⁰ all algorithms that have received much attention, and the lesser-known algorithms by Ten-no,¹¹ and by Ishida (ISH).¹²

HGP improved OS's method by applying the transfer relations, already noted in DRK's work, outside the quadruple sum over contractions, so that only primitive integrals where η_j and η_l are of s type need be computed inside the contraction loops. This idea was taken further in GHGP's method where additionally these authors borrowed the proposal of MD and limited the mandatory primitive integrals to those where only η_i is not an s orbital. Generation of other intermediate integrals is carefully accomplished via a so-called "scaling" of integrals, so that much intermediate work is moved stepwise outside the contraction loops. The floating-point operation count (FLOP) shows that GHGP's method is indeed superior for contracted functions. Note that the FLOP counts reported in Refs. 8, 9, 11 for the DRK method are all erroneous. Only the count given in Ref. 10 is approximately correct, in accord with the efficiency observed in practice with the DRK algorithm.^{13,14} Finally, Ishida's algorithm¹² has actually an even better FLOP count than GHGP's although it looks to be very complex to implement in practice.

The driving force for the present research was the belief that applying recursion formulas to two-dimensional integrals and then combining these integrals to form six-dimensional integrals ought to be more efficient than applying somewhat more complex recursion formulas directly to six-dimensional integrals. Comparison of the LRL method¹⁰ with the GHGP method⁸ supports this belief. It clearly shows that for low contraction level and high angular momentum functions, the Rys quadrature technique as implemented by LRL and in some cases as implemented by DRK remains the method of choice. The superiority of the GHGP algorithm⁹ over the LRL algorithm for even moderately contracted low angular momentum functions is however quite clear.

In the present work we aimed at devising an algorithm based on the Rys quadrature that would be especially efficient at handling highly contracted basis functions for all levels of basis function angular momentum. The algorithm that we outline below succeeds in doing this. In what follows we give a detailed description of how the new algorithm works for contracted s and p functions, because for such basis functions the presentation remains tractable and helps understand how and why the algorithm works. Modern basis sets do emphasize high angular momentum basis functions. The new algorithm will work in these cases as well. We note

however that the higher the angular momentum, the lower the effective contraction level for any group of four shells. Thus it appears that an optimal strategy might involve two (or more) different algorithms, one for moderately to highly contracted low angular momentum functions (s , p , most likely d), combined with one for lightly contracted high angular momentum basis functions (possibly d and higher angular momentum).

The "revisited" Rys quadrature algorithm described below combines the following features: (1) the numerical quadrature of DRK; (2) the "scaling" technique of GHGP; and (3) the transfer equations of HGP. The deterministic numerical quadrature step replaces the complex tree algorithm of GHGP, and relies on simple recursion relations for the two-dimensional integrals. In addition the numerical quadrature makes it possible to evaluate the needed elementary integrals on centers I and K directly rather than on the exponent-weighted charge-distribution centers. In this manner the "scaling" procedure is simplified. The resulting algorithm is anticipated to have a very attractive FLOP count.

In Sec. II we introduce the notations and give the basic equations of the numerical quadrature for completeness. The "scaling" procedure is described in Sec. III along with a detailed outline of the algorithm. Conclusions and outlook are given in Sec. IV. In what follows we adopt the notation of GHGP.⁹ Parentheses are used for integrals over contracted functions, square-brackets are used for primitive integrals, a mixed notation is used to indicate that the contraction summation has been carried out for only one of the two charge distributions, either $\chi_i\chi_j$ or $\chi_k\chi_l$.

II. THE RYS QUADRATURE

A. The numerical quadrature

It has been known since the early work of Boys¹ that electron repulsion integrals over primitive Gaussian functions can be expressed in the form,

$$[\eta_i\eta_j||\eta_k\eta_l] = \sum_{m=0}^L \text{Coef}_m F_m(X), \quad (5)$$

where

$$F_m(X) = \int_0^1 t^{2m} \exp(-Xt^2) dt, \quad (6)$$

and

$$L = \lambda_i + \lambda_j + \lambda_k + \lambda_l. \quad (7)$$

The value of X depends upon the exponential parameters a_i , a_j , a_k , and a_l , and the positions of the centers of the four Gaussians, but it is independent of the angular momentum indices, the n_x 's, n_y 's, and n_z 's. It follows immediately that Eq. (5) can be written,

$$[\eta_i\eta_j||\eta_k\eta_l] = \int_0^1 P_L(t^2) \exp(-Xt^2) dt, \quad (8)$$

where $P_L(t^2)$ is a polynomial of degree L in t^2 with coefficients Coef_m . The integral in Eq. (8) can be evaluated exactly by an N -point numerical quadrature formula,

$$[\eta_i \eta_j \| \eta_k \eta_l] = \sum_{\alpha=1}^N P_L(t_\alpha^2) W_\alpha, \tag{9}$$

where N is any integer satisfying the condition,

$$N > \frac{L}{2}, \tag{10}$$

t_α is a positive root of the N th polynomial $R_N(t, X)$,

$$R_N(t_\alpha, X) = 0, \tag{11}$$

and W_α is a weight factor, which depends upon the value of X . $R_N(t, X)$ is the member of degree $2N$ of a set of polynomials (the Rys polynomials) of even order in the variable t , orthonormal in the interval $[0, 1]$ with respect to the weight factor $\exp(-Xt^2)$,

$$\int_0^1 R_N(t, X) R_M(t, X) \exp(-Xt^2) dt = \delta_{NM}. \tag{12}$$

In the early work of King and Dupuis,¹⁵ the roots and weights were evaluated by polynomial approximations valid over finite intervals of X although interpolation approaches should prove more efficient.

B. The polynomial $P_L(t^2)$

We express the Coulomb operator $1/r_{12}$ as a Gaussian transform,

$$1/r_{12} = \frac{2}{\sqrt{\pi}} \int_0^\infty \exp(-u^2 r_{12}^2) du. \tag{13}$$

A change in order of integration leads to

$$[\eta_i \eta_j \| \eta_k \eta_l] = \frac{2}{\sqrt{\pi}} \int_0^\infty [\eta_i \eta_j | \exp(-u^2 r_{12}^2) | \eta_k \eta_l] du, \tag{14}$$

and the integrand in Eq. (14) factors into the product of three two-dimensional integrals associated with the three axes of a coordinate system,

$$[\eta_i \eta_j | \exp(-u^2 r_{12}^2) | \eta_k \eta_l] = I'_x I'_y I'_z. \tag{15}$$

Of the twelve nx , ny , nz indices, only four nx values enter into the I'_x term,

$$I'_x(nxi, nxj, nxk, nxl)$$

$$= \int \int (x_1 - x_i)^{nxi} (x_1 - x_j)^{nxj} (x_2 - x_k)^{nxk} (x_2 - x_l)^{nxl} \times \exp(-Q_x) dx_1 dx_2, \tag{16}$$

where

$$Q_x = a_i(x_1 - x_i)^2 + a_j(x_1 - x_j)^2 + a_k(x_2 - x_k)^2 + a_l(x_2 - x_l)^2 + u^2(x_1 - x_2)^2. \tag{17}$$

