Supporting Information (SI) for:

Fast evaluation of two-center integrals over Gaussian charge distributions and Gaussian orbitals with general interaction kernels

Mieke Peels and Gerald Knizia

J. Chem. Theory Comput. 2020, https://doi.org/10.1021/acs.jctc.9b01296

List of Appendices

Appendix A: Details of basis functions

A.1. Expressions for solid harmonics

While it is possible to use Cartesian Gaussian basis functions directly (such as in Eq. (4)), in practice it is advantageous to instead transform the Cartesian monomial prefactors into polynomials which represent an (almost)-irreducible representation of the rotation group around the center they are based on. The representation used here, as in most non-relativistic quantum chemistry programs, is given by the real solid harmonics $\{S^l_m(\mathbf{r}-\mathbf{A})\}$ (where $\mathbf{A}$ is the chosen center of the basis function). For a given angular momentum l around the basis function's center $\mathbf{A}$, there are $2l+1$ real solid harmonics $\{S^l_m(\mathbf{r}-\mathbf{A});\ m = -l, (-l+1), \ldots, (l-1), l\}$, each of which is a polynomial in the Cartesian components of vector (or vector-operator) $\mathbf{r} = (x, y, z)$ with monomials of total degree l:

$$S^l_m(\mathbf{r}) = S^l_m(x,y,z) = \sum_{i=0}^{l} \sum_{j=0}^{l-i} \sum_{k=0}^{l-i-j} s^{lm}_{ijk} x^i y^j z^k. \tag{A1}$$

As described around Eq. (6), for a given integer $l \geq 0$, this sum with the restriction $l = i + j + k$, will be abbreviated as

$$S^l_m(\mathbf{r}) = \sum_{ijk}^{(l)} s^{lm}_{ijk} x^i y^j z^k. \tag{A2}$$

The real solid harmonics, and their relationship to complex solid harmonics and spherical harmonics were provided in Ref. 1 and are discussed in detail in Ref. 2. For completeness, the recursive formulas used here are reproduced in Tab. I. While in some applications it is beneficial to employ the recursive formulas directly, we here use them to pre-compute and tabulate the set of expansion coefficients $\{s^{lm}_{ijk}\}$ from Eq. (6), using either symbolic algebra or high-precision arithmetic; Tab. II contains an explicit list of the first few $S^l_m(\mathbf{r})$ obtained, as polynomials in x, y, and z. From these, the s^{lm}_{ijk} coefficients can be inferred. In our programs, the coefficients are then used in generated code to perform both explicit and inlined solid harmonic transformations without addressing overhead.

A.2. Equivalence of Hermite and standard solid harmonics

Apart from the fact that for a given $l \geq 2$ there are less solid harmonics (namely, $2l+1$) than there are Cartesian functions (namely, $(l+1)(l+2)/2$), the solid harmonic basis has an important theoretical advantage, which is discussed in detail by Weniger[3,4] and later also by Reine et al.[5]: If applied to a

TABLE I. Recursive formulas for computing the real solid harmonics $S^l_m(\mathbf{r})$ used in this work. Taken from Ref. 2. Here δ_{1l} is 1 when $l = 1$ and 0 otherwise, and non-existent terms (i.e., S^l_m with $l < 0$ or $|m| > l$) are zero. Eqs. (A4) and (A5) are used for the terms with $|m| = l$ and Eq. (A6) is used for the remaining terms.

$$S^0_0 := 1 \tag{A3}$$

$$S^l_l = \sqrt{\frac{2^{\delta_{1l}}(2l-1)}{2l}}\left(xS^{l-1}_{l-1} - (1-\delta_{1l})\,yS^{l-1}_{-(l-1)}\right) \tag{A4}$$

$$S^l_{-l} = \sqrt{\frac{2^{\delta_{1l}}(2l-1)}{2l}}\left(yS^{l-1}_{l-1} + (1-\delta_{1l})\,xS^{l-1}_{-(l-1)}\right) \tag{A5}$$

$$S^l_m = \frac{1}{\sqrt{(l+m)(l-m)}}\Big((2l-1)zS^{l-1}_m - \sqrt{(l-1+m)(l-1-m)}\left(x^2+y^2+z^2\right)S^{l-2}_m\Big) \tag{A6}$$

Gaussian exponential function, the polynomial and gradient solid harmonics are *identical*:

$$S^l_m\left(\frac{1}{2\alpha}\nabla\right)\exp\left(-\alpha\|r\|^2\right) = S^l_m(\mathbf{r})\exp\left(-\alpha\|r\|^2\right). \tag{A7}$$

This is a result of the Hobson theorem discussed in Appx. C.1. Here the spherical tensor gradient operator[3] $S^l_m(\nabla)$ is defined analogously to Eq. (6), by inserting $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ instead of $\mathbf{r} = (x, y, z)$ as argument to S^l_m. This property shown in Eq. (A7) means that, as long as monomials in (x, y, z) are ultimately transformed into solid harmonics $S^l_m(x, y, z)$, we need not distinguish between Cartesian monomials as Gaussian prefactors:

$$x^{l_x}y^{l_y}z^{l_z}e^{-\alpha\|\mathbf{r}^2\|}$$

and derivative polynomial prefactors:

$$\left(\frac{1}{2\alpha}\frac{\partial}{\partial x}\right)^{l_x}\left(\frac{1}{2\alpha}\frac{\partial}{\partial y}\right)^{l_y}\left(\frac{1}{2\alpha}\frac{\partial}{\partial z}\right)^{l_z}e^{-\alpha\|\mathbf{r}^2\|},$$

as all monomial contributions $x^iy^jz^k$ with $i + j + k < l$ around the given center $\mathbf{A}$ are annihilated in the solid harmonic transformation. Only the contributions at the order of *exactly* $i + j + k = l$ matter!

This property is central to the integral algorithm presented in this work, which ultimately allows eliminating all exponent-dependent prefactors from the actual recursive integral computations. The property was also used in several prior works to simplify or unify[5] integral evaluation techniques, or to construct entirely diffent approaches based directly on angular momentum coupling algebra, rather than the conventional quantum chemical recursive Gaussian integration techniques.[6–12] Under certain circumstances, these simplifications also allowed massive increases in the computational efficiency compared to the computation of integrals over Cartesian Gaussian basis functions.[13]

TABLE II. Explicit expressions for the real solid harmonics $S^l_m(\mathbf{r})$ with $l \leq 4$ used in this work. These are obtained from the recurrence formulas in Tab. I. A Python script to evaluate these at arbitrary order is provided. The coefficients s^{lm}_{ijk} from Eq. (6) can be read off from this: For example, the row for S^4_4 shows that $s^{44}_{040} = \frac{\sqrt{35}}{8}$, $s^{44}_{220} = -\frac{3\sqrt{35}}{4}$, and $s^{44}_{400} = \frac{\sqrt{35}}{8}$, with all other s^{44}_{ijk} coefficients vanishing.

$$\begin{aligned}
S^0_0(\mathbf{r}) &= 1\\[1ex]
S^1_{-1}(\mathbf{r}) &= y\\
S^1_0(\mathbf{r}) &= z\\
S^1_1(\mathbf{r}) &= x\\[1ex]
S^2_{-2}(\mathbf{r}) &= \sqrt{3}xy\\
S^2_{-1}(\mathbf{r}) &= \sqrt{3}yz\\
S^2_0(\mathbf{r}) &= z^2 - \tfrac{1}{2}y^2 - \tfrac{1}{2}x^2\\
S^2_1(\mathbf{r}) &= \sqrt{3}xz\\
S^2_2(\mathbf{r}) &= -\tfrac{\sqrt{3}}{2}y^2 + \tfrac{\sqrt{3}}{2}x^2\\[1ex]
S^3_{-3}(\mathbf{r}) &= -\tfrac{\sqrt{10}}{4}y^3 + \tfrac{3\sqrt{10}}{4}x^2y\\
S^3_{-2}(\mathbf{r}) &= \sqrt{15}xyz\\
S^3_{-1}(\mathbf{r}) &= \sqrt{6}yz^2 - \tfrac{\sqrt{6}}{4}y^3 - \tfrac{\sqrt{6}}{4}x^2y\\
S^3_0(\mathbf{r}) &= z^3 - \tfrac{3}{2}y^2z - \tfrac{3}{2}x^2z\\
S^3_1(\mathbf{r}) &= \sqrt{6}xz^2 - \tfrac{\sqrt{6}}{4}xy^2 - \tfrac{\sqrt{6}}{4}x^3\\
S^3_2(\mathbf{r}) &= -\tfrac{\sqrt{15}}{2}y^2z + \tfrac{\sqrt{15}}{2}x^2z\\
S^3_3(\mathbf{r}) &= -\tfrac{3\sqrt{10}}{4}xy^2 + \tfrac{\sqrt{10}}{4}x^3\\[1ex]
S^4_{-4}(\mathbf{r}) &= -\tfrac{\sqrt{35}}{2}xy^3 + \tfrac{\sqrt{35}}{2}x^3y\\
S^4_{-3}(\mathbf{r}) &= -\tfrac{\sqrt{70}}{4}y^3z + \tfrac{3\sqrt{70}}{4}x^2yz\\
S^4_{-2}(\mathbf{r}) &= 3\sqrt{5}xyz^2 - \tfrac{\sqrt{5}}{2}xy^3 - \tfrac{\sqrt{5}}{2}x^3y\\
S^4_{-1}(\mathbf{r}) &= \sqrt{10}yz^3 - \tfrac{3\sqrt{10}}{4}y^3z - \tfrac{3\sqrt{10}}{4}x^2yz\\
S^4_0(\mathbf{r}) &= z^4 - 3y^2z^2 + \tfrac{3}{8}y^4 - 3x^2z^2 + \tfrac{3}{4}x^2y^2 + \tfrac{3}{8}x^4\\
S^4_1(\mathbf{r}) &= \sqrt{10}xz^3 - \tfrac{3\sqrt{10}}{4}xy^2z - \tfrac{3\sqrt{10}}{4}x^3z\\
S^4_2(\mathbf{r}) &= -\tfrac{3\sqrt{5}}{2}y^2z^2 + \tfrac{\sqrt{5}}{4}y^4 + \tfrac{3\sqrt{5}}{2}x^2z^2 - \tfrac{\sqrt{5}}{4}x^4\\
S^4_3(\mathbf{r}) &= -\tfrac{3\sqrt{70}}{4}xy^2z + \tfrac{\sqrt{70}}{4}x^3z\\
S^4_4(\mathbf{r}) &= \tfrac{\sqrt{35}}{8}y^4 - \tfrac{3\sqrt{35}}{4}x^2y^2 + \tfrac{\sqrt{35}}{8}x^4
\end{aligned}$$

A.3. Normalization factors for Gaussian primitives

As explained in Sec. 4, an integral code itself does not need to be concerned with the normalization factors of Gaussian basis functions, because those factors can be absorbed as constants into the (anyway completely general) basis function contraction coefficients $\omega_{(\tau,z)}$ of Eq. (7). Nevertheless, in most practical applications the basis function normalization is important—and since there are multiple feasible choices (depending on both usage context and free user choice), normalization is a common source of both confusion and of incompatibility of formulas and numerical techniques obtained from different sources. For this reason, we briefly discuss the two most important types of normalization conventions of primitive Gaussians and point out subtleties in their handling.

