

Efficient evaluation of the geometrical first derivatives of three-center Coulomb integrals

Gyula Samu, and Mihály Kállay

Citation: *The Journal of Chemical Physics* **149**, 124101 (2018); doi: 10.1063/1.5049529

View online: <https://doi.org/10.1063/1.5049529>

View Table of Contents: <http://aip.scitation.org/toc/jcp/149/12>

Published by the [American Institute of Physics](#)

PHYSICS TODAY

WHITEPAPERS

ADVANCED LIGHT CURE ADHESIVES

Take a closer look at what these environmentally friendly adhesive systems can do

READ NOW

PRESENTED BY
 **MASTERBOND**
ADHESIVES | SEALANTS | COATINGS

Efficient evaluation of the geometrical first derivatives of three-center Coulomb integrals

Gyula Samu^{a)} and Mihály Kállay^{b)}

MTA-BME Lendület Quantum Chemistry Research Group, Department of Physical Chemistry and Materials Science, Budapest University of Technology and Economics, P.O. Box 91, H-1521 Budapest, Hungary

(Received 23 July 2018; accepted 7 September 2018; published online 24 September 2018)

The calculation of the geometrical derivatives of three-center electron repulsion integrals (ERIs) over contracted spherical harmonic Gaussians has been optimized. We compared various methods based on the Obara–Saika, McMurchie–Davidson, Gill–Head–Gordon–Pople, and Rys polynomial algorithms using Cartesian, Hermite, and mixed Gaussian integrals for each scheme. The latter ERIs contain both Hermite and Cartesian Gaussians, and they combine the advantageous properties of both types of basis functions. Furthermore, prescreening of the ERI derivatives is discussed, and an efficient approximation of the Cauchy–Schwarz bound for first derivatives is presented. Based on the estimated operation counts, the most promising schemes were implemented by automated code generation, and their relative performances were evaluated. We analyzed the benefits of computing all of the derivatives of a shell triplet simultaneously compared to calculating them just for one degree of freedom at a time, and it was found that the former scheme offers a speedup close to an order of magnitude with a triple-zeta quality basis when appropriate prescreening is applied. In these cases, the Obara–Saika method with Cartesian Gaussians proved to be the best approach, but when derivatives for one degree of freedom are required at a time the mixed Gaussian Obara–Saika and Gill–Head–Gordon–Pople algorithms are predicted to be the best performing ones. *Published by AIP Publishing.* <https://doi.org/10.1063/1.5049529>

I. INTRODUCTION

Analytical derivative methods are essential tools of quantum chemistry^{1,2} since many of the most important molecular properties are related to energy derivatives. The geometrical gradient of the energy is necessary to find equilibrium structures of chemical systems, which makes it especially of interest. The speed of the evaluation of electron repulsion integral (ERI) derivatives is an important factor in the efficiency of such gradient calculations.³

When Cartesian Gaussian basis functions are applied, the ERI derivatives can be computed in a straightforward manner invoking the differentiation rule of the Gaussians after the evaluation of the ERIs. More elaborate techniques were also investigated in the literature that proved useful in accelerating four-center ERI derivative calculations. With the Rys polynomial method,^{4–7} where ERIs are calculated as the weighted sum of two-dimensional integral products, one can differentiate these intermediates and calculate the ERI derivatives directly.^{8,9} Using the McMurchie–Davidson (MD) expansion,¹⁰ the cost of the derivative integral computation can be reduced by differentiating the Hermite integrals and the corresponding expansion coefficients with respect to variables related to nuclear coordinates and then, after an assembly, transforming these derivatives into the desired quantities.¹¹ The Gill–Head–Gordon–Pople (GHP) algorithm,^{12,13} which

transforms the Hermite integrals of the MD scheme into ERIs over Cartesian overlap distributions using contracted level recursions, can be used to compute derivatives as well.^{14,15} The accompanying coordinate expansion and transferred recurrence relation method,¹⁶ efficient for the computation of ERIs with heavily contracted basis sets, was also extended to serve derivative calculation purposes.¹⁷ It was shown in Ref. 18 that Hermite Gaussians are also applicable to ERI and ERI derivative calculations when the angular parts of the basis functions are spherical harmonics, and the Obara–Saika¹⁹ (OS) recurrences for Hermite ERIs were also presented. The use of Hermite Gaussians offers the advantage of a more simple calculation of derivatives from the ERIs compared to Cartesian Gaussians, although the horizontal recurrence relation²⁰ (HRR) involves four terms instead of the original two. The translational and rotational invariance of ERIs and their application to derivative evaluations were also investigated by several authors.^{21–27}

Density fitting (DF) is an established procedure to efficiently and accurately approximate the four-center ERIs required by most quantum chemical methods. If the Coulomb metric is applied for DF,^{28–32} then three- and two-center ERI derivatives are necessary to evaluate the derivative of the energy. Here the efficiency is determined by the speed of the three-center ERI derivative calculation, for which, in principle, all of the above mentioned methods can be adapted. While this has not been done so far, several techniques have been developed for the evaluation of three-center ERIs, which are also useful in accelerating the computation of their derivatives.

^{a)}Electronic mail: gysamu@mail.bme.hu

^{b)}Electronic mail: kallay@mail.bme.hu

Ahlrichs showed³³ that the OS method can be considerably simplified if the fitting functions have spherical harmonic angular parts. In our recent work, we proved³⁴ that the computation of the two-dimensional integrals of the Rys method also becomes simpler in this case. The MD and GHP algorithms greatly benefit from the use of spherical harmonic Hermite Gaussians for the ket side of three-center ERIs since the time consuming transformation of Hermite Gaussians into the Cartesian ones can be avoided for the fitting functions.¹⁸ We should also mention the work of Köster, who combined the OS, MD, and GHP methods for three-center ERIs³⁵ and developed new recurrence formulas for fitting basis sets containing uncontracted Hermite Gaussian functions.³⁶ Efficient implementation of three-center ERIs and ERI derivatives,^{37–40} as well as options for prescreening these quantities,^{41–43} has also received attention.

It is worth mentioning that the explicit evaluation of the three-center ERI derivative list is not mandatory for the computation of the derivative of the Coulomb-energy in a direct self-consistent field (SCF) calculation. Utilizing the J-engine scheme and related ideas,^{36,44} it is possible to only partially calculate the integral-derivatives and, by reverse operations, transform the density matrix into a form appropriate for the contraction with intermediate differentiated ERIs rather than with those over atomic orbitals (AOs). The approach is especially advantageous for Kohn–Sham SCF gradients, but it can also be beneficial for DF Hartree–Fock or hybrid density functional calculations if the J-engine is applied along with reduced-cost schemes^{45–51} for the computation of the Fock-exchange contribution to the gradient. In the latter cases, however, three-center ERI derivatives still have to be computed. In addition, the ideas of the J-engine approach cannot be efficiently utilized for correlated gradient calculations. We also note that, for basis functions with sufficiently small overlap, the exact calculation of ERI derivatives is not necessary. Here significant speedups can be achieved by the application of approximations based on multipole or asymptotic expansions.^{52–54} Still, the evaluation of the near-field ERI derivatives demands considerable computational effort, especially in the case of integrals involving high angular momentum functions.

In this study, our goal is to find the most efficient method for evaluating three-center ERI derivatives using contracted spherical harmonic Gaussians. In Sec. II, we summarize the properties of Cartesian and Hermite Gaussian basis functions and three-center ERI derivatives and discuss the adaptation of the OS, MD, GHP, and Rys schemes for calculating the derivatives for Cartesian and Hermite Gaussian integrals. We give new recurrence relations for each method to evaluate mixed Gaussian ERIs, which contain both Cartesian and Hermite Gaussians. The motivation for using these integrals is to benefit from both the easier derivative calculation with Hermite Gaussians and the less expensive Cartesian recurrences. We note that, to our knowledge, the MD, GHP, and Rys methods have not yet been extended to the case of Hermite Gaussian basis functions, and this has been achieved in the present work. We explore schemes for evaluating the ERI derivatives separately for each degree of freedom and also the ones where all of the derivatives are computed at the same time. The latter

approach makes us able to exploit the translational invariance of the integrals and use shared recursion intermediates for the derivative calculations. We also consider different prescreening approaches, introducing an efficient approximation of the Cauchy–Schwarz upper bound for ERI derivatives. Section III presents and discusses floating point operation (FLOP) counts for the considered algorithms. In Sec. IV, implementation details are given, and in Sec. V, we compare the runtime performances of the selected schemes, which we implemented by automated tools, as well as the four prescreening approaches described in Sec. II F. We summarize our conclusions in Sec. VI.

II. THEORY

A. Geometrical derivatives of three-center Coulomb integrals

In this work, we investigate the efficient evaluation of the first-order geometrical derivatives of three-center Coulomb integrals over contracted solid harmonic Gaussian functions using integrals over Cartesian and Hermite Gaussians. The Cartesian Gaussians have the form of

$$G_{IJK}(\mathbf{r}, a, \mathbf{A}) = x_A^I y_A^J z_A^K \exp(-ar_A^2), \quad (1)$$

and the Hermite Gaussians will be defined as¹⁸

$$\tilde{H}_{\tilde{I}\tilde{J}\tilde{K}}(\mathbf{r}, a, \mathbf{A}) = \frac{\partial^{\tilde{L}} \exp(-ar_A^2)}{(2a)^{\tilde{L}} \partial A_x^{\tilde{I}} \partial A_y^{\tilde{J}} \partial A_z^{\tilde{K}}}, \quad (2)$$

where \mathbf{r} denotes the position vector of the electron, \mathbf{A} is the position of the center of the function, a is a constant Gaussian exponent, and r_A is the magnitude of the vector $\mathbf{r}_A = \mathbf{r} - \mathbf{A}$ with x_A being the x component of \mathbf{r}_A . $L = I + J + K$ will be called the angular momentum of the Gaussian, and the vector $\mathbf{L} = (I, J, K)$ will be referred to as its angular momentum vector. A tilde over the angular momentum vector and its components will be used for Hermite Gaussians to distinguish them from the corresponding Cartesian Gaussians. Functions with the same center, exponent, and angular momentum constitute a shell with $(L + 1)(L + 2)/2$ components. The primitive Gaussians are separable in the three Cartesian directions, that is, $G_{IJK} = G_I G_J G_K$ and $\tilde{H}_{\tilde{I}\tilde{J}\tilde{K}} = \tilde{H}_{\tilde{I}} \tilde{H}_{\tilde{J}} \tilde{H}_{\tilde{K}}$, where, for instance, $G_I = x_A^I \exp(-ax_A^2)$ and $\tilde{H}_{\tilde{I}} = \partial^{\tilde{I}} \exp(-ax_A^2) / (2a \partial A_x)^{\tilde{I}}$. Cartesian Gaussians obey the recurrence relations

$$x_A G_I = G_{I+1} \quad (3)$$

and

$$\frac{\partial G_I}{\partial A_x} = 2a G_{I+1} - I G_{I-1}, \quad (4)$$

while the corresponding relations for the Hermite Gaussians are

$$x_A \tilde{H}_{\tilde{I}} = \tilde{H}_{\tilde{I}+\tilde{1}} + \frac{\tilde{I}}{2a} \tilde{H}_{\tilde{I}-\tilde{1}} \quad (5)$$

and

$$\frac{\partial \tilde{H}_{\tilde{I}}}{\partial A_x} = 2a \tilde{H}_{\tilde{I}+\tilde{1}}. \quad (6)$$

We will also make use of the

$$x_B = x_A + X_{AB} \quad (7)$$

equation. In some cases, we will utilize the unscaled Hermite Gaussians, defined as

$$H_{\bar{I}\bar{J}\bar{K}}(\mathbf{r}, a, \mathbf{A}) = \frac{\partial^{\bar{L}} \exp(-a\mathbf{r}_A^2)}{\partial A_x^{\bar{I}} \partial A_y^{\bar{J}} \partial A_z^{\bar{K}}}. \quad (8)$$

A shell of differentiated Cartesian Gaussians can be transformed into a differentiated solid harmonic Gaussian shell consisting of $2L + 1$ functions as

$$\frac{\partial G_{Lm}(\mathbf{r}, a, \mathbf{A})}{\partial A_x} = \sum_{I+J+K=L} C_{IJK}^{Lm} \frac{\partial G_{IJK}(\mathbf{r}, a, \mathbf{A})}{\partial A_x}, \quad (9)$$

where m is an integer satisfying $-L \leq m \leq L$, and the C_{IJK}^{Lm} coefficients in Eq. (9) only depend on the angular momentum vector and the value of L and m .⁵⁵ Equation (9) also holds for undifferentiated Cartesian Gaussians. For the transformation of the Hermite Gaussians, the values of the coefficients are the same as in the Cartesian case,¹⁸ and the simple differentiation rule, Eq. (6), allows us to perform the differentiation and the

transformation in a single step as

$$\frac{\partial G_{Lm}(\mathbf{r}, a, \mathbf{A})}{\partial A_x} = \sum_{\bar{I}+\bar{J}+\bar{K}=\bar{L}} C_{\bar{I}\bar{J}\bar{K}}^{Lm} 2a\tilde{H}_{\bar{I}\bar{J}\bar{K}}(\mathbf{r}, a, \mathbf{A}), \quad (10)$$

where $C_{\bar{I}\bar{J}\bar{K}}^{Lm} = C_{IJK}^{Lm}$. We obtain contracted Gaussians (and differentiated contracted Gaussians analogously) by combining the corresponding elements of shells with different exponents as

$$\chi_{ALm}(\mathbf{r}, \mathbf{A}) = \sum_a G_{Lm}(\mathbf{r}, a, \mathbf{A}) d_{a\chi_A}, \quad (11)$$

where the contraction coefficients $d_{a\chi_A}$ also include the norm of the solid harmonic Gaussian function and are the same for a given shell. The solid harmonic transformation and the contraction of primitives are interchangeable operations even if we use Eq. (10) since in practice the multiplication with $2a$ is performed on earlier stages of the calculation.

Three-center ERIs over primitive Cartesian Gaussian functions are defined as

$$(\mathbf{L}_a \mathbf{L}_b | \mathbf{L}_c) = \iint \frac{G_{I_a J_a K_a}(\mathbf{r}_1, a, \mathbf{A}) G_{I_b J_b K_b}(\mathbf{r}_1, b, \mathbf{B}) G_{I_c J_c K_c}(\mathbf{r}_2, c, \mathbf{C})}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2, \quad (12)$$

where $\mathbf{L}_a = (I_a, J_a, K_a)$ stands for the angular momentum vector, and $L_a = I_a + J_a + K_a$ is the angular momentum of the function with exponent a . Likewise, the ERIs over primitive Hermite Gaussian functions are given as

$$(\tilde{\mathbf{L}}_a \tilde{\mathbf{L}}_b | \tilde{\mathbf{L}}_c) = \iint \frac{\tilde{H}_{I_a J_a K_a}(\mathbf{r}_1, a, \mathbf{A}) \tilde{H}_{I_b J_b K_b}(\mathbf{r}_1, b, \mathbf{B}) \tilde{H}_{I_c J_c K_c}(\mathbf{r}_2, c, \mathbf{C})}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2. \quad (13)$$

The term class will be used to refer to primitive or solid harmonic integrals with the same angular momenta, centers, and exponents; e.g., the primitive class **(111)** contains 27 primitive integrals. For contracted integrals, a class will mean a group of ERIs over the same contracted functions and with the same angular momenta and centers. A shell triplet will refer to all the integrals belonging to the same centers and angular momenta.