Here x_i is the x coordinate of the center I of primitive η_i and nxi is the corresponding nx angular momentum index, and similar notation is used for η_j , η_k , and η_l . We define some new quantities,

$$x_A = (a_i x_i + a_j x_j) / (a_i + a_j), \tag{18}$$

$$x_B = (a_k x_k + a_l x_l) / (a_k + a_l), \tag{19}$$

$$A = a_i + a_j, \tag{20}$$

$$B = a_k + a_l, \tag{21}$$

$$\rho = AB / (A + B), \tag{22}$$

$$D_x = \rho(x_A - x_B)^2, \tag{23}$$

$$G_x = (a_i a_j / (a_i + a_j))(x_i - x_j)^2 + (a_k a_l / (a_k + a_l))(x_k - x_l)^2. \tag{24}$$

In terms of these, Q_x becomes

$$Q_x = G_x + A(x_l - x_A)^2 + B(x_2 - x_B)^2 + u^2(x_1 - x_2)^2. \tag{25}$$

We then make a change of variable from u to t ,

$$u^2 = \rho t^2 / (1 - t^2), \tag{26}$$

$$t^2 = u^2 / (\rho + u^2), \tag{27}$$

$$dt = \rho(\rho + u^2)^{-3/2} du. \tag{28}$$

As u varies from zero to infinity, t varies from zero to unity. Finally we define a modified form of the two-dimensional integrals,

$$I_x = \exp(D_x t^2) (1 - t^2)^{-1/2} I'_x, \tag{29}$$

and similarly I_y related to I'_y and I_z to I'_z . Substitution of Eqs. (15), (26)–(29) into Eq. (14) yields Eq. (8), where

$$P_L(t^2) = 2(\rho/\pi)^{1/2} I_x I_y I_z, \tag{30}$$

and using y and z analogs of Eq. (23), it follows that

$$X = D_x + D_y + D_z. \tag{31}$$

Substitution of Eq. (30) into Eq. (9) yields the working formula for the electron repulsion integral,

$$[\eta_i \eta_j \| \eta_k \eta_l] = \sum_{\alpha=1}^N I_x(u_\alpha) I_y(u_\alpha) I_z^\#(u_\alpha), \tag{32}$$

where

$$I_z^\#(u_\alpha) = 2(\rho/\pi)^{1/2} I_z(u_\alpha) W_\alpha. \tag{33}$$

C. Recursion formulas for the two-dimensional integrals

We define

$$g_{n,m}^x(x_1, x_2) \equiv (x_1 - x_i)^n (x_2 - x_k)^m \exp[-A(x_1 - x_A)^2 - B(x_2 - x_B)^2 - u^2(x_1 - x_2)^2], \tag{34}$$

and the two-dimensional integral,

$$G_{n,m}^x = \int \int g_{n,m}^x(x_1, x_2) dx_1 dx_2. \tag{35}$$

$G_{n,m}^x$ is related to I'_x of Eq. (16) by placing the polynomial factors of the η_j and η_l primitives on center I and K . Thus,

$$G_{n,m}^x \exp(-G_x) = I'_x(nxi + nxj, 0, nxk + nxl, 0, u). \tag{36}$$

Rys *et al.*⁶ showed that

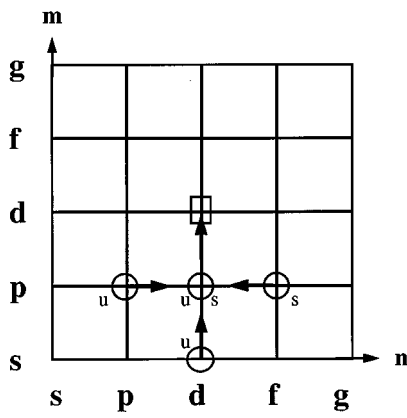


FIG. 1. Diagrammatic representation of the recursion formula Eq. (45). u stands for unscaled integral block, and s stands for scaled integral block.

$$\frac{m}{2} G_{n,m-1}^x + \{B(x_B - x_k) + u^2(x_i - x_k)\} G_{n,m}^x + u^2 G_{n+1,m}^x - (B + u^2) G_{n,m+1}^x = 0, \quad (37)$$

$$\frac{n}{2} G_{n-1,m}^x + \{A(x_A - x_i) + u^2(x_k - x_i)\} G_{n,m}^x - (A + u^2) G_{n+1,m}^x + u^2 G_{n,m+1}^x = 0. \quad (38)$$

We make use of Eqs. (18)–(21) and sum Eqs. (37) and (38) to obtain a new recursion formula,

$$B G_{n,m+1}^x = -A G_{n+1,m}^x + \frac{n}{2} G_{n-1,m}^x + \frac{m}{2} G_{n,m-1}^x + \{a_j(x_j - x_i) + a_l(x_l - x_k)\} G_{n,m}^x. \quad (39)$$

Equation (39) is the key equation that underlies the new formulation. It provides a means to transfer angular momentum from electron 1 to electron 2. Indeed, Eq. (39) says that, in the (n,m) plane, point $(n,m+1)$ which lies east of (n,m) can be obtained from (n,m) and its north $(n+1,m)$, west $(n,m-1)$, and south $(n-1,m)$ neighbors. Analogous equations hold for y and z . Equation (39) is depicted in Fig. 1.

In their earlier work DRK made use of other recursion relations, also derived from Eqs. (27) and (28). Of those we repeat only one that will be useful to us,

$$G_{n+1,0}^x = n B_{10} G_{n-1,0}^x + C_{00}^x G_{n,0}^x, \quad (40)$$

where C_{00}^x and B_{10} are given by

$$C_{00}^x = (x_A - x_i) + \frac{B(x_B - x_A)}{A+B} t^2, \quad (41)$$

$$B_{10} = \frac{1}{2A} - \frac{B}{2A(A+B)} t^2. \quad (42)$$

Finally we note the transfer equations that transfer angular momentum from center I to center J , and from center K to center L ,

$$I_x'(nxi, nxj, m, 0) = I_x'(nxi + 1, nxj - 1, m, 0) + (x_i - x_j) I_x'(nxi, nxj - 1, m, 0), \quad (43)$$

$$I_x'(nxi, nxj, nxk, nxl) = I_x'(nxi, nxj, nxk + 1, nxl - 1) + (x_k - x_l) I_x'(nxi, nxj, nxk, nxl - 1). \quad (44)$$

Similar equations hold for I_x , I_y , and I_z .

