AIP The Journal of Chemical Physics

Evaluation of molecular integrals over Gaussian basis functions

Michel Dupuis, John Rys, and Harry F. King

Citation: J. Chem. Phys. **65**, 111 (1976); doi: 10.1063/1.432807
View online: http://dx.doi.org/10.1063/1.432807
View Table of Contents: http://jcp.aip.org/resource/1/JCPSA6/v65/i1
Published by the American Institute of Physics.

Evaluation of molecular integrals over Gaussian basis functions

Michel Dupuis, John Rys, and Harry F. King

Department of Chemistry, State University of New York at Buffalo, Buffalo, New York 14214

(Received 22 January 1976)

This paper is concerned with the efficient computation of the ubiquitous electron repulsion integral in molecular quantum mechanics. Differences and similarities in organization of existing Gaussian integral programs are discussed, and a new strategy is developed. An analysis based on the theory of orthogonal polynomials yields a general formula for basis functions of arbitrarily high angular momentum. $(\eta_i\eta_j \| \eta_k\eta_l) = \Sigma_{\alpha=1,n} I_x(u_\alpha)I_y(u_\alpha)I_z^*(u_\alpha)$. By computing a large block of integrals concurrently, the same I factors may be used for many different integrals. This method is computationally simple and numerically well behaved. It has been incorporated into a new molecular SCF program HONDO. Preliminary tests indicate that it is competitive with existing methods especially for highly angularly dependent functions.

I. INTRODUCTION

Recent years have seen an increasing use of d orbitals and higher angular momentum functions for *ab initio* quantum chemical calculations, and it is almost certain that in future calculations these functions will be used even more. To date there have been relatively few *ab initio* studies of transition metal compounds but this is bound to change, and for these systems d and even f orbitals will play an important role. Many calculations in the current literature emphasize the need to include d-type polarization functions in basis sets for atoms in the first or second rows of the periodic table. For example, a reliable determination of the rotational barrier for hydrogen peroxide requires a highly flexible basis set,[1] and it is found that d-type functions on the nitrogen are essential for the description of the inversion barrier for ammonia.[2] In a study of pentacoordinated phosphorus, Keil and Kutzelnigg conclude that the axial bonds in PH_3F_2 are best described as being three-center, four-electron bonds[3] rather than sp^3d hybrids, but they note substantial stabilization of these bonds by the contribution of d orbitals.[4]

The necessary angular dependence of a basis function can be achieved explicitly through the use of spherical harmonics, or equivalently through use of integer powers of the Cartesian coordinates. When the latter are combined with a Gaussian radial factor, as suggested by S. F. Boys,[5] a primitive basis function takes the form

$$\eta = x^{n_x} y^{n_y} z^{n_z} \exp(-ar^2). \tag{1}$$

The sum of powers λ,

$$\lambda = n_x + n_y + n_z , \tag{2}$$

is closely related to the total angular momentum quantum number. The above η is the type of primitive Gaussian with which we are concerned in this paper. The most popular alternative, the Gaussian lobe function, employs only s-type factors, $\lambda = 0$, and achieves its angular dependence through the combination of two or more such functions located on different centers.[6] Of course there are also other basis functions, notably Slater functions, but the attractive feature of Gaussians resides in the fact that there exist practical, closed form solutions to the Coulomb repulsion integrals.[5,7] One is faced, however, with two computational problems as λ takes on higher integer values. The first is the very large number of two-electron integrals to be computed. For example, just one a parameter on each of four different centers gives rise to 10 000 distinct integrals if one includes all η with $\lambda \leq 2$. The second problem is that the analytical expression for an integral becomes increasingly complex for the larger λ values and hence integral evaluation on a computer becomes expensive in terms of both execution time and core storage requirements for the instruction code.

