

Call for another Seward: Optimization of F12 integral evaluation

Toru Shiozaki^{1, 2}

Institut für Theoretische Chemie, Universität Stuttgart, Pfaffenwaldring 55, D-70569 Stuttgart, Germany

Received 14.11.2010, accepted 20.11. 2010, published 08.12.2010

Explicitly correlated F12 theories have been developed in the past decades to rectify the slow convergence of dynamical electron correlation models with basis size, in which additional two-electron integrals over F12 kernels are required. This article reviews some existing algorithms for these integrals, including the author's attempt, and leave an open question: what will be the most efficient algorithm (and who wants to implement it into a tightly optimized code)?

1 Introduction

In 1867, William H. Seward (the 24th United States Secretary of State) purchased Alaska, the western frontier at that time, from Russia for 2 cents per acre. Back then, the purchase was mocked by the public as "Seward's Ice-box" [1]. After 150 years, however, his decision has proven right: Alaska is now known for its huge reserves of natural resources. Therefore, he has been a symbol of being bold to do what one believes is important yet is apparently useless.

In early 1990's, Lindh and coworkers have worked on electron repulsion integral (ERI) evaluation [2], which they wrote was "an exhausted scientific area with no room for innovation." They named the code SEWARD. Their effort has nonetheless turned out to be very valuable; all the users of MOLCAS and MOLPRO have benefited from his efficient integral routine, saving much time and perhaps increasing their number of publications. It is amazing that SEWARD is still among the most efficient integral routines after almost two decades.

There is yet another western frontier in theoretical chemistry: F12 integrals. Explicitly correlated R12 or F12 theories, which originate from Kutzelnigg's seminal work in 1985 [3], have been successful to ameliorate the slow convergence of the conventional electron correlation methods with respect to basis size [4, 5, 6]. They introduce the so-called correlation factor which explicitly depends on the electron-electron distance r_{ij} to properly describe electronic wave functions around the coalescence of two electrons. In early developments, a linear function [$f(r_{12}) = r_{12}$] was used for the correlation factor; in 2004, however, Ten-no introduced a short-ranged Slater-type geminal function (STG) [$f(r_{12}) = e^{-\gamma r_{12}}$] [7], which has been shown to improve both accuracy and numerical stability over a linear function and exclusively used in the community. In F12 theories with the so-called approximation C [8], one needs to evaluate four types of two-electron integrals,

$$(pq|rs)_S \equiv \iint d\mathbf{r}_1 d\mathbf{r}_2 \phi_p(\mathbf{r}_1) \phi_q(\mathbf{r}_1) f(r_{12}) \phi_r(\mathbf{r}_2) \phi_s(\mathbf{r}_2), \quad (1)$$

$$(pq|rs)_Y \equiv \iint d\mathbf{r}_1 d\mathbf{r}_2 \phi_p(\mathbf{r}_1) \phi_q(\mathbf{r}_1) \frac{f(r_{12})}{r_{12}} \phi_r(\mathbf{r}_2) \phi_s(\mathbf{r}_2), \quad (2)$$

$$(pq|rs)_U \equiv \iint d\mathbf{r}_1 d\mathbf{r}_2 \phi_p(\mathbf{r}_1) \phi_q(\mathbf{r}_1) U_{12} \phi_r(\mathbf{r}_2) \phi_s(\mathbf{r}_2), \quad (3)$$

$$(pq|rs)_X \equiv \iint d\mathbf{r}_1 d\mathbf{r}_2 \phi_p(\mathbf{r}_1) \phi_q(\mathbf{r}_1) f(r_{12})^2 \phi_r(\mathbf{r}_2) \phi_s(\mathbf{r}_2), \quad (4)$$

where U_{12} is the double commutator of $f(r_{12})$ with the kinetic operator

$$U_{12} = \frac{1}{2} [f(r_{12}), [-\frac{1}{2} \partial_1^2 - \frac{1}{2} \partial_2^2, f(r_{12})]]. \quad (5)$$

$\phi_p(\mathbf{r}_1)$ is a one-electron Cartesian Gaussian basis function centered at \mathbf{P}_p ,

$$\phi_p(\mathbf{r}_1) = x^l y^m z^n \exp(-\zeta_p \|\mathbf{r}_1 - \mathbf{P}_p\|^2). \quad (6)$$

This article is intended to provide a (non-self-contained) overview of the existing algorithms to evaluate these integrals and to leave a question on the best algorithm for those integrals.