Charge/multipole normalization. In the simplest case, the individual Gaussian basis functions *themselves* could be used to represent electronic charges or multipoles, and should be normalized accordingly. This may be relevant, for example, in the evaluation of bare or effective interactions between finite molecular charge distributions in the development of force fields, model potentials (e.g., similar developments to MB-pol[14]), solvation models, frozen-density embedding methods, or other techniques (discussed in Sec. 2). For these, the *molecular charge distributions* (as opposed to molecular orbitals, see below) could be represented as a linear combination of atomic Gaussian basis functions, and the presented integration techniques could be used to evaluate energy contributions from parameterized Coulomb, screened Coulomb, or exchange interactions, etc., and their nuclear gradients.[15]

For example, if a primitive basis function

$$\rho_c(\mathbf{r}) = q_C N_{\mathrm{chg}}(\gamma) e^{-\gamma\|\mathbf{r}-\mathbf{C}\|^2}, \tag{A8}$$

is to represent a finite spherically-symmetric Gaussian charge distribution of total electronic charge q_C at point $\mathbf{C}$, then the normalization factor $N_{\mathrm{chg}}(\gamma)$ is determined by the condition

$$\int \rho_c(\mathbf{r})\,\mathrm{d}^3 r = q_C. \tag{A9}$$

Note the absence of a norm-squared in the integrand (*cf* Eq. (A11)). This condition can be evaluated by applying the basic Gaussian integral Eq. (C2) once in each Cartesian direction, and yields the normalization factor

$$N_{\mathrm{chg}}(\gamma) = \left(\frac{\gamma}{\pi}\right)^{\frac{3}{2}}. \tag{A10}$$

This factor $q_C N_{\mathrm{chg}}(\gamma)$ could then either be used as prefactor of the basic integrals (with one contraction coefficient $\omega_{(\tau,z)} = 1$ in Eq. (7)), or absorbed into the correponding contraction coefficient. The same charge normalization constant Eq. (A10) would be used with basis functions of $l \geq 1$ to represent finite multipoles at points $\mathbf{C}$ in space (or, in general, finite linear expansions of arbitrary charge distributions).

We note in passing that the given formulas can be taken into the limit of the exponents $\zeta \to +\infty$ to represent point-charges or point-multipoles and the interactions of the same with finite charges and multipoles. To do this, separate code may not be necessary; as long as the target quantity is well-defined, the given integral formulas are stable when used with very large but finite exponents such as $\zeta = 10^{20}$; indeed, this is how our own DFT program evaluates nuclear attraction integrals with inline-contracted three-index integrals.

In some cases also the limit $\zeta \to +0$ may be useful, for example to make basis functions represent static external potentials $V(x,y,z) = V_0 \cdot z$ or similar. But this limit is more problematic numerically, and if used, the formulas and implementation need to be carefully checked for compatibility.

Orbital basis function normalization. This kind of normalization is relevant when the Gaussian basis functions are used to represent atomic orbitals over which molecular orbitals are to be expanded, which may apply to both semi-empirical (e.g., Refs. 16 and 17) and ab-initio quantum chemistry methods. In particular, in ab-initio quantum chemistry methods, frequently well-defined and standardized sets of Gaussian basis functions are used to expand molecular orbitals, with basis sets being researched and published independently of the methods they are meant to be used for, and carrying standardized names such as def2-nZVPP[18,19], ano-pVnZ[20], or cc-pCVnZ-F12.[21] A published basis set for an element then specifies all primitive exponents ζ_τ and corresponding modified contraction coefficients $\tilde{\omega}_{(\tau,z)}$ to be used for the radial basis functions in Eq. (7). However, the contraction coefficients are typically published with respect to *normalized* primitive Gaussians which fulfill

$$\int |\tilde{\psi}_a(\mathbf{r})|^2\,\mathrm{d}^3 r = 1, \tag{A11}$$

unlike the unnormalized raw primitive Gaussians discussed in Sec. 4 and the rest of this text. Consequently, the contraction coefficients $\tilde{\omega}_{(\tau,z)}$ published in the ab initio basis function literature (or obtained from libraries such as the former EMSL basis set exchange[22] or its successor[23]) need to be divided by the normalization integral of the raw Gaussian basis functions $S^l_m(\mathbf{r})e^{-\zeta\|\mathbf{r}\|^2}$,

$$N_{\mathrm{ao}}(\zeta,l) = \sqrt{\int \left|S^l_m(\mathbf{r})e^{-\zeta\|\mathbf{r}\|^2}\right|^2 \mathrm{d}^3 r} \tag{A12}$$

$$= \left(\frac{\pi}{2\zeta}\right)^{\frac{3}{4}} \cdot \sqrt{\frac{(2l-1)!!}{(4\zeta)^l}}, \tag{A13}$$

to obtain the coefficients $\omega_{(\tau,z)}$ as specified in Eq. (7):

$$\omega_{(\tau,z)} := \frac{\tilde{\omega}_{(\tau,z)}}{N_{\mathrm{ao}}(\zeta,l)}. \tag{A14}$$

In these, the double factorial $n!!$ for integers $n \geq -1$ is defined by the initial conditions and recurrence relation

$$(-1)!! := 1, \qquad 0!! := 1, \qquad n!! := (n-2)!! \cdot n. \tag{A15}$$

In a practical program, this re-normalization norm would typically be evaluated only when the basis function data is first loaded or converted from a basis set library, and then absorbed via Eq. (A14) into the resident data structures characterizing each element's basis functions.

The Gaussian primitive norm in Eq. (A13) is compatible with the solid harmonics as specified in Appendix A.1. If instead angular factors employing spherical normalization convention are used, or l-component weighting (e.g., the sphere integral of the used harmonics evaluates to 4π or $\frac{4\pi}{2l+1}$ instead of 1 as here), then additional factors of $\sqrt{4\pi}$ and/or $\frac{1}{\sqrt{2l+1}}$ may occur. For a detailed derivation of a related normalization condition, see see Eq. (6.6.14) in Ref. 2 (note that unlike the here presented formula, this derivation does include the additional $\sqrt{2l+1}$ and $\sqrt{4\pi}$ factors).

Appendix B: Aspects of code design

B.1. Basis function data representation

A suitable data structure representing all relevant details of a generally contracted shell of Gaussian basis functions (Sec. 4) in a program is given in Fig. 1. This can be implemented in a flat, C-like structure, providing for a simple basis function interface. These structures are simple enough that, if required, they could be formed on the fly based on the internal data structures of a host program.

Note that one such data structure represents a total of $N_{\mathrm{fn}} := (2l+1)\cdot N_{\mathrm{co}}$ contracted basis functions, which themselves are implicitly specified as contractions over $(2l+1)\cdot N_{\mathrm{exp}}$ primitive Gaussians. In a call to an integral routine, the integrals over all of those relevant basis functions would be computed together. This also means that a basis function order has to be decided upon, which defines which of the $(2l+1)\cdot N_{\mathrm{co}}$ represented basis functions appears at which place.

In the case of valence atomic orbital basis functions in semiempirical methods, substantial efficiency gains could be obtained by sharing primitive exponents across all basis functions of a given atom, including basis functions differing in angular momentum l. In this case, the data structure of Fig. 1 can be modified to include an additional N_{co}-length array of angular momenta $\{l_z, z=1,\ldots,N_{\mathrm{co}}\}$—one for each contraction—instead of a common l for all contractions. The algorithms presented can be easily adjusted to the shared-exponent case, but for simplicity we will not discuss it explicitly.

struct RawShell:

- l: integer, angular momentum of the shell
- nExp: integer, number of exponents (primitives)
- nCo: integer, number of contractions
- pExp: double pointer to array of nExp primitive exponents
- pCo: double pointer to array of (nExp × nCo) contraction matrix
- pCen: double pointer to shell center (array of length 3)

FIG. 1. Data structure used for defining basis function data. Integral evaluation routines take pointers to such structures as arguments. One structure describes the $(2l+1)\times N_{\mathrm{co}}$ basis functions $S^l_m(\mathbf{r}-\mathbf{C})\sum_{e=0}^{N_{\mathrm{exp}}-1}\mathrm{Co}[e+N_{\mathrm{exp}}\cdot c]\exp\big(-\mathrm{Exp[e]}\cdot\|\mathbf{r}-\mathbf{C}\|^2\big)$ (where $m=-l\ldots l$ and $c=0\ldots N_{\mathrm{co}}-1$).

B.2. Representation of interaction kernels

Routines for the presented integral algorithms can be written in such a way that *any* of the supported interaction kernels K_{12} (e.g., Tab. 1) can be handled with the same code and interface. For example, to evaluate the integrals $\{(a|K_{12}|c)\}$ via the base algorithm (Sec. 5.2), the routine only requires the ability to obtain $\tilde{G}_m(\rho,T)$ integrals with $0\leq m\leq L$ and arbitrary (ρ,T)—these functions are the *only* required information about the actual interaction K_{12}.

To represent the different interactions (see Tab. 1 and Appendix E.2), we recommend defining an abstract base class `IntegralKernel` as interface, from which classes for specific interactions (e.g., Coulomb kernel, Overlap kernel, contracted Gaussian kernel) are derived. These classes may contain data members for parameters of the interaction (e.g., for the generic Gaussian kernel of Tab. 1, the exponents $\{\omega_k\}$ and coefficients $\{c_k\}$ are required, while the Coulomb and Overlap kernels have no parameters).

As interface to the integral routines, the kernel class may supply a (pure) virtual function `EvalGmTilde` with parameters $(\rho,T,m_{\max},f)$, which computes and returns an array of $(m_{\max}+1)$ output quantities $\{(f\cdot\tilde{G}_m(\rho,T)), m = 0,1,\ldots,m_{\max}\}$. Most kernel functions involve recurrences between or common intermediates for $\tilde{G}_m(\rho,T)$ of identical ρ and T, but differing m. For this reason, all $(m_{\max}+1)$ output quantities are computed in a single routine call. In this interface, the parameter f represents a generic scalar prefactor with which all output quantities $\{\tilde{G}_m(\rho,T)\}$ are multiplied; while *conceptually* redundant, absorbing the prefactor f already at the kernel level reduces floating point and memory overhead if the $\tilde{G}_m(\rho,T)$ with $0\leq m\leq m_{\max}$ are computed recursively. To illustrate, a viable C-like protoype for such a routine is

```
void EvalTildeGm(double *pOut,
double rho, double T, unsigned MaxM,
double Factor) const;
```

This writes the output values to `pOut`[0],. . .,`pOut`[$m_{\max}$].

