Molecular Integrals and Information Processing

MICHAEL P. BARNETT

Department of Chemistry, Princeton University, Princeton, NJ 08544-1009

Received 6 May 2003; accepted 6 May 2003

DOI 10.1002/qua.10614

ABSTRACT: We report new applications of computer algebra (CA) to the evaluation of molecular integrals over STOs. Closed formulas were tabulated for all the overlap integrals with principal quantum numbers 1 to 12. These provide unrestricted precision results that are compared with earlier numerical work of other authors. All the auxiliary functions needed to compute two- and three-center integrals have been reduced to closed form. We describe novel features of the software infra-structure that have wider application in theoretical chemistry and computational science. © 2003 Wiley Periodicals, Inc. Int J Quantum Chem 95: 791–805, 2003

Key words: molecular integrals; Slater orbitals; computer algebra; functional programming; molecular computing

1. Introduction

We describe some new formulas for the symbolic and very accurate computation of molecular integrals over Slater-type orbitals (STOs), using the ϛ-function method that we introduced some time ago [1, 2]. Some of the mathematical infrastructure and all of the novel software methods can be used to facilitate coding and optimization in many other areas of computational science. Certain features of the coding are related to the characteristics of some speculative systems to process digital and analog information based on chemical principles.

Correspondence to: M. P. Barnett. e-mail: michaelb@princeton.edu

Although Gaussian orbitals have dominated quantum chemistry for several decades [3, 4, 5, 6], the physical "naturalness" of Slater orbitals, that includes the behavior when the electron is close to the nucleus and when it is far away, maintains interest in methods to overcome the difficulty of evaluating the multi-center integrals. STOs have been used directly in recent energy calculations for the H_2^+ ion [7, 8], the H_2, HF, CO_2, and HCN molecules [9], the LiH molecule [9, 10], the crystal-field properties of the Sm^{2+} ion [11], and the excitation energy of the Cd_2 dimer [12]. The many papers that use the STO-*n*G approach to ab initio calculations [5, 6] reflect the appeal of Slater orbitals. A recent *Chemical Abstracts* on-line search, by the author, with the key "STO-3G" produced nearly 4000 citations. Within the past few months, papers using STO-*n*G have discussed topics as diverse as nuclear

spin state mixing [13], ionization potentials [14], UV resonance Raman spectra [15], rotational barriers [16], and electrical conductivity [17].

The problem of converting products of exponentials based on several nuclei into a separable coordinate system dominates the evaluation of STO integrals. The approaches are categorized in [18], with citations to where they were first reported and to recent developments. Some of the methods that were abandoned because the analysis was overwhelming should be re-examined using CA.

CA is the key to our revival of work on the ς-function method during the past decade, after an interval of nearly 25 years. Problems of catastrophically slow convergence and loss of accuracy in particular circumstances were resolved [19, 20, 21]. General formulas were reported for two-center Coulomb and hybrid integrals, and closed formulas for all such Coulomb integrals containing *K* and *L* shell orbitals were tabulated [22]. Some numerical work on a few overlap integrals of high quantum number was critiqued [23]. The transformation of solid spherical harmonics has been mechanized [24]. Supporting software methodology was reported [25–27]. It uses MATHEMATICA© [28].

The large scale tabulation of formulas for overlap integrals and some further steps towards the systematic reduction of multi-center integrals are described in the sections that follow. Also, we discuss how some of this work may impact computer science and vice versa.

2. Reduction of Overlap Integrals

Assume right handed coordinates systems on atoms *A* and *B*, with *B* at (0, 0, ρ) in the O_aXYZ system and $OX_a \parallel OX_b$. Denote the overlap integral between real, normalized STO's with quantum numbers (n_a, l_a, m), (n_b, l_b, m) and screening constants (k_a, k_b) by $O_q^{(r)}(\kappa, \tau)$ where $\kappa = k_a/k_b$, $\tau = k_b\rho$ and q stands for the list n_a, l_a, n_b, l_b, m. The solid harmonic factor of the orbital on atom *B*, displayed here as formula (1),* is transfered to atom *A* using the classical formula (Eq. (25) in Sect. 89 of Ref. [35] with $-\rho$ in place of ρ). The residual $r_b^{n_b-l_b-1}e^{-k_br_b}$ is transferred to atom *A* by the ς-function expansion Eq. (2). Integration over (θ_a, ϕ) reduces the series to a finite sum because of the orthogonality of the Legendre functions. To streamline the subsequent calculations, we introduce the scaled integral $O_q^{(s)}$, by Eqs. (3) and (4). Hence the special case of Eq. (5) obtained by setting $\varpi = s$. This contains the *U* coefficients and $Z_{m,n,l}^{(s)}$ functions defined by Eqs. (6) and (7). Eq. (5) is very similar to the formula for overlaps containing complex STOs (Eq. (87) in [22]). The $Z_{m,n,l}^{(s)}(\kappa, \tau)$ function is $2\sqrt{\tau}$ times the function denoted by $Z_{m,n,l}(\kappa, \tau)$ in earlier papers. Refs. [21] and [22] contain schemes to compute the *U* coefficients and the $Z^{(s)}$ functions. Both have undergone considerable improvements, described below.

The *U*'s are the Gaunt coefficients of planetary theory. They are closely related to the Clebsch-Gordan, 3*j*, 6*j* and other coefficients of angular momentum theory. These are the subject of extensive literature that includes recurrence schemes and analyses of the numerical errors produced by iteration [29]. The *U*'s are rational. In consequence, using the rational arithmetic feature of a CA system makes the error analyses moot. In [22], we describe the calculation of the *U*'s from the built-in `ClebschGordan` function [30] of MATHEMATICA. We also coded a recurrence scheme and, as a check, we coded the direct integration of Eq. (6).

Surprisingly, this elementary approach has turned out to be the fastest in the working context that we use, as well as less demanding of prior knowledge. We expand the product of Legendre functions as an explicit polynomial, and add the weighted coefficients. The tactic can be extended to use the integral representations of the related coefficients in angular momentum theory.

$$r_b^{l_b} P_{l_b}^{m_b}(\cos\theta_b)(\sin \mid \cos)m\phi \tag{1}$$

$$e^{-r_b} r_b^{m-1} = \sum_{n=0}^{\infty} \frac{2n+1}{\sqrt{r_a\rho}} P_n(\cos\theta_a)\zeta_{m,n}(r_a, \rho) \tag{2}$$

$$O_q^{(r)}(\kappa, \tau) = s_q(\kappa, \tau) O_q^{(s)}(\kappa, \tau) \tag{3}$$

$$s_q(\kappa, \tau) = \frac{(-1)^m 2^{n_a+n_b} \kappa^{n_a+1/2}}{\tau} \times \sqrt{\frac{(l_b+m)!(l_a-m)!(2l_b+1)(2l_a+1)}{(2n_b)!(2n_a)!(l_a+m)!(l_b-m)!}} \tag{4}$$

*Displayed formulas and computer scripts are grouped as artwork that was reproduced photographically from output which the author generated and/or validated by computer. The long version of the paper [18] contains these in place.

$$O^{(\varpi)}_{n_a,l_a,n_b,l_b,m}(\kappa,\tau) = \sum_{\lambda=|m|}^{l_b}\left[\begin{pmatrix} l_b-m \\ \lambda-m \end{pmatrix}(-\tau)^{l_b-\lambda} \times \sum_{\nu=|\lambda-l_a|}^{\lambda+l_a} U\begin{pmatrix} m & -m & 0 \\ l_a & \lambda & \nu \end{pmatrix} \times Z^{(\varpi)}_{n_b-l_b,\nu,n_a+\lambda}(\kappa,\tau)\right] \quad (5)$$

$$U\begin{pmatrix} m & -m & 0 \\ l_1 & l_2 & n \end{pmatrix} = \frac{2n+1}{2} \times \int_{-1}^{1} P^m_{l_1}(x)P^m_{l_2}P_n(x)\,dx \quad (6)$$

$$Z^{(s)}_{m,n,l}(\kappa,\tau) = \frac{2}{\sqrt{\tau}}\int_0^\infty e^{-\kappa t}t^{l+\frac{1}{2}}\zeta_{m,n}(t,\tau)\,dt \quad (7)$$

3. Vectorized Recurrence

The Z functions have been computed by recurrence since they were introduced in [1]. Recently, we realized the benefits of the following "vectorizability." Consider a K-term recurrence in n for a sequence of functions $f_n(x)$, given by Eq. (8) and the starting formulas given by Eq. (9) for $n = 1$ to N. By induction, Eq. (9) also holds for $n > N$. Also, f can be replaced by $\beta^{(j)}$ in (8). This decouples the calculation of the β^j_n for separate values of j. In particular, when the $\alpha_{n,k}(x)$ and $\beta^{(j)}_n(x)$ are rational in x, the calculation of the $f_n(x)$ reduces to J sets of iterations using rationals, even if the $\chi^{(j)}$ are not rational.