The primitive integral where the total angular momentum is zero is of central importance for the evaluation of ERIs. For both types of basis functions, the value of this integral is given explicitly as³³

$$(\mathbf{000}) \equiv (\mathbf{000})^{(0)} = \theta_{pc} \kappa_{ab} F_0(\alpha R_{PC}^2), \quad (14)$$

with

$$\begin{aligned} \kappa_{ab} &= \exp(-\mu R_{AB}^2), & \theta_{pc} &= \frac{2\pi^{5/2}}{pc\sqrt{p+c}}, \\ \mu &= \frac{ab}{a+b}, & \mathbf{P} &= \frac{a\mathbf{A} + b\mathbf{B}}{p}, \\ \mathbf{R}_{AB} &= \mathbf{A} - \mathbf{B}, & \mathbf{R}_{PC} &= \mathbf{P} - \mathbf{C}, \\ p &= a + b, & \alpha &= \frac{pc}{p+c} \end{aligned}$$

and F_n being the Boys function of order n , defined as

$$F_n(x) = \int_0^1 t^{2n} \exp(-xt^2) dt. \quad (15)$$

Integrals $(\mathbf{000})^{(n)}$ calculated similar to Eq. (14) with the order of the Boys function not necessarily being zero serve as the starting values for the OS,¹⁹ MD,¹⁰ and GHP¹² schemes for the evaluation of the integrals of Eqs. (12) and (13) and their derivatives.

Generally, differentiation of the integrals defined by Eqs. (12) and (13) results in a linear combination of integrals over differentiated basis functions given by Eqs. (4) and (6) as

$$\frac{\partial (\mathbf{L}_a \mathbf{L}_b | \mathbf{L}_c)}{\partial A_x} = \left(\frac{\partial \mathbf{L}_a}{\partial A_x} \mathbf{L}_b | \mathbf{L}_c \right) + \left(\mathbf{L}_a \frac{\partial \mathbf{L}_b}{\partial A_x} | \mathbf{L}_c \right) + \left(\mathbf{L}_a \mathbf{L}_b | \frac{\partial \mathbf{L}_c}{\partial A_x} \right). \quad (16)$$

In the following, we will utilize the translational invariance of the integrals,

$$\left(\frac{\partial \mathbf{L}_a}{\partial A_x} \mathbf{L}_b | \mathbf{L}_c \right) + \left(\mathbf{L}_a \frac{\partial \mathbf{L}_b}{\partial B_x} | \mathbf{L}_c \right) + \left(\mathbf{L}_a \mathbf{L}_b | \frac{\partial \mathbf{L}_c}{\partial C_x} \right) = 0. \quad (17)$$

First of all, it trivially follows that Eq. (16) evaluates to zero if the three functions are located on the same center. Rearranging Eq. (17), we get

$$\left(\frac{\partial \mathbf{L}_a}{\partial A_x} \mathbf{L}_b | \mathbf{L}_c \right) + \left(\mathbf{L}_a \frac{\partial \mathbf{L}_b}{\partial B_x} | \mathbf{L}_c \right) = - \left(\mathbf{L}_a \mathbf{L}_b | \frac{\partial \mathbf{L}_c}{\partial C_x} \right), \quad (18)$$

from which two further things are apparent. First, if two of the three centers coincide, differentiating with respect to the coordinates of this center does not require the evaluation of the sum in Eq. (16), but rather we can differentiate the third function and take the negative of the result. This does not

necessarily reduce the operation count since performing the differentiation for this second center can be more expensive than to do so for the functions centered on the first one. However, for medium-sized and larger systems, such integrals are much less numerous than the ones with three distinct centers. For such shell triplets with coinciding centers, Eq. (18) only makes an optimized code-generated implementation simpler since one does not need to have separate subroutines to treat these cases. Second and more importantly, if we wish to produce the derivatives for the shell triplets with respect to all three centers, it is sufficient to perform the calculations only for two of the centers for a given shell triplet, and the derivatives for the third (most expensive) one can be constructed by using Eq. (18). The classes required for these two derivatives can be built up in the same recursive process, reducing the redundancy of the calculation. However, proceeding in such a way is not always the preferred choice. For example, producing derivatives with respect to a couple of degrees of freedom at a time may be advantageous for a coarse-grained parallelization scheme where the derivatives with respect to the coordinates of a center are evaluated on a separate node. The same holds when the solution of response equations is necessary for each degree of freedom, as is the case for the calculation of second derivatives, and we wish to avoid the storage and sorting of the integral-derivatives. For these reasons, we will investigate both approaches.

Utilizing Eq. (18), we only need to explicitly calculate 6 of the 9 derivatives of a three-center ERI. When Cartesian Gaussians are used, the classes $([L_a + \mathbf{1}_x]L_b|L_c)$, $([L_a - \mathbf{1}_x]L_b|L_c)$, $(L_a[L_b + \mathbf{1}_x]|L_c)$, and $(L_a[L_b - \mathbf{1}_x]|L_c)$, for example, have to be evaluated to construct the necessary derivatives. Kahn has shown²⁴ that, from the rotational invariance of the integrals, it is possible to recover all of the derivatives of an $(L_aL_b|L_c)$ integral by just computing, for example, $\partial(L_aL_b|L_c)/\partial A_x$, $\partial(L_aL_b|L_c)/\partial A_y$, $\partial(L_aL_b|L_c)/\partial B_z$, and three other auxiliary integrals which are linear combinations of the ERIs in the $(L_aL_b|L_c)$ class. The remaining derivatives can be calculated by solving three simultaneous linear equations, optionally at the contracted level. The possible advantage is that not all components of the classes with increased angular momenta have to be evaluated. The approach, however, has drawbacks. It is required to compute the extra class $(L_aL_b|L_c)$. When the HRR is applied, as is often the case, these ERIs appear as intermediates for $(L_a[L_b + \mathbf{1}_x]|L_c)$. However, this fact cannot be exploited if the buildup of the final angular momenta takes place at the contracted level since according to Eq. (4) the class with incremented L_b will be scaled by $2b$, meaning that the class required for the auxiliary integrals has to be computed by a separate recursion. Furthermore, none of the functions of the $(L_aL_b|L_c)$ ERIs can be transformed into the spherical harmonic Gaussian basis until the auxiliary integrals have been formed, while performing these operations on earlier intermediates is known⁷ to enhance the speed of the calculation. This also hinders the exploitation of $(L_aL_b|L_c)$ as an intermediate of another class. The formation of the auxiliary integrals is also relatively expensive, each requiring five additions and six multiplications. The solution of the system of linear equations is also considerably more expensive than the application of Eq. (18). Finally, the

advantage of not evaluating every component of the more expensive classes becomes less significant with the increase of the angular momenta. For example, when we wish to compute every $\partial(L_aL_b|L_c)/\partial B_z$, we need the same number of components from the $(L_a[L_b + \mathbf{1}_x]|L_c)$ class as the number of ERIs in $(L_aL_b|L_c)$, with the ratio of the number of the components in the two classes getting smaller with increasing L_b . Because of these considerations, we only deal with the invariance property defined by Eq. (17) in this work.

In the following, L_a will denote the final angular momentum we wish to build up on the function with exponent a . l_a will be used to represent the angular momenta of intermediate classes, that is, classes that are required by the recursions or other transformations to create the final angular momentum. The notation will be similar for the other two functions. The index ranges for the intermediates reported in this paper were determined on the basis of a method outlined in the [supplementary material](#). Even though the presentation suggests that the lower limits of the ranges can take negative values, for brevity, we will not use the notation $\max(0, l)$ for these cases but assume that the angular momenta are greater than or equal to zero. The notation for the various investigated methods will display the basic algorithm, the applied basis function, the center the coordinates of which we differentiate with respect to, and an additional index if we inspect different approaches for a given problem. For example, $\text{GHP}_{\text{Cart},A2}$ will denote the GHP-based scheme where we use Cartesian Gaussians and differentiate with respect to \mathbf{A} , while the approach to which “2” refers will be explained later. We will omit the indices when we talk about all the possible values for those indices, e.g., OS_{Cart} means all the discussed methods within the OS scheme that apply Cartesian Gaussians. We will use an upper left index “3” to denote schemes where all the derivatives of the shell triplets are produced by a common recursion. For example, ${}^3\text{OS}_{\text{Cart},AB}$ refers to the OS-based scheme using Cartesian Gaussians where we simultaneously evaluate the derivatives with respect to \mathbf{A} and \mathbf{B} by recursion and calculate the \mathbf{C} derivatives via Eq. (18).

B. Obara–Saika recursion

The OS scheme utilizes recurrence relations for auxiliary intermediate integrals defined as

$$\begin{aligned} (L_aL_b|L_c)^{(n)} &= \frac{2}{\pi^{1/2}} \int_0^\infty \iint G_{I_aJ_aK_a}(\mathbf{r}_1, a, \mathbf{A}) \\ &\quad \times G_{I_bJ_bK_b}(\mathbf{r}_1, b, \mathbf{B}) G_{I_cJ_cK_c}(\mathbf{r}_2, c, \mathbf{C}) \\ &\quad \times \exp(-|\mathbf{r}_1 - \mathbf{r}_2|^2 u^2) \left(\frac{u^2}{\alpha + u^2} \right)^n d\mathbf{r}_1 d\mathbf{r}_2 du \end{aligned} \quad (19)$$

to construct the true ERIs with $n = 0$ and with the required angular momenta for the calculation of the differentiated integrals. From now on, superscript (n) will be kept only when it is not equal to zero. For the derivation of the equations for four-center ERIs, we refer to the other work.^{19,55}

An efficient application of the OS method to three-center solid harmonic Gaussian ERIs was developed by Ahlrichs.³³ Our equations will slightly differ from those of Ref. 33 since it will prove useful to work with the starting integrals

$$(\mathbf{0}\mathbf{0}\mathbf{0})^{(n)} \equiv (\mathbf{0})^{(n)} = (-2\alpha)^n \theta_{pc} \kappa_{ab} F_n(\alpha R_{PC}^2). \quad (20)$$

To arrive at the undifferentiated ($L_a L_b L_c$) class, we first apply the vertical recurrence relation (VRR)

$$\begin{aligned} ([l_a + \mathbf{1}_x] \mathbf{0} \mathbf{0})^{(n)} &= X_{PA}(\mathbf{l}_a \mathbf{0} \mathbf{0})^{(n)} + \frac{1}{2p} X_{PC}(\mathbf{l}_a \mathbf{0} \mathbf{0})^{(n+1)} \\ &+ \frac{i_a}{2p} \left(([l_a - \mathbf{1}_x] \mathbf{0} \mathbf{0})^{(n)} \right. \\ &\left. + \frac{1}{2p} ([l_a - \mathbf{1}_x] \mathbf{0} \mathbf{0})^{(n+1)} \right). \end{aligned} \quad (21)$$

Here and from now on, $\mathbf{l}_a + \mathbf{1}_x$ means that the x component of the angular momentum vector \mathbf{l}_a has been increased by one. To reach an $(L_a \mathbf{0} \mathbf{0})^{(n)}$ class, Eq. (21) needs $(\mathbf{0}\mathbf{0}\mathbf{0})^{(m)}$ ERIs for $n \leq m \leq n + L_a$. Next, the simplified vertical recurrence for the ket function³³

$$\begin{aligned} (\mathbf{l}_a \mathbf{0} | [l_c + \mathbf{1}_x])^{(n)} &= -\frac{1}{2c} X_{PC}(\mathbf{l}_a \mathbf{0} | l_c)^{(n+1)} \\ &- \frac{i_a}{4pc} ([l_a - \mathbf{1}_x] \mathbf{0} | l_c)^{(n+1)} \end{aligned} \quad (22)$$

is applied, which requires $(\mathbf{l}_a \mathbf{0} \mathbf{0})^{(n)}$ intermediates for $L_a - L_c \leq l_a \leq L_a$ and $n = L_c$ to construct an $(L_a \mathbf{0} | L_c)$ class. Finally, the HRR

$$(\mathbf{l}_a | [l_b + \mathbf{1}_x] | L_c) = ([l_a + \mathbf{1}_x] | l_b | L_c) + X_{AB}(\mathbf{l}_a | l_b | L_c) \quad (23)$$

is employed, for which $(\mathbf{l}_a \mathbf{0} | L_c)$ -type intermediates for $L_a \leq l_a \leq L_a + L_b$ are needed to build $(L_a L_b | L_c)$ ERIs. We note that the $l_a \rightarrow l_c \rightarrow l_b$ order of building up the angular momenta is not the only possible one, and the electron transfer relation⁵⁶ can also be applied in the evaluation of the ERIs. In our previous study,³⁴ we found that the presented scheme is the most efficient OS-type algorithm for undifferentiated three-center Cartesian ERIs.

The application of this method is straightforward for the derivatives with respect to the coordinates of **A** and **B**, where we use Eqs. (20)–(23) to evaluate the required classes for Eq. (4). This slightly modifies the index ranges; e.g., for $([\partial L_a / \partial A_x] | L_b | L_c)$, the lower and upper limits that depend on L_a decrease and increase by one, respectively. Equation (22) can, however, not be applied if we wish to evaluate $(L_a L_b | [\partial L_c / \partial C_x])$. Instead, we shall use the original VRR equation¹⁹ reduced to the three-center case,

$$\begin{aligned} (\mathbf{l}_a \mathbf{0} | [l_c + \mathbf{1}_x])^{(n)} &= -\frac{1}{2c} X_{PC}(\mathbf{l}_a \mathbf{0} | l_c)^{(n+1)} \\ &- \frac{i_a}{4pc} ([l_a - \mathbf{1}_x] \mathbf{0} | l_c)^{(n+1)} \\ &+ \frac{i_c}{2c} (\mathbf{l}_a \mathbf{0} | [l_c - \mathbf{1}_x])^{(n)} \\ &+ \frac{1}{2c} (\mathbf{l}_a \mathbf{0} | [l_c - \mathbf{1}_x])^{(n+1)}. \end{aligned} \quad (24)$$

For undifferentiated ERIs, the last two terms of Eq. (24) cancel when we transform L_c to the solid harmonic Gaussian basis.³³ Now we are changing to the differentiated solid harmonic Gaussian basis, and from Eqs. (4) and (9), we can see that functions with angular momenta higher and lower than L_c get transformed by the coefficients belonging to L_c ; hence, the cancellation does not take place. The $(\mathbf{l}_a \mathbf{0} | \mathbf{0})^{(n)}$ ERIs required for the calculation of an $(L_a \mathbf{0} | L_c)$ class by using Eq. (24) are

those with the $\text{mod}[L_c, 2] \leq n \leq L_c$ and $L_a - \text{mod}[L_c, 2] - 2[\lfloor n - \text{mod}[L_c, 2] \rfloor / 2] \leq l_a \leq L_a$ index ranges, where $\lfloor x \rfloor$ denotes the integer part of x .