A number of computer programs are now in use. In addition to the GAUSSIAN 70 program which in its presently available version handles only s and p-type bases,[8,9] there are several which treat higher angular momentum functions. These include Whitten's and Ahlrichs' lobe programs[10] and the following which work with η given by Eq. (1): PHANTOM 75,[11] IBMOL,[12] and BIGGMOLI.[13] (PHANTOM 75 is the most recent version of POLYATOM modified specifically for the CDC 6000 series.) By combining two or more primitives on the same center into a suitably chosen, fixed, linear combination, i.e., a contracted Gaussian basis function

$$\chi = x^{n_x} y^{n_y} z^{n_z} \sum_k c_k \exp(-a_k r^2), \tag{3}$$

it is possible to reduce the number of integrals to be manipulated while only slightly reducing the flexibility of the basis set. Raffenetti has introduced a generalized contraction scheme in which a primitive Gaussian may contribute to more than just one basis function.[13] The other programs mentioned above employ what Raffenetti calls "segmented contraction" according to which each primitive is used only once. An important feature of the GAUSSIAN 70 program of Pople, *et al.*[8,9] is the so-called shell structure by which four contracted functions, s, p_x, p_y, and p_z are grouped together into a "shell" and treated simultaneously so as to make maximum use of intermediate computations which are common to as many as $4^4 = 256$ different integrals. Raffenetti, Ahlrichs and other make efficient use of charge density concepts.[10,13] In addition, Ahlrichs achieves a highly efficient evaluation of the special function F_0 which arises with great frequency in lobe calculations.

Following Pople, we define a shell of functions to be a collection of χ all on the same center and all made up from the same set of exponential parameters a_k. A shell structure is computationally advantageous and particularly simple if an integral over primitives can be written as a product of three factors corresponding to the three Cartesian coordinates. The overlap integral can be factored in this way but the Coulomb repulsion integral can not. We prove, however, that the two-electron, Coulomb repulsion integral can be expressed as a finite sum of such products of three factors. Each term corresponds to one root of a Rys polynomial[14] the degree of which depends upon the sum of the four λ values. In effect, the Coulomb repulsion integral is evaluated by an exact numerical quadrature formula.[14]

The method of Rys quadratures is applicable to a wide variety of molecular integrals over Gaussian basis functions including those for all the usual one-electron properties, as well as for the three- and four-electron integrals that arise in certain treatments of electron correlation. The method is simple, accurate and applies to any nonnegative integer values of n_x, n_y, and n_z in (1). This paper illustrates the mathematical principles by solving the two-electron, Coulomb repulsion integral. We report a new SCF program HONDO, which uses Rys quadratures, and compare it with GAUSSIAN 70 and PHANTOM 75. With respect to the evaluation of two-electron integrals for a typical basis of s-, p- and d-type contracted Gaussians the present version of HONDO is found to be as accurate as and about five times faster than PHANTOM 75.

II. PROGRAM ORGANIZATION

In this section, we discuss some of the options that we and others have weighed in programing molecular integral evaluation on a traditional computer. (In a future paper, we intend to re-examine some of these issues in the context of minicomputers and microprocessors.)

A. Lobe functions

In his classic 1950 paper[5] Boys pointed out that higher angular momentum functions can be generated by successively differentiating the 1s-type Gaussian function with respect to the coordinates of its center. He used this idea to obtain analytical expressions for integrals, but in modern times differentiation has been replaced by a finite difference approximation which leads to the Gaussian lobe method. We can not claim to be experienced with lobe calculations, but we would like to state why we looked for an alternative approach. In going to higher angular momentum functions the lobe method appears to present problems with respect to both numerical accuracy and computational efficiency. Note that there are choices to be made in the selection of the lobe centers. Letting them be too close together leads to loss of accuracy due to numerical round off error, and letting them be too far apart implicitly mixes higher angular momentum components into the basis functions, e.g., a p-type lobe function has some f-type components. In other words a p-shell is not rotationally invariant but depends to some extent upon the selection of lobe centers. The possibility exists that one might inadvertently pick the basis so as to bias the calculation of, say, a bond angle or a quadrupole coupling tensor. In practice this does not seem to have been a problem with p functions which only involve first differences, but it might become a problem in applications to, say, transition metal chemistry where high angular momentum functions play an important role. There is another aspect to the selection of lobe centers. One might ask where to place them so as to best represent all basis functions in a shell. For example, we could represent a p_x-type primitive by two lobes displaced slightly to opposite sides of the origin along the x axis, and similarly for p_y and p_z. Then the p-shell representation would be based on six points at the six corners of a small octahedron. Alternatively, one could use four points at the corners of a regular tetrahedron, all four contributing to each of p_x, p_y, and p_z and, if one chose, to an s function as well. Suppose one is to compute the 3^4 integrals corresponding to shells of three p functions on each of four centers. Using the octahedral representation each of these 81 integrals is the sum of $2^4=16$ terms since each p orbital contains two lobes. Altogether one has to calculate $81\times 16=1296$ $(ss\|ss)$ integrals. Using the tetrahedral representation each of the 81 integrals is the sum of $4^4=256$ terms since each p orbital is now a linear combination of four s functions, but these 256 $(ss\|ss)$ integrals are common to all $(\eta_i\,\eta_j\|\eta_k\,\eta_l)$ so one reduces the number of F_0 functions to be computed by a factor of about 5. Apparently only the octahedral representation is in use, but by taking advantage of an efficient n^5 transformation[15] one might make the tetrahedral method competitive. It is also possible to use an irregular tetrahedron which combines some of the features of each. In any case the number of terms required to represent any one η_i is roughly proportional to its λ_i and the number of terms contributing to one $(\eta_i\,\eta_j\|\eta_k\,\eta_l)$ is approximately proportional to the *product* of the λ's. For larger angular momentum functions we obviously prefer an expansion that increases only as the *sum* of the λ's.