D. Recursion formulas for primitive integrals

The recursion formulas given above are valid for the two-dimensional integrals that are used to build the primitive electron repulsion integrals as sums of products of the two-dimensional integrals for N roots, according to Eq. (32). These recursion relations can be turned into recursion formulas applicable directly to the six-dimensional primitive integrals. Multiplication of Eq. (39) by $\exp(-G_x)$, then by $\exp(D_x t_\alpha^2)(1 - t_\alpha^2)^{-1/2}$, and by I_y and I_z , followed by summation over roots and weights α , leads to the following equation:

$$B[n, 0 \| m + 1, 0] = -A[n + 1, 0 \| m, 0] + \frac{n}{2}[n - 1, 0 \| m, 0] + \frac{m}{2}[n, 0 \| m - 1, 0] + \{a_j(x_j - x_i) + a_l(x_l - x_k)\}[n, 0 \| m, 0]. \quad (45)$$

Recall that the bracket notation refers to six-dimensional integrals, in this case integrals over primitive Gaussian (uncontracted) functions. There is one more remark to make about Eq. (45). In contrast to Eq. (39) that is valid for the two-dimensional integral along the \mathbf{x} direction, and that offers no possibility of confusion, in Eq. (45) the direction for the increment or decrement $\mathbf{1}$ of angular momentum is dictated by the coefficient of the fourth term in the sum. We really ought to use the notation $\mathbf{1}_x$. However, to avoid complexity, we will keep the notation $\mathbf{1}$ without subscript, and we will have to remember the implication due to the coefficient $\{a_j(x_j - x_i) + a_l(x_l - x_k)\}$. Equation (45) is closely related to Eq. (30) of LRL.¹⁰ The difference lies in our working with centers I, J, K , and L directly, in contrast with these authors working with centers A and B .

Just as before, from Eqs. (43) and (44) substituted into Eq. (29) and then Eq. (32) it follows that

$$[ni, nj \| m, 0] = [ni + 1, nj - 1 \| m, 0] + (x_i - x_j) \times [ni, nj - 1 \| m, 0] \quad (46)$$

and

$$[ni, nj \| nk, nl] = [ni, nj \| nk + 1, nl - 1] + (x_k - x_l) \times [ni, nj \| nk, nl - 1]. \quad (47)$$

Equations (46) and (47) are the ‘‘horizontal recursion relations’’ of Head-Gordon and Pople⁸ applied in the \mathbf{x} direction. In these equations the variables ni , nj , nk , and nl have a collective meaning, for example, ni refers to the triplet (nxi, nyi, nzi) and similarly for nj , nk , and nl . The direction to which the transfer applied is implied by the factors $(x_i - x_j)$ and $(x_k - x_l)$. Note that these relations are independent of the Gaussian exponents, something that is not the case for

Eq. (45). Thus Eqs. (46) and (47) may be applied outside the sums over primitive functions, while Eq. (45) must be applied inside the summation loop.

Thus the computational strategy takes shape as follows: inside the loop over primitives, calculate the primitive integrals of the form $[n+m,0||0,0]$, use Eq. (45) to generate $[n,0||m,0]$, and sum up into the integrals over contracted functions $(n,0||m,0)$. Outside the loop over primitives, apply the transfer equations (46) and (47) to form the final integrals $(ni,nj||nk,nl)$. This algorithm is equivalent to the second LRL algorithm in Ref. 10.

III. THE RYS QUADRATURE REVISITED

Electron repulsion integrals over contracted basis functions can be written as a double sum over the $\eta_i\eta_j$ and $\eta_k\eta_l$ products of primitive Gaussians with the primitive integrals multiplied by the charge distribution coefficients $C_A = c_{ai}c_{aj}$ and $C_B = c_{ak}c_{al}$ [the c 's are the contraction coefficients of Eq. (4)]. Thus we write

$$(n,0||m,0) = \sum_B \sum_A [n,0||m,0] C_B C_A. \tag{48}$$

The notation is adapted from GHGP, the parentheses are the result of contraction of the square brackets for the bra $[n,0|$ as well as for the ket $|m,0]$.

Let us assume for the moment that we have at our disposal the primitive integrals $[n,0||m,0]$, $[n+1,0||m,0]$, $[n-1,0||m,0]$, and $[n,0||m-1,0]$, and that we would like to calculate not only the contracted integrals $(n,0||m,0)$, $(n+1,0||m,0)$, $(n-1,0||m,0)$, and $(n,0||m-1,0)$, but also the integrals $(n,0||m+1,0)$. For the first four groups of integrals we apply Eq. (48) directly. For the last group, $(n,0||m+1,0)$, we follow the procedure described below. This step is at the heart of the method. It corresponds to transferring angular momentum from electron 1 to electron 2.

Substitution of Eq. (45) into Eq. (48) yields

$$\begin{aligned} (n,0||m+1,0) &= \sum_B \frac{1}{B} C_B \sum_A -A [n+1,0||m,0] C_A \\ &+ \frac{n}{2} \sum_B \frac{1}{B} C_B \sum_A [n-1,0||m,0] C_A \\ &+ \frac{m}{2} \sum_B \frac{1}{B} C_B \sum_A [n,0||m-1,0] C_A \\ &+ (x_j - x_i) \sum_B \frac{1}{B} C_B \sum_A a_j [n,0||m,0] C_A \\ &+ (x_l - x_k) \sum_B \frac{a_l}{B} C_B \sum_A [n,0||m,0] C_A. \end{aligned} \tag{49}$$

The strategy to be followed is explicitly depicted in Eq. (49). Contraction of the primitive integrals $[n-1,0||m,0]$, $[n,0||m-1,0]$, and $[n,0||m,0]$ over $\eta_i\eta_j$ [terms 2, 3, and 5 in

Eq. (49)] can be carried out inside the loop over A . The contraction of the primitive integrals $[n+1,0||m,0]$ and $[n,0||m,0]$ [terms 1 and 4 in Eq. (49)] can be carried out also inside the loop over A , but only after they have been **scaled** by $-A$ and a_j , respectively. Inside the loop over B (loop over $\eta_k\eta_l$) contraction over B is performed after **scaling** the first four partial sums by $1/B$ and the fifth sum by a_l/B (in the present case the scaling factor $1/B$ could be absorbed into C_B , this is not true in general). The final contracted integrals $(n,0||m+1,0)$ can be generated outside the loop over primitive functions by adding the five terms as in Eq. (49) after multiplication by the appropriate factors $(x_j - x_i)$ and $(x_l - x_k)$, or their y and z analogs, factors that are independent of the primitive functions.

The situation is however a little more complicated than that. Indeed the various integral groups in Eq. (45) are themselves obtained through application of Eq. (45) for $0,1,\dots,n$, and $0,1,\dots,m$. Repeated application of Eq. (45) provides a means to compute any integral group $[i,0||k,0]$ from $[n+m,0||0,0]$ with the condition $i+k \leq n+m$. This is truly the heart of the method. In what follows we derive the expressions for the $[i,0||k,0]$ group directly in terms of the $[i,0||0,0]$ groups. These expressions are the key equations of the new algorithm and show precisely which scaled integrals need be calculated. Since there is no possibility of confusion we will use the notation $[n||m]$ to represent the $[n,0||m,0]$ integral group, and $[s\ p\ d\ f\ g\dots||m]$ to represent the ‘‘row vector’’ of integral groups $[s||m], [p||m], \dots$. The derivation that follows initially deals with the ‘‘diagonal’’ integrals where the angular momentum exponents are accumulated on x , or y , or z only. The generalization is not very much more complicated, it requires that one pays attention to the exact combination of angular momentum exponents that are being created. This point is illustrated towards the end of this section.