For integral screening purposes, the kernel class may support additional interfaces to allow for estimating the effective range of r_{12} for which $|K(r_{12})|\geq\varepsilon$ for a given threshold $\varepsilon\geq 0$, or to support additional interfaces for more efficient screening protocols.[24]

B.3. Interfaces of integral routines

How should one define an interface of integral routines to a host program, such that the routines can be called in a simple and general manner, but also allow for full efficiency and generality? This question is of considerable practical importance. For example, the base algorithm of Sec. 5.2 calculates (all) contracted integrals $(a|K_{12}|c)$ for a given kernel function K_{12} and generally contracted basis function shells. If we were to define a function `EvalInt2e2c` to execute this algorithm, how should its inputs and outputs be transmitted? There are many possible choices, and not all are equally good.

We found it most useful to define integral routine interfaces in the following manner:

- The routine is supplied two pointers/references to constant basis function objects as described in Sec. B.1: one for the a-shells (representing $N^a_{\mathrm{fn}} = (2l_a + 1) \cdot N^a_{\mathrm{co}}$ basis functions total), and one for the c-shells (representing $N^c_{\mathrm{fn}} = (2l_c + 1) \cdot N^c_{\mathrm{co}}$ basis functions total). It computes as output the $N^a_{fn} \cdot N^c_{fn}$ basis function integrals $\{(a|K_{12}|c)\}$ where $a = 0, 1, \ldots, (N^a_{\mathrm{fn}} - 1)$ and $c = 0, 1, \ldots, (N^c_{\mathrm{fn}} - 1)\}$.

- The routine is supplied a pointer/reference to an interaction kernel object as described in Sec. B.2. This specifies the $K(r_{12})$ interaction to evaluate.

- We allow the routine to *directly* write its computational result into a larger data structure (typically a matrix or tensor), rather than allocating and filling a separate intermediate data block for only the resulting shell integral block. To this end, the routine is supplied with an output data pointer `pOut` and two integer strides `StrideA` and `StrideC`, such that the concrete output integral $(a|K_{12}|c)$ for function indices $a \in \{0, 1, \ldots, (N_{\mathrm{fn}} - 1)^a\}$ and $c \in \{0, 1, \ldots, (N^c_{\mathrm{fn}} - 1)\}$ is addressed via memory location

  $$\texttt{pOut}[a \cdot \texttt{StrideA} + c \cdot \texttt{StrideC}]$$

  In most cases, this construction allows directly writing integrals into their final data structures without any need for intermediate memory allocations or separate copy operations. For example, to write the $(a|c)$ block into a $N \times N$ column-major overlap matrix, one would use `StrideA` $= 1$ and `StrideC` $= N$, and supply as `pOut` the memory address of the first element of the target block in the matrix. Apart from efficiency gains, this also simplifies the routine's use.

- The routine may be asked to absorb a common scalar prefactor (`Prefactor`) into all computed integrals, and/or to add its results to the previous data at the output location, rather than overwriting it (`Add`). Both aspects are commonly used in practice, and can be handled most efficiently at the integral routine level.

All combined, a viable C-like protoype for such a routine is

```
void EvalInt2e2c(double *pOut,
size_t StrideA, size_t StrideC, const
RawShell &A, const RawShell &C,
double Prefactor, bool Add, const
IntegralKernel &Kernel);
```

With such an interface, many common operations (e.g., assembly of an overlap matrix, or of the core Hamiltonian matrix from multiple components such as for kinetic energy or the nuclear potential) can be handled with simple code and minimal overhead.

Appendix C: Proofs, Derivations, Theorems

C.1. Basic results and theorems

For the sake of completeness and accessibility, we repeat some well-known mathematical results which have been used in this text. Additionally, this clarifies the concrete versions of the theorems used, in a form compatible with our notation.

Gaussian Product Theorem. A product of two s-type Gaussians $\chi_1(\mathbf{r}) = e^{-\sigma(\mathbf{r}-\mathbf{S})^2}$ and $\chi_2(\mathbf{r}) = e^{-\rho(\mathbf{r}-\mathbf{R})^2}$, placed on different centers $\mathbf{S}$ and $\mathbf{R}$, can be expressed as a scaled *single* Gaussian of $(\mathbf{r}-\mathbf{Z})$ located on a third center $\mathbf{Z}$:

$$e^{-\sigma(\mathbf{r}-\mathbf{S})^2}e^{-\rho(\mathbf{r}-\mathbf{R})^2} = e^{-(\sigma+\rho)\left(\mathbf{r}-\mathbf{Z}\right)^2}e^{-\left(\frac{\sigma\rho}{\sigma+\rho}\right)(\mathbf{S}-\mathbf{R})^2}, \tag{C1}$$

with $\mathbf{Z} := \frac{\sigma\mathbf{S}+\rho\mathbf{R}}{\sigma+\rho}$, which lies on the line segment between $\mathbf{R}$ and $\mathbf{S}$. Note that the second rhs exp-term is a constant prefactor, as it does not depend on $\mathbf{r}$.

Gaussian Integral.

$$\int_{-\infty}^{\infty} e^{-\sigma x^2}\mathrm{d}x = \sqrt{\frac{\pi}{\sigma}}. \tag{C2}$$

Order m binary product rule. Let $G : \mathbb{R} \mapsto \mathbb{R}$ and $H : \mathbb{R} \mapsto \mathbb{R}$ be two smooth scalar functions. Then the m-th $t-$derivative of the product $G(t)H(t)$ is obtained as:

$$\left(\frac{\partial}{\partial t}\right)^m G(t)H(t) = \sum_{i=0}^{m}\binom{m}{i}\left[\frac{\partial^i G(t)}{\partial t^i}\right]\left[\frac{\partial^{(m-i)}H(t)}{\partial t^{(m-i)}}\right]. \tag{C3}$$

The differential operators do not act beyond square brackets.

Hobson Theorem. A solid harmonic of $\mathbf{r}$ multiplied with a function $f(r^2)$ depending on $r^2 := \|\mathbf{r}\|^2$ alone can be expressed with a solid harmonic differential operator (aka the “spherical tensor gradient operator”) as follows:[3]

$$S^l_m(\mathbf{r})f(r^2) = S^l_m\left(\tfrac{1}{2}\nabla_r\right)f(r^2). \tag{C4}$$

As a corrolarry, we get

$$S^l_m(\mathbf{r}-\mathbf{A})\,e^{-\alpha(\mathbf{r}-\mathbf{A})^2} = S^l_m\left(\tfrac{1}{2\alpha}\nabla_A\right)\,e^{-\alpha(\mathbf{r}-\mathbf{A})^2}. \tag{C5}$$

C.2. Proof of the used McMurchie-Davidson Recurrence Relation (MDRR)

Commutator $[(\partial_x)^k, x]$: Notation: For brevity, we will here abbreviate $\frac{\mathrm{d}}{\mathrm{d}x}$ as ∂_x; additionally, differential operators do not act beyond square brackets. This paragraph establishes that for any integer $k' = 1, 2, 3, \ldots$, it holds that:

$$\partial_x^{k'}x = x\cdot\partial_x^{k'} + k'\cdot\partial_x^{(k'-1)}. \tag{C6}$$

For $k' = 0$, we will treat the second term on the rhs as zero (it has prefactor $k' = 0$, but an undefined derivative); in this case Eq. (C6) holds also for $k' = 0$. Proof of Eq. (C6) for $k' \neq 0$: First note that for any smooth function $g(x)$,

$$\begin{aligned}[\partial_x, x]\,g(x) &= \partial_x x g(x) - x\partial_x g(x) \\ &= \underbrace{[\partial_x x]}_{1}g(x) + \underbrace{x\partial_x g(x) - x\partial_x g(x)}_{0} = 1g(x).\end{aligned} \tag{C7}$$

This well-known relation shows that

$$\partial_x x = 1 + x\partial_x, \tag{C8}$$

and therefore the validity of Eq. (C6) for $k' = 1$. Now let $k \geq 2$ be an integer, and assume Eq. (C6) to hold for all integers $k' \leq k-1$. Then we have, inserting Eq. (C6) for $k' = k-1$ in step 2 and Eq. (C8) in step 3:

$$\begin{aligned}\partial_x^k x &= \partial_x\left(\partial_x^{k-1}x\right) \\ &= \partial_x\left(x\partial_x^{k-1} + (k-1)\partial_x^{k-2}\right) \\ &= (\partial_x x)\,\partial_x^{k-1} + (k-1)\partial_x^{k-1} \\ &= (1 + x\partial_x)\,\partial_x^{k-1} + (k-1)\partial_x^{k-1} \\ &= x\partial_x^k + k\partial_x^{k-1}.\end{aligned} \tag{C9}$$

By induction, this shows the general validity of Eq. (C6).

Scalar MDRR: Let $\partial_x := \left(\frac{\mathrm{d}}{\mathrm{d}x}\right)$, $t := \rho x^2$, and $\partial_t := \left(\frac{\partial}{\partial t}\right)$. Then for any smooth function $f(t)$, it holds that

$$\begin{aligned}\partial_x^k\partial_t^m f(t) = 2\rho\Big(&x\partial_x^{(k-1)}\partial_t^{(m+1)}f(t) \\ &+ (k-1)\partial_x^{(k-2)}\partial_t^{(m+1)}f(t)\Big).\end{aligned} \tag{C10}$$

We only need to show that for any smooth function $f(t)$

$$\partial_x^k f(t) = 2\rho\left[x\partial_x^{(k-1)}\partial_t f(t) + (k-1)\partial_x^{(k-2)}\partial_t f(t)\right]: \tag{C11}$$

If this holds for any generic function f', it also holds for $f'(t) := \partial_t^m f(t)$; so Eq. (C11) implies Eq. (C10).

To establish Eq. (C11), first note that

$$\partial_x f = \frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial f}{\partial t}\frac{\mathrm{d}t}{\mathrm{d}x} = \frac{\partial f}{\partial t}\frac{\mathrm{d}(\rho x^2)}{\mathrm{d}x} = 2\rho x\frac{\partial}{\partial t}f. \tag{C12}$$

Now invoke Eq. (C6) with $k' = k-1$, and insert Eq. (C12):

$$\begin{aligned}\partial_x^k f(t) &= \partial_x^{(k-1)}\left(\partial_x f(t)\right) = \partial_x^{(k-1)}\left(x\cdot 2\rho\,\partial_t f(t)\right) \\ &= \left(\partial_x^{(k-1)}x\right)\left(2\rho\,\partial_t f(t)\right) \\ &= \left(x\partial_x^{(k-1)} + (k-1)\partial_x^{(k-2)}\right)\left(2\rho\,\partial_t f(t)\right).\end{aligned} \tag{C13}$$

This establishes Eq. (C10).