The Z functions needed for the hybrid ($AAAB$) integrals in this formulation contain 4-element vectors. These vectors collapse to two elements for the overlap and Coulomb integrals [23]. The worst vectors for the $ABAC$ integrals contain 53 elements. The vectorization

1. Allows enormous increase in speed in symbolic computations run sequentially in MATHEMATICA—in fact it makes computations possible that rapidly grind to a halt, otherwise,
2. Puts the major part of the computation within the scope of systems that just handle rationals,
3. Allows completely decoupled parallel computation—both symbolic and numeric,
4. Avoids some major sources of digital erosion,
5. Often propagates to the quantities that are computed from the vectorized $f_n(x)$,
6. Permits the coding of the recurrence scheme as a single set of statements that computes either the $\beta^{(j)}_n(x)$ for a single value of j, or the list $\{\beta^{(1)}_n(x), \ldots, \beta^{(J)}_n(x)\}$, or the complete functions $f_n(x)$ under control of a software switch in a piece of polymorphic code.

This technique can expedite the calculation of integrals over Gaussian orbitals. Also, recent work [33] on the Feynman integrals of high energy particle physics leads to expressions that can be cast as dot products of vectors of "nested sums" and vectors of polylogarithms [34].

4. The Tabulation Formulas

The recurrence scheme for the Z functions described in [21, 22] is adapted to the vectorized approach by introducing the notation in Eqs. (10) and (11).

$$\alpha_{n,0}(x)f_n(x) + \ldots + \alpha_{n,K}(x)f_{n+K}(x) = 0 \quad (8)$$

$$f_n(x) = \beta^{(1)}_n(x)\chi^{(1)}(x) + \ldots + \beta^{(J)}_n(x)\chi^{(J)}(x) = |\beta^{(1)}_n(x), \ldots, \beta^{(J)}_n(x)|.|\chi^{(1)}(x), \ldots, \chi^{(J)}(x)| \quad (9)$$

$$Z^{(s)}_{h,i,j}(\kappa,\tau) = Z^{(v)}_{h,i,j}(\kappa,\tau).\left|e^{-\tau}, e^{-\kappa\tau}\right| \quad (10)$$

$$Z^{(v)}_{h,i,j}(\kappa,\tau) = \left|Z^{(0)}_{h,i,j}(\kappa,\tau), Z^{(2)}_{h,i,j}(\kappa,\tau)\right| \quad (11)$$

$$Z^{(\varpi)}_{1,i,j}(\kappa,\tau) = \frac{\tau}{2i+1}\left[Z^{(\varpi)}_{0,i-1,j+1}(\kappa,\tau) - Z^{(\varpi)}_{0,i+1,j+1}(\kappa,\tau)\right] \quad (12)$$

$$Z^{(\varpi)}_{h,i,j}(\kappa,\tau) = \tau^2 Z^{(\varpi)}_{h-2,i,j}(\kappa,\tau) + Z^{(\varpi)}_{h-2,i,j+2}(\kappa,\tau) - \frac{2\tau}{2i+1}\left[iZ^{(\varpi)}_{h-2,i-1,j+1}(\kappa,\tau) + (i+1)Z^{(\varpi)}_{h-2,i+1,j+1}(\kappa,\tau)\right] \quad (13)$$

$$O^{(s)}_q(\kappa,\tau) = O^{(v)}_q(\kappa,\tau).\left|e^{-\tau}, e^{-\kappa\tau}\right| \quad (14)$$

$$O^{(v)}_q(\kappa,\tau) = \left|O^{(0)}_q(\kappa,\tau), O^{(2)}_q(\kappa,\tau)\right| \quad (15)$$

$$Z^{(s)}_{0,i,j}(1,\tau) = e^{-\tau} Z^{(0)}_{0,i,j}(1,\tau) \tag{16}$$

The $Z^{(0)}_{0,i,j}$ and $Z^{(2)}_{0,i,j}$ functions are found from Bessel polynomials $b_n(x)$ and certain rational expressions that we denote by $a^{(0)}_{j,i}(\kappa)$ and $a^{(2)}_{j,i}(\kappa, \tau)$. The relevant formulas follow trivially from Eq. (79) of [22]. The precise details, and the computation of the rationals by the methods described in [18, 21] are not needed in the present discussion. The previously reported recurrence formulas for the Z functions [21] are rewritten in the polymorphic form of Eqs. (12) and (13). This scheme can be used with $\varpi = 0$, 2, v or s. Besides allowing the separate computation of the $Z^{(0)}_{h,i,j}$ and $Z^{(2)}_{h,i,j}$ functions in sequence or in parallel, the $Z^{(v)}_{h,i,j}$ are stored as "lists" that MATHEMATICA computes in much less time than the $Z^{(s)}_{h,i,j}$, which are stored as linear combinations of exponentials.

A simple combination of Eqs. (5), (10) and (11) extends the vector representation to the scaled overlap integrals defined by Eqs. (14) and (15). Then Eq. (5) computes the $O^{(s)}$, $O^{(v)}$, $O^{(0)}$ and $O^{(2)}$ when $\varpi = s$, v, 0, 2, respectively.

When $\kappa = 1$, Eq. (16) maintains the computational structure. The equation for $Z^{(0)}_{0,i,j}(1, \tau)$ follows trivially from Eq. (80) of [22]. The recurrence formulas (12) and (13) hold without change.

5. Telescoped Recurrence

The computation of the Z functions, with or without vectorization, involves a succession of subsidiary calculations that can be organized in many different ways. The relative efficiency of these alternatives differs enormously when computing different sets of integrals and on different computer systems. This kind of situation occurs frequently in computational science.

To analyze the recurrence scheme in a practical context, let S_o be the set of overlap integrals that are needed in a particular problem. Let S_z be the set of $Z^{(s)}$ functions in the equations for these integrals. Let S_r be the set of $Z^{(s)}$ functions that occur as intermediate results when recurring from $Z^{(s)}_{0,i,j}$ to the elements of S_z.

When S_o exhausts all combinations of principal quantum numbers (n_a, n_b) $\leq$ some $\bar{n}$, it can be shown that all the $Z^{(s)}$ functions with $m > 0$ in S_r belong to S_z, and the recurrence scheme is efficient. Chemical problems occur, however, that require just a few integrals with very high quantum numbers for particular values of the molecular parameters. S_z then is a very small subset of S_r, and mixed symbolic or numeric recurrence to the $Z^{(s)}$ functions for these integrals is very inefficient.

$$\alpha^{\nu,\lambda}_{2\varsigma,n,l} \tau^{\varsigma-\lambda} Z^{(s)}_{m-2\varsigma,n+\nu-\varsigma,l+\lambda}(\kappa,\tau) \tag{17}$$

$$\begin{aligned} &n_a = 1 \text{ to } \bar{n}, n_b = 1 \text{ to } n_a, \\ &l_a = 0 \text{ to } n_a - 1, l_b = 0 \text{ to } n_b - 1, \\ &m = 0 \text{ to } (l_a, l_b)_< \end{aligned} \tag{18}$$

$$\left\{ O^{(r)}_{1,0,1,0,0}(\kappa,\tau), O^{(r)}_{2,0,1,0,0}(\kappa,\tau), \ldots, O^{(r)}_{3,2,3,2,1}(\kappa,\tau), O^{(r)}_{3,2,3,2,2}(\kappa,\tau) \right\} \tag{19}$$

$$O^{(r)}_{n_a,l_a,n_b,l_b,m}(\kappa,\tau) = (-1)^{l_a+l_b} O^{(r)}_{n_b,l_b,n_a,l_a,m}\left(\frac{1}{\kappa}, \kappa\tau\right) \tag{20}$$

$$\left\{ O^{(r)}_{1,0,1,0,0}(\kappa,\tau) = s_{1,0,1,0,0}(\kappa,\tau) O^{(s)}_{1,0,1,0,0}(\kappa,\tau), \ldots, O^{(r)}_{3,2,3,2,2}(\kappa,\tau) = s_{3,2,3,2,2}(\kappa,\tau) O^{(s)}_{3,2,3,2,2}(\kappa,\tau) \right\} \tag{21}$$

$$\left\{ O^{(r)}_{1,0,1,0,0}(\kappa,\tau) = \frac{2\kappa^{\frac{3}{2}}}{\tau} O^{(v)}_{1,0,1,0,0}(\kappa,\tau).\left|e^{-\tau}, e^{-\kappa\tau}\right|, \ldots, O^{(r)}_{3,2,3,2,2}(\kappa,\tau) = \frac{4\kappa^{\frac{7}{2}}}{9\tau} O^{(v)}_{3,2,3,2,2}(\kappa,\tau).\left|e^{-\tau}, e^{-\kappa\tau}\right| \right\} \tag{22}$$

$$\left\{ O^{(v)}_{1,0,1,0,0}(\kappa,\tau), \ldots, O^{(v)}_{3,2,3,2,2}(\kappa,\tau) \right\} \tag{23}$$

$$\begin{aligned} &\left\{ O^{(r)}_{1,0,1,0,0}(\kappa,\tau) = \frac{8\kappa^{3/2}}{\tau\left(\kappa^2-1\right)^3} \right. \\ &\quad \times \left[\left\{ 4\kappa + \tau\left(\kappa^2-1\right) \right\} e^{-\kappa\tau} \right. \\ &\quad \left. \left. + \left\{ -4 + \tau\left(\kappa^2-1\right) \right\} \kappa e^{-\tau} \right], \\ &\cdots \\ &O^{(r)}_{3,2,3,2,2}(\kappa,\tau) = \frac{128\kappa^{7/2}}{\tau^5\left(\kappa^2-1\right)^7} \\ &\quad \times \left[\left\{ 2880\kappa + \ldots + \left(\kappa^2-1\right)^3 \tau^5 \right\} e^{-\kappa\tau} \right. \\ &\quad \left. \left. + \left\{ -2880 + \ldots + \left(\kappa^2-1\right)^3 \tau^5 \right\} \kappa e^{-\tau} \right] \right\} \end{aligned} \tag{24}$$

```
o[r][1,0,1,0,0,k_,t_] /; k=!=1 :>
 (8*k^(3/2)*((-4*k+k*(-1+k^2)*t)*exp[-t]
+(4*k+(-1+k^2)*t)*exp[-(k*t)]))/((-1+k^2)^3*t)
```