B. Elimination of integrals

In the remainder of this paper we confine the discussion to basis functions given by Eq. (3), of which (1) is a special case. Several programs were mentioned above that work with these functions, namely: IBMOL, GAUSSIAN 70, PHANTOM 75, BIGGMOLI, and HONDO. These programs differ in how they treat molecular symmetry. Some, e.g., GAUSSIAN 70, ignore point group symmetry, reasoning presumably, that many interesting chemical systems have no molecular symmetry at all. The option of using point group symmetry is built into HONDO, but that has little to do with the subject of this paper so we will discuss it elsewhere.[16] All of the above mentioned programs contain some provision for detecting and eliminating negligibly small integrals.[10] The tests and cut off criteria used in HONDO are mentioned in Sec. IV. More powerful methods are discussed by Whitten and Clementi.[17]

C. Shell structure

During integral evaluation all integrals for a set of four shells are computed concurrently and held in core. Raffenetti refers to this as a "block" of integrals.[13] Only after all members of the block are computed are they stored on a disk file for later use during SCF iterations. Table I provides a convenient nomenclature for the various shells. A D shell consists of the six functions d_{xx}, d_{yy}, d_{zz}, d_{xy}, d_{xz}, and d_{yz}. The combination $d_{xx}+d_{yy}+d_{zz}$ is, of course, not a true d function and is labeled $3s$ in the last column of Table I. Similarly, the F shell contains f_{xxx}, f_{xyy}, and f_{xzz} which when added together form a p_x function. It seems most appropriate to refer to this linear combination as a $4\,p_x$ function. The third column of Table I lists the number of contracted functions in a shell. For segmented S, P, D, ... shells this is given by $(\lambda+1)(\lambda+2)/2$, and for L, M, N, ... shells by $(\lambda_{\max}+1)(\lambda_{\max}+2)(\lambda_{\max}+3)/6$. A generalized contraction scheme increases these shell sizes. For example, suppose that a total of 24 primitives are used for an L shell, so that the radial factor in (3) is a linear combination of six Gaussians. Following Raffenetti's arguments,[13] one might construct three different s-type and two different p-type linear combinations giving an L shell with altogether nine members.

D. Integral evaluation strategies

Selection of the optimal method for computing integrals must take account of the composition of the basis set and how the integrals are to be used. It has been our experience with HONDO that integral evaluation rarely consumes as much computer time as do the subsequent SCF iterations, so in choosing the basis the first consideration is to achieve the necessary flexibility without making the χ basis needlessly large. These considerations usually dictate the construction of a small number of well-chosen, highly contracted, low angular momentum, inner shell functions, and a lightly contracted valence shell set with some high-λ functions. Keeping this in mind, let us consider the option of using either an M shell or an L and a D shell. The same number and type of functions are involved in either case, but for the M shell the set of a_k parameters is constrained to be the same for all functions. The advantages of doing this are: (i) time per integral computed is reduced, (ii) various derivatives can be evaluated exactly, (iii) the total number of shells and blocks is reduced. The disadvantages are: (iv) block sizes are increased, (v) flexibility of the basis is reduced, (vi) combining high-λ and low-λ functions into the same shell interferes with optimal contractions. To appreciate item (i) note that a set of four M shells give rise to a block of at least 10 000 integrals all of which are computed from a common collection of intermediate results. For certain applications item (ii) above may be important. It can be shown, for example, that an M shell contains an L subshell and first derivatives of these subshell functions with respect to displacement of the shell center. Conflicting with these favorable features is item (iv) which seriously affects core storage requirements and (v) and (vi) which can expand the size of the χ basis to the point where advantages due to (i) are nullified. The ideal program organization would be one that works efficiently under either option. Raffenetti's generalized contraction scheme, on the other hand, appears to be advantageous from all points of view except (iv).