¹shiozaki.toru@gmail.com, http://www.tshiozaki.net

²This work was done when the author was at Quantum Theory Project, University of Florida.

2 Background

2.1 Electron repulsion integrals

Before going into the details on the F12 integrals, let us briefly review some algorithms for ERI evaluation. Since the ERIs are fundamental quantities on which all the electronic structure theories are based, a large amount of effort has been devoted for the efficient evaluation of ERIs:

$$(pq|rs) \equiv \iint d\mathbf{r}_1 d\mathbf{r}_2 \phi_p(\mathbf{r}_1) \phi_q(\mathbf{r}_1) r_{12}^{-1} \phi_r(\mathbf{r}_2) \phi_s(\mathbf{r}_2). \quad (7)$$

The key is to use the integral representation of the integral kernel proposed by Boys more than a half century ago [9]:

$$r_{12}^{-1} = \frac{2}{\pi^{1/2}} \int_0^\infty e^{-r_{12}^2 u^2} du. \quad (8)$$

After simple algebra, one obtains [10],

$$(pq|rs) = \eta \int_0^1 w_{\text{eri}}(t) I_x(t) I_y(t) I_z(t) dt, \quad w_{\text{eri}}(t) = \frac{e^{-Tt}}{2\sqrt{t}}, \quad (9)$$

where I_x , I_y , and I_z are monic *polynomials* of t . η and T , as well as I 's, are dependent on the position, exponent, angular numbers, and contraction coefficient of the Gaussian functions. A recent paper by Flocke *et al.* [11] is pedagogical to understand this transformation.

One way of evaluating ERI is to rewrite Eq. (9) as

$$(pq|rs) = \eta \sum_{m=0}^N i_m F_m(T), \quad (10)$$

$$F_m(T) = \int_0^1 t^m w_{\text{eri}}(t) dt, \quad (11)$$

where N is the sum of angular numbers of the basis functions and i_m are dependent on the parameters of the basis functions. $F_m(T)$ are basic integrals called Boys functions, which can formally be obtained by the upward recurrence relation

$$F_m(T) = \frac{1}{2T} [(2m-1)F_{m-1}(T) - e^{-T}], \quad (12)$$

$$F_0(T) = \sqrt{\frac{\pi}{4T}} \text{erf}(\sqrt{T}). \quad (13)$$

although it is very numerically unstable in practice. Therefore, interpolation or Taylor expansion using tabulated values is often used in standard packages. The Head-Gordon-Pople algorithm (HGP) [12] has introduced an efficient way to compute i_m by rearranging the original recurrence formula of Obara and Saika (OS) [13].

The other approach is to rewrite Eq. (9) as a discrete sum over Gaussian quadrature grid [14], called Rys quadrature [15, 16, 17],

$$(pq|rs) = \eta \sum_g^{[N/2]} w_g I_x(t_g) I_y(t_g) I_z(t_g). \quad (14)$$

The positions t_g and weights w_g of quadrature grid are obtained formally by $F_m(T)$ with $0 \leq m \leq N$; owing to numerical instability, however, interpolation of tabulated data is usually used to determine the grid (see Sec. 3.1).

Among the fastest in the world is the SEWARD package by Lindh and coworkers [2]. They have introduced an efficient algorithm to build up T 's in the Rys-quadrature algorithm (Hamilton and Schaefer proposed a similar algorithm independently [18]).