(25)

To deal with this circumstance, we applied Eq. (13), with $m - 2$ in place of m, to the right hand side of the displayed form of Eq. (13). This gave an equation for $Z^{(s)}_{m,n,l}$ in terms of $Z^{(s)}_{m-4,n',l'}$ functions. ς applications of this process gave a formula for $Z^{(s)}_{m,n,l}$ as a finite sum over (λ, ν) of terms of the form (17). We used CA to construct recurrence formulas for the α coefficients in the terms (17) and to tabulate closed expressions for these in terms of n for $0 \le \varsigma \le 20$ and the appropriate ranges of (λ, ν). These improve the computational efficiency for isolated values of the Z functions with high n_b by as much as $O(n_b^2)$. The tactic is loosely analogous to the construction of Lommel polynomials [31] and the use of Clenshaw's algorithm (see, for example, Sect. 5.4 of Ref. [32]). It can expedite the symbolic and numerical application of many recurrence schemes.

6. The Overlap Tabulation

Vectorized and telescoped recurrence were used to construct a large table of formulas for overlap integrals, and to compute partially symbolic results for isolated combinations of high quantum numbers. The coding uses a novel principle that allows run time choice of computational strategy. Consider the tabulation of all nonzero overlap integrals with quantum numbers given by the iteration scheme (18). If, for example, $\bar{n} = 3$, the tabulation produces formulas for the 27 integrals in the list (19). Integrals with $n_a < n_b$ and with $(n_a = n_b, l_a < l_b)$ are found by transposing the atoms using Eq. (20). The tabulation is begun by applying the operation `makeEquationList[(3)]` to the list (19). This constructs the list (21). In general, `makeEquationList` converts a list of unevaluated functions $\{f(x_1), f(x_2), \ldots\}$ to the list of equations $\{f(x_1) = g(x_1), f(x_2) = g(x_2), \ldots\}$ which instantiate the equation $f(x) = g(x)$ specified by the operator. (Here, $f(x)$ is read literally, and $g(x)$ stands for a form in x.)

Next, the successive application of the operators `useEquation[(4)]` and `useEquation[(14)]` replaces all the s_q and $O^{(s)}_q$ objects, using Eqs. (4) and (14). Hence the list of equations (22). Next, `makeNameList[`$O^{(\nu)}$`]` extracts the names of the $O^{(\nu)}_q$ objects in the list (22). This produces the list (23). The process is continued in a cycle of steps that individually: (1) convert a list of names to a list of equations, (2) extract a list of names from the right hand sides of a list of equations, and (3) perform back substitution. Besides the parameterized operators used above, the process uses

1. `passWithName`[*label*] that assigns the parameter *label* to the operand for later reference, and passes the operand on to the next operator in the composition.
2. `backSubstituteIn`[*label*] that takes a list of equations $\{f_1 = e_1, f_2 = e_2, \ldots\}$ passed by the previous operator, and replaces $f_1, f_2, \ldots$ by $e_1, e_2, \ldots$ in the list that *label* identifies.
3. `makeRecursively`[*equationReference*] that forms a sequence of equations descending to the basis values.
4. `reduceRecursively` that works backwards through the sequence constructed by a `makeRecursively` operation.

The entire tabulation is performed in this way by the composition of 40 operators shown with their respective intermediary results in the longer account on the web site [18]. Hence the list (24).

This formulation proceeds from mathematical equations to executable code very easily, but it recomputes the same auxiliary functions when it is used on separate occasions. So we tabulated formulas for several of these functions and saved them on disc as `RuleDelayed` statements. For example, the result for the $1s^4$ integral in (24) is saved as the script (25). The statements are brought into play by `useTable`[*tableName*]. This operates on a list of names, and looks for table entries that begin with patterns that match the search items. The matched names are replaced by the appropriate instantiations of the right hand sides. Names that are not matched are left unchanged.

For faster operation, we load the low order part of each table at the start of a calculation, converting the ":>" to ":=" (`SetDelayed`) statements. Only a few tables are loaded in their entirety—the breakpoint depends on the trade-off between the load time and storage needs on the one hand, and the subsequent look-up time for individual entries in the unloaded portions.

We use the :> connective because it meets some needs directly, and is easily converted to := or == or = connectives for particular working contexts.

7. Some Results

The procedure OVLP computes overlap integrals for a variety of situations using the methods of Sections 2–6. The default output consists of closed expressions in (κ, τ). If, however, the name κ is assigned to another symbol, or to a number, or to a "nonatomic" expression, for example $1 + \epsilon$, before the procedure is invoked, then the assigned value is used in place of κ. The name τ can be assigned, too, with corresponding effects.

A production run begins by loading files of the $Z^{(v)}$ functions and the U, a and α coefficients. The user sets a limit on the indexes of each of the files that is loaded. OVLP deals with the entire overlap integral, and each intermediary function by (1) branching if $\kappa = 1$ (if this is relevant), (2) letting the `SetDelayed` statements in high-speed storage evaluate each of the objects under current attention that matches the pattern on the left of a ":=" symbol, (3) performing a table look-up for any of these objects that are unresolved, and (4) using a formula at the next level of the computational tree to calculate any of these objects that are still unresolved.

The OVLP procedure was used, in its default mode, to construct a table of individual formulas for all the distinct nonzero overlap integrals covered by the iterator (18) when $\bar{n} = 12$. The table contains 1, 6, 21, 55, 120, 231, 406, 666, 1035, 1540, 2211, and 3081 integrals for $n = 1$ to 12, respectively. It can be accessed over the web, together with simple procedures that use it [18]. Now that the table exists, the OVLP procedure refers to it whenever numbers or algebraic expressions have been assigned to κ and/or τ and the quantum numbers are in range. When these numbers are outside the range of the table, the OVLP procedure switches automatically to the telescoped recurrence scheme.

The expressions in the list (24) typify the results for unequal screening parameters. Each integral has been reduced to the form (26), where c and d are integers. Each p_1 and p_2 is a polynomial of degree $3n_a$ or less in κ and degree $4n_a$ or less in τ. The coefficients are integers. The formula (27) for $\langle 4d_{xz}|3d_{xz}\rangle$ is typical of the results when $\kappa = 1$. The general form is $ae^{-\rho}/b\sqrt{c}$, where a is a polynomial in ρ with integer coefficients, and b and c are integers.

Spot values of overlap integrals published by several authors in recent years are compared with the corresponding values from our formulas in Tables I and II. The agreement throughout these tables confirms, to a large extent, the accuracy that the respective authors claim for their results. However, the accuracy falls short occasionally and unpredictably. Some other work that contains more serious errors is discussed separately—the worst discrepancy occurs between a reported value that rounds to -5×10^{-8} and the value from our formulas that rounds to -8×10^{-78} [23].

$$\frac{\kappa^{n_a+1/2}\,(p_1 e^{-\tau} + p_2 e^{-\kappa\tau})}{c\sqrt{d}(\kappa^2-1)^{n_a+n_b+1}\tau^{l_a+l_b+1}} \tag{26}$$

$$\frac{e^{-\rho}}{210\sqrt{14}}\,\big(\,735 + 735\rho + 255\rho^2 + 10\rho^3 - 20\rho^4 - 7\rho^5 - \rho^6\big) \tag{27}$$

$$0.43385\ 6800488341\ 39558\ 87760\ 75035 \tag{28}$$

The values of all of our formulas for $(n_a, n_b) \le 10$ at $\kappa = 2$, $\tau = 3$ were compared by López [38] with output of his numerical program which aims at 15-digit accuracy. This covered 2677 of our results, and gave 15-digit agreement in 2180 cases. The worst disagreement was in the 13th digit, and only occurred for three integrals. Exploring the loss of precision that would occur if the formulas were evaluated in a fixed-precision system is easy. Exceptional cases lose about $2n_a$ digits for $\kappa = 2$, $\tau = 3$. Our comparisons so far do not include methods that use Gaussian expansions of the STOs. This is planned as a separate study that will include both free-standing software for the integrals and full-scale Gaussian packages.

Numerical methods in the literature give very inaccurate results for κ close to 1. We write $\kappa = 1 + \epsilon$, factor out $\exp(-\tau)$ and expand $\exp(-\epsilon\tau)$ as a power series. Taking 11 terms in this series for the integral $\langle 1s|2p_z\rangle$ with ($k_a = 102/100$, $k_b = 101/100$, $\rho = 14/10$) that Jones considered [39] gives 30-digit accuracy—see the displayed result (28). Assigning τ to an expression that contains trigonometric functions of the bond angles leads to precise expressions in the molecular geometry.