Using formulas developed in Sec. III, each integral over primitives is computed, multiplied by a product of contraction coefficients, and added into the appropriate element of the block of integrals. Thus, this method is compatible with any shell structure discussed above provided that there is sufficient core memory to hold an entire block of integrals, and even this restriction could be relaxed for uncontracted basis functions. Two numbers, K and N_b, naturally arise in the discussion of computational strategy. We define K to be the product $\Pi\, k_{\max}$, where $k_{\max}$ is the number of terms in the summation in (3), and the product is over the four shells for a given block of integrals. Let N_b be the block size, i.e., the product of the four shell sizes. Using the proposed integration method, the time to compute a block of integrals is the sum of three terms: (a) overhead, independent of K and N_b, (b) set up time, proportional to K but independent of N_b, and (c) time for assembly of primitive integrals, proportional to K times N_b. With good programing tactics and reasonably large block sizes we expect the third term to dominate, although the discussion in Sec. IV indicates that this is not presently the case in HONDO, and there are even circumstances in which (a) is dominant. Pople and coworkers[8,9] compute a block of integrals in an entirely different way. By employing a sequence of different Cartesian coordinate frames, no operation is carried out as often as KN_b times. If all four shells are L shells, for example, then calculations inside the innermost loop over the k indices of Eq. (3) are carried out in a Cartesian frame chosen so that only 22 of the 256 integrals are nonzero. In the final step of the process the N_b integrals over χ functions are obtained by a coordinate transformation. Our results indicate that this transformation requires roughly the same computational labor as computing N_b $(\eta_i\,\eta_j\,||\,\eta_k\,\eta_l)$ integrals by the Rys quadrature method, but the labor involved in this transformation increases as N_b rather than KN_b. For heavily contracted functions, the coor-

TABLE I. Shell terminology.

Type	Basis functions	Number of members	Angular momentum components[a]
L	$\lambda\leq 1$	4	s,p
M	$\lambda\leq 2$	10	$s,3s,p,d$
N	$\lambda\leq 3$	20	$s,3s,p,4p,d,f$
S	$\lambda=0$	1	s
P	$\lambda=1$	3	p
D	$\lambda=2$	6	$3s,d$
F	$\lambda=3$	10	$4p,f$
G	$\lambda=4$	15	$5s,5d,g$

[a] When $\lambda>1$ the basis functions are not necessarily pure angular momentum functions. The table gives the angular momentum functions that could be obtained by a linear transformation on the basis.

dinate rotations pay for themselves because most of the work is done outside loops over k-type labels. This strategy is most advantageous for the STO-nG basis sets for which it was invented.[8]

III. RYS QUADRATURE METHOD

In this section we derive the quadrature formula, based on Rys polynomials, for a two-electron Coulomb repulsion integral over primitive Gaussian basis functions.

It has been known since the early work of Boys that the two-electron integral can be expressed in the form[5,7,18]

$$(\eta_i\,\eta_j|r_{12}^{-1}|\eta_k\,\eta_l) = \sum_{m=0,\,L} C_m F_m(X) \ , \tag{4}$$

where

$$F_m(X) = \int_0^1 t^{2m}\exp(-Xt^2)\,dt \tag{5}$$

and

$$L = \lambda_i + \lambda_j + \lambda_k + \lambda_l \ . \tag{6}$$

The value of X depends upon the exponential parameters a_i, a_j, a_k, a_l, and the positions of the centers of the four Gaussians, but is independent of the values of the 12 n_x, n_y, n_z indices. It follows immediately that (4) is a one-dimensional integral,

$$(\eta_i\,\eta_j||\eta_k\,\eta_l) = \int_0^1 P_L(t)\exp(-Xt^2)\,dt \ , \tag{7}$$

where $P_L(t)$ is an even polynomial of degree $2L$ with coefficients C_m. The Rys polynomial, $R_n(t, X)$, is defined to be an even polynomial of degree $2n$ in the variable t. For any real value of X there exists an infinite set of such polynomials orthonormal in the following sense:

$$\int_0^1 R_n(t,X)\,R_m(t,X)\exp(-Xt^2)\,dt = \delta_{nm} \ . \tag{8}$$

It follows from the theory of orthogonal polynomials that (7) can be evaluated exactly by an n-point numerical quadrature formula.[14,19]

$$(\eta_i\,\eta_j||\eta_k\,\eta_l) = \sum_{\alpha=1,n} P_L(t_\alpha)\,W_\alpha \ , \tag{9}$$

where n is any integer satisfying

$$n > L/2, \tag{10}$$

where t_α is a positive zero of the nth Rys Polynomial

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

and W_α is an appropriate quadrature weight factor which depends upon the value of X. Two of us have recently discussed practical methods for computing the values of $t_\alpha(X)$ and $W_\alpha(X)$.[14]

Taketa gives an explicit but complicated algebraic formula for the C_m coefficients.[7] In principle, one could compute the values of X, C_m, t_α, and W_α, then by substitution evaluate $P_L(t_\alpha)$, and use (9) to obtain the desired integral. In this way one avoids the computation of $F_m(X)$ but instead has to obtain the roots and weights of a Rys polynomial. This method is not recommended. It would be more efficient to use (4) directly. We do recommend the use of (9) but only because one can obtain numerical values of $P_L(t_\alpha)$ without computing the C_m coefficients. We brought (4) into the discussion because it is the traditional formula and because it defines $P_L(t)$. Our analysis is based upon the fact that $P_L(t)$ exists and that is an even polynomial of degree $2L$.

Express the Coulomb operator, r_{12}^{-1}, as a Gaussian transform. After a change in order of integration (valid) one obtains

$$(\eta_i\,\eta_j||\eta_k\,\eta_l) = 2\pi^{1/2}\int_0^\infty du(\eta_i\,\eta_j|\exp(-u^2r_{12}^2)|\eta_k\,\eta_l). \tag{12}$$

The integrand in (12) factors into a product of three two-dimensional integrals associated with the three axes of a Cartesion coordinate system

$$(\eta_i\,\eta_j|\exp(-u^2r_{12}^2)|\eta_k\,\eta_l) = I_x' I_y' I_z' \ . \tag{13}$$

Of the 12 n_x, n_y, n_z indices only the four n_x values enter into the I_x' factor:

$$I_x'(n_i, n_j, n_k, n_l, u) = \iint dx_1\,dx_2\,(x_1-x_i)^{n_i}(x_1-x_j)^{n_j} \times (x_2-x_k)^{n_k}(x_2-x_l)^{n_l}e^{-Q_x} \ , \tag{14}$$

where

$$Q_x = a_i(x_1-x_i)^2 + a_j(x_1-x_j)^2 + a_k(x_2-x_k)^2 + a_l(x_2-x_l)^2 + u^2(x_1-x_2)^2. \tag{15}$$

Here x_i is the x coordinate of the center of η_i and n_i is the corresponding n_x index. Let us define some new quantities:

$$x_A = (a_i x_i + a_j x_j)/(a_i + a_j) \ , \tag{16}$$

$$x_B = (a_k x_k + a_l x_l)/(a_k + a_l) \ , \tag{17}$$

$$A = a_i + a_j \ , \tag{18}$$

$$B = a_k + a_l \ , \tag{19}$$

$$\rho = AB/(A+B) \ , \tag{20}$$

$$D_x = \rho\,(x_A - x_B)^2 \ , \tag{21}$$

$$G_x = a_i\,a_j(a_i+a_j)^{-1}(x_i-x_j)^2 + a_k\,a_l\,(a_k+a_l)^{-1}(x_k-x_l)^2 \tag{22}$$