It is easily confirmed that in the case of additional spatial dimensions, i.e., $t = \rho(x^2 + y^2 + z^2)$ instead of $t = \rho x^2$, Eq. (C10) holds for all partial derivatives ∂_x, ∂_y, ∂_z independently. In particular, if we define

$$T := \rho\left(R_x^2 + R_y^2 + R_z^2\right), \tag{C14}$$

$$\hat{D}_T := \left(+2\rho \frac{\partial}{\partial T}\right), \tag{C15}$$

$$\partial_i := \frac{\partial}{\partial R_i} \qquad \text{(where } i \in \{x,y,z\}\text{)}, \tag{C16}$$

then we get for any smooth function $G(T)$ that

$$\begin{aligned}\partial_i^k \hat{D}_T^m G(T) = &+R_i \partial_i^{(k-1)} \hat{D}_T^{(m+1)} G(T) \\ &+(k-1)\partial_i^{(k-2)} \hat{D}_T^{(m+1)} G(T).\end{aligned} \tag{C17}$$

This equation is a version of the well-known McMurchie-Davidson recurrence relation,[2,25,26] and allows building higher spatial derivatives of a function of $G(\rho R^2)$ from its scalar parameter derivatives incrementally. A similar MDRR for general smooth functions $f(r)$ has been previously provided by Cisneros and coworkers;[15] the current variant differs in being formulated in terms of $f(\rho R^2)$ instead of $f(\sqrt{R^2})$.

Appendix D: Evaluation of the I_0 integral

Following the original work of Ahlrichs,[27] we here reproduce the detailed derivations required to construct concrete base integral expressions relating Eqs. (12) and (13) of the main text.

D.1. General simplifications

We here consider the evaluation of I_0 in Eq. (12),

$$I_0 := \iint e^{-\alpha(\mathbf{r}_1-\mathbf{A})^2} K(r_{12}) e^{-\gamma(\mathbf{r}_2-\mathbf{C})^2}\, \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2. \tag{D1}$$

We first substitute the integration over $\mathbf{r}_1$ and $\mathbf{r}_2$ into center-of-mass and relative coordinates, by defining

$$\mathbf{r}_{12} := \mathbf{r}_1 - \mathbf{r}_2 \qquad \mathbf{R}_{12} := \frac{1}{2}(\mathbf{r}_1 + \mathbf{r}_2) \tag{D2}$$

$$\Rightarrow \qquad \mathbf{r}_1 = \mathbf{R}_{12} + \tfrac{1}{2}\mathbf{r}_{12} \qquad \mathbf{r}_2 = \mathbf{R}_{12} - \tfrac{1}{2}\mathbf{r}_{12} \tag{D3}$$

Inserting this yields

$$\begin{aligned} I_0 &= \iint e^{-\alpha\left((\mathbf{R}_{12}+\frac{1}{2}\mathbf{r}_{12})-\mathbf{A}\right)^2} K(r_{12}) e^{-\gamma\left((\mathbf{R}_{12}-\frac{1}{2}\mathbf{r}_{12})-\mathbf{C}\right)^2} \mathrm{d}\mathbf{R}_{12}\mathrm{d}\mathbf{r}_{12} \\ &= \iint e^{-\alpha\left(\mathbf{R}_{12}+(\frac{1}{2}\mathbf{r}_{12}-\mathbf{A})\right)^2} e^{-\gamma\left(\mathbf{R}_{12}-(\frac{1}{2}\mathbf{r}_{12}-\mathbf{C})\right)^2} K(r_{12})\, \mathrm{d}\mathbf{R}_{12}\mathrm{d}\mathbf{r}_{12} \end{aligned} \tag{D4}$$

We now invoke the Gaussian Product Theorem Eq. (C1) to transform the exponentials as

$$e^{-\alpha\left(\mathbf{R}_{12}+\left(\frac{1}{2}\mathbf{r}_{12}-\mathbf{A}\right)\right)^2} e^{-\gamma\left(\mathbf{R}_{12}-\left(\frac{1}{2}\mathbf{r}_{12}-\mathbf{C}\right)\right)^2} \tag{D5}$$

$$= e^{-(\alpha+\gamma)(\mathbf{R}_{12}-\mathbf{Z})^2} e^{-\rho((\mathbf{A}-\mathbf{C})-\mathbf{r}_{12})^2} \tag{D6}$$

where we defined

$$\begin{aligned} \rho &:= \frac{\alpha\gamma}{\alpha+\gamma} \\ \mathbf{Z} &:= \frac{1}{\alpha+\gamma}\left(\alpha\left(\mathbf{A}-\tfrac{1}{2}\mathbf{r}_{12}\right) + \gamma\left(\mathbf{C}+\tfrac{1}{2}\mathbf{r}_{12}\right)\right). \end{aligned} \tag{D7}$$

Inserting Eq. (D6) into Eq. (D4) then yields

$$\begin{aligned} I_0 = \int & e^{-\rho((\mathbf{A}-\mathbf{C})-\mathbf{r}_{12})^2} K(r_{12}) \cdot \\ & \left(\int e^{-(\alpha+\gamma)(\mathbf{R}_{12}-\mathbf{Z})^2} \mathrm{d}\mathbf{R}_{12}\right) \mathrm{d}\mathbf{r}_{12}. \end{aligned} \tag{D8}$$

In this expression, let us consider the inner integral over $\mathbf{R}_{12}$. Its dependence on $\mathbf{Z}$ can be eliminated by shifting the coordinate origin of the full-space integral: Substituting $\mathbf{r} := \mathbf{R}_{12} - \mathbf{Z}$ and $\sigma := \alpha + \gamma$, the $\mathbf{R}_{12}$ integral evaluates to

$$\begin{aligned} \int e^{-\sigma \mathbf{r}^2} \mathrm{d}\mathbf{r} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\sigma\left(x^2+y^2+z^2\right)} \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \\ &= \int_{-\infty}^{\infty} e^{-\sigma x^2}\,\mathrm{d}x \cdot \int_{-\infty}^{\infty} e^{-\sigma y^2}\,\mathrm{d}y \cdot \int_{-\infty}^{\infty} e^{-\sigma z^2}\,\mathrm{d}z \\ &= \sqrt{\frac{\pi}{\sigma}} \cdot \sqrt{\frac{\pi}{\sigma}} \cdot \sqrt{\frac{\pi}{\sigma}} = \left(\frac{\pi}{\sigma}\right)^{\frac{3}{2}} = \left(\frac{\pi}{\alpha+\gamma}\right)^{\frac{3}{2}}. \end{aligned} \tag{D9}$$

Inserting this back into Eq. (D8), we obtain

$$I_0 = \left(\frac{\pi}{\alpha+\gamma}\right)^{\frac{3}{2}} \int e^{-\rho(\mathbf{R}-\mathbf{r}_{12})^2} K(r_{12})\, \mathrm{d}\mathbf{r}_{12}, \tag{D10}$$

where $\mathbf{R} = (\mathbf{A} - \mathbf{C})$. Let us define

$$G_0(\rho, T) := \int e^{-\rho(\mathbf{R}-\mathbf{r}_{12})^2} K(r_{12})\, \mathrm{d}\mathbf{r}_{12}. \tag{D11}$$

As discussed by Ahlrichs,[27] for many kernels $K(r_{12})$, this expression can be evaluated in closed form. In particular, Eq. (D11) is a good starting point for integral kernels which factorize into the Cartesian space directions. We will treat various examples of those in Appendix E.2, where concrete expressions of $G_m(\rho,T)$ for various kernels $K(r_{12})$ are given, and their evaluation is discussed.

D.2. Radially non-separable kernels

Eq. (D11) is not necessarily a good starting point for all integral kernels. To proceed with Eq. (D11) for more general kernels, we introduce spherical coordinates (r,ϑ,φ), which are oriented in such a way that the z-axis faces in the direction of $\mathbf{R}$. Therefore the polar angle ϑ measures the angle between $\mathbf{r}_{12}$ and $\mathbf{R}$, so $\mathbf{r}_{12}\cdot\mathbf{R} = rR\cos(\vartheta)$. By noting that

$$\begin{aligned} e^{-\rho(\mathbf{R}-\mathbf{r}_{12})^2} &= e^{-\rho\left(R^2+r^2\right)}e^{2\rho rR\cos(\vartheta)} \\ &= e^{-T}e^{-\rho r^2}e^{2\rho rR\cos(\vartheta)}, \end{aligned} \tag{D12}$$

where $T := \rho R^2$, and inserting this into Eq. (D11), we obtain

$$G_0(\rho,T) = \int_0^\infty\int_0^\pi\int_{-\pi}^{+\pi} e^{-T}e^{-\rho r^2}e^{2\rho rR\cos(\vartheta)}\cdot \\ K(r)\,r^2\sin(\vartheta)\,\mathrm{d}\varphi\,\mathrm{d}\vartheta\,\mathrm{d}r.$$

As no terms depend on φ, its integral evaluates to 2π. The terms depending on ϑ can be collected, and the ϑ-integral can be evaluated by the substitution $x := \cos(\vartheta)$ ($\rightarrow$ $\mathrm{d}x = -\sin(\vartheta)\mathrm{d}\vartheta$):

$$\int_0^\pi e^{2\rho rR\cos(\vartheta)}\sin(\vartheta)\,\mathrm{d}\vartheta = \int_{-1}^1 e^{2\rho rRx}\mathrm{d}x = \frac{\sinh(2\rho rR)}{\rho rR}.$$

With this we obtain the one-dimensional integral

$$G_0(\rho,T) = 2\pi\int_0^\infty e^{-T}\frac{\sinh(2\rho rR)}{\rho rR}r^2e^{-\rho r^2}K(r)\,\mathrm{d}r. \tag{D13}$$

Noting $\rho R = \sqrt{\rho T}$, and substituting $y := \sqrt{\rho}r$ ($\rightarrow$ $\mathrm{d}r = \mathrm{d}y/\sqrt{\rho}$), this can be evaluated to

$$\begin{aligned} G_0 &= \frac{2\pi e^{-T}}{\sqrt{\rho T}}\int_0^\infty \sinh\left(2r\sqrt{\rho T}\right)r\,e^{-\rho r^2}K(r)\,\mathrm{d}r \\ &= \frac{2\pi e^{-T}}{\sqrt{\rho^3T}}\int_0^\infty \sinh\left(2y\sqrt{T}\right)y\,e^{-y^2}K\left(\frac{y}{\sqrt{\rho}}\right)\mathrm{d}y. \end{aligned} \tag{D14}$$