2.2 F12 integrals with an STG- nG

A popular approach to F12 integrals at the moment, which has been used in MOLPRO, TURBOMOLE, and MPQC, is to expand an STG to a linear combination of n Gaussian-type geminals or GTGs (STG- nG) as

$$e^{-\gamma r_{12}} = \sum_{i=1}^{N_g} c_i e^{-\alpha_i r_{12}^2}. \tag{15}$$

Usually $N_g = 6$ is used for practical applications. Among others, the one proposed by Alrichs [19] is very efficient; it is based on the fact that the Obara–Saika recurrence relation [13] (and hence HGP relation [12]) holds for any combined GTG and Coulomb integrals. The basic integrals can be explicitly written as follows [20]. For $(pq|rs)_S$, $(pq|rs)_X$, $(pq|rs)_Y$, and $(pq|rs)_U$

$$H_n^{(S)}(\rho, T) = \sum_i^{N_g} \eta_i e^{-\rho_i T}, \tag{16}$$

$$H_n^{(X)}(\rho, T) = \sum_{i=1}^{N_g} \sum_{j=1}^i (2 - \delta_{ij}) \eta_{ij} \rho_{ij}^n e^{-\rho_{ij} T}, \tag{17}$$

$$H_n^{(Y)}(\rho, T) = \sum_{i=1}^{N_g} \tilde{\eta}_i e^{-\rho_i T} \sum_{m=0}^n n C_m \rho_i^{n-m} \hat{\rho}_i^m F_m(\hat{\rho}_i T), \tag{18}$$

$$H_n^{(U)}(\rho, T) = \sum_{i=1}^{N_g} \sum_{j=1}^i (2 - \delta_{ij}) \tilde{\eta}_{ij} \rho_{ij}^{n-1} (a_{ij} + b_{ij} T - n \hat{\rho}_{ij}) e^{-\rho_{ij} T}. \tag{19}$$

ρ 's contain all the information on the exponents of GTGs. The definition of ρ_i , ρ_{ij} , etc. is out of the scope of this article; interested readers should refer to Ref. [20]. The F12 integrals can be obtained simply by replacing $F_m(T)$ with these quantities in ERI evaluation codes, since the recurrence formula stays the same. The implementation of F12 integrals based on Alrichs' algorithm has been reported by Höfener *et al.* [20] in TURBOMOLE and recently also by Knizia as an efficient F12 integral core of MOLPRO [21].

We note in passing that there has been an attempt called HYPER-PRISM by Lambrecht and Head-Gordon [22]. Valeev has implemented a similar scheme with computer algebra and automated tuning [23].

2.3 F12 integrals with a genuine STG

The evaluation of F12 integrals using an STG has been invented by Ten-no in his ingenious work in 2004 [7]. It is based on the integral representation of the Yukawa potential,

$$\frac{e^{-\gamma r_{12}}}{r_{12}} = \frac{2}{\pi^{1/2}} \int_0^\infty e^{-u^2 r_{12}^2 - \frac{\gamma^2}{4u^2}} du, \tag{20}$$

which can be derived by successive application of Fourier and Laplace transformation to the Yukawa potential. By a similar transformation to that for ERI, one obtains [7]

$$(pq|rs)_Y = \eta \int_0^1 t w_{\text{slater}}(t) I_x(t) I_y(t) I_z(t) dt, \quad w_{\text{slater}}(t) = \frac{e^{-Tr+U(1-t^{-1})}}{2t\sqrt{t}}. \tag{21}$$

T and U are dependent on the parameters of the basis functions, and U is proportional to γ^2 at the same time. It must be emphasized that I_x is identical to that in ERI evaluation. Moreover, Ten-no showed [7] that STG integrals can be computed by taking the derivative with respect to γ as

$$(pq|rs)_S = \eta' \int_0^1 (1-t) w_{\text{slater}}(t) I_x(t) I_y(t) I_z(t) dt. \tag{22}$$

using the relation $(pq|rs)_S = -\frac{\partial}{\partial \gamma} (pq|rs)_Y$. As in the ERI evaluation, we can rewrite Eqs. (21) and (22) as

$$(pq|rs)_Y = \eta \sum_{m=0}^N i_m G_m(T, U), \tag{23}$$

$$(pq|rs)_S = \eta' \sum_{m=0}^N i_m [G_m(T, U) - G_{m-1}(T, U)], \tag{24}$$

where i_m is the same quantity appearing in Eq. (10) and

$$G_m(T, U) = \int_0^1 t^{m-1} w_{\text{slater}}(t) dt, \tag{25}$$

which is often called Ten-no's function. He also gave a three-term recurrence relation to $G_m(T, U)$ [7, 24], and implemented the OS scheme based on $G_m(T, U)$.