The recursion relation Eq. (45) substituted into Eq. (48) can be conveniently represented in a matrix notation. Let us denote L the total angular momentum for a group of four shells, we define two matrices of order $L+1$ denoted I and $R(L,1_x)$. I stands for the identity matrix of order $L+1$ here (not the basis function center). We define the quantity labeled c_x ,

$$c_x = \{a_j(x_j - x_i) + a_l(x_l - x_k)\}. \tag{50}$$

The $R(L,1_x)$ matrix is a tridiagonal matrix with $-A$ on the lower diagonal, with $i/2$ on the upper diagonal where $i = 1 \dots L+1$ is the row index, and c_x on the diagonal. The notation reflects that $R(L,1_x)$ is a matrix function of L and of one angular momentum direction, x in the present case. This is because Eq. (45) raises the power of the x angular monomial by $\mathbf{1}$, and equivalent equations can be written for y and z . The angular momentum direction 1_x is linked to the diagonal term c_x . For the time being we may ignore the subscript in c_x for ease of notation. At some point however, we will not be able to ignore it any longer, as discussed later. Thus for $L=4$ we have

$$R(L,1) = \begin{bmatrix} c & \frac{1}{2} & 0 & 0 & 0 \\ -A & c & \frac{2}{2} & 0 & 0 \\ 0 & -A & c & \frac{3}{2} & 0 \\ 0 & 0 & -A & c & \frac{4}{2} \\ 0 & 0 & 0 & -A & c \end{bmatrix}. \quad (51)$$

Using this notation Eq. (49) takes the form

$$[s \ p \ d \cdots L \| m+1] = 1/B \{ [s \ p \ d \cdots L \| m] R(L,1) + [s \ p \ d \cdots L \| m-1] \} m/2. \quad (52)$$

Successive applications of Eq. (52) yield the expression relating $[s \ p \ d \cdots L \| m]$ to $[s \ p \ d \cdots L \| s]$ for any $m, m = 1 \cdots L/2$. We have

$$\begin{aligned} [s \ p \ d \cdots L \| p] &= [s \ p \ d \cdots L \| s] R_p, \\ [s \ p \ d \cdots L \| d] &= [s \ p \ d \cdots L \| s] R_d, \\ &\vdots \\ [s \ p \ d \cdots L \| L/2] &= [s \ p \ d \cdots L \| s] R_{L/2}. \end{aligned} \quad (53)$$

The notation in Eq. (53) indicates that each integral group $[s \| d], [p \| d], \dots, [L \| d]$ is, in a generic sense, equal to a linear combination of the integral groups $[s \| s], [p \| s], \dots, [L \| s]$, with the coefficients being the matrix elements of the matrix R_m , $m = 1 \cdots L/2$,

$$\begin{aligned} R_p &= \frac{1}{B} R, \\ R_d &= \frac{1}{B^2} \left(R^2 + \frac{B}{2} I \right), \\ R_f &= \frac{1}{B^3} \left(R^3 + \frac{3B}{2} R \right), \\ R_g &= \frac{1}{B^4} \left(R^4 + 3BR^2 + \frac{3B^2}{4} I \right), \\ &\dots \end{aligned} \quad (54)$$

In Eq. (54) we used the shortened notation R for the $R(L,1)$ matrix of Eq. (51). These equations provide a general recipe to express any group $[s \ p \cdots L \| m]$ in terms of $[s \ p \cdots L \| s]$ for any m and L . Symbolic algebra software can be conveniently used to obtain the matrices R_d, R_f, R_g, \dots . For an integral group $[ni, nj \| nk, nl]$ the target integral groups are $[\max(ni, nj) \cdots ni + nj \| \max(nk, nl)]$ up to $[\max(ni, nj) \cdots ni + nj \| nk + nl]$. As an example a $[pp \| pp]$ group requires the four groups $[p \| p], [p \| d], [d \| p], [d \| d]$. The formula for $[p \| p]$, in terms of $[s \| s], [p \| s]$, and $[d \| s]$ is found as the p, p elements of R_p , for $[d \| p]$ it is the element d, p of R_p , for $[p \| d]$ it is the element p, d of the matrix R_d , and for $[d \| d]$ it is the element d, d of R_d in accord with Eq. (54). We have

$$\begin{aligned} [p \| p] &= \frac{1}{B} \left(\frac{1}{2} [s \| s] + c [p \| s] - A [d \| s] \right), \\ [d \| p] &= \frac{1}{B} ([p \| s] + c [d \| s] - A [f \| s]), \end{aligned} \quad (55)$$

$$[p \| d] = \frac{1}{B^2} \left(c [s \| s] + \left\{ c^2 - \frac{3}{2} A + \frac{B}{2} \right\} [p \| s] - 2Ac [d \| s] + A^2 [f \| s] \right),$$

$$[d \| d] = \frac{1}{B^2} \left(\frac{1}{2} [s \| s] + 2c [p \| s] + \left\{ c^2 - \frac{5}{2} A + \frac{B}{2} \right\} [d \| s] - 2Ac [f \| s] + A^2 [g \| s] \right).$$

In this example, Eq. (55) says that in order to get the contracted $(p \| p)$ block we need the $[s \| s]$ block, the $[p \| s]$ block **scaled** with a_j [recall the definition of c given earlier Eq. (50)] and the $[d \| s]$ block **scaled** with $-A$. All these are computed inside the loop over the pairs of $\eta_i \eta_j$ primitives. Inside the loop over the $\eta_k \eta_l$ primitives, all the $\eta_i \eta_j$ -contracted blocks get scaled with a_l or B -dependent factors. While these formulas may appear cumbersome, it must be remembered that the algorithm makes extensive reuse of the calculated basic integral blocks $[s \| s], [p \| s], \dots, [g \| s]$ and that scaling is relatively inexpensive, some of the scaling occurring outside the inner loops as seen in Eq. (49).

The target integrals blocks for the low angular momentum integrals up to $(dd \| dd)$ are given in the Appendix. The Appendix provides a global view of block scaling, i.e., which block must be computed, so that it is possible to generate all needed individual integrals. We do find there the terms in Eq. (55) for the $(d \| d)$ integrals. The complexity of the formulas grows quickly with the target integral blocks, i.e., with the angular momentum. As the effective degree of contraction K decreases with increasing angular momentum we may want to compromise between ease of programming and computational gains. In practice we can decide to write a computer code that covers all cases with basis functions up to d type. The recursion formula is represented diagrammatically as in Fig. 1.