The OS method can be developed for ERIs over Hermite Gaussians as well.¹⁸ The bra side VRR is very similar to its Cartesian counterpart, having the form of

$$\begin{aligned} ([\tilde{l}_a + \tilde{\mathbf{1}}_x] \mathbf{0} \mathbf{0})^{(n)} &= X_{PA}(\tilde{\mathbf{l}}_a \mathbf{0} \mathbf{0})^{(n)} + \frac{1}{2p} X_{PC}(\tilde{\mathbf{l}}_a \mathbf{0} \mathbf{0})^{(n+1)} \\ &+ \frac{\tilde{i}_a}{2p} \left(-\frac{b}{a} ([\tilde{l}_a - \tilde{\mathbf{1}}_x] \mathbf{0} \mathbf{0})^{(n)} \right. \\ &\left. + \frac{1}{2p} ([\tilde{l}_a - \tilde{\mathbf{1}}_x] \mathbf{0} \mathbf{0})^{(n+1)} \right). \end{aligned} \quad (25)$$

The VRR for the ket function is the same as Eq. (22). The HRR for the Hermite ERIs,

$$\begin{aligned} (\tilde{\mathbf{l}}_a | \tilde{l}_b + \tilde{\mathbf{1}}_x | \tilde{L}_c) &= ([\tilde{l}_a + \tilde{\mathbf{1}}_x] | \tilde{l}_b | \tilde{L}_c) + X_{AB}(\tilde{\mathbf{l}}_a | \tilde{l}_b | \tilde{L}_c) \\ &+ \frac{\tilde{i}_a}{2a} ([\tilde{l}_a - \tilde{\mathbf{1}}_x] | \tilde{l}_b | \tilde{L}_c) - \frac{\tilde{i}_b}{2b} (\tilde{\mathbf{l}}_a | [\tilde{l}_b - \tilde{\mathbf{1}}_x] | \tilde{L}_c), \end{aligned} \quad (26)$$

can be obtained with the help of Eqs. (5) and (7). The range for the necessary $(\tilde{\mathbf{l}}_a \mathbf{0} | \tilde{L}_c)$ intermediates for Eq. (26) is $\tilde{L}_a - \tilde{L}_b \leq \tilde{l}_a \leq \tilde{L}_a + \tilde{L}_b$. The ERIs required for **A** and **B** derivatives are computed with the above described scheme with modified index limits; e.g., for $([\partial \tilde{L}_a / \partial A_x] | \tilde{L}_b | \tilde{L}_c)$, the lower and upper limits that depend on \tilde{L}_a increase by one according to Eq. (6). Note that the multiplication with the double of the exponent appearing in Eq. (6) can be built into Eq. (20) if we calculate a derivative with respect to one center at a time. When we differentiate with respect to **C**, the necessary ket VRR equation is analogous to Eq. (24); however, the third term is zero. This is because in the corresponding equation for four-center ERIs¹⁸ this third term is multiplied by $-d/c$, where d is the exponent for the fourth function, and d is zero for three-center integrals. Hence the relation becomes

$$\begin{aligned} (\tilde{\mathbf{l}}_a \mathbf{0} | [\tilde{l}_c + \tilde{\mathbf{1}}_x])^{(n)} &= -\frac{1}{2c} X_{PC}(\tilde{\mathbf{l}}_a \mathbf{0} | \tilde{l}_c)^{(n+1)} \\ &- \frac{\tilde{i}_a}{4pc} ([\tilde{l}_a - \tilde{\mathbf{1}}_x] \mathbf{0} | \tilde{l}_c)^{(n+1)} \\ &+ \frac{\tilde{i}_c}{4c^2} (\tilde{\mathbf{l}}_a \mathbf{0} | [\tilde{l}_c - \tilde{\mathbf{1}}_x])^{(n+1)}, \end{aligned} \quad (27)$$

for which we need $(\tilde{\mathbf{l}}_a \mathbf{0} \mathbf{0})^{(n)}$ -type ERIs to build an $(\tilde{L}_a \mathbf{0} | \tilde{L}_c)$ class for $\tilde{L}_c - \lceil \tilde{L}_c / 2 \rceil \leq n \leq \tilde{L}_c$ and $\tilde{L}_a + \tilde{L}_c - 2n \leq \tilde{l}_a \leq \tilde{L}_a$.

We see that both kinds of Gaussian basis functions have advantageous properties. With Hermite Gaussians, the explicit evaluation of the primitive derivatives can be avoided and the recursion for \tilde{L}_c during the computation of $(\tilde{L}_a \tilde{L}_b | [\partial \tilde{L}_c / \partial C_x])$ is simpler, while for the Cartesian Gaussians the HRR is cheaper, independent of the exponents, and requires a smaller range of intermediates. For higher L_b values, this latter property compensates for the widening of the index ranges due to Eq. (4), while if the differentiated function is of s type, the direct application of Eq. (4) can be avoided similar to the Hermite case. In the following, we exploit that, for first derivatives, only one of the functions has to be Hermite Gaussian to utilize Eq. (6), and we shall investigate how calculating such mixed

Gaussian ERIs affects the complexity of the task. The derivation of the following recurrences can be found in Subsection 1 of the [Appendix](#).

For the **A** derivatives, we choose the first and second bra functions to be a Hermite and a Cartesian Gaussian, respectively, and use Eq. (25) for the bra VRR. Since the choice for the function in the ket side is irrelevant now, we apply Eq. (22) to gain the $(\tilde{l}_a\mathbf{0}|\tilde{l}_c)$ -type intermediates. For the $\tilde{l}_a \rightarrow l_b$ translation, we need a new HRR,

$$(\tilde{l}_a[l_b+\mathbf{1}_x]|\tilde{l}_c) = ([\tilde{l}_a+\tilde{l}_x]l_b|\tilde{l}_c) + X_{AB}(\tilde{l}_al_b|\tilde{l}_c) + \frac{\tilde{l}_a}{2a}([\tilde{l}_a-\tilde{l}_x]l_b|\tilde{l}_c), \quad (28)$$

which is one term less expensive than Eq. (26). In the case of the derivatives for **B**, we reverse the type of the three functions and use Eqs. (21) and (22) for the $(l_a\mathbf{0}|\tilde{l}_c)$ ERIs and then the

$$(l_a[\tilde{l}_b+\tilde{l}_x]|\tilde{l}_c) = ([l_a+\mathbf{1}_x]\tilde{l}_b|\tilde{l}_c) + X_{AB}(l_a\tilde{l}_b|\tilde{l}_c) - \frac{\tilde{l}_b}{2b}(l_a[\tilde{l}_b-\tilde{l}_x]|\tilde{l}_c) \quad (29)$$

HRR is applied, for which the range of the necessary starting intermediates is the same as for Eq. (23). The obvious choice for the **C** derivatives is to use the $(l_al_b|\tilde{l}_c)$ ERIs since this would allow us to employ Eq. (23) for the HRR. The appropriate recurrence to construct the $(l_a\mathbf{0}|\tilde{l}_c)^{(n)}$ -type integrals is

$$\begin{aligned} (l_a\mathbf{0}|\tilde{l}_c+\tilde{l}_x)^{(n)} &= \frac{1}{2c}X_{PC}(l_a\mathbf{0}|\tilde{l}_c)^{(n+1)} \\ &\quad - \frac{i_a}{4pc}([l_a-\mathbf{1}_x]\mathbf{0}|\tilde{l}_c)^{(n+1)} \\ &\quad + \frac{\tilde{l}_c}{4c^2}(l_a\mathbf{0}|\tilde{l}_c-\tilde{l}_x)^{(n+1)}. \end{aligned} \quad (30)$$

Equation (30) is analogous to Eq. (27), but since the function on the first center is now a Cartesian Gaussian, we can use Eq. (23) to build up L_b . In addition to the OS methods for Cartesian (OS_{Cart}) or Hermite (OS_{Herm}) Gaussian ERI derivatives, we will also analyze the new schemes based on these mixed Gaussian ERIs (OS_{Mixed}) and conclude that the use of such mixed integrals is often superior to the pure Cartesian and always to the pure Hermite algorithms. For the case when the derivatives of all three centers are evaluated at the same time, we will compare the purely Cartesian Gaussian based methods (${}^3OS_{\text{Cart}}$) with ${}^3OS_{\text{Herm,AB}}$, ${}^3OS_{\text{Mixed,AC}}$, and ${}^3OS_{\text{Mixed,BC}}$. As

we will see, the ${}^3OS_{\text{Cart,AB}}$ route is very competitive because of the simple form of Eq. (23). The OS-based algorithms are summarized in Table I.

C. McMurchie–Davidson method

The essence of the MD scheme for undifferentiated ERIs is to recover Gaussian overlap distributions (Cartesian or Hermite) from Hermite Gaussian functions, for which two-center Hermite integrals have to be calculated first. Exploiting the translational invariance of two-center integrals (that is, $-\partial/\partial P_x = \partial/\partial C_x$), these can be expressed with one-center ERIs as¹⁸

$$\begin{aligned} (\tilde{l}_p|\tilde{l}_c)^{(n)} &= (2p)^{-\tilde{l}_p}(2c)^{-\tilde{l}_c} \frac{\partial^{\tilde{l}_p+\tilde{l}_c}(\mathbf{0})^{(n)}}{\partial P_x^{\tilde{l}_p} \partial P_y^{\tilde{l}_p} \partial P_z^{\tilde{l}_p} \partial C_x^{\tilde{l}_c} \partial C_y^{\tilde{l}_c} \partial C_z^{\tilde{l}_c}} \\ &= (2p)^{-\tilde{l}_p}(-2c)^{-\tilde{l}_c} \frac{\partial^{\tilde{l}_p+\tilde{l}_c}(\mathbf{0})^{(n)}}{\partial P_x^{\tilde{l}_p+\tilde{l}_c} \partial P_y^{\tilde{l}_p+\tilde{l}_c} \partial P_z^{\tilde{l}_p+\tilde{l}_c}} \\ &= (2p)^{-\tilde{l}_p}(-2c)^{-\tilde{l}_c}(\tilde{l}_p+\tilde{l}_c)^{(n)}. \end{aligned} \quad (31)$$

The one-center quantities on the rightmost side of Eq. (31) can be calculated with the McMurchie–Davidson recurrence¹⁰

$$(\tilde{l}_c+\tilde{l}_p+\tilde{l}_x)^{(n)} = X_{PC}(\tilde{l}_c+\tilde{l}_p)^{(n+1)} + (\tilde{l}_p+\tilde{l}_c)(\tilde{l}_c+\tilde{l}_p-\tilde{l}_x)^{(n+1)}. \quad (32)$$

This relation requires $(\mathbf{0})^{(n)}$ ERIs for $[\tilde{l}_p/2] + \text{mod}[\tilde{l}_p, 2] \leq n \leq \tilde{l}_p$ to evaluate (\tilde{l}_p) integrals. From these, one way to recover the three-center integrals is the use of the expression⁵⁵

$$\begin{aligned} (l_al_b|\tilde{l}_c)^{(n)} &= (-2c)^{-L_c} \sum_{\tilde{l}_p=0}^{l_a+l_b} E_{i_p}^{l_a,l_b} \sum_{\tilde{l}_p=0}^{J_a+J_b} E_{j_p}^{J_a,J_b} \\ &\quad \times \sum_{k_p=0}^{K_a+K_b} E_{k_p}^{K_a,K_b}(\tilde{l}_p+\tilde{l}_c)^{(n)}, \end{aligned} \quad (33)$$

where it has been utilized that for three-center ERIs over solid harmonic Gaussians we only have to expand the bra side since the Hermite Gaussian in the ket side can be readily transformed into the solid harmonic Gaussian basis. The recursive evaluation of the E expansion coefficients that transform \tilde{l}_p functions into l_al_b products is described in Ref. 55, and the coefficients to construct $\tilde{l}_a\tilde{l}_b$ distributions from \tilde{l}_p functions are presented in Ref. 18. It is more efficient³⁴ to use recurrences to transform

TABLE I. Serial number of the equations to be applied to calculate the necessary classes in the various OS-based algorithms. Equation (20) is omitted since its use is common to all routes. Separate recursion denotes the schemes where only derivatives with respect to one center are evaluated, while in the common recursion schemes, derivatives for all the centers are calculated at the same time for a shell triplet.

Separate recursion		Common recursion	
$OS_{\text{Cart,A}}$ and $OS_{\text{Cart,B}}$	(21), (22), (23)	${}^3OS_{\text{Cart,AB}}$	(21), (22), (23)
$OS_{\text{Cart,C}}$	(21), (24), (23)	${}^3OS_{\text{Cart,AC}}$ and ${}^3OS_{\text{Cart,BC}}$	(21), (24), (23)
$OS_{\text{Mixed,A}}$	(25), (22), (28)	${}^3OS_{\text{Herm,AB}}$	(25), (22), (26)
$OS_{\text{Mixed,B}}$	(21), (22), (29)	${}^3OS_{\text{Mixed,AC}}$	(25), (27), (28)
$OS_{\text{Mixed,C}}$	(21), (30), (23)	${}^3OS_{\text{Mixed,BC}}$	(21), (30), (29)
$OS_{\text{Herm,A}}$ and $OS_{\text{Herm,B}}$	(25), (22), (26)		
$OS_{\text{Herm,C}}$	(25), (27), (26)		

the Hermite polynomials in x_P to monomials in x_A and x_B for Cartesian ERIs and to Hermite polynomials in x_A and x_B for integrals over Hermite Gaussians. In the Cartesian case, we work with the recursions⁵⁵

$$(\Omega_{l_a+1_x, l_b}^{\bar{l}_p} | = \bar{i}_p (\Omega_{l_a, l_b}^{\bar{l}_p-1_x} | + X_{PA} (\Omega_{l_a, l_b}^{\bar{l}_p} | + \frac{1}{2p} (\Omega_{l_a, l_b}^{\bar{l}_p+1_x} |, \quad (34)$$

$$(\Omega_{l_a, l_b+1_x}^{\bar{l}_p} | = \bar{i}_p (\Omega_{l_a, l_b}^{\bar{l}_p-1_x} | + X_{PB} (\Omega_{l_a, l_b}^{\bar{l}_p} | + \frac{1}{2p} (\Omega_{l_a, l_b}^{\bar{l}_p+1_x} | \quad (35)$$

(where the unaffected ket side of the integrals has been omitted from the formulas), which use integrals over hybrid functions of the form

$$\Omega_{l_a, l_b}^{\bar{l}_p} = x_A^{i_a} y_A^{j_a} z_A^{k_a} x_B^{i_b} y_B^{j_b} z_B^{k_b} \frac{\partial^{\bar{l}_p} \kappa_{ab} \exp(-pr_p^2)}{\partial P_x^{\bar{l}_p} \partial P_y^{\bar{l}_p} \partial P_z^{\bar{l}_p}}. \quad (36)$$

To get integrals with an $(\Omega_{l_a, 0}^{\bar{l}_p} |$ bra side with Eq. (34), we need starting integrals over $\Omega_{0, 0}^{\bar{l}_p}$ functions for $0 \leq \bar{l}_p \leq L_a$.