Given these equations, it is rather straightforward to introduce the Gaussian quadrature scheme to F12 integrals on which our development was based [25]. It replaces Eqs. (21) and (22) by finite sums

$$(pq|rs)_Y = \eta \sum_{g=1}^{\lceil N+1/2 \rceil} w_g t_g I_x(t_g) I_y(t_g) I_z(t_g), \tag{26}$$

$$(pq|rs)_S = \eta' \sum_{g=1}^{\lceil N+1/2 \rceil} w_g (1-t_g) I_x(t_g) I_y(t_g) I_z(t_g). \tag{27}$$

Again I_x is the same as in ERI evaluation. The quadrature grid can be shared between Yukawa and Slater integrals of the same exponent. Owing to the additional factor of t_g and $1-t_g$, the rank of quadrature is larger by one for even N . We will return to this scheme in Sec. 3.

Using a genuine STG, one can easily show that [26]

$$(pq|rs)_X = (pq|rs)'_S, \tag{28}$$

$$(pq|rs)_U = \gamma^2 (pq|rs)'_S, \tag{29}$$

in which \prime means that the integrals are evaluated with STG of twice the exponent 2γ , i.e., $f(r_{12}) = e^{-2\gamma r_{12}}$. As we will see later, this serves a potential advantage of using a genuine STG.

3 Our attempt using Rys-Like quadrature

In this section, the author's attempt with a genuine STG is reviewed. Inspired by Flocke's work [see Sec. 3.1] [11, 27], we sought for an algorithm to evaluate the Rys-like quadrature grid on the fly by Wheeler's algorithm (see below). We hoped that it could be the most efficient algorithm, which nonetheless has not been realized so far [Sec. 3.2].

We therefore resorted to a (less elegant) interpolation scheme based on tabulated data in Ref. [25], which is still quite efficient especially for high-angular batches [Sec. 3.3].

3.1 Wheeler's algorithm and ERI

Let us start with the general relationship between orthogonal polynomials and Gaussian quadrature [28]. Given a weight function $w(x)$ and an interval $[a, b]$ with $w(x) > 0$, $x \in (a, b)$, there is a set of monic orthogonal polynomials $[P_n(x)]$ so that

$$\int_a^b P_m(x) P_n(x) w(x) dx = 0 \quad (m \neq n), \tag{30}$$

where n and m are the rank of polynomials. There are several classical ones, for example, the Hermite polynomials $w(x) = e^{-x^2}$ on $[-\infty, \infty]$ and Laguerre polynomials $w(x) = e^{-x}$ on $[0, \infty]$. It is very easy to show that any orthogonal polynomials have a three-term recurrence formula:

$$P_n(x) = (x - \alpha_{n-1}) P_{n-1}(x) - \beta_{n-1} P_{n-2}(x). \tag{31}$$

The classical orthogonal polynomials have a closed form for α and β , while the others do not.

An n_G -point Gaussian quadrature integrates exactly the polynomials $[f(x)]$ of ranks up to $2n_G - 1$, i.e.,

$$\int_a^b f(x) w(x) dx = \sum_{g=1}^{n_G} P(x_g) w_g. \tag{32}$$

The positions x_g and weights w_g of quadrature grid points are connected to α_n and β_n of the underlying orthogonal polynomials: Given α_n and β_n , x_g are the eigenvalues of the tridiagonal matrix \mathbf{Z} ,

$$Z_{ij} = \begin{cases} \alpha_i & i = j, \\ \sqrt{\beta_k} & |i - j| = 1, k = \max(i, j), \end{cases} \quad (33)$$

and the weights are calculated using the first element of the associated eigenvectors (x_{j0}) as $w_g = x_{g,0}^2$. In principle, one can obtain the quadrature grid from the moment $M_n(x) = \int_a^b x^n w(x) dx$ up to $n = n_G$:

$$\{M_n(x)\} (0 \leq n \leq n_G) \rightarrow \{\alpha_n, \beta_n\} (0 \leq n \leq n_G) \rightarrow \{w_g, x_g\}. \quad (34)$$

Practically, however, the first mapping from $M_n(x)$ to α_n and β_n is ill conditioned and terribly numerically unstable. The Chebyshev algorithm is usually used instead which maps $\{M_n(x)\} (0 \leq n \leq 2n_G)$ to $\{\alpha_n, \beta_n\}$; this is much better than Eq. (34), but still unstable and unpractical in the current context.