Before returning to the discussion of the efficiency of the proposed algorithm, we must recall that equations such as Eq. (55) were derived for the case when the angular momentum grows along one direction only. For example, Eq. (55) is correct when both p 's are identical, let us say the $(s \| s)$ and $(p_x \| s)$ integrals lead to the $(s \| p_x)$ integrals and the $(s \| s)$, $(p_x \| s)$, and $(d_{xx} \| s)$ integrals lead to the $(s \| d_{xx})$ integral. To obtain cross integrals such as $(s \| d_{xy})$ we must raise the angular momentum in the x direction and in the y direction. We accomplish this by combining the recursion formulas in two directions,

$$[s p d \cdots L \| m_x + 1] = 1/B \{ [s p d \cdots L \| m_x] R(L, 1_x) + [s p d \cdots L \| m_x - 1] \} (m_x/2), \tag{56}$$

$$[s p d \cdots L \| m_y + 1] = 1/B \{ [s p d \cdots L \| m_y] R(L, 1_y) + [s p d \cdots L \| m_y - 1] \} (m_y/2). \tag{57}$$

Recall that $R(L, 1_x)$ and $R(L, 1_y)$ differ through the diagonal terms, c_x and c_y , respectively. Application of Eq. (56) leads from $(p_y \| s)$ and $(d_{xy} \| s)$ to $(p_y \| p_x)$, and from $(s \| s)$ and $(p_x \| s)$ to $(s \| p_x)$. Application of Eq. (57) then leads from $(p_y \| p_x)$ and $(s \| p_x)$ to the desired integral $(s \| d_{xy})$. In place of the second equation in Eq. (53) we can write

$$[s p d \cdots L \| pp'] = [s p d \cdots L \| s] R_{pp'}, \tag{58}$$

with

$$R_{pp'} = \frac{1}{B^2} R R' \equiv \frac{1}{B^2} R(L, 1_x) R(L, 1_y). \tag{59}$$

Equation (59) is closely related to the second equation in Eq. (54) giving R_d for like-angular momentum raising. For higher angular momenta we can define other needed transformation matrices R_{dp} , $R_{pp'p''}$, $R_{dd'}$, $R_{dpp'}$, and so on. The list of scaled blocks for “diagonal” and “nondiagonal” integrals are identical, albeit the individual integral expressions are different. The approach is completely general and the formulas can be obtained with a symbolic algebra software.

IV. DISCUSSION

The key factor that makes the proposed algorithm computationally efficient lies in the very small number of terms in the “row vector” of integral groups $[s p d \cdots L \| s]$ that must be computed in the inner loop. This number is equal to $(L+1)(L+2)(L+3)/6$. In the $(pp \| pp)$ the “row vector” of blocks $[s p d f g \| s]$ contains 35 integrals, in the $(dd \| dd)$ case it contains 165 integrals. In contrast the number of individual target integrals grows as $\{(l+1)[7(l+1)^2 - 1]/6\}^2$ where $l=L/4$. This number is $9^2=81$ for $(pp \| pp)$ and $31^2=961$ for $(dd \| dd)$. These latter numbers correspond to the work done inside all four loops over primitives in the LRL algorithm. To the extent that the additional work due to scaling and building of the target integrals from the scaled integrals stays at a reasonable level, this algorithm will be superior to LRL’s for moderately and highly contracted functions. Indeed this should be the case. We note as well that the computational dependence for the $[s p d \cdots L \| s]$ of $(L+1)(L+2)(L+3)/6$ is smaller than the one reported by Ishida.¹²

Comparison of the present algorithm with the GHGP algorithm shows that both require computation of the same number of basic integrals inside the inner loop over primitives, mainly $[s p d \cdots L \| s]$ integrals just as the MD method. For MD and GHGP these primitive integrals are centered on the exponent-weighted charge distribution center of the primitives, labeled A and B in Eqs. (18) and (19). A step required for MD and GHGP involves the transferring of the target primitive integrals from the exponent-weighted primi-

TABLE I. FLOP counts for various shell quartets using different algorithms.

		PH Ref. 3	GHGP Ref. 9	LRL Ref. 10	DRK Ref. 5	DM This work ^a
$(pp \ pp)$	x	220	300	691	904	440
	y	2300	610	30	30	140
	z	4000	680	330	0	650
$(dd \ dd)$	x		2450	10 255	14 660	2575
	y		25 800	30	30	2550
	z		28 900	11 300	0	18 800
$(sp sp \ sp sp)$	x	220	450	753	1056	
	y	2300	1300	30	30	
	z	4000	1700	800	0	

^aEstimates. See text for explanations.

tive centers to the actual primitive centers. Although creating the target integrals $[s p d \cdots \| s p d \cdots]$ from the basic integrals $[s p d \cdots L \| s]$ is trivial in their approaches, the transfer from exponent-weighted centers to primitive centers requires additional work. In our approach this step is completely bypassed as the numerical quadrature yields primitive integrals directly centered on the primitive centers. An added benefit of the numerical quadrature is the extreme simplicity of computing the basic integrals, in contrast to the complexity of the tree search in the GHGP algorithm. Of course once we have calculated $(n0 \| m0)$, then we can use the best possible strategy to generate the $(i j \| kl)$ integrals.¹⁶

It has become usual to express the performance of algorithms for the calculation of electron repulsion integrals by expressing the number of floating-point operations (FLOP) needed to compute $(pp \| pp), (dd \| dd), \dots$ integrals as a function of the degree of contraction \mathbf{K} , assuming that all four shells p, d, \dots have the same degree of contraction. The expression is

$$\text{FLOP} = xK^4 + yK^2 + z. \tag{60}$$

Equation (60) reflects the loop structure of the algorithm, x corresponds to the work done inside all four loops over primitives, y corresponds to the work done outside the two inner loops ($\eta_i \eta_j$ -contraction) but inside the two outer loops ($\eta_k \eta_l$ -contraction), and z corresponds to the work done outside all four contraction loops. Table I gives $x, y,$ and z for the PH, GHGP, LRL, and DRK algorithms, along with estimates for the present algorithm. These estimates are based on the analysis that follows. The analysis becomes quickly complex and higher angular momentum cases are not considered here.

A. Estimates for $(pp \| pp)$ and $(dd \| dd)$

To gain insight in the potential of the present approach, we consider the cases of $(pp \| pp)$ and $(dd \| dd)$ integrals.

1. Evaluation of the roots and weights of the Rys polynomials

The evaluation of the Rys roots and weights is done inside the K^4 loop since the argument X [Eq. (31)] depends on the exponents of the primitive functions. For the $(pp \| pp)$ case we need 3 roots and 3 weights, and for the $(dd \| dd)$ case 5 roots and 5 weights. We can adopt an efficient

TABLE II. Algorithm for constructing the $[g||s]$ primitive integrals for the $(pp||pp)$ integrals.