The formulas for the overlap integrals, for $\kappa \neq 1$, $n = 1$ to 6 occupy 465K bytes, and for $n = 7$ to 12 occupy 729K, 1.6M, 3.2M, 6.1M and 18M bytes, respectively. Loading the overlap integral files for $n = 1$ to 5 in a typical MATHEMATICA session in our current working environment takes roughly 23 seconds, and for $n = 6$ to 8 take 48, 120, and 270 seconds, respectively. A very simple experiment

TABLE I

Comparison of results from our formulas (mpb) with results of Mekelleche and Baba-Rahman [71] (mba), Jones [72] (jca: by computer algebra, jdp: completely numeric), Maslov and Niukkanen [73] (man), Rico, López and Ramirez [43] (rlr), Bhattacharya and Dhabal [74] (bhd), and derived from Weniger and Steinborn [75] by Rico et al. [43] (wes). Notes: (1) We assume that the association of this result with $\rho = 2$ in Ref. [73] is a misprint. (2) Jones forces the sign to be positive—see foot of page 751 in Ref. [39]. (3) The labeling of this result as $\langle 4s|1p1\rangle$ in [71] is a misprint. (4) $k_a = 65197/10000$, $k_b = 20387/10000$, $\rho = 375803/100000$.

$< 1s\|1s > (22/10, 15/10, 2)$		
2.31371 10583 27531 09767 2	(-1)	mpb
2.31371 10583 2753	(-1)	mba
2.31371 10583 275	(-1)	man
2.31371 10583 2753	(-1)	rlr
2.31371 10583 2752	(-1)	wes
$< 1s\|1s > (55/10, 15/10, 2)$		
5.91259 38608 93374 11286 7	(-2)	mpb
5.91259 38608 9337	(-2)	mba
5.91259 38608 934	(-2)	man
5.91259 38608 9339	(-2)	rlr
5.91259 38608 9333	(-2)	wes
$< 1s\|1s > (77/10, 15/10, 2)$		
3.50548 00488 25174 54583 1	(-2)	mpb
3.50548 00488 2518	(-2)	mba
$< 1s\|1s > (99/10, 15/10, 2)$		
2.38363 49699 89141 11521 9	(-2)	mpb
2.38363 49699 8914	(-2)	mba
2.38363 49989 1	(-2)	man
2.38363 49989 13	(-2)	rlr
2.38363 49989 13	(-2)	wes
$< 1s\|1s > (10, 10, 14/10)$		
6.67994 73767 98662 00198 1	(-5)	mpb
6.67994 73767 9866	(-5)	mba
6.67994 73767 987	(-5)	man [1]
6.67994 73767 9866	(-5)	rlr
6.67994 73767 98662 0020	(-5)	bhd
$< 5s\|5s > (1/10, 1/10, 14/10)$		
9.99637 18941 03858 56139 3	(-1)	mpb
9.99637 18941 0387	(-1)	mba
9.99637 18941 03858	(-1)	jca
9.99637 18941 03862	(-1)	jdp
9.99637 18941 03858 5615	(-1)	bhd
$< 1s\|2p0 > (10, 2, 14/10)$		
-1.17413 78968 66282 84854 9	(-1)	mpb
-1.17413 78968 6628	(-1)	mba
1.17413 78968 66283	(-1)	jca [2]
1.17413 78968 66282	(-1)	jdp [2]
-1.17413 78968 663	(-1)	man
-1.17413 78968 6628	(-1)	rlr
-1.17413 78968 66282 8486	(-1)	bhd
$< 4s\|4p0 > (5/10, 4/10, 14/10)$		
-1.23035 08689 42323 08052 3	(-1)	mpb
-1.23035 21353 1177	(-1)	mba
-1.23035 08689 42323 0805	(-1)	bhd
$< 2p0\|2p0 > (2, 2, 14/10)$		
-1.00740 38214 74410 89530 3	(-1)	mpb
-1.00740 38214 7441	(-1)	mba
-1.00740 38214 743	(-1)	man
-1.00740 38214 740	(-1)	rlr
-1.00740 38214 74410 8953	(-1)	bhd
$< 2p0\|5d0 > (2, 3/10, 14/10)$		
-2.33230 08171 99429 89706 2	(-3)	mpb
-2.33230 08171 9943	(-3)	mba
-2.33230 08171 99429 8971	(-3)	bhd
$< 3d0\|5d0 > (15/10, 3/10, 14/10)$		
1.22836 35964 31262 49448 3	(-2)	mpb
1.22836 35964 3125	(-2)	mba
1.22836 35964 31262 4953	(-2)	bhd
$< 1s\|1s > (77/10, 15/10, 14/10)$		
8.41736 21969 73076 54318 9	(-2)	mpb
8.41736 21969 7208	(-2)	mba
$< 3d1\|3d1 > (65/10, 2, 14/10)$		
-9.59577 53440 88398 15431 1	(-2)	mpb
-9.59577 53440 8845	(-2)	mba
$< 4s\|4p0 > (15/10, 3/10, 14/10)$		
-4.76529 56826 56991 40022 2	(-2)	mpb
-4.76529 56826 5699	(-2)	mba [3]
$< 3d0\|3d0 > [4]$		
1.25476 38617 81621	(-2)	mpb
1.25476 38617 818	(-2)	mba
1.25476 38617 8162	(-2)	rlr
$< 3d1\|3d1 > [4]$		
-6.99117 44322 62868	(-3)	mpb
-6.99117 44322 628	(-3)	mba
-6.99117 44322 6285	(-3)	rlr
$< 3d2\|3d2 > [4]$		
1.10496 50932 97633	(-3)	mpb
1.10496 50932 974	(-3)	mba
1.10496 50932 9763	(-3)	rlr
$< 1s\|2p0 > (102/100, 101/100, 1)$		
-4.33856 80048 83413 96	(-1)	mpb
4.33856 80048 83414	(-1)	jca
4.33856 80048 83410	(-1)	jdp
$< 5d2\|5d2 > (15/10, 14/10, 1)$		
9.38978 63304 49060 78	(-1)	mpb
9.38978 63304 49060	(-1)	jca
9.38978 63304 49057	(-1)	jdp
$< 744\|744 > (4, 3, 1)$		
1.10110 47776 49702 17	(-1)	mpb
1.10110 47776 49702	(-1)	jca
1.10110 47776 49702	(-1)	jdp
$< 10s\|10s > (8, 25/10, 3)$		
2.19686 18105 63723 82	(-1)	mpb
2.19686 18105 63724	(-1)	jca
2.19686 18105 63940	(-1)	jdp
$< 1000\|1090 > (0, 25/10, 1)$		
1.82992 53777 82952 91	(-5)	mpb
1.82992 53777 82953	(-5)	jca
1.82992 53777 90326	(-5)	jdp

that computed the numerical values of about 9,000 integrals with n_a, $n_b = 1$ to 12 averaged about 0.018 seconds per integral. Work is in hand to achieve very much greater speeds by converting the formulas to FORTRAN. The production of optimized code by CA has been reported in connection with Gaussian integrals [44].

There are many ways to avoid the redundant computation of sub-expressions that are repeated in different integrals, and to streamline the polynomial evaluation, vector multiplication, file handling, and table lookup. Some of these are introduced in the demonstration file `DEMO_NOTES.dta` on the web site [18]. The computation of 100,000 integrals per second that was reported for a purely numerical process [45] provides a challenging benchmark. We need to explore the construction of Chebyshev and rational approximations from the closed formulas for the integrals, and the accuracy and efficiency of multivariate interpolation on a grid of rational arguments.

Other authors have used our tables already, in work on magnetic semiconductors using a Hubbard model and polaron theory [46], and on conductivity and electron transfer in electroluminescent polymers using Hückel theory [47].

The one-electron two-center resonance integrals and kinetic energy integrals can be derived from the main table of overlap integrals and an auxiliary table of overlap integrals containing the fictional orbitals with $l_b = n_b \geq 0$—see, for example, Eqs. (5) to (7) of [43]. The 0*s* orbital has been of longstanding interest as a basis function [40], and recent uses include work on Li_3 [41].

The operation of the OVLP procedure illustrates another kind of elementary parallelization. Using a network of multiple processor UNIX platforms that share disc storage, we started background jobs in close succession, to tabulate formulas for the *a* coefficients, the *Z* functions, and the overlap integrals. Further jobs to compute numerical values for individual integrals then took different routes, depend-

TABLE II

Comparison of results from our formulas with results of Magnasco, Rapallo and Casanova [76]. Each sequence of four lines consists of the name of the integral, the values of (k_a, k_b, ρ), the value from our formula multiplied by $(-1)^{l_b+m}$ to allow for inward pointing z axes, and the value in the reference just cited. Notes: (1) The transposition of digits in this entry in [76] is a misprint.