Appendix E: Expressions for concrete kernel functions

E.1. Coulomb kernel $K(r_{12}) = 1/r_{12}$

For some kernels $K(r_{12})$, Eq. (D14) can be explicitly evaluated. In particular, if we substitute in the Coulomb kernel, $K(r_{12}) = \frac{1}{r_{12}}$, we obtain an integral which can be evaluated by standard techniques in closed form:

$$G_0^{\mathrm{coul}} = \frac{2\pi e^{-T}}{\sqrt{\rho^3T}}\int_0^\infty \sinh\left(2y\sqrt{T}\right)y\,e^{-y^2}\frac{\sqrt{\rho}}{y}\mathrm{d}y. \tag{E1}$$

$$= \frac{2\pi}{\rho}\frac{e^{-T}}{\sqrt{T}}\underbrace{\int_0^\infty \sinh\left(2y\sqrt{T}\right)e^{-y^2}\mathrm{d}y}_{=\frac{1}{2}\sqrt{\pi}e^T\,\mathrm{erf}\left(\sqrt{T}\right)}. \tag{E2}$$

$$= \frac{\pi}{\rho}\frac{\sqrt{\pi}}{\sqrt{T}}\mathrm{erf}\left(\sqrt{T}\right) = \frac{2\pi}{\rho}F_0(T). \tag{E3}$$

Here $\mathrm{erf}(x) = \frac{1}{\sqrt{\pi}}\int_{-x}^x e^{-t^2}\mathrm{d}t$ is the Error Function, and $F_0(T)$ is the first Boys function ($m = 0$):

$$F_0(T) := \frac{1}{2}\frac{\sqrt{\pi}}{\sqrt{T}}\mathrm{erf}\left(\sqrt{T}\right). \tag{E4}$$

The practical evaluation of $F_0(T)$ and its derivatives $F_m(T) = \left(-\frac{\partial}{\partial T}\right)^m F_0(T)$ will be discussed in Appendix G. With the given Boys function, we get

$$G_m^{\mathrm{coul}}(\rho,T) = \frac{2\pi}{\rho}F_m(T) \tag{E5}$$

as kernel function for the Coulomb kernel $K(r_{12}) = \frac{1}{r_{12}}$.

E.2. $G_m(\rho,T)$ for various factorizable integral kernels

By directly invoking Eq. (D11), we get for the simplest integral kernels:

$$K(\mathbf{r}_{12}) = \delta(\mathbf{r}_{12}) \quad \Rightarrow G_0(\rho,T) = e^{-T} \tag{E6}$$

$$K(\mathbf{r}_{12}) = 1 \quad \Rightarrow G_0(\rho,T) = \left(\frac{\pi}{\rho}\right)^{3/2} \tag{E7}$$

$$K(\mathbf{r}_{12}) = \mathbf{r}_{12}^2 \quad \Rightarrow G_0(\rho,T) = \sqrt{\frac{\pi^3}{\rho^5}}\left(T+\frac{3}{2}\right) \tag{E8}$$

$$K(\mathbf{r}_{12}) = e^{-\omega r_{12}^2} \quad \Rightarrow G_0(\rho,T) = \left(\frac{\pi}{\rho+\omega}\right)^{3/2}e^{-\left(\frac{\omega}{\rho+\omega}\right)T}. \tag{E9}$$

All of these kernels factorize into the three Cartesian directions. Eq. (E6) is a useful kernel; if inserted into $(a|c)$, it evaluates overlap integrals. Also kinetic energy integrals and related kernels can be obtained from this kernel, as discussed above in Sec. 6.1. Eq. (E7) is easily evaluated by shifting the coordinate system origin in the full-space integral Eq. (D11) to $\mathbf{R}$, and factorizing the integral into the three Cartesian directions. It is not a particularly useful kernel, though: it results in $(a|c) = \left(\int a(\mathbf{r}_1)\,\mathrm{d}^3 r_1\right)\cdot\left(\int c(\mathbf{r}_2)\,\mathrm{d}^3 r_2\right)$, and could at best be used to compute a's multipole-normalization integrals if combined with a function $c(\mathbf{r}) = 1$ (by setting $\gamma = 0$ and $l_c = 0$). Eq. (E8) will be derived in Appendix F.2, and the Gaussian kernel Eq. (E9) next in Appendix E.3.

E.3. $G_m(\rho,T)$ for Gaussian fits of general $K(r_{12})$

Eq. (E9) is one of the most useful kernels, as it allows (approximately) computing integrals over almost arbitrary interaction potentials $K(r_{12})$—by first non-linearly fitting them into a fixed linear combination of multiple Gaussians in r_{12}:

$$K(r_{12}) \approx \sum_{k=1}^{N} c_k e^{-\omega_k r_{12}^2}, \tag{E10}$$

where the $\{c_k\}$ and $\{\omega_k\}$ denote suitable coefficients and exponents. As a direct extension of (E9), the corresponding $G_m(\rho,T)$ function is then

$$\begin{aligned} G_m(\rho,T) &= \left(-\frac{\partial}{\partial T}\right)^m \sum_{k=1}^{N} c_k \left(\frac{\pi}{\rho+\omega}\right)^{3/2} e^{-\tilde{\rho}_k T} \\ &= \sum_{k=1}^{N} c_k \left(\frac{\pi}{\rho+\omega}\right)^{3/2} \tilde{\rho}_k^m e^{-\tilde{\rho}_k T}, \end{aligned} \tag{E11}$$

where $\tilde{\rho}_k = \left(\frac{\omega_k}{\rho+\omega_k}\right)$. In F12 theory, this is commonly used to compute integrals over the Slater geminals $K(r_{12}) = -\frac{1}{\gamma}\exp(-\gamma r_{12})$ (or other F_{12} correlation factors) by fitting them with a fixed number of Gaussians;[28] however, also arbitrary other kernels can be fitted, and there may be reasons to do so, particularly in semi-empirical methods. The $G_m(\rho,T)$ in Eq. (E9) is obtained by noticing that the integral factors into the three Cartesian directions, and using the Gaussian product theorem Eq. (C1) (with $S = 0$ and Z irrelevant)

$$e^{-\rho(R_x-x)^2} e^{-\omega x^2} = e^{-(\rho+\omega)(x-Z)^2} e^{-\frac{\rho\omega}{\rho+\omega}R_x^2}. \tag{E12}$$

With this and the standard Gaussian integral Eq. (C2), we get

$$\begin{aligned} \int_{-\infty}^{\infty} e^{-\rho(R_x-x)^2} e^{-\omega x^2}\mathrm{d}x &= \int_{-\infty}^{\infty} e^{-(\rho+\omega)(x-Z)^2} e^{-\left(\frac{\rho\omega}{\rho+\omega}\right)R_x^2}\mathrm{d}x \\ &= e^{-\left(\frac{\rho\omega}{\rho+\omega}\right)R_x^2}\sqrt{\frac{\pi}{\rho+\omega}}. \end{aligned} \tag{E13}$$

This directly yields Eq. (E9) if multiplied by the equivalent results from the other Cartesian directions. This kernel has been previously presented by Höfener and coworkers.[29]

Appendix F: Transformations of integral kernels

As pointed out by Ahlrichs,[27] if a concrete expression $G_0(\rho,T)$ for an integral kernel $K(r_{12})$ is already known, it is in some cases possible to construct from this the corresponding $\tilde{G}_0(\rho,T)$ quantity for a related kernel $\tilde{K}(r_{12})$. We here outline and derive the known general kernel transformations.

F.1. Transformation $\tilde{K}(r_{12}) = e^{-\omega r_{12}^2} K(r_{12})$

If the new kernel $\tilde{K}(r_{12})$ is given by

$$\tilde{K}(r_{12}) = e^{-\omega r_{12}^2} K(r_{12}), \tag{F1}$$

then the corresponding $\tilde{G}_0(\rho,T)$ can be obtained as follows: Via Eq. (D11) we obtain

$$\begin{aligned} \tilde{G}_0(\rho,T) &= \int e^{-\rho(\mathbf{R}-\mathbf{r}_{12})^2} \tilde{K}(r_{12})\,\mathrm{d}\mathbf{r}_{12} \\ &= \int e^{-\rho(\mathbf{R}-\mathbf{r}_{12})^2} e^{-\omega \mathbf{r}_{12}^2} K(r_{12})\,\mathrm{d}\mathbf{r}_{12}. \end{aligned} \tag{F2}$$

For the integrand we can invoke the Gaussian product theorem Eq. (C1) (with $\mathbf{S} = 0$):

$$e^{-\rho(\mathbf{R}-\mathbf{r}_{12})^2} e^{-\omega \mathbf{r}_{12}^2} = e^{-(\rho+\omega)(\mathbf{r}-\mathbf{Z})^2} e^{-\frac{\rho\omega}{\rho+\omega}\mathbf{R}^2} \tag{F3}$$

$$\mathbf{Z} = \frac{\rho\mathbf{R}+\omega\mathbf{0}}{\rho+\omega} = \left(\frac{\rho}{\rho+\omega}\right)\mathbf{R}. \tag{F4}$$

Inserting this back into Eq. (F2) and subsequently substituting in Eq. (D11), we get

$$\begin{aligned} \tilde{G}_0(\rho,T) &= \int e^{-\left(\frac{\rho\omega}{\rho+\omega}\right)\mathbf{R}^2} e^{-(\rho+\omega)(\mathbf{r}-\mathbf{Z})^2} K(r_{12})\,\mathrm{d}\mathbf{r}_{12} \\ &= e^{-\left(\frac{\omega}{\rho+\omega}\right)T} \int e^{-(\rho+\omega)\left(\mathbf{r}-\left(\frac{\rho}{\rho+\omega}\right)\mathbf{R}\right)^2} K(r_{12})\,\mathrm{d}\mathbf{r}_{12} \\ &= e^{-\left(\frac{\omega}{\rho+\omega}\right)T} G_0\!\left(\rho+\omega, \left(\frac{\rho}{\rho+\omega}\right)T\right) \qquad \text{(F5)} \\ &= e^{-\tilde{\rho}T} G_0(\rho+\omega, \hat{\rho}T) \qquad \text{(F6)} \end{aligned}$$

where we have defined

$$\tilde{\rho} := \left(\frac{\omega}{\rho+\omega}\right) \qquad\qquad \hat{\rho} := \left(\frac{\rho}{\rho+\omega}\right). \tag{F7}$$

So the $\tilde{G}_0(\rho,T)$ function of the new kernel can be obtained by a simple transformation of the function $G_0(\rho,T)$ of the original kernel. This allows attenuating arbitrary kernels with a Gaussian function $e^{-\omega r^2}$. The simple structure of the prefactor also allows computing the needed derivatives in a simple way. The transformed $\tilde{G}_m(\rho,T)$ is

$$\begin{aligned}\tilde{G}_m(\rho,T) &:= \left(-\frac{\partial}{\partial T}\right)^m \tilde{G}_0(\rho,T)\\ &= \left(\frac{-\partial}{\partial T}\right)^m e^{-\tilde{\rho}T} G_0(\rho+\omega,\hat{\rho}T)\,. \end{aligned}\tag{F8}$$

First note that

$$\left(-\frac{\partial}{\partial T}\right)^i e^{-\tilde{\rho}T} = \tilde{\rho}^i e^{-\tilde{\rho}T}\tag{F9}$$

$$\left(-\frac{\partial}{\partial T}\right)^m G_0(\rho+\omega,\hat{\rho}T) = \hat{\rho}^m G_m(\rho+\omega,\hat{\rho}T)\,.\tag{F10}$$

Inserting this into the order-m product rule Eq. (C3), we get

$$\begin{aligned}\tilde{G}_m(\rho,T) &= \sum_{i=0}^{m}\binom{m}{i}\left[\left(-\frac{\partial}{\partial T}\right)^i e^{-\tilde{\rho}T}\right]\cdot\\ &\qquad\left[\left(-\frac{\partial}{\partial T}\right)^{(m-i)} G_0(\rho+\omega,\hat{\rho}T)\right]\\ &= \sum_{i=0}^{m}\binom{m}{i}\tilde{\rho}^i\,\hat{\rho}^{(m-i)}\,e^{-\tilde{\rho}T}G_{m-i}(\rho+\omega,\hat{\rho}T)\,.\end{aligned}\tag{F11}$$

Here the $G_m(\rho,T)$ on the rhs of Eq. (F11) is the G_m function of the original kernel $K(r_{12})$.