1)	s			21)	$xxxx$			$= 1) * G(xxxx)$
2)	x			22)	xxx	y		$= 3) * G(xxx)$
3)		y		23)	xx		z	$= 4) * G(xxz)$
4)			z	24)	xx	yy		$= 5) * G(yy)$
5)	xx			25)	xx	y	z	$= 9) * G(x)$
6)	x	y		26)	xx		zz	$= 10) * G(xx)$
7)	x		z	27)	x	yyy		$= 2) * G(yyy)$
8)		yy		28)	x	yy	z	$= 18) * G(x)$
9)		y	z	29)	x	y	zz	$= 19) * G(x)$
10)			zz	30)	x		zzz	$= 20) * G(x)$
11)	xxx			31)		yyy		$= 1) * G(yyyy)$
12)	xx	y		32)		yy	z	$= 4) * G(yyy)$
13)	xx		z	33)		yy	zz	$= 10) * G(yy)$
14)	x	yy		34)		y	zzz	$= 20) * G(y)$
15)	x	y	z	35)			$zzzz$	$= G(zzzz)$
16)	x		zz					
17)		yyy						
18)		yy	z					
19)		y	zz					
20)			zzz					

approach¹⁷ involving lookup tables and interpolations. A 3-point interpolation, with 3 multiplications and 2 additions, can yield roots and weights of sufficient accuracy for a total of $6 * 5 = 30$ FLOPs for $(pp||pp)$ and $10 * 5 = 50$ FLOPs for $(dd||dd)$. No additional work is necessary for sp shells (Table II).

2. Evaluation of the two-dimensional integrals

$I_x(nx, 0, t_\alpha)$, $I_y(ny, 0, t_\alpha)$, and $I_z(nz, 0, t_\alpha)$

This step is done inside the K^4 loop. G_x , G_y , and G_z are calculated by means of the 2-point recursion formula of Eq. (40) for each root t_α . There are 3 roots for the $(pp||pp)$ and $(spsp||spsp)$ cases, and 5 roots for the $(dd||dd)$ case. For $(pp||pp)$ integrals the indices nx , ny , and nz run up to 4. The cost of this step is $3 * (7 \text{ multiplications plus } 6 \text{ additions}) = 39$ FLOPs. For the $(dd||dd)$ case we have $5 * (15 \text{ multiplications plus } 14 \text{ additions}) = 145$ FLOPs. About 16 FLOPs per root are needed to create the coefficients in Eqs. (41) and (42), about 48 FLOPs for the $(pp||pp)$ case and 80 FLOPs for the $(dd||dd)$ case.

3. Evaluation of the primitive integrals $[n||s]$

For $(pp||pp)$ we need $[nx ny nz || 0]$ for $nx, ny, nz = 1$ to 4, i.e., $[g||s]$ integrals. There are 35 such integrals. This K^4 step is carried out according to Eq. (32) in the most general case. In the case at hand, we can take advantage of the fact that whenever nx , or ny , or nz is equal to zero, the corresponding G_x , or G_y , or G_z factor is equal to 1.0. We do not have to multiply by that factor. The cost per integral is essentially one multiplication to form the product $G_x * G_y * G_z$ for each root plus $(nroots - 1)$ additions for the summation over roots (Table II). For $(pp||pp)$ this amounts to $(35 - 5) * 3 + 35 * 2 = 160$ FLOPs. For $(dd||dd)$ it amounts to $(165 - 9) * 5 + 165 * 4 = 1440$ FLOPs.

4. Evaluation of the scaled integrals

Tables III and IV illustrate the block scaling, at the K^4 level and at the K^2 level, respectively. Scaling has to be

performed to generate the complete set of terms needed for the $(pp||pp)$ integrals. The tables are based on the formulas given in the Appendix. The work amounts to 79 (multiply+add) inside the K^4 loop and 53 (multiply+add) inside the K^2 loop. For sp shells these numbers are 90 and 59, respectively. For $(dd||dd)$ integrals, a similar analysis shows that the K^4 scaling can be realized with 856 (multiply+add), while the K^2 scaling amounts to 1258 (multiply+add).

5. K^0 work and horizontal recursion formulas

Once the scaled integrals have been calculated, the final integrals $(n, 0 || m, 0)$ have to be assembled before the horizontal recursion formulas can be applied. Inspection of the expressions given in the Appendix and of the actual expres-

TABLE III. K^4 scaling for the $(pp||pp)$ and $(dd||dd)$ integrals.^a

	$(pp pp)$		$(dd dd)$	
	K^4	K^2	K^4	K^2
a_j	sp d		sp d f g	
A	sp d f		sp d f g h	
a_j^2	sp d		sp d f g	
$A a_j$	p d f		p d f g h	
A^2	d f g		d f g h i	
a_j^3			p d f g	
$A a_j^2$			d f g h	
$A^2 a_j$			f g h i	
A^3			g h i j	
a_j^4			d f g	
$A a_j^3$			f g h	
$A^2 a_j^2$			g h i	
$A^3 a_j$			h i j	
A^4			i j k	

^aEntry n means that a primitive block $[n || 0]$ gets scaled by the coefficient in column 1. For example in the $(pp||pp)$ case, the first entry in the table indicates that the $[s || 0]$, $[p || 0]$, and $[d || 0]$ blocks get scaled with the a_j coefficients to form three scaled blocks. Entries in italics are needed for sp shells.

TABLE IV. K^2 scaling for the $(pp||pp)$ and $(dd||dd)$ integrals.^a

	$(pp pp)$		$(dd dd)$	
	K^4	K^2	K^4	K^2
a_i		s p d		s p d f
$(a_j)a_i$		s p d		s p d f g
$(A)a_i$		p d f		p d f g h
$(a_j^2)a_i$				p d f g
$(Aa_j)a_i$				d f g h
$(A^2)a_i$				f g h i
$(a_j^3)a_i$				d f g
$(Aa_j^2)a_i$				f g h
$(A^2a_j)a_i$				g h i
$(A^3)a_i$				h i j
$B a_i$				p d f g
$(a_j)B a_i$				d f g
$(A)B a_i$				f g h
a_i^2		s p d		s p d f g
$(a_j)a_i^2$				p d f g
$(A)a_i^2$				d f g h
$(a_j^2)a_i^2$				d f g
$(Aa_j)a_i^2$				f g h
$(A^2)a_i^2$				g h i
$B a_i^2$				d f g
a_i^3				p d f g
$(a_j)a_i^3$				d f g
$(A)a_i^3$				f g h
B		s p d		s p d f g
$(a_j)B$				p d f g
$(A)B$				d f g h
$(a_j^2)B$				d f g
$(A^2)B$				g h i
B^2				d f g

^aSee explanations in Table III.

sions in the $(pp||pp)$ case suggest that in average each of the $(pd||pd)$ integrals needed for a $(pp||pp)$ block is the sum of about 4 unscaled or scaled contributions, and each of the $(dfg||dfg)$ integrals needed for a $(dd||dd)$ block is the sum of about 8 contributions, for a total of about 320 FLOPs and 7500 FLOPs, respectively. Finally the horizontal recursion transformation from $(n,0||m,0)$ to $(i,j||kl)$ can be performed with the most efficient algorithm of LRL.¹⁰ Together these two steps yield the K^0 coefficient in Table I.