$< 1s\|1s >$	$< 2p0\|2p0 >$	$< 1s\|1s >$	$< 3d2\|3d2 >$
(13/10, 31/5, 1/5)	(3/5, 23/10, 5/2)	(13/10, 31/5, 87/10)	(63/10, 21/5, 87/10)
4.12584 97424 892 (-1)	1.76576 74187 482 (-2)	1.01269 23680 410 (-5)	2.55649 01005 738 (-14)
4.12584 97424 9 (-1)	1.76576 74187 5 (-2)	1.01269 23680 4 (-5)	2.55649 01005 7 (-14)
$< 1s\|2p0 >$	$< 2p0\|3d0 >$	$< 1s\|2p0 >$	$< 3d1\|2p1 >$
(13/10, 23/10, 1/5)	(3/5, 51/10, 5/2)	(13/10, 23/10, 87/10)	(6/5, 41/10, 87/10)
1.43974 18882 206 (-1)	1.58946 11613 894 (-3)	2.18047 10225 153 (-4)	4.11368 90226 628 (-4)
1.43974 18882 2 (-1)	1.58946 11613 9 (-3)	2.18047 10225 2 (-4)	4.11368 90226 6 (-4)
$< 1s\|3d0 >$	$< 2p1\|2p1 >$	$< 1s\|3d0 >$	$< 3d1\|3d1 >$
(13/10, 51/10, 1/5)	(21/5, 21/5, 5/2)	(13/10, 51/10, 87/10)	(6/5, 5/2, 87/10)
1.72737 42371 169 (-2)	3.65615 20624 127 (-3)	1.24334 78612 300 (-5)	3.57247 56661 740 (-3)
1.72737 42371 2 (-2)	3.65615 20624 1 (-3)	1.24334 78612 3 (-5)	3.57247 56661 7 (-3)
$< 2p1\|2p1 >$	$< 2p1\|3d1 >$	$< 2p1\|2p1 >$	$< 3d0\|1s >$
(12/5, 41/10, 1/5)	(21/5, 22/5, 5/2)	(12/5, 41/10, 87/10)	(23/10, 31/5, 87/10)
8.05762 60233 329 (-1)	1.29213 64533 706 (-2)	1.84773 74348 048 (-8)	8.50964 50965 6402 (-7)
8.05762 60233 3 (-1)	1.29213 64533 7 (-2)	1.84773 74348 0 (-8) [1]	8.50964 50965 67 (-7)
$< 2p1\|3d1 >$	$< 3d2\|3d2 >$	$< 2p1\|3d1 >$	$< 3d0\|2p0 >$
(12/5, 5/2, 1/5)	(63/10, 21/5, 5/2)	(12/5, 5/2, 87/10)	(23/10, 23/10, 87/10)
1.95021 57628 541 (-1)	1.52743 00235 442 (-3)	3.91718 84139 202 (-6)	1.05290 98672 538 (-4)
1.95021 57628 5 (-1)	1.52743 00235 4 (-3)	3.91718 84139 2 (-6)	1.05290 98672 5 (-4)
$< 3d2\|3d2 >$	$< 3d1\|2p1 >$	$< 2p0\|1s >$	$< 3d0\|3d0 >$
(63/10, 21/5, 1/5)	(6/5, 41/10, 5/2)	(3/5, 31/5, 87/10)	(23/10, 51/10, 87/10)
8.04662 84833 219 (-1)	1.39961 89316 812 (-1)	6.82744 98785 428 (-3)	3.23123 69551 376 (-6)
8.04662 84833 2 (-1)	1.39961 89316 8 (-1)	6.82744 98785 4 (-3)	3.23123 69550 8 (-6)
$< 3d2\|3d2 >$	$< 3d1\|3d1 >$	$< 2p0\|2p0 >$	$< 3d1\|2p1 >$
(29/5, 9/2, 1/5)	(6/5, 5/2, 5/2)	(3/5, 23/10, 87/10)	(2, 21/5, 87/10)
8.78358 70505 647 (-1)	1.87418 62254 391 (-1)	2.53762 17505 712 (-2)	3.14775 83049 634 (-6)
8.78358 70505 6 (-1)	1.87418 62254 4 (-1)	2.53762 17505 7 (-2)	3.14775 83049 6 (-6)
$< 2p1\|2p1 >$	$< 3d0\|1s >$	$< 2p0\|3d0 >$	$< 3d1\|3d1 >$
(12/5, 41/10, 5/2)	(23/10, 31/5, 5/2)	(3/5, 51/10, 87/10)	(2, 22/5, 87/10)
2.54205 62259 533 (-2)	8.74662 82104 029 (-2)	1.14181 91760 793 (-3)	6.31466 12612 457 (-6)
2.54205 62259 5 (-2)	8.74662 82104 0 (-2)	1.14181 91760 8 (-3)	6.31466 12612 5 (-6)
$< 2p1\|3d1 >$	$< 3d0\|2p0 >$	$< 2p1\|2p1 >$	$< 3d2\|3d2 >$
(12/5, 5/2, 5/2)	(23/10, 23/10, 5/2)	(21/5, 21/5, 87/10)	(29/5, 9/2, 87/10)
2.02429 73113 878 (-1)	2.25219 66653 811 (-1)	5.16900 98114 407 (-13)	7.93431 05494 730 (-15)
2.02429 73113 9 (-1)	2.25219 66653 8 (-1)	5.16900 98114 4 (-13)	7.93431 05494 8 (-15)
$< 2p0\|1s >$	$< 3d0\|3d0 >$	$< 2p1\|3d1 >$	
(3/5, 31/5, 5/2)	(23/10, 51/10, 5/2)	(21/5, 22/5, 87/10)	
7.81284 63131 971 (-2)	8.01680 73315 669 (-2)	3.30166 95806 842 (-12)	
7.81284 63132 0 (-2)	8.01680 73315 7 (-2)	3.30166 95806 8 (-12)	

ing on the extent of the tables that were available when they were run. This technique is suited to grid computing [48]. Calculations would have to be time and platform stamped to create audit trails in production.

Other recent large scale tabulations of formulas include hyperspherical harmonics [49] and a table of formulas for Feynman integrals that has already reached 700MB [50, 51].

8. Checking the Formulas

The need to check algebraic computations, repeating the calculation, if possible, by alternative mathematical methods has been stressed recently [42]. The following checks were run on the formulas for the overlap integrals.

(1) Each integral $O^{(s)}_{n_a,n_a-1,n_b,n_b-1,m}(\kappa, \tau)$ was recalculated symbolically using elliptic coordinates. Then each $O^{(s)}_{n_a,l_a,n_b,l_b,m}(\kappa, \tau)$ with $n_a > l_a + 1$ and/or $n_b > l_b + 1$ was recalculated by differentiating the integral with the appropriate lower value of n_a or n_b. The $O^{(s)}_q$ integrals were found by removing the normalizing factors from the corresponding $O^{(r)}_q$ integrals. The quantum number n_b was raised by switching the names A and B using Eq. (20), increasing the new n_a then switching back.

(2) Each formula for $\kappa \neq 1$ was taken to the limit $\kappa = 1$ using the built-in MATHEMATICA`Limit` function and compared with the result that had been computed separately.

(3) Each formula for $\kappa \neq 1$ was taken to the limit $\tau = 0$, correspondingly, and compared with the value for the resulting one-center integral given by the MATHEMATICA `Integrate` function after some simple manipulations.

The calculation using elliptic coordinates was coded in a very simple fashion. As a result, it is very slow. On the positive side, however, it is easy to

follow and it is free of any dependence on auxiliary functions that are used in the s-function calculation. The three-term recurrence formula between overlap integrals of fixed (n_a, l_a, n_b, l_b) and consecutive m that is discussed by Rico et al. [43] would provide a further test of our results. So would the recalculation of nonsymmetrical integrals with the nuclei transposed.

Some of the ways that we used to check could be used to compute the integrals in the first place. This would still leave the need for multiple routes to the integrals to reinforce their validity, and the need to simplify the results. Simplifying the formulas has been an important and time consuming part of the main calculation. We were able to bypass the simplification of the check results when showing their equivalence to the primary results. This saved considerable time.

In the infrastructure of this work, the formulas used in the tabulation and reported in all our other papers with mathematical content over the past 8 years are part of a growing body produced by control files of coded derivations. These files were developed interactively and then run in batch mode. The discussion of the pieces of MATHEMATICA code (32)–(39) illustrate the style. The control files generate complete documents as well as individual formulas [27]. Each theorem that is copied from a reference source and each new definition is typed just once. Extensive checks are applied to the formulas that are copied from monographs and to the results that are derived. New files refer to earlier files by simple naming and addressing conventions. The files serve an extra role as mathematical audit trails. Whilst these measures cannot exclude all error, they provide far greater confidence than we find possible otherwise, and make inconsistencies much easier to track down and correct. We hope that similar measures will be adopted in other fields, too, leading to a coalescence of rigorously dependable material that can be used in computations, much as networks of standards have emerged in the laboratory.

9. Optimization

The OVLP procedure uses the equations of Section 2 in the order they were presented. Performance can be improved substantially by (1) changing the order in which the equations are used and (2) coalescing certain combinations. The handling of the scale factor $s_{n_a,l_a,n_b,l_b,m}$ illustrates the trade-offs. The scheme discussed earlier incorporates the closed expressions from Eq. (4) at the start of the calculation and carries these through the later operations. The scale factor can be kept as a name until the final stage of the calculation, however, by moving the operation `useEquation`[(4)] to the end of the 40-item composition. If the names of the scale factors are shorter than the actual expressions, this change reduces intermediate "swelling." It amounts to reordering the sibling nodes in the tree that charts the computation.

A related change coalesces Eqs. (3), (4), and (14) into Eq. (29). Then `makeEquationList`[(29)] replace the steps that produced the list (22). This amounts to combining the node for a parent with the nodes of several consecutive children in the computation tree.

The enumeration of alternatives such as these and the assessment of their relative merits has long bedeviled large-scale scientific computing. Quite often, a possible improvement becomes evident only in the course of production work. Quite often, too, the production runs require an investment of programming effort that precludes significant change in the design of the program.