Formulas of this kind have been presented by Höfener[29] in the context of F12 theory, where they are used in conjunction with the Gaussian fit of Slater geminal correlation factors

$$-\frac{1}{\beta}\exp(-\beta r_{12}) \approx \sum_{k=1}^{N} c_k e^{-\omega_k r_{12}^2} =: F_{12}(r_{12})\tag{F12}$$

(see Eq. (E10)) to compute integrals over their products with Coulomb interaction potentials $1/r_{12}$:

$$\tilde{K}(r_{12}) = \frac{F_{12}(r_{12})}{r_{12}} = \sum_{k=1}^{N} c_k \frac{\exp\left(-\omega_k r_{12}^2\right)}{r_{12}}\tag{F13}$$

However, the Gaussian transformation is very general, and may find uses in many other cases, for example in the representation of interaction pseudo-potentials.

F.2. Transformation $\tilde{K}(r_{12}) = r_{12}^2 K(r_{12})$

As pointed out by Ahlrichs,[27] the Gaussian kernel transformation $\tilde{K}(r_{12}) = e^{-\omega r_{12}^2}K(r_{12})$ of the previous section affords a systematic way to construct the transformed $\tilde{G}_0(\rho,T)$ function for the kernel

$$\tilde{K}(r_{12}) = r_{12}^2 K(r_{12}),\tag{F14}$$

if $K(r_{12})$ is associated with a known $G_0(\rho,T)$ function.

First, note that

$$\tilde{K}(r_{12}) = r_{12}^2 K(r_{12})\tag{F15}$$

$$= \lim_{\omega\to 0}\left(\frac{-\partial}{\partial\omega}\right) e^{-\omega r_{12}^2}K(r_{12}),\tag{F16}$$

and that, according to Eq. (D11), for any kernel $K(r_{12})$, the associated kernel function $G_0(\rho,T)$ is obtained as a linear transformation of this $K(r_{12})$. Consequently, we can obtain $\tilde{G}_0(\rho,T)$ for $\tilde{K}(r_{12}) = r_{12}^2 K(r_{12})$ by combining Eq. (F16) with Eqs. (F6) and (F7) from the Gaussian transformation of the original kernel function:

$$\begin{aligned}\tilde{G}_0(\rho,T) &= \lim_{\omega\to 0}\left(\frac{-\partial}{\partial\omega}\right) e^{-\tilde{\rho}T}G_0(\rho+\omega,\hat{\rho}T) \qquad\text{(F17)}\\ &= \lim_{\omega\to 0}\left(\frac{-\partial}{\partial\omega}\right) e^{-\left(\frac{\omega}{\rho+\omega}\right)T} G_0\Big(\rho+\omega,\left(\frac{\rho}{\rho+\omega}\right)T\Big)\,.\end{aligned}$$

Explicitly evaluating the ω-derivatives at $\omega = 0$ then yields Ahlrichs' transformation formula[27]

$$\tilde{G}_0(\rho,T) = \frac{T}{\rho}\left(\frac{\partial G_0(\rho,T)}{\partial T} + G_0(\rho,T)\right) - \frac{\partial G_0(\rho,T)}{\partial\rho}.\tag{F18}$$

Unfortunately, this transformed $\tilde{G}_0(\rho,T)$ includes a ρ-derivative of $G_0(\rho,T)$; consequently, the ability to numerically evaluate the original kernel's $G_m(\rho,T)$ functions for arbitrary m (and therefore arbitrary T-derivatives) is generally not sufficient to also evaluate the transformed kernel's $\tilde{G}_m(\rho,T)$ via product rules—unlike for the discussed Gaussian transformation, where Eq. (F11) affords a general abstract numerical implementation. That means that generally this $\tilde{K}(r_{12}) = r^2K(r_{12})$ transformation requires explicit symbolic computations to derive and implement $\tilde{G}_m(\rho,T)$ based on a known symbolic expression for the original $G_0(\rho,T)$.

There are two special integral kernels of interest which this transformation gives access to; both were pointed out by Ahlrichs.[27] First, we have

$$\begin{aligned}&\tilde{K}(r_{12}) = r_{12}\\ &\quad\Rightarrow \tilde{G}_0(\rho,T) = \frac{2\pi}{\rho^2}\left((1+T)F_0(T) - T\cdot F_1(T)\right),\end{aligned}\tag{F19}$$

which is obtained from Eqs. (E5) and (F18) by applying the r_{12}^2 transformation to the Coulomb kernel $K(r_{12}) = 1/r_{12}$. $F_m(T)$ denotes the Boys function described in Appendix G. Note also that higher $\tilde{G}_m(\rho,T)$ of this $\tilde{G}_0(\rho,T)$ can be evaluated with product rule Eq. (C3). Second,

$$\tilde{K}(r_{12}) = r_{12}^2 \quad\Rightarrow\quad \tilde{G}_0(\rho,T) = \sqrt{\frac{\pi^3}{\rho^5}}\left(T+\frac{3}{2}\right),\tag{F20}$$

which is obtained from Eqs. (E7) and (F18) by applying the r_{12}^2 transformation to the identity $K(r_{12}) = 1$.

F.3. Attenuated Coulomb kernel $\tilde{K}(r_{12}) = \mathrm{erf}(r_{12})/r_{12}$

In ab-initio quantum chemistry, the Coulomb interaction kernel $K(r_{12}) = 1/r_{12}$ (see Appendix E.1) is occasionaly split into short-range and long-range parts via[30]

$$K(r_{12}) = \frac{1}{r_{12}} =: \underbrace{\frac{\mathrm{erfc}(\theta \cdot r_{12})}{r_{12}}}_{\text{short-range}} + \underbrace{\frac{\mathrm{erf}(\theta \cdot r_{12})}{r_{12}}}_{\text{long-range}}, \qquad \text{(F21)}$$

where $\theta > 0$ is a chosen range separation parameter (with unit of inverse length, and often called "ω"). Here $\mathrm{erf}(z)$ and $\mathrm{erfc}(z)$ denote the 'Error Function' and 'Complementary Error Function', which are defined as

$$\mathrm{erf}(z) := \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2}\,\mathrm{d}t, \qquad \text{(F22)}$$

$$\mathrm{erfc}(z) := 1 - \mathrm{erf}(z). \qquad \text{(F23)}$$

This split is particularly prevalent in various range-separated density functional theory methods, where the split is used for one of two distinct reasons. First, in range-separated hybrid functionals (e.g., Refs. 31 and 32) to correct the asymptotic long-range behavior of Kohn-Sham orbitals relevant for excitation energies, by using only long-range exact exchange. Secondly, in the reverse, to save computational cost in large molecules, by using only short-range exact exchange,[30] because many properties of chemical systems turn out to be rather insensitive to the long-range part of the Coulomb interaction.[33,34] Also, other kinds of attenuation strategies have been discussed.[35,36]

We only need to consider the $\tilde{G}_m(\rho,T)$ construction for one of the terms on the rhs of Eq. (F21), because the other one is trivially obtained as difference from an un-attenuated Coulomb kernel. We consider

$$\tilde{K}(r_{12}) := \frac{\mathrm{erf}(\theta \cdot r_{12})}{r_{12}} \qquad \text{(F24)}$$

and aim to construct the corresponding $\tilde{G}_0(\rho,T)$. To this end, note that

$$\mathrm{erf}(\theta \cdot r_{12}) = \frac{2}{\sqrt{\pi}} \int_0^{(\theta \cdot r_{12})} e^{-t^2}\,\mathrm{d}t \qquad \text{(F25)}$$

$$= \frac{2 r_{12}}{\sqrt{\pi}} \int_0^{\theta} e^{-u^2 r_{12}^2}\,\mathrm{d}u \qquad \text{(F26)}$$

$$= \frac{r_{12}}{\sqrt{\pi}} \int_0^{\theta^2} \frac{1}{\sqrt{\omega}} e^{-\omega r_{12}^2}\,\mathrm{d}\omega. \qquad \text{(F27)}$$

Into Eq. (F22) we insert $z := \theta \cdot r_{12}$ to yield Eq. (F25); then change the integration variable via $t =: u \cdot r_{12}$ to yield Eq. (F26); then insert $\omega := \sqrt{u}$ to yield Eq. (F27). Consequently, Eq. (F24) can be re-expressed as

$$\tilde{K}(r_{12}) = \frac{\mathrm{erf}(\theta \cdot r_{12})}{r_{12}} = \frac{1}{\sqrt{\pi}} \int_0^{\theta^2} \frac{1}{\sqrt{\omega}} e^{-\omega r_{12}^2}\,\mathrm{d}\omega. \qquad \text{(F28)}$$