B. Discussion

It is clear from Table I that DRK remains the method of choice for uncontracted functions. This is not really unexpected. The other methods are more efficient for contracted functions. They were designed to take advantage of the concept of charge distribution to varying extents. This can be seen in Table I. MD's method falls in that category as well. Our goal was to devise an algorithm based on the Rys quadrature that would deal efficiently with contracted functions. The critical factor for success resides in the reduction in the number of fundamental integrals that need to be computed inside the K^4 loop. In the spirit of minimizing this number, it is clear that PH's method remains the most successful (many are equal to zero due to using local axis frames), albeit at the cost of much higher K^2 and K^0 coefficients. It is clear also from Table I that the present algorithm is superior to LRL's method for all the cases. The reason lies

in this minimum number of integrals computed in the K^4 loop. The trick about transferring angular momentum between electrons, a technique that we borrowed from MD and GHGP, is responsible for the gain, since both the LRL method and the present method use the same numerical quadrature.

Table I shows that the K^4 coefficient in the present method is close to the same coefficient in the GHGP's method, while the K^2 and K^0 coefficients are much smaller than those in GHGP. The similar magnitude of the K^4 coefficients between the two methods comes in part from the fact that both methods compute the same number of fundamental integrals. The much-reduced K^2 and K^0 coefficients in the present method is a direct reflection of the advantage of using a numerical quadrature and working directly with the basis function centers, rather than the charge distribution centers. The "scaling" step is much simpler because of that. That the new method does not do quite as well as GHGP for $(pp||pp)$ integrals shows that there is some inherent overhead in the numerical quadrature when dealing with low angular momentum functions. We do not seem to have succeeded in overcoming this point. However the present algorithm appears to be largely superior to GHGP's for $(dd||dd)$ integrals. We expect this to hold true for higher angular momentum functions. We expect the present method to be competitive for $(spsp||spsp)$ integrals, although we have not fully analyzed the effect of the different contraction coefficients for the s function and the p functions of an sp shell.

V. SUMMARY

We have outlined a novel approach to the calculation of electron repulsion integrals over Gaussian basis functions. It takes advantage of the best features of several approaches: the angular momentum transfer of McMurchie–Davidson, the scaling technique of Gill–Head–Gordon–Pople, the horizontal transfer equations proposed by Head–Gordon–Pople, and the Rys numerical integration of Dupuis–Rys–King. The resulting algorithm is anticipated to be quite competitive for contracted Gaussian functions. The derivation and implementation are simplified through the use of symbolic algebraic software. The method generalizes to high angular momentum integrals. Implementation is in progress and performance data will be presented in future reports. The extension of the method to integral (first- and second-) derivatives with respect to nuclear coordinates should also benefit from the advances devised in the present work.

ACKNOWLEDGMENTS

We dedicate this work to the memory of John Rys. We thank Dr. A. Komornicki for stimulating discussions and his careful reading of the paper. We thank also Dr. Graham Fletcher for useful discussions. This work was supported in part by the U.S. Department of Energy's Office of Biological and Environmental Research, and by the Office of Basic Energy Sciences, Chemical Physics Program. The Pacific Northwest National Laboratory is a multiprogram national laboratory operated for the U.S. Department of Energy by Battelle Memorial Institute under Contract No. DE-AC06-76RLO-1830.

APPENDIX: INTEGRAL BLOCK SCALING FOR ($d||d$) CHARGE DISTRIBUTIONS [$(pp||pp)$ INTEGRALS] AND ($g||g$) CHARGE DISTRIBUTIONS [$(dd||dd)$ INTEGRALS] (VALID FOR "DIAGONAL" INTEGRALS INVOLVING $p_x, d_{x^2}, f_{x^3}, g_{x^4}, \dots$)

A-1		A-5	
$[s p] \times B$		$[g p] \times B$	
$[s s]$	c	$[s s]$	
$[p s]$	$-A$	$[p s]$	
$[d s]$		$[d s]$	
$[f s]$		$[f s]$	2
$[g s]$		$[g s]$	c
$[h s]$		$[h s]$	$-A$
$[i s]$		$[i s]$	
$[j s]$		$[j s]$	
$[k s]$		$[k s]$	
A-2		A-6	
$[p p] \times B$		$[s d] \times B^2$	
$[s s]$	1/2	$[s s]$	$\{c^2 - 1/2A\} + 1/2B$
$[p s]$	c	$[p s]$	$-2Ac$
$[d s]$	$-A$	$[d s]$	A^2
$[f s]$		$[f s]$	
$[g s]$		$[g s]$	
$[h s]$		$[h s]$	
$[i s]$		$[i s]$	
$[j s]$		$[j s]$	
$[k s]$		$[k s]$	
A-3		A-7	
$[d p] \times B$		$[p d] \times B^2$	
$[s s]$		$[s s]$	c
$[p s]$	1	$[p s]$	$\{c^2 - 3/2A\} + 1/2B$
$[d s]$	c	$[d s]$	$-2Ac$
$[f s]$	$-A$	$[f s]$	A^2
$[g s]$		$[g s]$	
$[h s]$		$[h s]$	
$[i s]$		$[i s]$	
$[j s]$		$[j s]$	
$[k s]$		$[k s]$	
A-4		A-8	
$[f p] \times B$		$[d d] \times B^2$	
$[s s]$		$[s s]$	1/2
$[p s]$		$[p s]$	$2c$
$[d s]$	3/2	$[d s]$	$\{c^2 - 5/2A\} + 1/2B$
$[f s]$	c	$[f s]$	$-2Ac$
$[g s]$	$-A$	$[g s]$	A^2
$[h s]$		$[h s]$	
$[i s]$		$[i s]$	
$[j s]$		$[j s]$	
$[k s]$		$[k s]$	

A-9

	$[f d] \times B^2$
$[s s]$	
$[p s]$	$3/2$
$[d s]$	$3c$
$[f s]$	$\{c^2 - 7/2A\} + 1/2B$
$[g s]$	$-2Ac$
$[h s]$	A^2
$[i s]$	
$[j s]$	
$[k s]$	

A-10

	$[g d] \times B^2$
$[s s]$	
$[p s]$	
$[d s]$	3
$[f s]$	$4c$
$[g s]$	$\{c^2 - 9/2A\} + 1/2B$
$[h s]$	$-2Ac$
$[i s]$	A^2
$[j s]$	
$[k s]$	

A-11

	$[s f] \times B^3$
$[s s]$	$\{c^3 - 3/2Ac\} + 3/2Bc$
$[p s]$	$\{-3Ac^2 + 3/2A^2\} - 3/2BA$
$[d s]$	$3A^2c$
$[f s]$	$-A^3$
$[g s]$	
$[h s]$	
$[i s]$	
$[j s]$	
$[k s]$	

A-12

	$[p f] \times B^3$
$[s s]$	$\{3/2c^2 - 3/4A\} + 3/2B1/2$
$[p s]$	$\{c^3 - 9/2Ac\} + 3/2Bc$
$[d s]$	$\{-3Ac^2 + 3A^2\} - 3/2BA$
$[f s]$	$3A^2c$
$[g s]$	$-A^3$
$[h s]$	
$[i s]$	
$[j s]$	
$[k s]$	