We can also work with mixed or Hermite Gaussian integrals in the MD scheme. In the case of Hermite ERIs, the auxiliary functions have the form of

$$\Omega_{l_a, l_b}^{\bar{l}_p} = \frac{\partial^{\bar{l}_p+1_x+1_b} \kappa_{ab} \exp(-pr_p^2)}{(2a)^{\bar{l}_a} \partial A_x^{\bar{l}_a} \partial A_y^{\bar{l}_b} \partial A_z^{\bar{l}_b} (2b)^{\bar{l}_b} \partial B_x^{\bar{l}_b} \partial B_y^{\bar{l}_b} \partial B_z^{\bar{l}_b} \partial P_x^{\bar{l}_p} \partial P_y^{\bar{l}_p} \partial P_z^{\bar{l}_p}}, \quad (37)$$

and the two recurrences (see Subsection 2 of the Appendix) are

$$(\Omega_{l_a+1_x, l_b}^{\bar{l}_p} | = \frac{1}{2p} (\Omega_{l_a, l_b}^{\bar{l}_p+1_x} | + X_{PA} (\Omega_{l_a, l_b}^{\bar{l}_p} | - \frac{\tilde{i}_a}{2p} \frac{b}{a} (\Omega_{l_a-1_x, l_b}^{\bar{l}_p} | + \frac{\tilde{i}_b}{2p} (\Omega_{l_a, l_b-1_x}^{\bar{l}_p} | \quad (38)$$

and

$$(\Omega_{l_a, l_b+1_x}^{\bar{l}_p} | = \frac{1}{2p} (\Omega_{l_a, l_b}^{\bar{l}_p+1_x} | + X_{PB} (\Omega_{l_a, l_b}^{\bar{l}_p} | + \frac{\tilde{i}_a}{2p} (\Omega_{l_a-1_x, l_b}^{\bar{l}_p} | - \frac{\tilde{i}_b}{2p} \frac{a}{b} (\Omega_{l_a, l_b-1_x}^{\bar{l}_p} |. \quad (39)$$

To calculate ERIs over $\Omega_{L_a, \tilde{L}_b}^{\bar{l}_p}$ with Eq. (39), we need $\Omega_{l_a, 0}^{\bar{l}_p}$ integrals for $\tilde{L}_a - \tilde{L}_b \leq \tilde{l}_a \leq \tilde{L}_a$ and $0 \leq \bar{l}_p \leq \tilde{L}_b - (\tilde{L}_a - \tilde{l}_a)$, while the $\Omega_{l_a, 0}^{\bar{l}_p}$ ERIs are gained by applying Eq. (38) starting from $\Omega_{0, 0}^{\bar{l}_p}$ ERIs for $\bar{l}_p \leq \tilde{l}_o \leq \bar{l}_p + \tilde{l}_a$. For the mixed integrals, we use the hybrid functions

$$\Omega_{l_a, l_b}^{\bar{l}_p} = x_B^{i_b} y_B^{j_b} z_B^{k_b} \frac{\partial^{\bar{l}_p+1_x} \kappa_{ab} \exp(-pr_p^2)}{(2a)^{\bar{l}_a} \partial A_x^{\bar{l}_a} \partial A_y^{\bar{l}_b} \partial A_z^{\bar{l}_b} \partial P_x^{\bar{l}_p} \partial P_y^{\bar{l}_p} \partial P_z^{\bar{l}_p}}, \quad (40)$$

$$\Omega_{l_a, l_b}^{\bar{l}_p} = x_A^{i_a} y_A^{j_a} z_A^{k_a} \frac{\partial^{\bar{l}_p+1_b} \kappa_{ab} \exp(-pr_p^2)}{(2b)^{\bar{l}_b} \partial B_x^{\bar{l}_b} \partial B_y^{\bar{l}_b} \partial B_z^{\bar{l}_b} \partial P_x^{\bar{l}_p} \partial P_y^{\bar{l}_p} \partial P_z^{\bar{l}_p}}$$

and the recurrences

$$(\Omega_{l_a+1_x, l_b}^{\bar{l}_p} | = \frac{1}{2p} (\Omega_{l_a, l_b}^{\bar{l}_p+1_x} | + X_{PA} (\Omega_{l_a, l_b}^{\bar{l}_p} | - \frac{\tilde{i}_a}{2p} \frac{b}{a} (\Omega_{l_a-1_x, l_b}^{\bar{l}_p} |, \quad (41)$$

$$(\Omega_{l_a, l_b+1_x}^{\bar{l}_p} | = \frac{1}{2p} (\Omega_{l_a, l_b}^{\bar{l}_p+1_x} | + X_{PB} (\Omega_{l_a, l_b}^{\bar{l}_p} | + \bar{i}_p (\Omega_{l_a, l_b}^{\bar{l}_p-1_x} | + \frac{\tilde{i}_a}{2p} (\Omega_{l_a-1_x, l_b}^{\bar{l}_p} |, \quad (42)$$

$$(\Omega_{l_a+1_x, l_b}^{\bar{l}_p} | = \frac{1}{2p} (\Omega_{l_a, l_b}^{\bar{l}_p+1_x} | + X_{PA} (\Omega_{l_a, l_b}^{\bar{l}_p} | + \bar{i}_p (\Omega_{l_a, l_b}^{\bar{l}_p-1_x} | + \frac{\tilde{i}_b}{2p} (\Omega_{l_a, l_b-1_x}^{\bar{l}_p} |, \quad (43)$$

$$(\Omega_{l_a, l_b+1_x}^{\bar{l}_p} | = \frac{1}{2p} (\Omega_{l_a, l_b}^{\bar{l}_p+1_x} | + X_{PB} (\Omega_{l_a, l_b}^{\bar{l}_p} | - \frac{\tilde{i}_b}{2p} \frac{a}{b} (\Omega_{l_a, l_b-1_x}^{\bar{l}_p} |. \quad (44)$$

Using Eqs. (41) and (42), the index range requirements are the same as for Eqs. (38) and (39), respectively. Reaching $\Omega_{L_a, \tilde{L}_b}^{\bar{l}_p}$ ERIs with Eq. (44) demands integrals over $\Omega_{L_a, 0}^{\bar{l}_p}$ for $0 \leq \bar{l}_p \leq \tilde{L}_b$, while these are evaluated with Eq. (43) from $\Omega_{0, 0}^{\bar{l}_p}$ for $\bar{l}_p - L_a \leq \tilde{l}_o \leq \bar{l}_p + L_a$. To use Eqs. (38), (39), and (41)–(44), we have to start from $(\Omega_{0, 0}^{\bar{l}_p} | \tilde{l}_c) = (\bar{l}_p + \tilde{l}_c)$ integrals; that is, the scaling with $(2p)^{-\bar{l}_p}$ should be left out from Eq. (31). Also, when all the target classes have the same \tilde{L}_c , the necessary scaling with $(-2c)^{-\tilde{L}_c}$ can be done at the level of the Boys functions. These newly presented equations will be mostly useful in setting up the GHP scheme for Hermite and mixed Gaussian integral-derivatives.

It is also possible to use the MD scheme along with OS-type recursions. For undifferentiated Cartesian ERIs, we concluded³⁴ that the application of both Eq. (23) for l_b and Eq. (22) for l_c enhances the performance compared to using only Eqs. (32), (34), and (35). In this route, one first calculates the $(\mathbf{0})^{(n)}$ ERIs with Eq. (20), applies Eq. (32) to construct the $(\bar{l}_p)^{(\tilde{L}_c)} = (\Omega_{0, 0}^{\bar{l}_p} | \mathbf{0})^{(\tilde{L}_c)}$ integrals, transforms these into the $(\Omega_{l_a, 0}^{\bar{l}_p} | \mathbf{0})^{(\tilde{L}_c)} = (l_a \mathbf{0} | \mathbf{0})^{(\tilde{L}_c)}$ ones via Eq. (34), and then proceeds as described for the OS method.

We will consider the following approaches within the MD scheme: when calculating **A** or **B** derivatives, we will produce the $(l_a \mathbf{0} | \mathbf{0})^{\tilde{L}_c}$ or $(\tilde{l}_a \mathbf{0} | \mathbf{0})^{\tilde{L}_c}$ intermediates via the MD recursions and then proceed as the corresponding OS scheme. If **C** derivatives are needed, this is not necessarily the best approach since it might be more efficient to increment the ket side function with Eq. (32) rather than with the VRR, which is more expensive now. Here we will produce the $(L_a L_b | \tilde{L}_c)$ target class from $(\bar{l}_p | \tilde{L}_c)$ by either directly with Eqs. (34) and (35) (MD_{Mixed,C1}) or using Eq. (34) for $(l_a \mathbf{0} | \tilde{L}_c)$ ERIs and then Eq. (23) (MD_{Mixed,C2}), or we apply Eq. (34) for the $(l_a \mathbf{0} | \mathbf{0})^{(n)}$ classes followed by Eqs. (30) and (23) (MD_{Mixed,C3}). Only the mixed Gaussian scheme is considered since transforming \tilde{L}_c into L_c does not offer any advantage. When we wish to evaluate the derivatives with respect to all three centers at the same time, we will only investigate the scheme applying the OS VRR for the ket side since in the majority of the above mentioned cases (that is, differentiating with respect to a single center) this turned out to be the most efficient choice. In these cases, we can also utilize the approach of Helgaker and Taylor¹¹ (further denoted as ³MD_{PR}). Here we produce derivatives with respect to **P** and **R**_{AB} using the equations

$$\frac{\partial (L_a L_b | \tilde{L}_c)^{(n)}}{\partial P_x} = \kappa_{ab} (-2c)^{-\tilde{L}_c} \sum_{\bar{i}_p=0}^{I_a+I_b} E_{\bar{i}_p}^{I_a, I_b} \sum_{\bar{j}_p=0}^{J_a+J_b} E_{\bar{j}_p}^{J_a, J_b} \times \sum_{\bar{k}_p=0}^{K_a+K_b} E_{\bar{k}_p}^{K_a, K_b} (\bar{l}_p + \bar{1}_x + \bar{L}_c)^{(n)} \quad (45)$$

and, with $\partial E_{\tilde{i}_p}^{I_a, I_b} \equiv \partial(\kappa_{ab} E_{\tilde{i}_p}^{I_a, I_b})/\partial X_{AB}$,

$$\frac{\partial(\mathbf{L}_a \mathbf{L}_b | \tilde{\mathbf{L}}_c)^{(n)}}{\partial X_{AB}} = (-2c)^{-\tilde{L}_c} \sum_{\tilde{i}_p=0}^{I_a+I_b} \partial E_{\tilde{i}_p}^{I_a, I_b} \sum_{\tilde{j}_p=0}^{J_a+J_b} E_{\tilde{j}_p}^{J_a, J_b} \times \sum_{\tilde{k}_p=0}^{K_a+K_b} E_{\tilde{k}_p}^{K_a, K_b} (\tilde{\mathbf{L}}_p + \tilde{\mathbf{L}}_c)^{(n)}. \quad (46)$$

The advantage of this scheme is that the summation limits remain the same as for undifferentiated ERIs. Note that using Eqs. (45) and (46), one should omit κ_{ab} from Eq. (20). The evaluation of the differentiated E coefficients is explained in Ref. 11. The **A** and **B** derivatives are then recovered by the expressions

$$\frac{\partial}{\partial A_x} = \frac{a}{p} \frac{\partial}{\partial P_x} + \frac{\partial}{\partial X_{AB}} \quad (47)$$

and

$$\frac{\partial}{\partial B_x} = \frac{\partial}{\partial P_x} - \frac{\partial}{\partial A_x}. \quad (48)$$

The derivatives with respect to **C** are trivially obtained by the translational invariance, $\partial/\partial C_x = -\partial/\partial P_x$. The considered routes are summarized in Table II. For the ³MD algorithms where **C** is differentiated, Cart or Mixed refers to the bra-side functions, while $\tilde{\mathbf{L}}_c$ is always a Hermite Gaussian.

D. Gill–Head–Gordon–Pople algorithm

Here we will investigate the method of Gill, Head-Gordon, and Pople¹⁴ for three-center ERI derivatives with Cartesian, Hermite, and mixed Gaussian integrals. The basis of the algorithm for Cartesian ERIs is that Eqs. (34) and (35) can be applied to contracted integrals if we work with the scaled hybrid ERIs

$$(\Omega_{I_a, I_b}^{\tilde{i}_p} |_{\lambda, \beta, \zeta} = \frac{(2a)^\lambda (2b)^\beta}{(2p)^\zeta} (\Omega_{I_a, I_b}^{\tilde{i}_p} |. \quad (49)$$

With $X_{PA} = -b/pX_{AB}$, we can recast Eq. (34) as

$$(\Omega_{I_a+1_x, I_b}^{\tilde{i}_p} |_{\lambda, \beta, \zeta} = \tilde{i}_p (\Omega_{I_a, I_b}^{\tilde{i}_p-1_x} |_{\lambda, \beta, \zeta} - X_{AB} (\Omega_{I_a, I_b}^{\tilde{i}_p} |_{\lambda, \beta+1, \zeta+1} + (\Omega_{I_a, I_b}^{\tilde{i}_p+1_x} |_{\lambda, \beta, \zeta+1}, \quad (50)$$

which is now not explicitly dependent on the Gaussian exponents. Similarly, since $X_{PB} = a/pX_{AB}$, Eq. (35) becomes

$$(\Omega_{I_a, I_b+1_x}^{\tilde{i}_p} |_{\lambda, \beta, \zeta} = \tilde{i}_p (\Omega_{I_a, I_b}^{\tilde{i}_p-1_x} |_{\lambda, \beta, \zeta} + X_{AB} (\Omega_{I_a, I_b}^{\tilde{i}_p} |_{\lambda+1, \beta, \zeta+1} + (\Omega_{I_a, I_b}^{\tilde{i}_p+1_x} |_{\lambda, \beta, \zeta+1}. \quad (51)$$

Thus the strategy is to first calculate $(\tilde{\mathbf{L}}_p + \tilde{\mathbf{L}}_c)$ integrals using Eq. (32), perform the necessary scalings according to Eq. (49) to produce all the λ , β , ζ -scaled classes of these ERIs that are going to be required by Eqs. (50) and (51), perform the primitive contraction on these scaled classes, and then apply Eqs. (50) and (51) to build up I_a and I_b , respectively. We refrain from presenting the necessary index ranges for the intermediates here since the presence of the scaling indices makes them quite complicated. The reader can find these index ranges in the [supplementary material](#). At the contracted level, we can work in two directions: in GHP_{Cart,1}, we build up I_a and I_b by using Eqs. (50) and (51), respectively, while the GHP_{Cart,2} scheme uses Eq. (50) to construct the appropriate $(\Omega_{I_a, 0}^0 |_{\tilde{\mathbf{L}}_c})_{\lambda, \beta, \zeta}$ intermediates for Eq. (23). The target classes are $([\mathbf{L}_a + \mathbf{1}_x] \mathbf{L}_b | \tilde{\mathbf{L}}_c)_{1,0,0}$ and $([\mathbf{L}_a - \mathbf{1}_x] \mathbf{L}_b | \tilde{\mathbf{L}}_c)_{0,0,0}$ for the **A** derivatives and $(\mathbf{L}_a [\mathbf{L}_b + \mathbf{1}_x] | \tilde{\mathbf{L}}_c)_{0,1,0}$ and $(\mathbf{L}_a [\mathbf{L}_b - \mathbf{1}_x] | \tilde{\mathbf{L}}_c)_{0,0,0}$ for the **B** derivatives. For the same reason as in MD, we will only consider mixed ERI based schemes for the **C** derivatives, where Eqs. (50) and (51) are employed to build up $(\mathbf{L}_a \mathbf{L}_b | \tilde{\mathbf{L}}_c + \tilde{\mathbf{1}}_x)_{0,0,0}$. Also, in the ³GHP algorithms where **C** is differentiated, the function on **C** will always be a Hermite Gaussian, and the ³GHP_{Cart} and ³GHP_{Mixed} notations will refer to the functions in the bra side.