As noted earlier, Eq. (D11) implies that for any kernel $K(r_{12})$, the associated kernel function $G_0(\rho,T)$ is obtained as a linear transformation of this $K_{12}(\mathbf{r})$. Consequently, we can obtain the kernel function $\tilde{G}_0(\rho,T)$ associated with $\tilde{K}(r_{12})$ from Eq. (F28) via

$$\tilde{G}_0(\rho,T) = \frac{1}{\sqrt{\pi}} \int_0^{\theta^2} \frac{1}{\sqrt{\omega}} G_0^{\mathrm{gauss}(\omega)}(\rho,T)\,\mathrm{d}\omega \qquad \text{(F29)}$$

where the quantity

$$G_0^{\mathrm{gauss}(\omega)}(\rho,T) = \left(\frac{\pi}{\rho+\omega}\right)^{3/2} e^{-\left(\frac{\omega}{\rho+\omega}\right)T} \qquad \text{(F30)}$$

denotes the kernel function for the Gaussian interaction kernel $K(r_{12}) = e^{-\omega r_{12}^2}$ described in Eq. (E9). Inserting this into Eq. (F29) and evaluating the integral, we get

$$\tilde{G}_0(\rho,T) = \frac{1}{\sqrt{\pi}} \int_0^{\theta^2} \sqrt{\frac{1}{\omega}\left(\frac{\pi}{\rho+\omega}\right)^3} \cdot e^{-\left(\frac{\omega}{\omega+\rho}\right)T}\,\mathrm{d}\omega$$

$$= \frac{\pi^{3/2}}{\rho\sqrt{T}} \mathrm{erf}\left(\eta \sqrt{\frac{T}{\eta^2+\rho}}\right). \qquad \text{(F31)}$$

Comparing this result to Eq. (E3), and substituting as intermediate $\kappa := \frac{\eta^2}{\eta^2+\rho}$, we obtain

$$\tilde{G}_0(\rho,T) = \frac{\pi}{\rho} \frac{\sqrt{\pi}}{\sqrt{T}} \mathrm{erf}\left(\sqrt{\left(\frac{\eta^2}{\eta^2+\rho}\right)}\sqrt{T}\right) \qquad \text{(F32)}$$

$$= \sqrt{\kappa} \cdot \left(\frac{\pi}{\rho} \frac{\sqrt{\pi}}{\sqrt{\kappa T}} \mathrm{erf}\left(\sqrt{\kappa T}\right)\right) \qquad \text{(F33)}$$

$$= \sqrt{\frac{\eta^2}{\eta^2+\rho}} \cdot G_0^{\mathrm{coul}}\left(\rho, \left(\frac{\eta^2}{\eta^2+\rho}\right)T\right), \qquad \text{(F34)}$$

where $G_0^{\mathrm{coul}}(\rho,T)$ denotes the Coulomb kernel function from Eq. (E3). Consequently, the integrals over the range-separated Coulomb kernels on the rhs of Eq. (F21) can be easily obtained from regular Coulomb kernel function $G_0^{\mathrm{coul}}(\rho,T)$ by variable transformation. The result Eq. (F34) is equivalent to Eq. (52) in Ref. 27.

Appendix G: Evaluation of the Boys function $F_m(T)$

Eq. (E4) introduced the first Boys function $F_0(T)$, given by

$$F_0(T) := \frac{1}{2}\frac{\sqrt{\pi}}{\sqrt{T}}\operatorname{erf}\left(\sqrt{T}\right). \tag{G1}$$

$F_0(T)$ and its derivatives $F_m(T)$ play an important role in the evaluation of molecular integrals over both raw and attenuated Coulomb potentials. Namely, the $G_m(\rho,T)$ kernel function for interaction potential $K(r_{12}) = \frac{1}{r_{12}}$, is given by[27]

$$\begin{aligned} G_m(\rho,T) &= \left(\frac{1\partial}{\partial T}\right)^m G_0(\rho,T) \\ &= \left(\frac{1\partial}{\partial T}\right)^m \frac{2\pi}{\rho} F_0(T) \\ &=: \frac{2\pi}{\rho} F_m(T). \end{aligned} \tag{G2}$$

Here we denoted the Boys function of order m with $F_m(T)$:

$$\begin{aligned} F_m(T) &= \left(-\frac{\partial}{\partial T}\right)^m F_0(T) \\ &= \left(-\frac{\partial}{\partial T}\right)^m \frac{1}{2}\frac{\sqrt{\pi}}{\sqrt{T}}\operatorname{erf}\left(\sqrt{T}\right). \end{aligned} \tag{G3}$$

Due to its relevance in Coulomb integral evaluation, its analytical properties have been investigated at length (see Ref. 2 ch. 9.8), and several dozen research articles have been written about the efficient and accurate evaluation of $F_m(T)$ in Eq. (G3). For this reason, we do not discuss all approaches proposed, but only refer to Ref. 37 for a recent article also discussing earlier references.

In the following, we will only describe the technique used in our program; this technique is reasonably efficient on modern CPU architectures and obtains results with full double-precision accuracy for arbitrary arguments. An example implementation in C++ is provided in the electronic supporting information to this article, as well as the Python scripts used for computing the data tables used in the C++ code. The techniques described next are by no means unique or unprecedented (indeed, most quantum chemistry programs evaluate the Boys function with similar methods in practice). They are just described for the sake of completeness and transparency.

First, a direct calculation on Eq. (G3) shows that the higher Boys functions $F_m(T)$ with integer $m \geq 1$ fulfill the following “upward” recurrence relation:

$$F_m(T) = \frac{1}{T}\left((2m-1)F_{m-1}(T) - e^{-T}\right). \tag{G4}$$

This relation looks promising at first glance, as it appears to allow computing all higher $F_m(T)$ from $F_0(T)$ via a simple recursion. However, it cannot be used in production-level code directly, because it becomes numerically unstable for either small T or large m—in these cases (some of) the e^{-T} and $(2m-1)F_{m-1}(T)$ terms become equal in magnitude and opposite in sign, leading to a near complete cancellation of significant digits in the subtraction. The corresponding “downward” recurrence formula

$$F_m(T) = \frac{2TF_{m+1}(T) + e^{-T}}{2m+1}. \tag{G5}$$

does not suffer from this problem; however, it requires computing at least a single $F_M(T)$ value, namely, the one for the largest M required, via alternative means. The $F_m(T)$ with $m < M$ can then be obtained by repeatedly appyling Eq. (G5).

While, due to its numerical instability, Eq. (G4) cannot be directly used in production code, it is entirely feasible to employ it in conjunction with *arbitrary-precision arithmetic* to *precompute and tabulate* full-precision values of $F_m(T)$ on a fixed grid of points $T_i = i\cdot h$ (where h is a constant step width and i is an integer) and for a given set of $m = 0\ldots M_{\max}$. This is a simple brute force approach; it does not incur any runtime overhead, as the corresponding tables only have to be constructed once. To this end, we precompute and tabulate the values of $F_m(i\cdot h)$ for $m = 0\ldots M_{\max}$ where $M_{\max} = 31$, table step $h = \frac{1}{8}$, and $i = 0,1,\ldots,(N_{\text{tab}}-1)$ (corresponding to $T \leq 12.0$). This is done offline, in a Python script using arbitrary precision arithmethic with 1000 significant decimal digits via the `mpmath` library. The values thus computed are then saved to a static constant $M_{\max}\times N_{\text{tab}}$ array of double precision numbers in a C++ source file (with $M_{\max}$ as the fast dimenion, and N_{tab} as the slow dimension), which is simply included in the compilation of our actual program.

Case 1: T inside the tabulated range ($T \leq 12.0$). With the stored values of $F_m(T_i)$, we can then proceed as follows to compute $F_m(T)$ for any T within the table range and for all $m = 0,1,\ldots,M$ for a given integer M:

- Find the tabulated $T_i = i\cdot h$ which is closest to our given T, by computing $i = \text{floor}(T/h + 0.5)$.
- Compute $\delta T := T - T_i$, the difference between T and the closest tabulated point $T_i = i\cdot h$.
- Approximate the value of $F_M(T)$ for the highest required integer M via a $P = 8$'th order Taylor expansion of $F_M(T)$ around the tabulated $T_i = i\cdot h$. As $F_{M+k}(T_i) = \left(-\frac{\partial}{\partial T}\right)^k F_M(T_i)$ (see Eq. (G3)), all the required information is contained in the precomputed table:

$$\begin{aligned} F_M(T) &\approx \sum_{k=0}^{P} \frac{(-\delta T)^k}{k!} F_{M+k}(T_i) \\ &= p_0 - \delta T\cdot\Big(p_1 - \delta T\cdot\Big(\frac{p_2}{2!} - \delta T\cdot \\ &\quad \Big(\frac{p_3}{3!} - \delta T\cdot\Big(\frac{p_4}{4!} - \delta T\cdot\Big(\ldots\Big)\Big)\Big)\Big)\Big). \end{aligned} \tag{G6}$$

In the second line, the coefficients $p_k = F_{(M+k)}(T_i)$ denote the expansion coefficients taken from the embedded Boys-function data table at index $i\cdot(M_{\max}+1)+$

$M+k$. With the memory layout described, they lie linearly in memory. $k! = k\cdot(k-1)\cdot(k-2)\cdot\ldots\cdot 1$ denotes the factorial. We also tabulate the $\{\frac{1}{k!}\}$ prefactors.

- Compute $F_m(T)$ for the remaining $m = (M-1), (M-2), \ldots, 1, 0$ from $F_M(T)$ via the (numerically stable) downward recurrence formula Eq. (G5). In this, we pre-tabulate the constant factors $1/(2m+1)$ to avoid divisions at runtime, and compute the value of e^{-T} only once.

With these combinations, $F_m(T)$ for $m = 0\ldots M$ and $i < N_{\text{tab}}$ (i.e., technically $T < 12.0 + h/2$) can be computed with only a single evaluation of an exponential function (namely, e^{-T}) and otherwise only linear-memory table lookups, and floating point multiplications and additions. In practice, the evaluation of e^{-T} is the most expensive partial operation. Additionally, with the given dimensions of N_{tab} and $M_{\max}$, the entire pretabulated table has a size of only 24.25 kiB, and therefore fits into the L1 data cache of most current computing architectures. As the lower derivatives are obtained from the exact recurrence relation Eq. (G5), furthermore all $F_m(T)$ for a given T and $m = 0,\ldots,M$ are automatically consistent with each other, even if there should be a residual approximation error for $F_M(T)$.

Case 2: Intermediate range ($12.0 < T \le 36.0$). If a given T is outside of the tabulated range, we proceed as follows. Let us define the function

$$S(x) = x\,\mathrm{erfc}(x)\,e^{x^2}. \qquad \text{(G7)}$$

With this and $\mathrm{erf}(x) + \mathrm{erfc}(x) = 1$, we have

$$\mathrm{erf}(x) = 1 - \frac{S(x)\,e^{-x^2}}{x} \qquad \text{(G8)}$$

$$\Rightarrow \mathrm{erf}(\sqrt{T}) = 1 - \frac{S(\sqrt{T})e^{-T}}{\sqrt{T}}. \qquad \text{(G9)}$$

This can be inserted into $F_0(T)$ from Eq. (G1),

$$F_0(T) = \frac{1}{2}\frac{\sqrt{\pi}}{\sqrt{T}}\,\mathrm{erf}\left(\sqrt{T}\right) \qquad \text{(G10)}$$

where $\mathrm{erf}(\sqrt{T})$ is evaluated by means of Eq. (G9).