A much more stable way to compute the α_n and β_n (and hence quadrature grid) is called Wheeler's algorithm which uses a similar set of orthogonal polynomials. Suppose there exists a classical set of orthogonal polynomials $\mathcal{P}_n(x)$ which obeys $\mathcal{P}_n(x) = (x - a_{n-1})\mathcal{P}_{n-1}(x) - b_{n-1}\mathcal{P}_{n-2}(x)$, which is similar to the target polynomials $P_n(x)$ (in other words, they are defined under a similar weight and integral range). We then compute the so-called modified moments $\mathcal{M}_n(x) = \int_b^a \mathcal{P}_n(x)w(x)dx$, which are in turn mapped to the quadrature grid as

$$\left[\begin{array}{l} \{\mathcal{M}_n(x)\} (0 \leq n \leq 2n_G) \\ \{a_n\} \{b_n\} (0 \leq n \leq 2n_G) \end{array} \right] \rightarrow \{\alpha_n, \beta_n\} (0 \leq n \leq n_G) \rightarrow \{w_g, x_g\}. \quad (35)$$

In the work of Flocke [11, 27], it has been shown that, to generate modified moments for Rys quadrature grid, one can use the generalized Laguerre polynomials $L^{-1/2}(Tt)$ and shifted Jacobi polynomials $G(\frac{1}{2}, \frac{1}{2}, t)$ for large and small T , respectively. The weight and integral range of $L^{-1/2}(Tt)$ are $e^{-Tt}/2\sqrt{t}$, $t \in [0, \infty)$ which are exact for $T \rightarrow \infty$, while those of $G(\frac{1}{2}, \frac{1}{2}, t)$ are $1/2\sqrt{t}$, $t \in [0, 1]$ which are exact for $T \rightarrow 0$. Moreover, he showed that there are efficient and stable ways to compute modified moments (based on the three-term recurrence relation),

$$\mathcal{M}_n^{(L)} = \int_0^1 w_{\text{eri}}(t)L_n^{-1/2}(Tt)dt, \quad (36)$$

$$\mathcal{M}_n^{(J)} = \int_0^1 w_{\text{eri}}(t)G_n(\frac{1}{2}, \frac{1}{2}, t)dt. \quad (37)$$

This finding allows us to compute Rys quadrature grid on the fly. The resulting code, which is the integral core of ACES III, is at least as efficient as SEWARD [11].

3.2 Difficulty with F12 integrals

Inspired by the work above, the author spent some time, hoping to establish a similar scheme for F12 integrals, which has turned out to be extremely hard (if by any chance possible) from several reasons. One is the non-trivial expression for modified moments with the weight function $w_{\text{slater}}(t)$. As one can see, in the limit of $U \rightarrow 0$, the Yukawa integrals reduce to ERIs (with an additional factor t in the denominator), from which one might expect that $L^{-1/2}(Tt)$ and $G(\frac{1}{2}, \frac{1}{2}, t)$ above are also useful for a certain range of T and U . Even though it is possible to obtain (after some algebra) the four-term recurrence relation of Eq. (36) with $w_{\text{eri}}(t)$ being replaced by $w_{\text{slater}}(t)$, it seems not possible to derive a similar recurrence formula for Eq. (37).

Another problem is the complicated shape of the weight function, as shown in Fig. 1. Unlike in the Rys quadrature, the shape of the weight function dramatically changes with parameters T and U with a moving pole, which makes it difficult to find appropriate classical orthogonal polynomials for modified moments.

3.3 Brute-force two-dimensional interpolation

One possible (and brute-force) way of overcoming the numerical instability is to use multiple precision floating points beyond doubles. There are standard packages GMP and MPFR, and a sophisticated wrapper MPFR C++ that provides overloaded arithmetics and mathematical functions [29]. In

our experience, 1024-bit floating points are sufficient to determine quadrature grid by means of the naive Chebyshev algorithm for any T and U values.