If we wish to work with Hermite integrals, we have to rewrite Eqs. (38) and (39) as we did so with Eqs. (34) and (35). That is, when a term is multiplied by $2a$, $2b$, or $(2p)^{-1}$, index λ , β , or ζ , respectively, is incremented by one. The resulting equations are

$$(\Omega_{I_a+1_x, I_b}^{\tilde{i}_p} |_{\lambda, \beta, \zeta} = (\Omega_{I_a, I_b}^{\tilde{i}_p+1_x} |_{\lambda, \beta, \zeta+1} - X_{AB} (\Omega_{I_a, I_b}^{\tilde{i}_p} |_{\lambda, \beta+1, \zeta+1} - \tilde{i}_a (\Omega_{I_a-1_x, I_b}^{\tilde{i}_p} |_{\lambda-1, \beta+1, \zeta+1} + \tilde{i}_b (\Omega_{I_a, I_b-1_x}^{\tilde{i}_p} |_{\lambda, \beta, \zeta+1} \quad (52)$$

and

$$(\Omega_{I_a, I_b+1_x}^{\tilde{i}_p} |_{\lambda, \beta, \zeta} = (\Omega_{I_a, I_b}^{\tilde{i}_p+1_x} |_{\lambda, \beta, \zeta+1} + X_{AB} (\Omega_{I_a, I_b}^{\tilde{i}_p} |_{\lambda+1, \beta, \zeta+1} + \tilde{i}_a (\Omega_{I_a-1_x, I_b}^{\tilde{i}_p} |_{\lambda, \beta, \zeta+1} - \tilde{i}_b (\Omega_{I_a, I_b-1_x}^{\tilde{i}_p} |_{\lambda+1, \beta-1, \zeta+1}. \quad (53)$$

Transforming Eq. (26) the same way, we get

$$(\Omega_{I_a, I_b+1_x}^0 |_{\lambda, \beta, \zeta} = (\Omega_{I_a+1_x, I_b}^0 |_{\lambda, \beta, \zeta} + X_{AB} (\Omega_{I_a, I_b}^0 |_{\lambda, \beta, \zeta} + \tilde{i}_a (\Omega_{I_a-1_x, I_b}^0 |_{\lambda-1, \beta, \zeta} - \tilde{i}_b (\Omega_{I_a, I_b-1_x}^0 |_{\lambda, \beta-1, \zeta}. \quad (54)$$

TABLE II. Serial numbers of the equations to be applied to calculate the necessary classes in the various MD-based algorithms. Equations (20) and (32) are omitted since all the schemes start with the use of these relations. See also the caption of Table I.

	Separate recursion	Common recursion
MD _{Cart,A} and MD _{Cart,B}	(34), (22), (23)	³ MD _{Cart,AB} (34), (22), (23)
MD _{Mixed,A}	(41), (22), (28)	³ MD _{Cart,AC} and ³ MD _{Cart,BC} (34), (24), (23)
MD _{Mixed,B}	(34), (22), (29)	³ MD _{Herm,AB} (41), (22), (26)
MD _{Mixed,C1}	(34), (35)	³ MD _{Mixed,AC} (41), (27), (28)
MD _{Mixed,C2}	(34), (23)	³ MD _{Mixed,BC} (34), (30), (29)
MD _{Mixed,C3}	(34), (30), (23)	³ MD _{PR} (45), (46)

We will use these equations for the ${}^3\text{GHP}_{\text{Herm,AB}}$ schemes. In ${}^3\text{GHP}_{\text{Herm,AB1}}$, \tilde{l}_a and \tilde{l}_b are built up by using Eqs. (52) and (53), respectively, while in ${}^3\text{GHP}_{\text{Herm,AB2}}$, Eq. (54) is applied to increment \tilde{l}_b .

To employ mixed integrals for **A** derivatives, we transform Eqs. (41) and (42), respectively, as

$$\begin{aligned} (\Omega_{\tilde{l}_a+\tilde{l}_x,\tilde{l}_b}^{\tilde{l}_p} | \lambda, \beta, \zeta) &= (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p+\tilde{l}_x} | \lambda, \beta, \zeta+1 - X_{\text{AB}} (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p} | \lambda, \beta+1, \zeta+1 \\ &\quad - \tilde{i}_a (\Omega_{\tilde{l}_a-\tilde{l}_x,\tilde{l}_b}^{\tilde{l}_p} | \lambda-1, \beta+1, \zeta+1 \end{aligned} \quad (55)$$

and

$$\begin{aligned} (\Omega_{\tilde{l}_a,\tilde{l}_b+\tilde{l}_x}^{\tilde{l}_p} | \lambda, \beta, \zeta) &= (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p+\tilde{l}_x} | \lambda, \beta, \zeta+1 + X_{\text{AB}} (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p} | \lambda+1, \beta, \zeta+1 \\ &\quad + \tilde{i}_a (\Omega_{\tilde{l}_a-\tilde{l}_x,\tilde{l}_b}^{\tilde{l}_p} | \lambda, \beta, \zeta+1 + \tilde{i}_p (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p-\tilde{l}_x} | \lambda, \beta, \zeta, \end{aligned} \quad (56)$$

and also Eq. (28) as

$$\begin{aligned} (\Omega_{\tilde{l}_a,\tilde{l}_b+\tilde{l}_x}^0 | \lambda, \beta, \zeta) &= (\Omega_{\tilde{l}_a+\tilde{l}_x,\tilde{l}_b}^0 | \lambda, \beta, \zeta + X_{\text{AB}} (\Omega_{\tilde{l}_a,\tilde{l}_b}^0 | \lambda, \beta, \zeta \\ &\quad + \tilde{i}_a (\Omega_{\tilde{l}_a-\tilde{l}_x,\tilde{l}_b}^0 | \lambda-1, \beta, \zeta. \end{aligned} \quad (57)$$

In the $\text{GHP}_{\text{Mixed,A}}$ algorithms, Eq. (55) is exploited to build up \tilde{l}_a , while to \tilde{l}_b , we apply Eqs. (56) and (57) in $\text{GHP}_{\text{Mixed,A1}}$ and $\text{GHP}_{\text{Mixed,A2}}$, respectively. For **B** derivatives, Eqs. (43) and (44) are rewritten, respectively, as

$$\begin{aligned} (\Omega_{\tilde{l}_a+\tilde{l}_x,\tilde{l}_b}^{\tilde{l}_p} | \lambda, \beta, \zeta) &= (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p+\tilde{l}_x} | \lambda, \beta, \zeta+1 - X_{\text{AB}} (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p} | \lambda, \beta+1, \zeta+1 \\ &\quad + \tilde{i}_b (\Omega_{\tilde{l}_a,\tilde{l}_b-\tilde{l}_x}^{\tilde{l}_p} | \lambda, \beta, \zeta+1 + \tilde{i}_p (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p-\tilde{l}_x} | \lambda, \beta, \zeta \end{aligned} \quad (58)$$

and

$$\begin{aligned} (\Omega_{\tilde{l}_a,\tilde{l}_b+\tilde{l}_x}^{\tilde{l}_p} | \lambda, \beta, \zeta) &= (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p+\tilde{l}_x} | \lambda, \beta, \zeta+1 + X_{\text{AB}} (\Omega_{\tilde{l}_a,\tilde{l}_b}^{\tilde{l}_p} | \lambda+1, \beta, \zeta+1 \\ &\quad - \tilde{i}_b (\Omega_{\tilde{l}_a,\tilde{l}_b-\tilde{l}_x}^{\tilde{l}_p} | \lambda+1, \beta-1, \zeta+1, \end{aligned} \quad (59)$$

while Eq. (29) is recast as

$$\begin{aligned} (\Omega_{\tilde{l}_a,\tilde{l}_b+\tilde{l}_x}^0 | \lambda, \beta, \zeta) &= (\Omega_{\tilde{l}_a+\tilde{l}_x,\tilde{l}_b}^0 | \lambda, \beta, \zeta + X_{\text{AB}} (\Omega_{\tilde{l}_a,\tilde{l}_b}^0 | \lambda, \beta, \zeta \\ &\quad - \tilde{i}_b (\Omega_{\tilde{l}_a,\tilde{l}_b-\tilde{l}_x}^0 | \lambda, \beta-1, \zeta. \end{aligned} \quad (60)$$

Equation (58) is utilized for \tilde{l}_a in the $\text{GHP}_{\text{Mixed,B}}$ schemes, and \tilde{l}_b is built up by Eqs. (59) and (60) in $\text{GHP}_{\text{Mixed,B1}}$ and $\text{GHP}_{\text{Mixed,B2}}$, respectively. The considered GHP-based schemes are given in Table III.

E. Rys quadrature

From the above discussed recursive methods, it follows that ERIs can be written as a linear combination of Boys functions and thus can be expressed as

$$\begin{aligned} (\mathbf{L}_a \mathbf{L}_b | \mathbf{L}_c) &= \sum_{r=0}^{L_a+L_b+L_c} Z_r F_r(\alpha R_{\text{PC}}^2) \\ &= \int_0^1 \sum_{r=0}^{L_a+L_b+L_c} Z_r t^{2r} \exp(-\alpha R_{\text{PC}}^2 t^2) dt, \end{aligned} \quad (61)$$

where the Z_r coefficients can be obtained, for example, by backtracking the OS recursions until the ERI is expanded only in Boys functions. Since this expression is an integral over a polynomial $f(t^2) = \sum_{r=0}^{L_a+L_b+L_c} Z_r t^{2r}$ multiplied by the weight function $\exp(-\alpha R_{\text{PC}}^2 t^2)$, it can be evaluated by the Rys quadrature^{6,55} with the expression

$$\int_0^1 f(t^2) \exp(-\alpha R_{\text{PC}}^2 t^2) dt = \sum_{r=1}^{N_{\text{rts}}} f(t_r^2) \omega_r(\alpha R_{\text{PC}}^2, t_r^2), \quad (62)$$

with $N_{\text{rts}} > \lceil (L_a + L_b + L_c)/2 \rceil$, t_r is the r th positive root of the $(2N_{\text{rts}})$ th Rys polynomial⁴ in t , and ω_r is the weight factor of the quadrature associated with t_r^2 . $f(t_r^2)$ can be written as

$$f(t_r^2) = 2 \left(\frac{\alpha}{\pi} \right)^{1/2} \Theta_x^{l_a, l_b, l_c}(t_r^2) \Theta_y^{j_a, j_b, j_c}(t_r^2) \Theta_z^{k_a, k_b, k_c}(t_r^2), \quad (63)$$

where $\Theta_x^{l_a, l_b, l_c}(t_r^2)$ is a two-dimensional (2D) integral with respect to the x coordinates of the two electrons.⁵ These 2D integrals can be computed recursively.⁵⁵ If we use $\Theta_x^{0,0,0}(t_r^2) = \Theta_y^{0,0,0}(t_r^2) = 1$ and $\Theta_z^{0,0,0}(t_r^2) = \theta_{pc} \kappa_{ab} \omega_r(\alpha R_{\text{PC}}^2, t_r^2)$ as the starting values for the recursion, the final working equation becomes

$$\begin{aligned} \int_0^1 f(t^2) \exp(-\alpha R_{\text{PC}}^2 t^2) dt \\ = \sum_{r=1}^{N_{\text{rts}}} \Theta_x^{l_a, l_b, l_c}(t_r^2) \Theta_y^{j_a, j_b, j_c}(t_r^2) \Theta_z^{k_a, k_b, k_c}(t_r^2). \end{aligned} \quad (64)$$

TABLE III. Serial numbers of the equations to be applied to calculate the necessary classes in the various GHP-based algorithms. Equations (20) and (32) are omitted since all the schemes start with the use of these relations. See also the caption of Table I.

Separate recursion	Common recursion
GHP _{Cart,A1} , GHP _{Cart,B1} , and GHP _{Mixed,C1} (50), (51)	${}^3\text{GHP}_{\text{Cart,AB1}}$, ${}^3\text{GHP}_{\text{Cart,AC1}}$, and ${}^3\text{GHP}_{\text{Cart,BC1}}$ (50), (51)
GHP _{Cart,A2} , GHP _{Cart,B2} , and GHP _{Mixed,C2} (50), (23) (55), (56)	${}^3\text{GHP}_{\text{Cart,AC2}}$, ${}^3\text{GHP}_{\text{Cart,AC2}}$, and ${}^3\text{GHP}_{\text{Cart,BC2}}$ (50), (23) (52), (53)
GHP _{Mixed,A1} (55), (56)	${}^3\text{GHP}_{\text{Herm,AB1}}$ (55), (56)
GHP _{Mixed,A2} (55), (57)	${}^3\text{GHP}_{\text{Mixed,AC1}}$ (55), (56)
GHP _{Mixed,B1} (58), (59)	${}^3\text{GHP}_{\text{Mixed,BC1}}$ (58), (59)
GHP _{Mixed,B2} (58), (60)	${}^3\text{GHP}_{\text{Herm,AB2}}$ (52), (54) ${}^3\text{GHP}_{\text{Mixed,AC2}}$ (55), (57) ${}^3\text{GHP}_{\text{Mixed,BC2}}$ (58), (60)