In terms of these, Q_x becomes

$$Q_x = G_x + A\,(x_1-x_A)^2 + B(x_2-x_B)^2 + u^2(x_1-x_2)^2. \tag{23}$$

Let us make a change of variable from u to t:

$$u^2 = \rho t^2/(1-t^2) \ , \tag{24}$$

$$t^2 = u^2/(\rho + u^2) \ , \tag{25}$$

$$dt = \rho(\rho + u^2)^{-3/2}\,du \ . \tag{26}$$

Note that as u varies from zero to infinity, t varies from zero to unity. Finally, we define a modified form of the two-dimensional integral

$$I_x = e^{D_x t^2}(1-t^2)^{-1/2}I_x' \ . \tag{27}$$

Similarly, I_y is related to I_y' and I_z to I_z'. By substituting (13) and (24)–(27) into (12) one recovers (7) where

$$P_L(t) = 2(\rho/\pi)^{1/2} I_x I_y I_z \ , \tag{28}$$

and using the y and z analogues of (21) one obtains

$$X = D_x + D_y + D_z \ . \tag{29}$$

Although it is far from being obvious, I_x, I_y, and I_z are even polynomials[20] in the variable t, so that $P_L(t)$ is a polynomial as is required for the validity of (9). To make this result plausible, consider the special case $n_i = n_j = n_k = n_l = 0$. Substitute (23) into (14) and express the quadratic form in the exponent as

$$Q_x = (B + u^2)(x_2 - \bar{x})^2 + Q'(x_1) \ , \tag{30}$$

where $Q'(x_1)$ is a quadratic function of x_1, independent of x_2. For the special case under consideration, the integral over x_2 is independent of the value of $\bar{x}$, and the second integration over x_1 yields

$$I_x'(0,\ 0,\ 0,\ 0,\ u) = e^{-G_x}\pi(A+B)^{-1/2}(\rho + u^2)^{-1/2} e^{-D_x t^2} \ , \tag{31}$$

where we have made use of (25) in the exponential factor. Substituting (31) and (25) into (27) reveals that I_x' is a polynomial of degree zero, i.e., is in this special case independent of t:

$$I_x(0,\ 0,\ 0,\ 0,\ u) = \pi e^{-G_x}(AB)^{-1/2} \ . \tag{32}$$

One could work out algebraic formulas for the I_x polynomials, but we prefer a numerical approach. Special methods for the computation of these two-dimensional integrals will be discussed elsewhere.[20] To obtain the working formula for the two-electron Coulomb repulsion integral simply substitute (28) into (9) to obtain

$$(\eta_i\,\eta_j\,||\,\eta_k\,\eta_l) = \sum_{\alpha=1,n} I_x(u_\alpha)\, I_y(u_\alpha)\, I_z^*(u_\alpha) \ , \tag{33}$$

where

$$I_z^*(u_\alpha) = 2(\rho/\pi)^{1/2} I_z(u_\alpha)\, W_\alpha \ . \tag{34}$$

To evaluate (33) one first computes ρ and X using (16)–(20) and (29). The value of X determines t_α, u_α, and W_α for $\alpha = 1, \ldots, n$, where n is chosen to be the smallest integer which satisfies (2), (6), and (10) for all $(\eta_i\,\eta_j\,||\,\eta_k\,\eta_l)$ in the block of integrals. The weight factor $W_\alpha(X)$ is evaluated using a fitting function based on Chebyshev polynomial approximation.[14] We do not explicitly require the value of t_α but compute directly, also by Chebyshev approximation, the dimensionless quantity $u_\alpha^2/\rho = t_\alpha^2/(1 - t_\alpha^2)$. (This is called u_α in Ref. 14). All of this replaces the computation of $F_m(X)$ in traditional Gaussian integral programs. Note that these quantities do not have to be computed for each $(\eta_i\,\eta_j\,||\,\eta_k\,\eta_l)$ because they are common to all such integrals in the block. Lists of the two-dimensional integrals I_x, I_y, I_z^* are then prepared[20] with the weight factor W_α combined with the I_z factor according to (34). Each $(\eta_i\,\eta_j\,||\,\eta_k\,\eta_l)$ is computed by picking I_x, I_y, I_z^* factors out of the lists and combining them according to (33). As each $(\eta_i\,\eta_j\,||\,\eta_k\,\eta_l)$ is generated it is multiplied by a product of contraction coefficients and added into the corresponding $(\chi\chi\,||\,\chi\chi)$ integral stored in core. Note that $(\eta_i\,\eta_j\,||\,\eta_k\,\eta_l)$ depends upon 12 indices, three for each η. The appropriate I_x factor depends only upon the n_x index of each η, thus, the same I factor may contribute to many different two-electron integrals. Note that the three I lists are unaffected if D shells are replaced by M shells, or F by N, and independent of whether the shells are segmented or not.