Converting a set of formulas for a hierarchy of functions to a computational tree just requires a simple resolution of cross references. Each function reference $f(x)$ on the right-hand side of an equation points to the equation $f(\ldots) = \ldots$ that gives the prescription to compute it. Cycles connote recurrence. Algorithms (1) to convert the tree representation to the composition of the elementary operators, and (2) to enumerate (i) the arrangements of a set of siblings, (ii) the coalescences of two or more adjacent siblings, and (iii) the coalescences of two or more adjacent members of a lineage are elementary data structure exercizes. Recursion complicates the issue, but the components of an automatic optimization procedure are in place.

10. Two-Electron Integrals

We plan to use the methods of Sections 3–6 to construct extensive tables of the two-center Coulomb and hybrid integrals, using the formulas in [22].

Three- and four-center integrals require the translation of solid spherical harmonics. This translation is needed in the reduction of Gaussian integrals, too. Steinborn, Ruedenberg and Filter discussed this problem exhaustively some 30 years ago

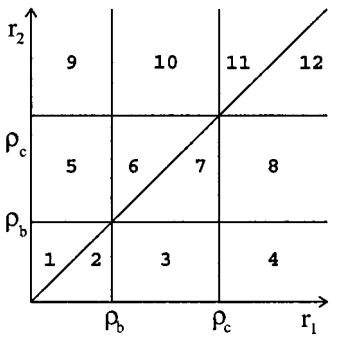

FIGURE 1. Piecewise integration regions for 3-center integral, $\rho_b < \rho_c$.

[36]. They used subtle reasoning and warn of the dangers of error due to misunderstanding and the inconsistencies of notation between different authors. CA provides an extremely simple solution. Given parallel coordinate systems (x, y, z) and (x_c, y_c, z_c) on atoms A and C. First, find the coefficients in the surface harmonic expansions of $t_{\lambda,\mu,\nu} = x_c^{\lambda} y_c^{\mu} z_c^{\nu}$ for $0 \le \nu_x + \nu_y + \nu_z \le$ some $\bar{l}$ by integrating $t_{\lambda,\mu,\nu}$ times each harmonic (30) for $0 \le l \le \lambda + \mu + \nu$, $0 \le m \le \mu + \nu$. Solve the resulting left triangular system for the harmonics in terms of the $t_{\lambda,\mu,\nu}$. Translate these expressions to the (x, y, z) system, expand, and replace the multinomials that result by harmonics in (r, θ, ϕ). To rotate harmonics, convert to Cartesian coordinates, rotate axes, and translate back. Full details are given in [24]. Tables of coefficients for $\lambda + \nu + \mu \le 20$ can be downloaded from [18].

$$O^{(r)}_{n_a,l_a,n_b,l_b,m}(\kappa,\tau) = \frac{(-1)^m 2^{n_a+n_b}\kappa^{n_a+1/2}}{\tau} \times\sqrt{\frac{(2l_b+1)(2l_a+1)(l_b+m)!(l_a-m)!}{(2n_b)!(2n_a)!(l_b-m)!(l_a+m)!}} \times O^{(v)}_{n_a,l_a,n_b,l_b,m}(\kappa,\tau).\,|e^{-\tau}, e^{-\kappa\tau}| \tag{29}$$

$$r_c^l P_l^m(\cos\theta_c)(\sin\,|\,\cos)m\phi_c \tag{30}$$

$$T_{n,m_1,m_2,l_1,\nu_1,l_2,\nu_2}(k_1,k_2,k_3,k_4,\rho_b,\rho_c) = \int_0^\infty\int_0^\infty e^{-k_1 r_1 - k_3 r_2} r_1^{l_1+1/2} \times\zeta_{m_1,n+\nu_1}(k_2 r_1, k_2\rho_b)\,\delta_n(r_1,r_2) \times\zeta_{m_2,n+\nu_2}(k_4 r_2, k_4\rho_c)\,dr_1\,dr_2 \tag{31}$$

```
eqn[zeta] =
 zeta[0,n][r,rho] ==
   if[ 0<=r<=rho,
    Ib[n+1/2,r] Kb[n+1/2,rho],
    Kb[n+1/2,r] Ib[n+1/2,rho]]        (32)

eqn[delta] =
 delta[n][r1,r2] ==
 if[0<=r1<=r2, r1^n/r2^(n+1), r2^n/r1^(n+1)]
                                      (33)
```

The reduction of the exchange and multi-center integrals is outlined in [2], where we show that the two- and three-center exchange integrals can be expressed as an infinite series of double integrals defined by Eq. (31) where ρ_b and ρ_c are the interatomic distances AB and AC. The T integrals are weighted by extended U coefficients which include a fourth Legendre function in the integrand of Eq. (6). The construction of recurrence formulas to compute the T integrals for $(m_1, m_2) > 0$ from the $T_{n,0,0,\ldots}$ is trivial. To construct the starting formulas we represent the $\varsigma_{0,n}$-functions by the statement (32), where the spherical Bessel functions $I_{n+1/2}$ and $K_{n+1/2}$ are denoted by `Ib` and `Kb`. The δ functions that come from the expansion of $1/r_{12}$ are represented correspondingly by the statement (33).

We assume, without loss of generality, that $\rho_b < \rho_c$. Figure 1 shows the quarter plane divided into the regions labeled 1 to 12 by the lines $r_1 = \rho_b$, $r_1 = \rho_c$, $r_2 = \rho_b$, $r_2 = \rho_c$, and $r_1 = r_2$. The analytical form of the integrand is uniform in each of these. The integration is split into 20 regions in the corresponding analysis for the four-center integrals, which suffers from a further change of form of the integrand at $r_2 = \rho_d$. Figure 2 enumerates these for the case $\rho_b < \rho_c < \rho_d$.

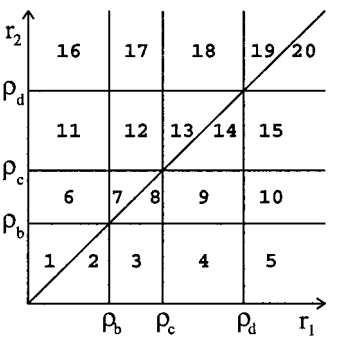

FIGURE 2. Piecewise integration regions for 4-center integral, $\rho_b < \rho_c < \rho_d$.

The $T_{n,\ldots,\nu}$ integrals were computed by double numerical quadrature during the 1960s, integrating over r_1 and then r_2. Some efforts were also made to compute the integral over r_1 from auxiliary functions with known closed form. This consists of exponentials and polynomials of r_2 in regions where $r_1 \le r_2$, but it contains *Ei* and log functions elsewhere.

For full analytic reduction, accordingly, we integrate over r_1 first in regions 2, 3, 4, 7, 8, 12 of Figure 1, and over r_2 first in the other regions. The construction of the piecewise integrations illustrates another kind of reasoning that is mechanized with ease. For the three-center case, begin with the ordering $\mathbb{O} = (0 \le \rho_b \le \rho_c < \infty)$. Each of the 12 regions is specified by the insertion of r_1 and r_2 into this sequence. For example, $\mathbb{O}_3 = 0 \le \rho_b \le r_1 \le \rho_c \le r_2 < \infty$ in region 3. A short MATHEMATICA statement constructs the list of these 12 regions. Two more very short statements construct the list of $\int\int_{\mathbb{O}_i}$ operators. This list is represented by the MATHEMATICA expression (34). Next, the sub-expression `[f1 f2 f3 f4]` is attached to each of these operators, to give (35). Then `f1` is replaced by the conditional expression for the δ function.

A very short procedure `resolve` is applied to each of the 12 pieces of the integral. It evaluates the test `0<=r1<=r2` by considering the relative positions of the items `(0,r1,r2)` in the `currentOrder` sub-expression, and returns r_1^n/r_2^{n+1} or r_2^n/r_2^{n+1} accordingly. Then `f2` is replaced by the conditional expression for the ς-function, and the `resolveScaled[k2]` function evaluates the test and returns $I_{n+1/2}(k_2 r_1) K_{n+1/2}(k_2 \rho_b)$ or $K_{n+1/2}(k_2 r_1) I_{n+1/2}(k_2 \rho_b)$ depending on the outcome. The factor `f3` is replaced by the ς-function containing $k_4 r_2$, and this is resolved. The factor `f4` is replaced by `exp[-k2 r1-k4 r2]` and the integrands are complete.

The `currentOrder` sub-expression is dropped from each of the 12 piece-expressions, and the list is changed to a `Plus`. This constitutes an acceptable piece-wise formulation of the entire integral. It is simplified slightly by coalescing pairs of pieces that have contiguous limits and contain the same integrand, e.g., pieces 5 and 6, leading to a 9-piece result. The entire process is performed by less than 50 short lines of simple code. It is checked, in part, by detaching the heads and showing that these coalesce to the quarter plane.

A preliminary exploration showed that the 9 pieces combine to give the dot product of a vector of 53 rational expressions and a vector of exponentials, exponential integrals and logarithms in the worst case [52]. The mathematical methods of [21] are being extended to construct a recurrence scheme for the rationals, using the coding methods that the derivation of one of the basic formulas in the next section illustrates.