To calculate the integral-derivatives, we consider two options. The first one is to evaluate the undifferentiated ERIs necessary for the direct construction of the derivatives by either Eq. (4) or Eq. (6). In our earlier investigation of the algorithm for ERIs,³⁴ we concluded that it is always more advantageous to calculate only the $(\mathbf{I}_a \mathbf{0} \mathbf{0})^{(n)}$ classes with the quadrature and proceed with OS recursions thereafter. For this, the 2D integrals $\Theta_x^{i_a,0,0}$ (and their y and z counterparts) for $0 \leq i_a \leq L_a + 1$ have to be evaluated. For brevity, in this subsection, we present the recursions for the Cartesian, Hermite, and mixed Gaussian 2D ERIs in a single equation because of their very similar form. The necessary recursion here is (see Subsection 3 of the Appendix)

$$\Theta_x^{i_a+1,0,0}(t_r^2) = \left(X_{\text{PA}} - \frac{\alpha}{p} X_{\text{PC}} t_r^2 \right) \Theta_x^{i_a,0,0}(t_r^2) + \frac{i_a}{2p} \left(w_{ba} - \frac{\alpha}{p} t_r^2 \right) \Theta_x^{i_a-1,0,0}(t_r^2), \quad (65)$$

where $w_{ba} = -b/a$ when the function centered on \mathbf{A} is a Hermite Gaussian and 1 otherwise. From these, we can compute $(\mathbf{I}_a \mathbf{0} \mathbf{0})^{(n)}$ by using Eq. (64), but note that each term in the summation has to be multiplied by t_r^{2n} for these classes.³⁴ When we can use Eq. (22), this is easily accounted for by starting the recursion for the z direction from $\Theta_z^{0,0,0}(t_r^2) = \theta_{pc} \kappa_{ab} \omega_r (\alpha R_{\text{PC}}^2, t_r^2) t_r^{2L_c}$. When \mathbf{C} is differentiated, we have, however, more than one n values to consider due to the fact that Eq. (24) or Eq. (27) have to be used. Here we have to apply³⁴ the more expensive assembly

$$\int_0^1 f(t^2) \exp(-\alpha R_{\text{PC}}^2 t^2) t^{2n} dt = \sum_{r=1}^{N_{\text{rs}}} \Theta_x^{i_a, j_b, j_c}(t_r^2) \Theta_y^{j_a, j_b, j_c}(t_r^2) \Theta_z^{k_a, k_b, k_c}(t_r^2) t_r^{2n}. \quad (66)$$

Thus in these cases, we will also inspect the schemes where Eq. (64) is used to construct the target ERIs and also when the $(\mathbf{I}_a \mathbf{0} \mathbf{l}_c)$ integrals are calculated by it. The 2D recurrence for incrementing i_c is (see Subsection 3 of the Appendix)

$$\Theta_x^{i_a, 0, i_c+1}(t_r^2) = \frac{\alpha}{c} X_{\text{PC}} t_r^2 \Theta_x^{i_a, 0, i_c}(t_r^2) + \frac{i_a t_r^2}{2(p+c)} \Theta_x^{i_a-1, 0, i_c}(t_r^2) + \frac{i_c}{2c} \left(w_c - \frac{\alpha t_r^2}{c} \right) \Theta_x^{i_a, 0, i_c-1}(t_r^2), \quad (67)$$

where $w_c = 1$ if the function on \mathbf{C} is a Cartesian and 0 if a Hermite Gaussian. Finally, for i_b , we use

$$\Theta_x^{i_a, i_b+1, i_c}(t_r^2) = \Theta_x^{i_a+1, i_b, i_c}(t_r^2) + X_{\text{AB}} \Theta_x^{i_a, i_b, i_c}(t_r^2) + w_a \frac{i_a}{2a} \Theta_x^{i_a-1, i_b, i_c}(t_r^2) - w_b \frac{i_b}{2b} \Theta_x^{i_a, i_b-1, i_c}(t_r^2), \quad (68)$$

where $w_a = 1$ if the \mathbf{A} function is a Hermite and 0 if a Cartesian Gaussian, and analogously for w_b . As for the MD and GHP methods, there are no advantages the Cartesian Gaussian ERIs offer for calculating \mathbf{C} derivatives compared to the mixed integrals; hence, in these cases, only the mixed algorithms will be characterized. Similar to the MD schemes, it turns out that the route applying Eq. (66) and then the HRR and VRR equations appropriate for the applied Gaussians is the most efficient

choice for the \mathbf{C} derivatives; hence, only this method will be investigated for the ³RYS algorithms.

The other option is the approach of Flocke and Lotrich,⁹ where the 2D integrals are differentiated, and Eq. (64) directly produces the differentiated ERIs,

$$\frac{\partial}{\partial A_x} \int_0^1 f(t^2) \exp(-\alpha R_{\text{PC}}^2 t^2) dt = \sum_{r=1}^{N_{\text{rs}}} \frac{\partial \Theta_x^{i_a, j_b, j_c}(t_r^2)}{\partial A_x} \Theta_y^{j_a, j_b, j_c}(t_r^2) \Theta_z^{k_a, k_b, k_c}(t_r^2). \quad (69)$$

This scheme will be further referred to as RYS_{2D} (e.g., RYS_{Cart,A2D} when we use Cartesian Gaussians and differentiate \mathbf{L}_a). Since the nuclear coordinate dependent parts of the 2D integrals are the x, y, or z components of the Cartesian or Hermite Gaussian basis functions, they follow the differentiation rule of Eq. (4) or Eq. (6). Concerning the methods for the calculation of the roots of the Rys polynomials and the weight factors, we refer to other work,^{4,9,57} but we note that the associated computational cost is of the same magnitude as that for the Boys function evaluations with the other algorithms. The discussed schemes are summarized in Table IV, with the exception of the RYS_{2D} schemes: these algorithms always calculate the 2D integrals with Eqs. (65), (67), and (68), compute their derivatives, and then apply Eq. (69).

F. Algorithmic considerations

The contraction of the primitive functions for a shell triplet is performed inside a nested loop structure by paying specific attention to the treatment of uncontracted AOs and the use of cache-friendly buffer arrays. For a detailed description, please see our previous work.³⁴

The main algorithmic difference between the approaches based on Cartesian and Hermite Gaussians for the OS, MD, and RYS methods is the applicability of the HRR at the contracted level. The fact that Eq. (23) does not depend on any Gaussian exponents is an obvious advantage of the Cartesian Gaussians. However, for the $(\tilde{\mathbf{L}}_a \tilde{\mathbf{L}}_b)$ and $(\tilde{\mathbf{L}}_a \mathbf{L}_b)$ bra sides with $L_b = 1$ and for the $(\mathbf{L}_a \tilde{\mathbf{L}}_b)$ ones with $L_b = 2$, only one of the

TABLE IV. Serial numbers of the equations to be applied to calculate the necessary classes in the various Rys-based algorithms. See also the caption of Table I.

Separate recursion	
RYS _{Cart,A} and RYS _{Cart,B}	(65), (64), (22), (23)
RYS _{Mixed,A}	(65), (64), (22), (28)
RYS _{Mixed,B}	(65), (64), (22), (29)
RYS _{Mixed,C1}	(65), (67), (68), (66)
RYS _{Mixed,C2}	(65), (67), (66), (23)
RYS _{Mixed,C3}	(65), (66), (30), (23)
Common recursion	
³ RYS _{Cart,AB}	(65), (64), (22), (23)
³ RYS _{Herm,AB}	(65), (64), (22), (26)
³ RYS _{Cart,AC} and ³ RYS _{Cart,BC}	(65), (66), (24), (23)
³ RYS _{Mixed,AC}	(65), (66), (30), (28)
³ RYS _{Mixed,BC}	(65), (66), (30), (29)

classes in the HRR recursion tree will be multiplied by a factor with an exponent, that is, $(\tilde{\mathbf{L}}_a - \mathbf{1}_x | \tilde{\mathbf{L}}_c)$ using Eqs. (26) and (28) and $(\mathbf{L}_a \mathbf{0} | \tilde{\mathbf{L}}_c)$ using Eq. (29). If these classes are multiplied by their respective exponent dependent factors at the primitive level, the HRR can be performed on the contracted basis in these cases. We note this because AOs with angular momenta greater than 2 are rarely contracted. While working with contracted integrals generally lowers the FLOP count, performing the operations at the primitive level is usually more cache friendly. The reason for this, as we concluded for ERIs,³⁴ is that most cache misses, that is, events where the referenced memory address is not represented in the cache memory of the central processing unit (CPU), happen during the reading or writing of large arrays that store the partially and fully contracted integrals. The calculation steps which are performed inside the primitive loops use the same small and fixed length arrays for every exponent triplet, while, at the contracted level, the data have to be read from and written into arrays that store all the classes for a shell triplet. The size of these arrays and thus the number of cache misses³⁴ can be reduced by applying the spherical harmonic transformation at the primitive level. Such algorithms are especially advantageous when prescreening is applied for the evaluation of the primitive classes, which can greatly reduce the actual work performed inside the primitive loops. We also note that the HRR work can be reduced in the cases when we evaluate \mathbf{B} derivatives for $L_a = L_b$ shell triplets by switching the two functions in the bra and then proceeding by differentiating the first function.

For the prescreening of significant ERI derivative batches, we consider two approaches. A straightforward strategy is to apply the Cauchy–Schwarz inequality, which, for example, for the first derivative of the first function, is written as⁴³

$$\left| \left(\frac{\partial \mathbf{L}_a}{\partial A_x} \mathbf{L}_b | \mathbf{L}_c \right) \right| \leq \sqrt{\left| \left(\frac{\partial \mathbf{L}_a}{\partial A_x} \mathbf{L}_b | \frac{\partial \mathbf{L}_a}{\partial A_x} \mathbf{L}_b \right) \right| \sqrt{|\mathbf{L}_c | \mathbf{L}_c |}}. \quad (70)$$

The calculation of the second derivatives appearing in Eq. (70) is a challenging task, and the efficiency of evaluating the quantities necessary for prescreening becomes important, especially when the three-center derivative integral list is only computed once for a given geometry. We can reduce the cost of this type of prescreening if we use Eq. (70) to determine the upper bounds for primitive ERI derivatives instead of derivatives of ERIs over AOs and multiply these with the maximal contraction coefficients and contraction degrees belonging to the primitives in question. A batch of AOs are then screened out using the largest of the associated primitive bounds. This method has the advantage that the calculation of the required second derivative integrals scales with the second power of the number of primitive functions, which would be fourth power if we aimed to give upper bounds for AO ERI derivatives. Furthermore, \mathbf{R}_{PQ} becomes 0 in these cases, which makes the Boys functions trivial to calculate.⁵⁵ An MD based calculation is even more simplified since the one-center ERIs given by Eq. (32) reduce to

$$(\tilde{\mathbf{L}}_p) = (\tilde{i}_p - 1)!! (\tilde{j}_p - 1)!! (\tilde{k}_p - 1)!! \kappa_{ab}^2 \theta_{pp}^2 (-p)^{l_p/2} F_{l_p/2}(0) \quad (71)$$

for the cases when \tilde{i}_p , \tilde{j}_p , and \tilde{k}_p (and therefore also \tilde{l}_p) all have even parity, otherwise $(\tilde{\mathbf{L}}_p) = 0$. This reduces the cost of an assembly of the type of Eq. (33). It is also advantageous to use Hermite Gaussians to evaluate the second derivatives. Note that for the application of Eq. (10) we only need to calculate the cases where $\tilde{\mathbf{L}}_a$ and $\tilde{\mathbf{L}}_b$ denote the same solid harmonic or differentiated solid harmonic Gaussians in the bra and the ket. In addition to screening entire AO batches, the calculated primitive bounds are readily used to screen the evaluation of primitive integral-derivatives. In the following, we will refer to this prescreening scheme as **C–S** when it is only applied to screen shell triplets, and when primitive ERI derivatives are also screened, the notation will be **C–S prim**. We note that in a scheme where we compute all the derivatives of a shell triplet at the same time, we can use Eq. (70) to see if the derivatives with respect to all three centers will be significant. If not, a less expensive algorithm computing only the derivatives with respect to a single center can be applied, along with Eq. (18). We will not pursue this approach in the present work.

Another option we could use for the prescreening is to work with the assumption that

$$|\mathbf{L}_a \mathbf{L}_b | \mathbf{L}_c | \leq |(\mathbf{ss} | \mathbf{s})|. \quad (72)$$

For three-center ERIs, we concluded that this approximation results in an efficient bound with satisfactory accuracy.³⁴ From Eq. (4), we can write for the derivatives of the first function

$$\left| \frac{\partial}{\partial A_x} (\mathbf{L}_a \mathbf{L}_b | \mathbf{L}_c) \right| \leq |[2 \max(a, b, c) + \max(l_a, l_b, l_c)] (\mathbf{ss} | \mathbf{s})| \quad (73)$$

and use this for the screening of AO batches and primitive ERI derivatives as well. The screening method using this approximate bound will be denoted as **sss** and **sss prim**. We will analyze the efficiency of both methods and the error introduced by this latter screening strategy.

It is not trivial to decide at what stage of the algorithms the spherical harmonic transformations and the construction of the derivatives should be carried out. The spherical harmonic transformations reduce the number of components for further operations, but their own costs also depend on what number of intermediates they are performed on. Constructing the derivatives of a Hermite Gaussian function with angular momentum $\tilde{\mathbf{L}}$ with Eq. (10) always increases the number of the intermediates compared to the $\tilde{\mathbf{L}} + \mathbf{1}_x$ shell, which is required for this operation. This is not necessary true for Cartesian Gaussians because applying Eq. (4) and then Eq. (9) decreases the number of components compared to the sum of the components in the $\mathbf{L} + \mathbf{1}_x$ and $\mathbf{L} - \mathbf{1}_x$ shells if $L > 3$. The possible stages where these transformations can be performed also vary from scheme to scheme. For example, in a $\text{GHP}_{\text{Cart}, A1}$ algorithm, if we first build up \mathbf{L}_a with Eq. (50) and then \mathbf{L}_b with Eq. (51), we can perform both the differentiation and the solid harmonic transformation of \mathbf{L}_a after using Eq. (50). With a $\text{GHP}_{\text{Mixed}, A1}$ scheme, where Eqs. (55) and (56) are used, we do not have this option since Eq. (56) depends on $\tilde{\mathbf{L}}_a$. Note also that, when Cartesian Gaussian derivatives are calculated with an algorithm employing Eq. (23), moving the HRR outside of the primitive loops makes it impossible to use common intermediates for the buildup of the two classes that appear in

Eq. (4) since the intermediates for the term with the increased angular momentum have to be multiplied by an exponent-dependent factor before the contraction. These considerations make it very complicated to say which order of operations will be the most preferable for a given algorithm. The most efficient positions for the various computational steps were thus determined by calculating floating point operation counts.

III. FLOATING POINT OPERATION COUNTS

To decide which of the various discussed methods is the most beneficial, we performed a preliminary FLOP counting. The required number of FLOPs for the considered schemes were compared by estimating the necessary number of operations for the evaluation of the derivatives with respect to the coordinates of the three centers for shell triplets up to (*hhl*). For this purpose, a simple program was developed that counts the FLOPs for a model system of three separate carbon atoms, with Dunning's⁵⁸ correlation consistent cc-pVXZ ($X = D, T, Q, 5$) basis sets (XZ for short) for the bra side and the corresponding auxiliary basis sets of Weigend⁵⁹ (cc-pVXZ-RI) for the ket side. The considered computational steps include the evaluation of the two- and three-center auxiliary quantities, the buildup of the primitive Gaussian ERIs, the construction of derivatives for Cartesian integrals by Eq. (4), and the transformation into the solid harmonic and contracted bases. The sparsity of the last two transformations was taken into consideration. We refrained from the estimation of the cost of the Boys function and Rys root evaluations since both depend heavily on the geometry of the system. For the same reason, we also did not consider the effect of primitive prescreening. The program counts the FLOP requirements for every possible route which can be derived by varying the order of the operations that can be performed at different parts of the algorithms. The spherical harmonic transformation of L_a , L_b , or L_c can be carried out at any place where no more operations depend on l_a , l_b , or l_c , respectively. For the derivatives, Eq. (4) or Eq. (10) can be applied after there is no more dependence on the shell to be differentiated for any recursion, even at the contracted level if the multiplication with the double of the exponent is performed before the contraction. Equation (23) can be carried out anywhere after the ($l_a \mathbf{0} l_c$) intermediates have been produced. As explained in Sec. II F, the same is possible for Eqs. (26) and (28) with $\tilde{L}_b = 1$ and for Eq. (29) with $L_b \leq 2$. For the MD₁ and GHP₁ type schemes, the order in which the bra-side angular momenta are built up is also arbitrary. For the GHP schemes, we only considered routes where at least one of the bra-side recursions is performed at the contracted level since, when both angular momenta are constructed in the primitive basis, the scheme is equal to an MD algorithm. The VRR equations, the one-center recursion of the MD and GHP schemes, and the 2D recursions and the quadrature assembly of the Rys method are executed at the primitive level.