The obvious drawback of using multiple precisions is that computation using them is *terribly slow* compared to doubles (one should recall that the current CPU's are designed to perform double precisions efficiently). It is so inefficient that it cannot be used at runtime. Therefore, we decided to compute and tabulate grid weights and positions of some selected T and U in compile time, and resort to interpolation at runtime [25]. The runtime interpolation is a two-dimensional one for each set of T and U , or more precisely, two successive one-dimensional interpolations. The cost of evaluating quadrature grids with interpolation is almost negligible for high-angular batches [less than 4% of the entire costs for (33|33)], but it can be a noticeable overhead (and hence not optimal) for low-angular batches. See details in Ref. [25].

4 Discussions and Prospect

4.1 Comparison of STG and STG-nG algorithms

It is not clear which is better: the use of an STG or an STG-nG. Let us first consider the cost of evaluating all four types of (00|00) integrals (i.e., integrals over s -type Gaussian basis functions). For (00|00) integrals, we could perhaps conclude that approaches with a genuine STG are more efficient than those based on an STG-nG. With a genuine STG, one needs to evaluate two sets of $G_0(T, U)$ and $G_{-1}(T, U)$ with the same T , which involve 4 calls of $\exp(x)$, 4 calls of $\text{erfc}(x)$, 3 calls of \sqrt{x} , and some double float divisions. With an STG-6G, one needs 21 calls of $\exp(x)$ and 6 calls of $\text{erfc}(x)$, 21 calls of \sqrt{x} and some double float divisions.

Next, we consider the large angular momentum limit [say, a (44|44) batch]. In this case, the algorithm with an STG is again potentially more efficient. With a genuine STG, one could evaluate all four integrals with the cost three times as expensive as that of the ERI evaluation thanks to the relation $(pq|rs)_U = \gamma^2(pq|rs)_X$ [Eqs. (28) and (29)]. This has already nearly achieved in Ref. [25]. On the other hand, the STG-6G approach is four times as expensive as the ERI evaluation.

For the intermediate angular momentum range, it is not clear which performs better, since there is no simple way to obtain $G_m(T, U)$ that contains two parameters T and U . Ten-no has implemented a code to set up $G_m(T, U)$ with various combination of algorithms, which are based on the Taylor series or continued fractions with many calls to $\exp(x)$, $\text{erfc}(x)$, and double float divisions [24]. The Rys-like quadrature algorithm with interpolation is still applicable, but the overhead is not negligible here. The STG-6G algorithms might be competitive, or more efficient.

The fair comparison between them is further complicated since one does not need certain classes of integrals, such as $(pq|rs)_U$ and $(pq|rs)_X$ with one and two auxiliary functions, respectively. Moreover, the use of density fitting [30, 31] makes the advantage of the STG approach less pronounced for large angular quantum numbers.

4.2 An apparently useless backdoor: inhomogeneous Helmholtz equation

It is well known that the Yukawa potential is Green's function of the imaginary mass inhomogeneous Helmholtz equation (or simply, the screened Poisson equation):

$$(\nabla^2 - \gamma^2)V(\mathbf{r}) = -4\pi\rho(\mathbf{r}), \quad (38)$$

which is analogous to the Coulomb potential which is Green's function of the Poisson equation. Similar to ERI evaluations through the Poisson equation (for instance, see Ref. [32]), one can obtain $(pq|rs)_Y$ by the formula [25]

$$(pq|rs)_Y = \int V_{rs}(\mathbf{r}_1)\phi_p(\mathbf{r}_1)\phi_q(\mathbf{r}_1)d\mathbf{r}, \quad (39)$$

where $V_{rs}(\mathbf{r}_1)$ satisfies $(\nabla^2 - \gamma^2)V_{rs}(\mathbf{r}) = -4\pi\phi_r(\mathbf{r})\phi_s(\mathbf{r})$. One can also evaluate $(pq|rs)_S$ by taking a numerical derivative of $(pq|rs)_Y$ with respect to γ . We have implemented and confirmed this formula [25] using Becke's fuzzy cell grid [33, 34]. This approach is nevertheless efficient only for Coulomb integrals in pure DFT calculations where r and s are contracted by the density; the author has not been aware of any possible use in F12 calculations.