One of the appealing aspects of (33) is that each term in the sum is simply a product of three numbers which happen to be the values of certain polynomials. In a traditional algebraic analysis the three polynomials would have to be combined into a single polynomial the coefficients of which would eventually become the C_m in (4). In effect, we replace polynomial multiplication by numerical multiplication, thereby avoiding the proliferation of binomial coefficients which are so evident in Taketa's formulas.[7] At the same time, we manage to preserve intermediate quantities which are common to different integrals, simplify the computation, and eliminate the round off error commonly associated with binomial expansion.

IV. RESULTS

A copy of PHANTOM 75 was obtained from QCPE during the Summer of 1975.[11] A version of GAUSSIAN 70, believed to be essentially the same as the current QCPE program, was already available on our computer system at that time.[9] Ten different SCF wavefunctions for the hydrogen peroxide molecule were computed using both of these programs as well as using the newly written HONDO. This version of the program accepts segmented S, P, D, and L shells. It can only inefficiently simulate a generalized contraction scheme by including the same primitive basis function in more than one shell. It can reasonably be assumed that comparisons of these programs are machine independent since all calculations were carried out on the same CDC 6400 computer under the KRONOS 2.0 system, using the FTN compiler, OPT = 2. Several hundred integrals of various kinds were printed out after the run and compared. Values obtained with HONDO and PHANTOM coincided to at least 12 significant decimal figures. GAUSSIAN 70 values were generally good to eight figures in keeping with its less accurate $F_m(X)$ subroutine. In HONDO a block of $(\eta\eta\,||\,\eta\eta)$ integrals was skipped only if the product of e^{-G} factors, see Eq. (32), was less than 10^{-20}. A computed $(\chi\chi\,||\,\chi\chi)$ integral was omitted from the integral file if its absolute value did not exceed 10^{-9}, thus, HONDO was forced to compute at least as many integrals as did the others. The basis functions were ordered so as to achieve maximum efficiency in PHANTOM.[21]

Table II gives execution times for evaluation of the Coulomb repulsion integrals in each case. Set 3 is an STO-3G minimal basis,[8] i.e., a ($6s$, $3p$; $3s$) basis contracted to [$2s$, $1p$; $1s$]. (In this notation the semicolon separates atoms.) Set 5 is the same ($6s$, $3p$; $3s$) basis uncontracted. Set 8 is an STO-3G set with an additional D shell on oxygen. A number of characteristics are rerevealed by inspection of Table II. Times for sets 1–3 reinforce the popular opinion held by quantum chemists that GAUSSIAN 70 is just about impossible to beat on its home turf. It is even faster than HONDO for sets 4 and 5, but this is not significant, and results partially from its rapid, and less accurate $F_m(X)$ routines as opposed to the Rys roots and weights routines. HONDO times

TABLE II. Two-electron integral evaluation time in seconds for the hydrogen peroxide molecule.[a]

Basis set	N[b]	Description	GAUSSIAN	HONDO	PHANTOM	GAUSSIAN/HONDO	PHANTOM/HONDO
1	12	$1G$	1.4	1.4	3.5	1.0	2.5
2	24	$2G$	3.3	8.9	30.6	0.34	3.4
3	36	$3G$	8.4	39.9	132.7	0.21	3.3
4	24	$2G$, unc	18.2	20.2	51.5	0.90	2.5
5	36	$3G$, unc	79.1	95.0	244.	0.83	2.6
6	24	$1G+D$	···	17.7	95.3	···	5.4
7	36	$2G+D$	···	55.7	306.3	···	5.5
8	48	$3G+D$	···	152.2	775.6	···	5.1
9	36	$2G+D$, unc	···	89.3	420.2	···	4.7
10	48	$3G+D$, unc	···	279.7	1203.	···	4.3

[a] Point group symmetry is not used.

[b] N equals the total number of primitive basis functions.

increase roughly as N^4 for sets 1–3, whereas GAUSSIAN 70 times vary roughly as $N^{2.3}$ in keeping with remarks in Sec. II. D. One might conclude that special subroutines should be employed in HONDO for highly contracted low angular momentum functions, but comparison of sets 3 and 8 shows that putting just one D shell on oxygen results in 74% of the computation time being spent on integrals involving at least one d orbital, (83% in the case of PHANTOM), so the additional labor would not be greatly rewarded. The present version of HONDO obviously spends a lot of time doing something other than solving Eq. (33). This is evident from a comparison of sets 3 and 5, or 2 and 4 which involve exactly the same collection of $(\eta\eta\|\eta\eta)$ integrals but different $(\chi\chi\|\chi\chi)$. Perhaps inefficient file manipulation is responsible. In any case the problem has little to do with the use of the Rys quadrature method. In spite of the problem just mentioned, HONDO is still roughly five times faster than PHANTOM if the basis contains even a modest number of d functions. Incidently, it is interesting to compare sets 5 and 9 which contain the same number of η and of χ. HONDO runs faster for set 9 which contains d functions but the opposite is true for PHANTOM.