Tables V–VIII display the FLOP counts for the various schemes. The numbers for each shell triplet with the best order of the computational steps are presented in the [supplementary material](#), and the conclusions regarding the efficiency of

TABLE V. FLOP counts for the various OS-based algorithms with the cc-pVXZ ($X = D, T, Q, 5$) basis sets.

Algorithm	X			
	D	T	Q	5
OS _{Cart,A}	834 039	3 586 356	19 552 557	90 066 913
OS _{Cart,B}	878 444	3 774 088	20 114 653	90 629 426
OS _{Cart,C}	963 485	4 685 749	28 813 487	145 680 780
OS _{Herm,A}	781 668	3 397 430	19 121 889	91 472 513
OS _{Herm,B}	904 647	4 135 692	23 578 592	111 565 131
OS _{Herm,C}	793 992	3 960 789	24 936 806	130 619 870
OS _{Mixed,A}	778 308	3 333 764	18 431 265	86 727 029
OS _{Mixed,B}	858 221	3 691 859	19 875 872	90 166 857
OS _{Mixed,C}	749 055	3 445 287	20 089 055	98 798 104
³ OS _{Cart,AB}	1 331 840	5 712 048	30 371 707	136 326 843
³ OS _{Cart,AC}	1 782 179	8 575 091	51 458 794	253 017 583
³ OS _{Cart,BC}	1 914 024	9 009 709	52 763 347	255 133 654
³ OS _{Herm,AB}	1 419 613	6 213 209	33 738 005	154 821 887
³ OS _{Mixed,AC}	1 508 292	7 288 759	43 730 208	217 161 994
³ OS _{Mixed,BC}	1 628 374	7 579 886	43 448 054	207 249 588

the methods for different angular momenta will rely on these results.

Table V contains the FLOP requirements of the OS-based algorithms. Here, using the mixed Gaussian integrals for each shell triplet reduces the FLOP count by 7%-4% for the **A** derivatives, 2%-1% for the **B** derivatives, and 22%-32% for the **C** derivatives for the DZ-5Z basis sets compared to applying only the OS_{Cart} schemes. These savings are 1%-5%, 5%-19%, and 6%-24%, respectively, compared to the case when we only use the OS_{Herm} algorithms. As L_b increases, the simplicity of Eq. (23) compensates for the extra work required for the shell triplet with the decreased angular momentum stemming from the last term of Eq. (4). Generally, when $L_b \geq 3$, OS_{Cart} becomes the best choice for **A** and **B** derivatives. In the latter case, the L_b that we have to build up is increased by one, but Eq. (29) is also more efficient than Eq. (28) since it requires fewer intermediates. For **C** derivatives, OS_{Mixed,C} is always the

TABLE VI. FLOP counts for the various MD-based algorithms with the cc-pVXZ ($X = D, T, Q, 5$) basis sets.

Algorithm	X			
	D	T	Q	5
MD _{Cart,A}	916 213	4 043 754	21 895 647	100 400 942
MD _{Cart,B}	958 288	4 218 790	22 394 767	100 711 632
MD _{Mixed,A}	865 972	3 827 843	21 051 221	98 842 722
MD _{Mixed,B}	941 824	4 146 520	22 178 126	100 298 776
MD _{Mixed,C1}	909 103	5 157 925	35 351 909	197 374 785
MD _{Mixed,C2}	842 553	4 521 333	28 852 865	150 054 265
MD _{Mixed,C3}	787 388	3 780 076	22 521 838	111 879 063
³ MD _{Cart,AB}	1 413 790	6 166 368	32 693 989	146 570 332
³ MD _{Cart,AC}	2 032 851	9 988 407	59 788 722	293 139 967
³ MD _{Cart,BC}	2 160 810	10 406 048	61 017 238	294 977 623
³ MD _{Herm,AB}	1 511 629	6 721 972	36 405 233	167 086 412
³ MD _{Mixed,AC}	1 721 805	8 452 087	50 373 647	249 640 343
³ MD _{Mixed,BC}	1 825 007	8 630 926	49 302 884	234 896 789
³ MD _{PR}	3 111 352	19 451 574	142 582 639	848 103 741

TABLE VII. FLOP counts for the various GHP-based algorithms with the cc-pVXZ ($X = D, T, Q, 5$) basis sets.

Algorithm	X			
	D	T	Q	5
GHP _{Cart,A1}	893 738	5 206 588	34 696 058	181 827 684
GHP _{Cart,A2}	886 662	4 718 681	28 260 468	136 426 246
GHP _{Cart,B1}	932 694	5 488 016	37 012 933	195 655 585
GHP _{Cart,B2}	944 946	4 993 045	29 474 616	140 521 684
GHP _{Mixed,A1}	841 451	4 782 519	31 904 972	168 362 436
GHP _{Mixed,A2}	847 333	4 577 449	28 616 778	143 174 999
GHP _{Mixed,B1}	909 815	5 259 732	35 075 833	183 173 164
GHP _{Mixed,B2}	939 050	5 002 224	29 675 460	142 225 345
GHP _{Mixed,C1}	662 485	4 118 817	30 732 628	178 382 077
GHP _{Mixed,C2}	671 760	3 855 113	25 984 570	138 067 909
³ GHP _{Cart,AB1}	1 380 773	8 382 919	55 370 303	288 455 800
³ GHP _{Cart,AB2}	1 605 952	8 784 298	54 930 312	280 549 055
³ GHP _{Cart,AC1}	1 343 819	8 593 244	60 728 861	332 513 082
³ GHP _{Cart,AC2}	1 353 354	8 067 045	52 558 001	267 957 804
³ GHP _{Cart,BC1}	1 383 307	8 878 565	62 975 698	342 948 786
³ GHP _{Cart,BC2}	1 467 546	8 511 933	54 385 447	273 898 475
³ GHP _{Herm,AB1}	1 297 987	8 213 273	57 886 000	318 368 393
³ GHP _{Herm,AB2}	1 573 955	8 192 973	49 124 752	242 974 909
³ GHP _{Mixed,AC1}	1 298 470	8 393 091	61 268 253	344 809 058
³ GHP _{Mixed,AC2}	1 314 025	7 925 813	52 946 831	275 481 605
³ GHP _{Mixed,BC1}	1 365 938	8 882 016	64 397 121	357 490 370
³ GHP _{Mixed,BC2}	1 466 426	8 502 294	54 433 543	275 552 831

best choice. The FLOP counts using mixed integrals for differentiating with respect to one center are nearly optimal; that is, they are on average about 1% higher than the counts resulting from the best suited Gaussian combination (OS_{Cart}, OS_{Herm}, or OS_{Mixed}) for every shell triplet. The most efficient way of evaluating all three nuclear derivatives is ³OS_{Cart,AB}. Here we only have one L_c to build up, so there are more common

TABLE VIII. FLOP counts for the various RYS-based algorithms with the cc-pVXZ ($X = D, T, Q, 5$) basis sets.

Algorithm	X			
	D	T	Q	5
RYS _{Cart,A}	909 536	3 821 275	20 459 245	93 163 062
RYS _{Cart,A2D}	1 584 986	7 524 841	46 468 848	242 012 893
RYS _{Cart,B}	953 406	4 002 579	20 982 691	93 546 498
RYS _{Cart,B2D}	1 618 478	7 631 728	46 915 887	243 064 918
RYS _{Mixed,A}	847 432	3 543 792	19 249 913	89 536 499
RYS _{Mixed,A2D}	1 562 392	7 428 420	46 092 369	240 372 510
RYS _{Mixed,B}	932 616	3 918 769	20 740 050	93 074 206
RYS _{Mixed,B2D}	1 622 394	7 643 554	46 907 993	242 640 950
RYS _{Mixed,C1}	1 107 643	4 917 716	27 808 428	135 735 251
RYS _{Mixed,C2}	990 588	4 423 282	24 829 490	118 200 624
RYS _{Mixed,C3}	839 702	3 841 094	22 218 908	108 540 338
RYS _{Mixed,C2D}	2 091 075	9 351 259	54 608 020	274 107 117
³ RYS _{Cart,AB}	1 409 731	5 954 407	31 297 787	139 483 332
³ RYS _{Cart,AC}	2 060 190	9 785 375	57 785 166	281 094 275
³ RYS _{Cart,BC}	2 193 980	10 211 841	58 993 050	282 670 260
³ RYS _{Herm,AB}	1 497 616	6 457 450	34 677 023	158 057 580
³ RYS _{Mixed,AC}	1 721 569	8 170 373	48 084 554	235 930 334
³ RYS _{Mixed,BC}	1 857 226	8 509 113	47 938 028	226 251 462

intermediates for Eq. (23) that are shared among the different required shell triplets than in the ³OS_{Cart,AC} and ³OS_{Cart,BC} cases. The ket side VRR is also cheaper, while the HRR is more efficient than for ³OS_{Herm,AB}. Applying the ³OS_{Cart,AB} scheme requires 12%-20% fewer FLOPs for calculating all the derivatives for each shell triplet in the DZ-5Z basis sets than for evaluating the derivatives with respect to the two cheapest centers separately and recovering the third one via Eq. (18).

The results for the MD algorithms are presented in Table VI. The conclusions for the schemes are very similar to those for OS. For the C derivatives of the shell triplets with small bra angular momenta and high \tilde{L}_c , the MD_{Mixed,C2} scheme demands the fewest FLOPs; however, using MD_{Mixed,C3} for every shell triplet only results in 3%-1% increase in the cost. The ³MD_{PR} algorithm is only competitive in the (ssls) case.

The FLOP counts of the investigated GHP paths are displayed in Table VII. With the GHP schemes, the use of the best Gaussian type results in 6%-3% and 3%-1% (0%-7% and 0%-2%) lower FLOP requirements than using only Cartesian Gaussian (mixed Gaussian) integrals for the A and B derivatives, respectively. Applying only GHP₁-type schemes costs 2%-21%, 2%-24%, and 1%-23% more than using the best route for the A, B, and C derivatives, respectively. For GHP₂, these values are 2%-0%, 4%-1%, and 2%-0%. As L_b or \tilde{L}_b increases, the application of the HRR becomes more favored since it requires fewer kinds of λ, β, ζ -scaled intermediates than the GHP recursions. Cartesian Gaussians are preferred for higher bra sides, especially for A derivatives. This is because for these classes it is more advantageous to perform the recursion for the first angular momentum at the primitive level, and while for B derivatives, the necessary intermediates for the HRR are the same in this case for the mixed and the Cartesian ERIs, and for the A derivatives, there are more HRR intermediates for the mixed integrals. When differentiating A, the preferred algorithm is GHP_{Cart,A2} for $L_a \geq 3$ and $L_b \geq 1$ and also for (ddl) bras, and GHP_{Mixed,A1} for the rest of the shell triplets. For B, GHP_{Cart,B2} is the best choice for (ddl), (ffl), (gfl), (ggl), (hgl), and (hhl) bras, GHP_{Mixed,B1} for bra sides up to (dpl) and also (fsl), and GHP_{Mixed,B2} for the other shell triplets. In the C case, where we only investigated the mixed Gaussian algorithms, the GHP_{Mixed,C2} route is the best when $L_a \geq 3$ and $L_b \neq 0$ and also for (ddl) bra sides, otherwise GHP_{Mixed,C1} is the method of choice. If we wish to calculate the three derivatives in a common algorithm, ³GHP_{Mixed,AC1} is preferred for $L_b = 0$, ³GHP_{Herm,AB1} for $\tilde{L}_a = \tilde{L}_b = 1$, and ³GHP_{Herm,AB2} for most of the remaining cases. For a number of shell triplets with (ggl), (hgl), and (hhl) bras, ³GHP_{Cart,AC2} is the most efficient because of the simplicity of Eq. (23). Calculating the derivatives with the ³GHP schemes reduces the cost by 14%-8% compared to evaluating the two cheapest derivatives and applying Eq. (18). Unlike in the OS and MD cases, this method becomes less efficient as the cardinal number of the basis set increases since performing operations at the contracted level limits the use of common intermediates in the recursions because of the different exponent scalings for the derivatives coming from Eqs. (4) and (6).

The results for the RYS schemes are presented in Table VIII. For the RYS_C algorithms, we observe that the

schemes applying VRR for \tilde{L}_c are always the most efficient; thus, the results are very similar to the OS case. Compared to Cartesian ERIs, the mixed Gaussian schemes reduce the cost by 7% and 2% for the **A** and **B** derivatives, respectively. As with OS, the ${}^3\text{RYS}_{\text{Cart,AB}}$ route is the best way of evaluating all three derivatives at the same time.