A-13

	$[d f] \times B^3$
$[s s]$	$3/2c$
$[p s]$	$\{3c^2 - 3A\} + 3/2B$
$[d s]$	$\{c^3 - 15/2Ac\} + 3/2Bc$
$[f s]$	$\{-3Ac^2 + 9/2A^2\} - 3/2BA$
$[g s]$	$3A^2c$
$[h s]$	$-A^3$
$[i s]$	
$[j s]$	
$[k s]$	

A-14

	$[f f] \times B^3$
$[s s]$	$3/4$
$[p s]$	$9/2c$
$[d s]$	$\{9/2c^2 - 27/4A\} + 3/2B3/2$
$[f s]$	$\{c^3 - 21/2Ac\} + 3/2Bc$
$[g s]$	$\{-3Ac^2 + 6A^2\} - 3/2BA$
$[h s]$	$3A^2c$
$[i s]$	$-A^3$
$[j s]$	
$[k s]$	

A-15

	$[g f] \times B^3$
$[s s]$	
$[p s]$	3
$[d s]$	$9c$
$[f s]$	$\{6c^2 - 12A\} + 3/2B2$
$[g s]$	$\{c^3 - 27/2Ac\} + 3/2Bc$
$[h s]$	$\{-3Ac^2 + 15/2A^2\} - 3/2BA$
$[i s]$	$3A^2c$
$[j s]$	$-A^3$
$[k s]$	

A-16

	$[s g] \times B^4$
$[s s]$	$\{c^4 - 3Ac^2 + 3/4A^2\} + 3B\{c^2 - 1/2A\} + 3/4B^2$
$[p s]$	$\{-4Ac^3 + 6A^2c\} + 3B\{-2Ac\}$
$[d s]$	$\{6A^2c^2 - 3A^3\} + 3B\{A^2\}$
$[f s]$	$-4A^3c$
$[g s]$	A^4
$[h s]$	
$[i s]$	
$[j s]$	
$[k s]$	

A-17

$[p\ g] \times B^4$	
$[s\ s]$	$\{2c^3 - 3Ac\} + 3B\{c\}$
$[p\ s]$	$\{c^4 - 9Ac^2 + 15/4A^2\} + 3B\{c^2 - 3/2A\} + 3/4B^2$
$[d\ s]$	$\{-4Ac^3 + 12A^2c\} + 3B\{-2Ac\}$
$[f\ s]$	$\{6A^2c^2 - 5A^3\} + 3B\{A^2\}$
$[g\ s]$	$-4A^3c$
$[h\ s]$	A^4
$[i\ s]$	
$[j\ s]$	
$[k\ s]$	

A-18

$[d\ g] \times B^4$	
$[s\ s]$	$\{3c^2 - 3/2A\} + 3B\{1/2\}$
$[p\ s]$	$\{4c^3 - 12Ac\} + 3B\{2c\}$
$[d\ s]$	$\{c^4 - 15Ac^2 + 39/4A^2\} + 3B\{c^2 - 5/2A\} + 3/4B^2$
$[f\ s]$	$\{-4Ac^3 + 18A^2c\} + 3B\{-2Ac\}$
$[g\ s]$	$\{6A^2c^2 - 7A^3\} + 3B\{A^2\}$
$[h\ s]$	$-4A^3c$
$[i\ s]$	A^4
$[j\ s]$	
$[k\ s]$	

A-19

$[f\ g] \times B^4$	
$[s\ s]$	$3c$
$[p\ s]$	$\{9c^2 - 15/2A\} + 3B\{3/2\}$
$[d\ s]$	$\{6c^3 - 27Ac\} + 3B\{3c\}$
$[f\ s]$	$\{c^4 - 21Ac^2 + 75/4A^2\} + 3B\{c^2 - 7/2A\} + 3/4B^2$
$[g\ s]$	$\{-4Ac^3 + 24A^2c\} + 3B\{-2Ac\}$
$[h\ s]$	$\{6A^2c^2 - 9A^3\} + 3B\{A^2\}$
$[i\ s]$	$-4A^3c$
$[j\ s]$	A^4
$[k\ s]$	

A-20

$[g\ g] \times B^4$	
$[s\ s]$	$3/2$
$[p\ s]$	$12c$
$[d\ s]$	$\{18c^2 - 21A\} + 3B\{3\}$
$[f\ s]$	$\{8c^3 - 48Ac\} + 3B\{4c\}$
$[g\ s]$	$\{c^4 - 27Ac^2 + 123/4A^2\} + 3B\{c^2 - 9/2A\} + 3/4B^2$
$[h\ s]$	$\{-4Ac^3 + 30A^2c\} + 3B\{-2Ac\}$
$[i\ s]$	$\{6A^2c^2 - 11A^3\} + 3B\{A^2\}$
$[j\ s]$	$-4A^3c$
$[k\ s]$	A^4

¹S. F. Boys, Proc. R. Soc. London, Ser. A **200**, 542 (1950).²H. Taketa, S. Huzinaga, and K. O. Ohata, J. Phys. Soc. Jpn. **21**, 2313 (1966).³J. A. Pople and W. J. Hehre, J. Comput. Phys. **27**, 161 (1978).⁴L. E. McMurchie and E. R. Davidson, J. Comput. Phys. **26**, 218 (1978).⁵M. Dupuis, J. Rys, and H. F. King, J. Chem. Phys. **65**, 111 (1976).⁶J. Rys, M. Dupuis, and H. F. King, J. Comput. Chem. **4**, 154 (1983).⁷S. Obara and A. Saika, J. Chem. Phys. **26**, 218 (1978).⁸M. Head-Gordon and J. A. Pople, J. Chem. Phys. **89**, 5777 (1988).⁹P. M. W. Gill, M. Head-Gordon, and J. A. Pople, Int. J. Quantum Chem. **23**, 269 (1989).¹⁰R. Lindh, U. Ryu, and B. Liu, J. Chem. Phys. **95**, 5889 (1991).¹¹S. Ten-no, Chem. Phys. Lett. **211**, 3963 (1993).¹²K. Ishida, Int. J. Quantum Chem. **59**, 209 (1996).¹³H. F. King and A. Komornicki, in *Geometrical Derivatives of Energy Surfaces and Molecular Properties*, edited by P. Jorgensen and J. Simons (Reidel, New York, 1986), p. 207.¹⁴M. Dupuis, S. Chin, and A. Marquez, in *Relativistic and Electron Correlation Effects in Molecules And Clusters*, NATO ASI Series, edited by G. Malli (Plenum, New York, 1992).¹⁵H. F. King and M. Dupuis, J. Comput. Phys. **21**, 144 (1976).¹⁶B. G. Johnson, P. M. W. Gill, and J. A. Pople, Chem. Phys. Lett. **206**, 229 (1993).¹⁷A. Komornicki (private communication).