The table of overlap integrals and the ability to go to high quantum numbers for numeric κ and/or τ invites further study of multipole expansions [53] and related approaches to the evaluation of multicenter integrals [54]. Our preliminary efforts to construct asymptotic formulas for the overlap integrals gave good results for $l = 0$ that deteriorated rapidly when l was increased.

```
{integral[{r2,0,rhoB}, {r1,r2,rhoB}][
  currentOrder[0,r1,r2,rhoB,rhoC,infinity],...
  integral[{r1,r2,rhoB}, {r2,0,rhoB}][
    currentOrder[0,rhoB,rhoC,r2,r1,infinity]}
```
(34)

```
{integral[{r2,0,rhoB}, {r1,r2,rhoB}][
  currentOrder[0,r1,r2,rhoB,rhoC,infinity][
   f1 f2 f3 f4], ... }
```
(35)

```
eqn[zexpan] = r[b]^(m-1)exp[-r[b]] ==
  sum[n,0,infinity][(2n+1)/sqrt[r[a] rho]*
   zeta[m,n,r[a], rho] P[n, cos[theta[a]]]]
```
(36)

```
eqn[cosRule] =
 r[b]^2==r[a]^2+rho^2-2 r[a] rho cos[theta[a]]
```
(37)

```
eqn[recurrenceLegendre] =
 x P[n, x] ==
   n P[n-1,x]/(2n+1)+(n+1) P[n+1,x])/(2n+1)
```
(38)

11. Deriving a Recurrence Formula

The recurrence formula that relates $\varsigma_{m,n}$ to $\varsigma_{m-2,n'}$, $n' = n$, $n \pm 1$ is derived from Eq. (2), the cosine rule and the recurrence formula for the Legendre functions. These are input as the MATHEMATICA statements (36–38). By convention, after an assignment `eqn[`*id*`]=`*lhs*`==`*rhs*, MATHSCAPE converts `brule[`*id*`]` to *lhs*`:>`*rhs* automatically, and it converts `grule[`*id*`]` to the rule that replaces any pattern of the **form** of *lhs* by the corresponding adaptation of *rhs*. These features are used in lines (4), (7) and (15) of the statement (39) that constructs the recurrence formula.

```
eqn[zeta[m]] =
 eqn[zexpan] //
  pipe[
   toBothSides[times[r[b]^2]],                (* 1*)
   toTheRhs[                                  (* 2*)
    moveCoefficientRight,                     (* 3*)
    brule[cosRule],                           (* 4*)
    toTheSummand[                             (* 5*)
     distribution[1,2,4,5][2][3],             (* 6*)
     grule[recurrenceLegendre],               (* 7*)
     toTermFreeOf[_^2][                       (* 8*)
      distribution[1,2,3,5,6,7][][4]]],       (* 9*)
    toTheSum[Distribute],                     (*10*)
    toSum[containing[P[n-1,_]]][
     reindex[n,n-1]],                         (*11*)
    toThe[sum[n,-1,_][_]][leftExpand],        (*12*)
    toSum[containing[P[n+1,_]]][
     reindex[n,n+1]],                         (*13*)
    toThe[sum[n,1,_][_]][leftExtend]],        (*14*)
    bruleFor[eqn[zexpan]/.m->m+2],            (*15*)
   zeroRight,                                 (*16*)
   toTheLhs[                                  (*17*)
    toEachTerm[moveCoefficientRight],         (*18*)
    joinTheSummands,                          (*19*)
    toTheSummand[factor]],                    (*20*)
   sum[n_,0,infinity][
    s_ P[n_,x_]]==0 -> s=0,                   (*21*)
   clearFractions,                            (*22*)
   m->m-2,                                    (*23*)
   solveLinear[zeta[m,n,r_a,rho]],            (*24*)
   toTheRhs[                                  (*25*)
    Distribute,                               (*26*)
    toSetOfTermsFreeOf[_^2][factor],          (*27*)
    to[Plus][containing[zeta],
             innermost][factorOut[-1]],       (*28*)
    ReleaseHold],                             (*29*)
   r[a] -> r]                                 (*30*)
```

(39)

The numbered lines act as follows.

1. Produces the representation of Eq. (40).
2. Focuses attention on the right hand side.
3. Moves the factor r_b^2 inside the sum.
4. Replaces r_b^2 using the cosine rule.
5. Focuses on the summand. MATHEMATICA arranges the factors in the order shown in (41).
6. Distributes factors 1, 2, 4, and 5 onto term 2 of factor 3 and onto the bracketed plus of the other terms. This converts the summand to (43).

$$e^{-r_b} r_b^{m+1} = r_b^2 \sum_{n=0}^{\infty} \frac{2n+1}{\sqrt{r_a\rho}} P_n(\cos\theta_a)\zeta_{m,n}(r_a,\rho) \quad (40)$$

$$(2n+1) \times P_n(\cos\theta_a) \times(\rho^2 - 2r_a\rho\cos\theta_a + r_a^2) \times\frac{1}{\sqrt{r_a\rho}} \times \zeta_{m,n}(r_a,\rho) \quad (41)$$

$$-\frac{2(2n+1)}{\sqrt{r_a\rho}} r_a\rho\cos\theta_a P_n(\cos\theta)\zeta_{m,n}(r_a,\rho) + \frac{2n+1}{\sqrt{r_a\rho}}(r_a^2+\rho^2)P_n(\cos\theta)\zeta_{m,n}(r_a,\rho) \quad (42)$$

$$-2\times(2n+1)\times\rho\times\left(\frac{n}{2n+1}P_{n-1}(\cos\theta_a) + \frac{n+1}{2n+1}P_{n+1}(\cos\theta_a)\right) \times r_a \times \frac{1}{\sqrt{r_a\rho}} \times \zeta_{m,n}(r_a,\rho) \quad (43)$$

$$\zeta_{m,n}(r,\rho) = (r^2+\rho^2)\zeta_{m-2,n}(r,\rho) - \frac{2r\rho}{2n+1}[n\zeta_{m-2,n-1}(r,\rho) + (n+1)\zeta_{m-2,n+1}(r,\rho)]. \quad (44)$$

7. Changes the term containing $\cos\theta_a P_l(\cos\theta_a)$ to the expression that MATHEMATICA puts in the order shown in (43). (`sqrt[ra rho]` sequesters the $\sqrt{r_a\rho}$ factor).
8. Focuses on this term.
9. Distributes factors 1, 2, 3, 5, 6, 7 onto the entirety of factor 4. This converts the summand to a "plus" of three terms that contain the Legendre functions of degrees $n-1$, n and $n+1$, respectively.
10. Splits the right hand side into three sums.
11. Converts $\sum_{n=0}^{\infty} f(n)$ to $\sum_{n=-1}^{\infty} f(n+1)$.
12. Converts $\sum_{n=-1}^{\infty} f(n)$ to $f(-1) + \sum_{n=0}^{\infty} f(n)$. In the present context, the term corresponding to $f(-1)$ is identically zero. Hence a series containing $P_n(\cos\theta_a)\mathfrak{s}_{m,n+1}(r_a,\rho)$.
13. Converts $\sum_{n=0}^{\infty} f(n)$ to $\sum_{n=1}^{\infty} f(n-1)$.
14. Converts $\sum_{n=1}^{\infty} f(n)$ to $-f(0) + \sum_{n=0}^{\infty} f(n)$. In the present context, the term corresponding to $f(0)$ is identically zero. Hence a series containing $P_n(\cos\theta_a)\mathfrak{s}_{m,n-1}(r_a,\rho)$.
15. Expands the left hand side.
16. Transposes *lhs* = *rhs* to *lhs* − *rhs* = 0.
17. Focuses on the new left hand side.
18. Changes the $-\Sigma(u)$ terms to $+\Sigma(-u)$.
19. Combines sums with unit external multiplier and identical ranges.

20. Factors the summand to $P_n(\cos\theta)$ times an expression containing four ς-functions.
21. Equates the general coefficient to zero.
22. Clears fractions in the resulting equation.
23. Changes $m + 2$ and m to m and $m - 2$.
24. Solves for $\varsigma_{m,n}$.
25. Focuses on the right hand side.

Lines 26–30 make the right hand side compact, to give Eq. (44).

12. Discussion

The derivation of Eq. (44) uses a programming style that we evolved about 10 years ago as a natural way to write symbolic calculations. The 40-element composition that leads to Eq. (24) uses the same principle of composing unary operators. This commonality transcends the contrast between operands that are about 300 and up to 3 million characters long in the two cases. The compositional style is a major feature of functional programming (FP). This is a well-established field of computer science. Readers who want to explore other possible applications of FP in writing numerical and non-numerical programs for chemical problems can refer to the classical paper [55], the recent instructional overview [56] and the conference proceedings [57].

Besides the practical aspect of FP in chemistry, we want to bring the idiom to the attention of a wide chemical audience for another reason that could have far greater long-term significance. The challenge of molecular computing has elicited a number of suggestions, that include photon induced digital switching and combinatorial exploitation of DNA behavior. An early involvement with questions of information storage and processing in biological systems [58] led us to speculate on ways in which analog information could be represented by variations in the relative local density of side groups in a copolymer that could serve as an associative memory unit [59]. This led to the specification of an extremely simple abstract computing device (the "primitive string transformer" or "PST") with powerful algorithmic capabilities [60, 61]. Its behavior is represented by a pattern replacement notation. Pattern replacements are explicit in much of our CA coding and we think that the coding can be put entirely in this form. Also, numerical operations can be expressed as character string manipulations.

Our resumption of work on molecular integrals interrupted a simulation of putative large scale recursively structured systems of PSTs, using an early, conventional supercomputer. The PST work did suggest the characteristics of molecular and biomolecular systems to perform several classes of information processing tasks of increasing complexity that will be reported separately.