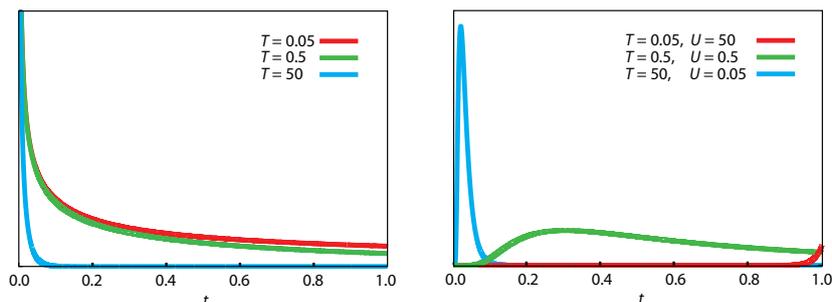


Figure 1: Weight functions for Rys and Rys-like F12 quadrature with some sets of parameters.

4.3 Possible Criticism

Unfortunately, one can easily formulate some criticisms to the exploration for efficient F12 integral codes. First, one could argue that the efficiency of the F12 integral evaluation is not important since the integral kernels are all short-ranged and the number of non-zero integrals is much smaller than that of ERIs. This is certainly true: for a large molecule, most of the F12 integrals can be skipped by prescreening [35]. However, it is still important to have it efficient for small systems and perhaps even for large systems when calculated by the integral-direct mode or with the local correlation approximation.

Second criticism would be that with density fitting the most time-consuming step could be the assembly step of three center quantities, e.g., $(ia|jb) = \sum_D (ia|D)(D|jb)$, where D labels auxiliary basis functions for DF, and not the integral evaluation itself. This is again certainly true but only for large systems. Note that CC-F12 calculations are limited to relatively small systems. Furthermore, the use of local fitting domain [35] reduces the cost of the assembly step, which may legitimate the development of efficient algorithms for F12 integrals.

5 Concluding Remarks

Given the success of the F12 theories, it is now of fundamental importance to implement a tightly optimized integral routines for F12 integrals (like SEWARD for ERI). This would be much appreciated by increasing the productivity of all the people in the theoretical chemistry community.

In this article, we intentionally have not concluded which (i.e., a genuine STG or an STG-6G) should be used for the best algorithm. Although it looks to the author that the former is favorable, it is still open for discussions. In addition, the efficiency of integral codes is determined not only by FLOP counts and the numbers of memory queries of the underlying algorithm but also by the way *how it is written* [36]. One needs to consider the cache efficiency, the numbers of function calls, loops, and condition branches, and the affinity to recent computer hardware (such as streaming SIMD extensions, general-purpose computing on graphics processing units, and so on). As pointed out by Lindh [37], it can affect the efficiency of molecular integral evaluation sometimes by an order of magnitude. In this sense, automated implementation with heuristics employed by Valeev [23] may be the way to go in a long run.

Last but not the least, we would like to emphasize that it could be an enjoyable task to realize such an optimized code; and you will be named “real programmer” [38].