The significance of this construction is that $S(x)$ is a well-behaved function, which can be accurately approximated as

$$S(x) \approx R(x) \qquad \left(\text{for } \sqrt{T_{\min}} \le x \le \sqrt{T_{\max}}\,\right) \qquad \text{(G11)}$$

using a low-order polynomial $R(x)$:

$$R(x) := c_0 + x\cdot(c_1 + x\cdot(c_2 + x\cdot(c_3 + x\cdot(\ldots)))).$$

Concretely, we obtained the following approximation of $\mathrm{erf}(\sqrt{T})$ by inserting Eq. (G11) into Eq. (G9) for $x = \sqrt{T}$:

$$\mathrm{erf}(x) = 1 - \frac{e^{-x^2}}{x}R(x) \qquad \text{(G12)}$$

The numerical coefficients $\{c_k\}$ are given by

$$\begin{aligned}
c_0 &= +1.6057555044955896\cdot 10^{-01}\\
c_1 &= +4.8560557460742598\cdot 10^{-01}\\
c_2 &= -3.1344877124388754\cdot 10^{-01}\\
c_3 &= +1.3230181851498005\cdot 10^{-01}\\
c_4 &= -3.8962506318633815\cdot 10^{-02}\\
c_5 &= +8.1949930595442508\cdot 10^{-03}\\
c_6 &= -1.2313078185300554\cdot 10^{-03}\\
c_7 &= +1.2945938765877632\cdot 10^{-04}\\
c_8 &= -9.0663179364002847\cdot 10^{-06}\\
c_9 &= +3.8045328994203999\cdot 10^{-07}\\
c_{10} &= -7.2445454685049530\cdot 10^{-09}.
\end{aligned}$$

In the range of $T = 12\ldots 36$, this approximation has a numerical error of $\le 1.16\cdot 10^{-16}$ for $\mathrm{erf}(\sqrt{T})$ (Eq. (G12)).

The other required $F_m(T)$ for $m = 1, 2, \ldots, M$ are then obtained by the upward recurrence relation Eq. (G4). For large T, which is the case invoked here, this relation is numerically reasonably stable.

Case 3: Long range ($T > 36.0$). We proceed with the same code as for the intermediate range, but make two adjustments to reduce numerical effort:

- For $T > 36$, the factor e^{-T} in Eq. (G9) is so small that we can approximate $\mathrm{erf}(\sqrt{T}) = 1$ in Eq. (G10). We therefore need not evaluate Eq. (G12) to compute $\mathrm{erf}(\sqrt{T})$, but can just set its result variable to 1.
- However, for $36 \le T \le 36 + 2M$, we still need to evaluate e^{-T}, as it may become relevant in the upward recurrence Eq. (G4) to compute higher $F_m(T)$ from $F_0(T)$.
- For $T \le 36 + 2M$, we can then also set $e^{-T} = 0$ for purposes of the upward recurrence.

In practice, these aspects are handled by having separate variables representing $\exp(-T)$ and $\mathrm{erf}(\sqrt{T})$, which are initialized to their boundary values (0 and 1, respectively), and only explicitly evaluated if required.

References

[1] Schlegel, H. B.; Frisch, M. J. Transformation between Cartesian and pure spherical harmonic Gaussians. *Int. J. Quantum. Chem.* **1995**, *54*, 83–87.

[2] Helgaker, T.; Jørgensen, P.; Olsen, J. *Molecular Electronic Structure Theory*; Wiley: Chichester, 2000.

[3] Weniger, E. J.; Steinborn, E. O. New representations for the spherical tensor gradient and the spherical delta function. *J. Math. Phys.* **1983**, *24*, 2553–2563.

[4] Weniger, E. J. The spherical tensor gradient operator. *Collect. Czechoslov. Chem. Commun.* **2005**, *70*, 1225–1271.

[5] Reine, S.; Tellgren, E.; Helgaker, T. A unified scheme for the calculation of differentiated and undifferentiated molecular integrals over solid-harmonic Gaussians. *Phys. Chem. Chem. Phys.* **2007**, *9*, 4771–4779.

[6] Giese, T. J.; York, D. M. Contracted auxiliary Gaussian basis integral and derivative evaluation. *J. Chem. Phys.* **2008**, *128*, 064104.

[7] Giese, T. J.; York, D. M. Spherical tensor gradient operator method for integral rotation: A simple, efficient, and extendable alternative to Slater–Koster tables. *J. Chem. Phys.* **2008**, *129*, 016102.

[8] Golze, D.; Benedikter, N.; Iannuzzi, M.; Wilhelm, J.; Hutter, J. Fast evaluation of solid harmonic Gaussian integrals for local resolution-of-the-identity methods and range-separated hybrid functionals. *J. Chem. Phys.* **2017**, *146*, 034105.

[9] Dunlap, B. I. Three-center Gaussian-type-orbital integral evaluation using solid spherical harmonics. *Phys. Rev. B* **1990**, *42*, 1127.

[10] Dunlap, B. I. Direct quantum chemical integral evaluation. *Int. J. Quantum Chem.* **2001**, *81*, 373–383.

[11] Hu, A.; Dunlap, B. I. Three-center molecular integrals and derivatives using solid harmonic Gaussian orbital and Kohn-Sham potential basis sets. *Can. J. Phys.* **2013**, *91*, 907–915.

[12] Dunlap, B. I. Angular momentum in solid-harmonic-Gaussian integral evaluation. *J. Chem. Phys.* **2003**, *118*, 1036–1043.

[13] Ahlrichs, R. Efficient evaluation of three-center two-electron integrals over Gaussian functions. *Phys. Chem. Chem. Phys.* **2004**, *6*, 5119–5121.

[14] Babin, V.; Leforestier, C.; Paesani, F. Development of a “first principles” water potential with flexible monomers: Dimer potential energy surface, VRT spectrum, and second virial coefficient. *J. Chem. Theory Comput.* **2013**, *9*, 5395–5403.

[15] Cisneros, G. A.; Piquemal, J.-P.; Darden, T. A. Generalization of the Gaussian electrostatic model: Extension to arbitrary angular momentum, distributed multipoles, and speedup with reciprocal space methods. *J. Chem. Phys.* **2006**, *125*, 184101.

[16] Grimme, S.; Bannwarth, C.; Shushkov, P. A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (Z=1–86). *J. Chem. Theory Comput.* **2017**, *13*, 1989–2009.

[17] Bannwarth, C.; Ehlert, S.; Grimme, S. GFN2-xTB—An accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *J. Chem. Theory Comput.* **2019**, *15*, 1652–1671.

[18] Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297–3305.

[19] Weigend, F.; Furche, F.; Ahlrichs, R. Gaussian basis sets of quadruple zeta valence quality for atoms H–Kr. *J. Chem. Phys.* **2003**, *119*, 12753.

[20] Neese, F.; Valeev, E. F. Revisiting the atomic natural orbital approach for basis sets: Robust systematic basis sets for explicitly correlated and conventional correlated ab initio methods? *J. Chem. Theory Comput.* **2010**, *7*, 33–43.

[21] Hill, J. G.; Mazumder, S.; Peterson, K. A. Correlation consistent basis sets for molecular core-valence effects with explicitly correlated wave functions: The atoms B–Ne and Al–Ar. *J. Chem. Phys.* **2010**, *132*, 054108.

[22] Schuchardt, K. L.; Didier, B. T.; Elsethagen, T.; Sun, L.; Gurumoorthi, V.; Chase, J.; Li, J.; Windus, T. L. Basis set exchange: a community database for computational sciences. *J. Chem. Inf. Model* **2007**, *47*, 1045–1052.

[23] Pritchard, B. P.; Altarawy, D.; Didier, B.; Gibson, T. D.; Windus, T. L. New Basis Set Exchange: An Open, Up-to-Date Resource for the Molecular Sciences Community. *J. Chem. Inf. Model* **2019**, *59*, 4814–4820.

[24] Hollman, D. S.; Schaefer, H. F.; Valeev, E. F. A tight distance-dependent estimator for screening three-center Coulomb integrals over Gaussian basis functions. *J. Chem. Phys.* **2015**, *142*, 154106.

[25] McMurchie, L. E.; Davidson, E. R. One-and two-electron integrals over Cartesian Gaussian functions. *J. Comput. Phys.* **1978**, *26*, 218–231.

[26] Gill, P. M.; Head-Gordon, M.; Pople, J. A. An efficient algorithm for the generation of two-electron repulsion integrals over gaussian basis functions. *Int. J. Quantum Chem.* **1989**, *36*, 269–280.

[27] Ahlrichs, R. A simple algebraic derivation of the Obara-Saika scheme for general two-electron interaction potentials. *Phys. Chem. Chem. Phys.* **2006**, *8*, 3072–3077.

[28] May, A. J.; Manby, F. R. An explicitly correlated second order Møller-Plesset theory using a frozen Gaussian geminal. *J. Chem. Phys.* **2004**, *121*, 4479–4485.

[29] Höfener, S.; Bischoff, F. A.; Glöß, A.; Klopper, W. Slater-type geminals in explicitly-correlated perturbation theory: application to n-alkanols and analysis of errors and basis-set requirements. *Phys. Chem. Chem. Phys.* **2008**, *10*, 3390–3399.

[30] Dombroski, J. P.; Taylor, S. W.; Gill, P. M. KWIK: Coulomb energies in O (N) work. *J. Phys. Chem* **1996**, *100*, 6272–6276.

[31] Yanai, T.; Tew, D. P.; Handy, N. C. A new hybrid exchange–correlation functional using the Coulomb-attenuating method (CAM-B3LYP). *Chem. Phys. Lett.* **2004**, *393*, 51–57.

[32] Refaely-Abramson, S.; Sharifzadeh, S.; Govind, N.; Autschbach, J.; Neaton, J. B.; Baer, R.; Kronik, L. Quasiparticle spectra from a nonempirical optimally tuned range-separated hybrid density functional. *Phys. Rev. Lett.* **2012**, *109*, 226405.

[33] Adamson, R. D.; Dombroski, J. P.; Gill, P. M. Chemistry without Coulomb tails. *Chem. Phys. Lett.* **1996**, *254*, 329–336.

[34] Gill, P. M.; Adamson, R. D.; Pople, J. A. Coulomb-attenuated exchange energy density functionals. *Mol. Phys.* **1996**, *88*, 1005–1009.

[35] Gill, P. M.; Adamson, R. D. A family of attenuated Coulomb operators. *Chem. Phys. Lett.* **1996**, *261*, 105–110.

[36] Lee, A. M.; Taylor, S. W.; Dombroski, J. P.; Gill, P. M. Optimal partition of the Coulomb operator. *Phys. Rev. B* **1997**, *55*, 3233.

[37] Weiss, A. K.; Ochsenfeld, C. A rigorous and optimized strategy for the evaluation of the Boys function kernel in molecular electronic structure theory. *J. Comput. Chem.* **2015**, *36*, 1390–1398.