From a more immediate perspective, the use of CA as a powerful and necessary tool to determine the accuracy of numerical algorithms and to guard against errors is reinforced by the work reported here. Unrestricted-precision arithmetic was of paramount importance in the construction and evaluation of the formulas, and we think that this resource will change scientific computing as dramatically as floating point arithmetic did 40 years ago. Symbolic and numerical computation are likely to become increasingly intertwined and we expect users to write in ways that mimic their pencil and paper mathematics to an increasing extent.

The mechanized evaluation of molecular integrals actually dates back to two of the earliest papers on computer algebra [62, 63]. Recent work on Gaussian and Slater integrals for energy and scattering calculations has used AXIOM [64], MAPLE [65–68], MATHEMATICA [7, 8, 69, 72] and REDUCE [10, 70]. We look forward to the mechanized pooling of the results of different software systems, and their incorporation in the further steps of quantum chemical studies that are supported by CA.

ACKNOWLEDGEMENTS

Leland Allen provided the environment to do the work reported here. His encouragement, support and advice is acknowledged with considerable gratitude. We are grateful also to Joseph Capitani for considerable ongoing advice and encouragement, I. I. Guseinov for the provision of data that helped in the early stages of debugging our procedures, R. López for running lengthy comparisons of our final results, R. Montagnani for the provision of extensive data, A. Brill, L. Kocbach, V. Magnasco, S. M. Mekelleche, J. Mohallem, W. J. Pietro, N. Sunel, H. Tai, and Y. Shen for answering our queries about their methods and results, J. A. Gracey and J. A. M. Vermasseren for information about Feynman integrals, G. Hutton for information about functional programming, and A. A. Kapadia and J. Whiting for specialized systems support.

References

1. Barnett, M. P.; Coulson, C. A. Phil Trans Roy Soc London A, 1951, 243, 221.
2. Barnett, M. P. in Alder, B.; Fernbach, S.; Rotenberg, M., Eds., Methods of Computational Physics, vol 2, 95, Academic Press, New York, 1963.
3. Boys, S. F. Proc Roy Soc London A 1950, 200, 542.
4. Shavitt, I. in Alder, B.; Fernbach, S.; Rotenberg, M., Eds., Methods of Computational Physics, vol 2, 1, Academic Press, New York, 1963.
5. Hehre, W. J.; Stewart, R. F.; Pople, J. A. J Chem Phys 1969, 51, 2657.
6. Pople, J. A. Rev Mod Phys 1999, 71, 1267.
7. Jones, H. W.; Etemadi, B. Phys Rev A 1993, 47, 3430.
8. Bouferguene, A.; Weatherford, C. A.; Jones, H. W. Phys Rev E 1999, 59, 2412.
9. Bouferguene, A.; Fares, M.; Hoggan, P. E. Int J Quant Chem 1996, 57, 801.
10. Righi, A. F. M.; Kuhnen, C. A. Int J Mod Phys C 1997, 8, 1159.
11. Shen, Y.; Bray, K. L. Phys Rev B 1998, 58, 5305.
12. de la Vega, J. M. G.; Miguel, B. Theor Chem Acta 2000, 104, 189.
13. Airne, A. R.; Brill, A. S. Phys Rev A 2001, 63, 052511.
14. Shinoda, H.; Mori, Y.; Kitagawa, T. J Mol Struct (Theochem) 2001, 546, 195.
15. Jarzecki, A. A.; Spiro, T. G. J Raman Spectrosc 2001, 32, 599.
16. Sharma, A.; Gupta, V. P.; Virdi, A.; Thakur, S. Indian J Pure Appl Phys 2001, 39, 525.
17. Wang, R. S.; Wang, C. G.; Hu, L. H.; Su, Z. M.; Zhu, D. X.; Zhang, X. Y. Synth Metals 2001, 119, 223.
18. Barnett, M. P. 2002, http://www.princeton.edu/~allen-grp/ms/overlap
19. Barnett, M. P. Chem Phys Lett 1990, 166, 65.
20. Barnett, M. P. Comp Math Applic 1991, 21, 79.
21. Barnett, M. P. Int J Quant Chem 2000, 76, 464.
22. Barnett, M. P. J Chem Phys 2000, 113, 9419.
23. Barnett, M. P. Theor Chem Acta 2002, 107, 241.
24. Barnett, M. P. 2003, J Chem Inf Comp Sci in course of publication.
25. Barnett, M. P. Comp Math Appl 1993, 26, 35.
26. Barnett, M. P.; Perry, K. R. Comp Math Appl 1994, 28, 17.
27. Barnett, M. P. TUGboat 1998, 19, 147.
28. Wolfram, S. The Mathematica book, 4th edn. Cambridge University Press, New York, 1999.
29. Sebilleau, D. J Phys A 1998, 31, 7157.
30. Feagin, J. M.; Kocbach, L.; Liska, R.; Nunez, A. 1994, www.fi.uib.no/AMOS/mathME
31. Watson, G. N. Theory of Bessel functions, Cambridge University Press, New York, 1944.
32. Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; Vetterling, W. T. Numerical recipes in C, Cambridge University Press, New York, 1988.
33. Moch, S.; Ewer, P.; Weinzierl, S. J Math Phys 2002, 43, 3363.
34. Gracey, J. A. 2002, private communication.
35. Hobson, E. W. Theory of spherical and ellipsoidal harmonics. Cambridge University Press, New York, 1931.
36. Steinborn, E. O.; Ruedenberg, K. Adv Quant Chem 1973, 7, 1.
37. Steinborn, E. O.; Filter, E. Int J Quant Chem Symp 1975, 9, 435.
38. López, R. 2001, private communication.
39. Jones, H. W. Int J Quant Chem 1992, 41, 749.
40. Zung, J. T.; Parr, R. G. J Chem Phys 1964, 41, 2888.
41. de C Bastone, É.; Quintão, A. D.; Vianna, R. O.; Mohallem, J. R. Zeit Phys D 1997, 39, 69.
42. Gracey, J. A. Comp Phys Comm 1998, 115, 113.
43. Rico, J. F.; López, R.; Ramirez, G. J Comp Chem 1988, 9, 790.
44. Gomez, C.; Scott, T. Comp Phys Comm 1998, 115, 548.
45. Hierse, W.; Oppeneer, P. M. Int J Quant Chem 1994, 52, 1249.
46. Angelescu, D. E.; Bhatt, R. N. Phy Rev B 2002, 65, 075211.
47. Pietro, W. J. 2003, private communication.
48. Buyya, R. Ed. High performance cluster computing: architectures and systems, Prentice Hall, Englewood Cliffs, 1999.
49. Wang, D.; Kupperman, A. J Chem Phys 2001, 115, 9184.
50. Vermaserren, J. 2003, private communication.
51. Moch, S.; Vermasseren, J. A. M.; Vogt, A. Nucl Phys B 2002, 646, 181.
52. Barnett, M. P. 2002, http://www.princeton.edu/~allen-grp/ms/axcat.ps
53. Zheng, X. H.; Zerner, M. C. Int J Quant Chem Suppl 1993, 27, 431.
54. Guseinov, I. I. J Mol Struct (Theochem) 1997, 417, 117.
55. Hughes, J. Comp J 1989, 32, 98.
56. Bird, R. Introduction to Functional Programming using Haskell, 2nd ed. Prentice Hall, 1998.
57. Proceedings of the ACM SIGPLAN International Conference on Functional Programming, Pittsburgh, Pennsylvania, October 2002.
58. Barnett, M. P. in Schmitt, F. O. Ed. Macromolecular Specificity and Biological Memory, 87, MIT Press, Cambridge, MA, 1962.
59. Barnett, M. P. in Carter, F. L.; Siatkowski, R. E.; Wohlten, H. Eds. Molecular Electronic Devices, Proceedings of the 3rd International Symposium on Molecular Electronic Devices, 229, North-Holland, 1988.
60. Barnett, M. P. ACM SIGPLAN Notices 1990, 25, 29.
61. Barnett, M. P.; Xu, R. ACM SIGPLAN Notices 1990, 25, 34.
62. Gray, B. F.; Pritchard, H. O.; Sumner, F. H. J Chem Soc 1956, 2631.
63. Boys, S. F.; Cook, G. B.; Reeves, C. M.; Shavitt, I. Nature (London) 1956, 1207.
64. Safouhi, H. J Math Chem 2001, 29, 213.
65. Bracken, P.; Bartlett, R. Int Journ Quantum Chem 1997, 62, 557.
66. Harris, F. E. Phys Rev A 1997, 55, 1820.
67. Harris, F. E. Int J Quant Chem 2002, 88, 701.
68. Sunel, N. 2002, private communication.
69. Jones, H. W. J Comp Chem 1991, 12, 1217.

70. Noble, C. J. Comp Phys Comm 1980, 19, 327.
71. Mekelleche, S. M.; Baba-Ahmed, A. Int J Quantum Chem 1997, 63, 843.
72. Jones, H. W. Int J Quant Chem 1997, 61, 881.
73. Maslov, I. V.; Niukkanen, A. N. J Struct Chem 1994, 35, 194.
74. Bhattacharya, A. K.; Dhabal, S. C. J Chem Phys 1986, 84, 1598.
75. Weniger, E. J.; Steinborn, E. O. Phys Rev A 1983, 28, 2026.
76. Magnasco, V.; Rapallo, A.; Casanova, M. Int J Quant Chem 1999, 73, 881.