For a final comparison we used a $(9s, 5p, 1d; 4s, 1p)$ basis contracted to $[4s, 3p, 1d; 4s, 1p]$ similar to that used by Raffenetti.[13] Although he also used a CDC 6400 the comparison is not perfect. Yet it is interesting to note that BIGGMOLI required 677 sec exclusive of labels time, where HONDO ran in 350 sec. including computation of labels.

V. SUMMARY

The method of Rys quadratures appears to offer remarkable advantages for the evaluation of molecular integrals over Gaussian basis functions with high angular momenta. The method is simple, accurate and efficient. Although the present HONDO program does not have the benefit of the many man years of refinement that have gone into programs based on traditional formulas, it already appears to be highly competitive for calculations which employ even a modest number of d functions in the basis. GAUSSIAN 70 is now, and probably will remain, the method of choice for a highly contracted basis of s- and p-type functions. Raffenetti's generalized contraction scheme appears to be very attractive and will be incorporated into a future version of HONDO.

ACKNOWLEDGMENT

This work was supported in part by an award from the University at Buffalo Foundation, and by a Graduate Fellowship to one of us (M. D.). Computer time was made available by the SUNY at Buffalo computer center.

[1] A. Veillard, Chem. Phys. Lett. **4**, 51 (1969).

[2] A. Rauk, L. C. Allen, and E. Clementi, J. Chem. Phys. **52**, 4133 (1970).

[3] J. I. Musher, in *Sulfur Research Trends*, edited by R. F. Gould (American Chemical Society, Washington D. C. 1972); Adv. Chem. Ser. **110**, 44 (1972).

[4] F. Keil and W. Kutzelnigg, J. Am. Chem. Soc. **97**, 3623 (1975).

[5] S. F. Boys, Proc. R. Soc. (London) A **200**, 542 (1950).

[6] J. L. Whitten, J. Chem. Phys. **44**, 359 (1966).

[7] H. Taketa, S. Huzinaga, and K. O. Ohato, J. Phys. Soc. Japan **21**, 2313 (1966).

[8] M. D. Newton, W. A. Lathan, W. J. Hehre, and J. A. Pople, J. Chem. Phys. **51**, 3927 (1969).

[9] W. J. Hehre, W. A. Lathan, R. Ditchfield, M. D. Newton, and J. A. Pople, GAUSSIAN 70, Program 236, QCPE, Indiana Univ., Bloomington, IN.

[10] R. Alhrichs, Theor. Chim. Acta. **33**, 157 (1974).

[11] D. Goutier, R. Macauley and A. J. Duke, PHANTOM, Program 241, QCPE, Indiana Univ., Bloomington, IN.

[12] D. R. Davis and E. Clementi, IBMOL, Program 92, QCPE, Indiana Univ., Bloomington, IN.

[13] R. C. Raffenetti, J. Chem. Phys. **58**, 4452 (1973).

[14] H. F. King and M. Dupuis, J. Comput. Phys. **21**, 1 (1976).

[15] K. C. Tang and C. Edmiston, J. Chem. Phys. **52**, 997 (1970).

[16] M. Dupuis and H. F. King, Int. J. Quant. Chem. (to be published).

[17] J. L. Whitten, J. Chem. Phys. **58**, 4496 (1973); E. Clementi, Bull. Natl. Acad. Sci. **69**, 2942 (1972).

[18] I. Shavitt, in *Methods in Computational Physics*, edited by B. Alder, S. Fernbach and M. Rotenburg (Academic, New York, 1963), Vol. 2, p. 1.

[19] G. Szegö, *Orthogonal Polynomials* (American Mathematics Society, New York, 1959).

[20] J. Rys, M. Dupuis, and H. F. King J. Comput. Phys. (to be submitted).

[21] This is in keeping with program notes distributed with the PHANTOM program.