References

- [1] http://en.wikipedia.org/wiki/William_H._Seward (accessed December 5, 2010).
- [2] R. Lindh, U. Ryu, B. Liu, *J. Chem. Phys.* 95 (1991) 5889.
- [3] W. Kutzelnigg, *Theor. Chim. Acta* 68 (1985) 445.
- [4] H.-J. Werner, T. B. Adler, G. Knizia, F. R. Manby, Efficient explicitly correlated coupled-cluster approximations, in: P. Čárský, J. Paldus, J. Pittner (Eds.), *Recent Progress in Coupled Cluster Methods*, Springer, 2010.
- [5] D. P. Tew, C. Hättig, R. A. Bachorz, W. Klopper, Explicitly correlated coupled-cluster theory, in: P. Čárský, J. Paldus, J. Pittner (Eds.), *Recent Progress in Coupled Cluster Methods*, Springer, 2010.
- [6] T. Shiozaki, E. F. Valeev, S. Hirata, in: R. A. Wheeler (Ed.), *Annual Reports in Computational Chemistry*, Vol. 5, Elsevier, Amsterdam, 2009, pp. 131–148.
- [7] S. Ten-no, *Chem. Phys. Lett.* 398 (2004) 56.
- [8] S. Kedžuch, M. Milko, J. Noga, *Int. J. Quantum Chem.* 105 (2005) 929.
- [9] S. F. Boys, *Proc. R. Soc. London, Ser. A* 200 (1950) 542.
- [10] The right-hand side is identical to $\int_0^1 e^{-Ts^2} I_x(s^2) I_y(s^2) I_z(s^2) ds$, which can be seen by changing the integration variable to $s = \sqrt{t}$.
- [11] N. Flocke, V. Lotrich, *J. Comp. Chem.* 29 (2008) 2722–2736.
- [12] M. Head-Gordon, J. A. Pople, *J. Chem. Phys.* 89 (1988) 5777.
- [13] S. Obara, A. Saika, *J. Chem. Phys.* 84 (1986) 3963.
- [14] See http://en.wikipedia.org/wiki/Gaussian_quadrature (accessed December 5, 2010).
- [15] M. Dupuis, J. Rys, H. F. King, *J. Chem. Phys.* 65 (1976) 111.
- [16] H. F. King, M. Dupuis, *J. Comput. Phys.* 21 (1976) 144.
- [17] J. Rys, M. Dupuis, H. F. King, *J. Comput. Chem.* 4 (1983) 154.
- [18] T. P. Hamilton, H. F. Schaefer III, *Chem. Phys.* 150 (1991) 163.
- [19] R. Ahlrichs, *Phys. Chem. Chem. Phys.* 8 (2006) 3072.
- [20] S. Höfener, F. A. Bischoff, A. Glöb, W. Klopper, *Phys. Chem. Chem. Phys.* 10 (2008) 3390.
- [21] AIC: A***** Integral Core written by G. Knizia (2010). See also <http://www.molpro.net>.
- [22] D. Lambrecht, M. Head-Gordon, a poster presentation in the 13th International Congress on Quantum Chemistry, Helsinki, 2009.
- [23] E. F. Valeev, LIBINT, machine-generated library for efficient evaluation of molecular integrals over Gaussian (2007).
- [24] S. Ten-no, *J. Chem. Phys.* 126 (2007) 014108.
- [25] T. Shiozaki, *Chem. Phys. Lett.* 479 (2009) 160.
- [26] J. Noga, S. Kedžuch, J. Šimunek, S. Ten-no, *J. Chem. Phys.* 128 (2008) 174103.
- [27] N. Focke, *J. Chem. Phys.* 131 (2009) 064107.
- [28] W. Gautschi, *Orthogonal Polynomials, Computation and Application*, Oxford University Press, Oxford, 2004.
- [29] The GMP and MPFR libraries are used with a wrapper called MPFR c++ written by P. Holoborodko.

- [30] F. R. Manby, *J. Chem. Phys.* 119 (2003) 4607.
- [31] S. Ten-no, F. R. Manby, *J. Chem. Phys.* 119 (2003) 5358.
- [32] F. R. Manby, P. J. Knowles, *Phys. Rev. Lett.* 87 (2001) 163001.
- [33] A. D. Becke, R. M. Dickson, *J. Chem. Phys.* 89 (1988) 2993.
- [34] T. Shiozaki, S. Hirata, *Phys. Rev. A* 76 (2007) 040503(R).
- [35] T. B. Adler, H.-J. Werner, F. R. Manby, *J. Chem. Phys.* 130 (2009) 054106.
- [36] K. Goto, R. A. V. De Geijn, *ACM Trans. Math. Software* 34 (2008) 12.
- [37] R. Lindh, Electron repulsion integrals, in: H. F. Schaefer III (Ed.), *Encyclopedia of Computational Chemistry*, Vol. 4, Wiley, Chichester, 1998.
- [38] E. Post, *Real Programmers Don't Use PASCAL*, *Datamation* July (1983) 263, see <http://www.ee.ryerson.ca/~elf/hack/realmen.html> (accessed December 5, 2010).