Concerning the order of the various operations, we observe that it is preferred to carry out the HRR step at the contracted stage for $L_b = 1$ or $\tilde{L}_b = 1$. For the considered basis sets, the higher shell triplets are uncontracted, and in these instances, applying Eq. (23) on contracted ERIs results in higher FLOP counts for the $L_b > 2$ cases than doing so at the primitive level, except for the **C** derivatives. This is because the HRR for the two terms in Eq. (4) cannot share any intermediates if the scaling with $2a$ or $2b$ has already been performed for the primitive ERIs. For example, if an ${}^3\text{OS}_{\text{Cart,AB}}$ scheme is applied, the $([I_a + \mathbf{1}_x]I_b|I_c)$ and $(I_a[I_b - \mathbf{1}_x]|I_c)$ classes appear as recursion intermediates for $(I_a[I_b + \mathbf{1}_x]|I_c)$. We cannot exploit this at the contracted stage since these classes will have different scalings. Our results also suggest that it is the most advantageous to perform the spherical harmonic transformation of the bra side functions, Eq. (4) for Cartesian ERIs, and the spherical harmonic transformation of the differentiated shell as the last operations in every investigated algorithm. In the case of the OS, MD, and RYS algorithms, both the costs of the spherical harmonic transformation of the ket function and the HRR are reduced if we transform L_c or \tilde{L}_c before the HRR in most of the cases. The spherical harmonic transformation is slightly more expensive this way for $(\tilde{L}_a\tilde{L}_b|)$ and $(\tilde{L}_aL_b|)$ bra sides, with $\tilde{L}_b = 1$ or $L_b = 1$, and for $(L_a\tilde{L}_b|)$ bra sides with $\tilde{L}_b = 1$, but the reduction in the FLOP count of the HRR compensates for this. In the most efficient ${}^3\text{OS}_{\text{Cart,AB}}$ algorithm, the L_c spherical harmonic transformation is always preferred to be carried out at the primitive stage since the intermediates required by Eq. (23) are less numerous before the $2a$ and $2b$ scaling of Eq. (4) is applied. For the spherical harmonic transformation of \tilde{L}_c in the GHP schemes, we found two options to be efficient for the majority of shell triplets: before the scaling of the $(\Omega_{0,0}^{\tilde{L}_c}|\tilde{L}_c) = (\tilde{I}_p + \tilde{L}_c)$ classes or after the contraction of the $(\Omega_{0,0}^{\tilde{L}_c}|\tilde{L}_c)_{\lambda,\beta,\zeta}$ ones. Transforming \tilde{L}_c before the scaling means that we do not have to perform this operation for all the necessary λ, β, ζ scaled classes. However, in this case, first we have to transform the $(\tilde{I}_p + \tilde{L}_c)$ integrals into the $(\Omega_{0,0}^{\tilde{L}_c}|\tilde{L}_c)_{0,0,0}$ ones according to Eq. (31), so the scaling itself and the contraction of the primitive functions will become more expensive because of the increased number of intermediates. In most of the cases, the second route is more efficient when the second function in the bra side is s and also for $(pp|)$ bras. Otherwise the first option is more efficient. Picking the better route for each shell triplet is 15%-5% and 3%-33% cheaper than going with only the first or second option, respectively, for **A** derivatives. These savings are 15%-4% and 2%-36% for **B** derivatives.

Tables V–VIII imply that for most scenarios the OS based algorithms provide the lowest FLOP count. Let us take a look at if any benefit can be predicted from using the best suited algorithm for each shell triplet compared to applying exclusively OS based schemes. Our results show that the MD schemes

do not offer significant advantage in any of the cases. It turns out that for a lot of $L_c = 0$ or $\tilde{L}_c = 0$ shell triplets the Rys scheme provides the smallest FLOP count. However, the gain is rather limited since the saving it introduces is less than 1%; thus, in the following, we do not consider this algorithm. The GHP method, in turn, considerably lowers the FLOP count for certain shell triplets. If we use the most advantageous OS or GHP algorithm for each triplet, the FLOP requirement is reduced by 6%-1%, 7%-1%, and 17%-5% for the **A**, **B**, and **C** derivatives, respectively. For the **A** and **B** derivatives, the $\text{GHP}_{\text{Mixed,A1}}$ and $\text{GHP}_{\text{Mixed,B1}}$ schemes are preferred for most of the shell triplets with (ssl) , (psl) , and $(pp|)$ bras, while for higher bra angular momenta, the GHP recursions become inefficient with the investigated basis sets. In the case of **C** derivatives, the $\text{GHP}_{\text{Mixed,C1}}$ algorithm is the best method for a number of $L_b = 0$ and $L_b = 1$ cases. As \tilde{L}_c increases, the advantage of Eq. (32) over Eq. (30) becomes more pronounced, although for higher L_b values, $\text{OS}_{\text{Mixed,C}}$ is still superior. Compared to using only ${}^3\text{OS}_{\text{Cart,AB}}$, the combined application of ${}^3\text{OS}_{\text{Cart,AB}}$ and ${}^3\text{GHP}_{\text{Herm,AB1}}$ reduces the cost of calculating the derivatives by 16%-2%. Up to $(pp|)$ bras, the GHP method is the better choice, and for derivatives with (dsl) bra side, the \tilde{L}_c from which ${}^3\text{GHP}_{\text{Herm,AB1}}$ is more efficient depends on the basis set. As the cardinal number increases, the ratio of the primitive and contracted functions becomes smaller, so recursions at the contracted stage are less preferred. However, with increasing \tilde{L}_c , building up this angular momentum with Eq. (32) becomes more advantageous, and the primitive contraction is also cheaper when performed on the $(\tilde{I}_p + \tilde{L}_c)_{\lambda,\beta,\zeta}$ intermediates. Applying the most efficient ${}^3\text{OS}$ and ${}^3\text{GHP}$ schemes costs 22%-20% fewer FLOPs for the DZ-5Z bases than calculating the derivatives with respect to the cheapest two centers for every shell triplet and then using Eq. (18). This is an improved result compared to applying only the OS or GHP schemes since now we utilize both the advantage of GHP for small-angular-momentum bra sides and the superiority of OS for higher angular momenta. If we only consider ${}^3\text{OS}_{\text{Cart,AB}}$ and ${}^3\text{GHP}_{\text{Herm,AB1}}$, this reduction in cost changes to 20% for each basis set. For this reason, in the following, we only deal with the implementation of the ${}^3\text{OS}_{\text{Cart,AB}}$ schemes along with the ${}^3\text{GHP}$ algorithms that are predicted to be the best performing ones. It should be noted, however, that we are comparing very dissimilar algorithms with a different amount of memory operations. For example, the work associated with the transformation of one-center Hermite ERIs to the two-center ones via Eq. (31) is not accounted for since it does not require any FLOPs if the exponent scalings have been taken care of beforehand. For this reason, it is necessary to compare the actual wall time performances of the two approaches to decide which one is more beneficial. We also note that the most advantageous GHP algorithms turned out to be the ones using mixed or Hermite Gaussian ERIs. With the exception of the **C** derivatives, these always use the newly derived recursions, Eqs. (52)–(60).

IV. IMPLEMENTATION

We implemented the ${}^3\text{OS}_{\text{Cart,AB}}$ algorithms as well as the best performing ${}^3\text{GHP}$ schemes for each shell triplet, which

Algorithm 1. Overview of the separate recursion algorithms. D denotes the center of differentiation.

```

Loop for  $D$ 
  Loop for  $A$ 
    Loop for  $l_a$ 
      Loop for  $B$ 
        Loop for  $l_b$ 
          Loop for  $C$ 
            If  $D = A, B,$  or  $C$ 
              Loop for  $l_c$ 
                sss or C-S screening
                Call generated codes

```

Algorithm 2. Overview of the common recursion algorithms.

```

Loop for  $A$ 
  Loop for  $l_a$ 
    Loop for  $B$ 
      Loop for  $l_b$ 
        Loop for  $C$ 
          Loop for  $l_c$ 
            sss or C-S screening
            Call generated codes

```

include the ${}^3\text{GHP}_{\text{Mixed,AC1}}$, ${}^3\text{GHP}_{\text{Herm,AB1}}$, ${}^3\text{GHP}_{\text{Herm,AB2}}$, and ${}^3\text{GHP}_{\text{Cart,AC2}}$ routes, by means of automated code generation. There is an optimized, generated subroutine for every shell triplet. The spherical harmonic transformations, the contraction of the primitives, and the recursions are as vectorized as possible, and the unnecessary components of the intermediate shells are omitted from the calculation.⁶⁰ The efficiency of this implementation method was demonstrated for our integral-direct local MP2 approach,⁵¹ which calculates three-center ERIs using optimized subroutines³⁴ designed in the same manner as in this work. Schematic representations of the algorithms applying separate and common recursions are presented in Algorithms 1 and 2, respectively. Both of the algorithms use a general driver subroutine that handles three-center ERI and ERI derivative calculations for various methods. This driver contains the loops from A (first atom) to l_c (third angular momentum), initializes the necessary arrays and variables, and computes the scaled contraction coefficients for efficient primitive contraction.³⁴ The order of the loops for the primitive functions in the generated codes is a , b , and c . When prescreening is carried out at the primitive stage, Eq. (70) is applied before the loop for c using the maximal $\sqrt{|l_c|l_c|}$ and

$\sqrt{|(\partial l_c / \partial C_x | \partial l_c / \partial C_x)|}$ for the given C and l_c or the smallest and largest c when Eq. (73) is used. The prescreening ERIs and ERI derivatives for Eq. (70) are calculated using a general routine which applies the MD scheme with the four-center variant of Eq. (33) and Hermite Gaussian basis functions.¹⁸ This allows us to exploit most of the simplifications discussed in Sec. II F.

V. PERFORMANCE TESTS

In the following tests, a medium-sized molecule of 94 atoms, acetoacetyl-CoA, was used as the model system. Wall time and cache simulation measurements were performed using a single core of an Intel Core i7-7700K 4.2 GHz CPU. The cache memory of this hardware includes 32-32 kB of data and instruction level 1, 256 kB of level 2, and 8 MB of level 3 cache.

Table IX compares the different prescreening approaches using a threshold of $10^{-10} E_h$. We see that the C-S screening is more efficient when we apply screening only for the shell triplets, while with primitive level prescreening, the **sss prim** method gives a tighter bound. **sss prim**, however, increases the error introduced by roughly an order of magnitude, which reaches $10^{-7} E_h$ with the QZ basis. This could be prohibitive when one wishes to converge a geometry optimization to high accuracy.

Figure 1 shows the estimated FLOP counts and wall times for the shell triplets in the cc-pVTZ basis, where the counts of the GHP method are superior to those of the OS algorithms (the related FLOP counts can be found in the [supplementary material](#), namely, in Table SIV for the ${}^3\text{OS}_{\text{Cart,AB}}$ route, Tables SXXX to SXXXIX for the ${}^3\text{GHP}_{\text{Cart}}$ schemes, Tables SXL and SXLI for the ${}^3\text{GHP}_{\text{Herm}}$, and Tables SXLII to SLI for the ${}^3\text{GHP}_{\text{Mixed}}$ methods). The predictions correlate well with the timings without prescreening up to the (*pslg*) triplets. For the other cases, the discrepancy can be attributed to work that is not accounted for in the FLOP counts, e.g., the rearrangement of one-center ERIs to the two-center ones. The GHP based methods also require more Boys function evaluations. Also, ${}^3\text{OS}_{\text{Cart,AB}}$ always uses the Cartesian HRR, Eq. (23), which is more vectorizable than Eq. (52) or Eq. (55). When prescreening is applied on the primitive level, the ${}^3\text{OS}_{\text{Cart,AB}}$ algorithm always performs better than the GHP ones. This is understandable since the main merit of the GHP scheme is to move much of the work to the contracted stage.

TABLE IX. Comparison of the C-S and sss prescreening approaches for acetoacetyl-CoA with the cc-pVXZ ($X = D, T, Q$) basis sets. Efficiency refers to the percentage of derivatives screened out, while accuracy is the maximum error larger than 10^{-10} a.u. in the final AO ERI derivatives.

	DZ		TZ		QZ	
	Efficiency	Accuracy	Efficiency	Accuracy	Efficiency	Accuracy
sss	35	$4.7 \cdot 10^{-10}$	40	$9.3 \cdot 10^{-9}$	43	$6.6 \cdot 10^{-8}$
C-S	45	...	46	...	47	...
sss prim	78	$7.6 \cdot 10^{-9}$	78	$3.7 \cdot 10^{-8}$	80	$3.5 \cdot 10^{-7}$
C-S prim	70	...	74	...	76	...

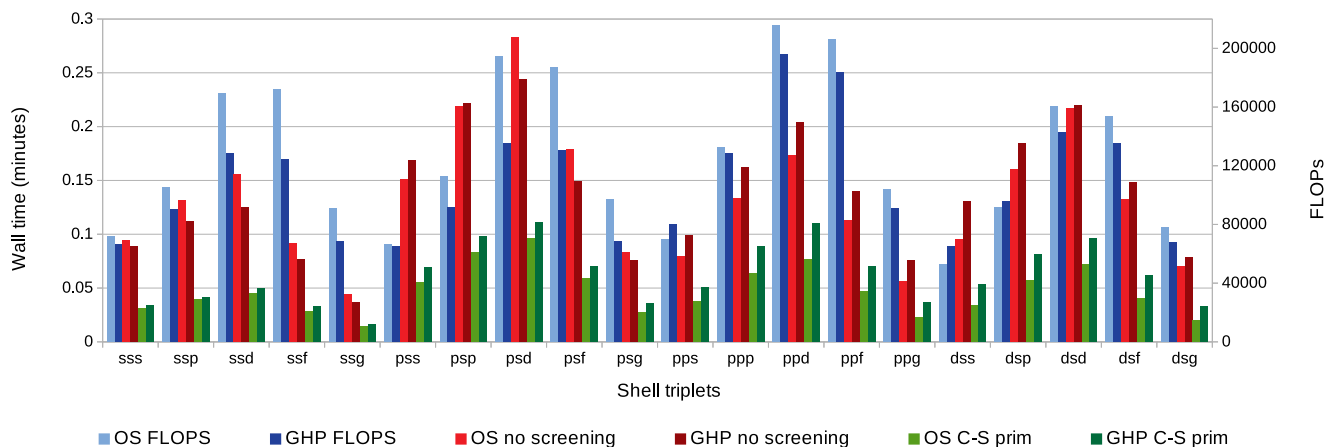


FIG. 1. FLOP counts and wall times for the derivatives of the (sss) – (dsf) shell triplets calculated with the implemented ^3OS and ^3GHP algorithms for acetoacetyl-CoA with the cc-pVTZ basis set.

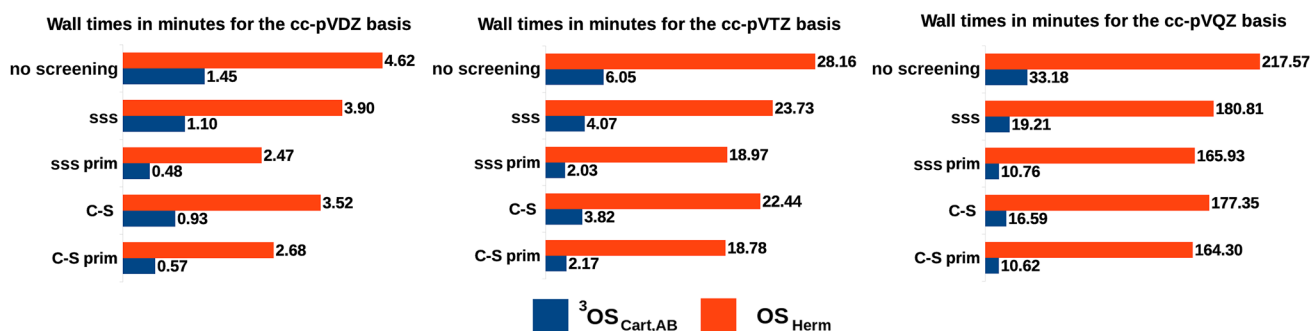


FIG. 2. Wall times in minutes for the execution of Algorithm 1 (red) and Algorithm 2 (blue) for acetoacetyl-CoA with the cc-pVXZ ($X = D, T, Q$) basis sets.

To compare the separate and common recursion algorithms, we measured the wall times required for the $^3\text{OS}_{\text{Cart,AB}}$ and the OS_{Herm} algorithms, of which the latter we previously implemented by automated code generation, to calculate the derivatives of every shell triplet for our test system. The wall times for the full execution of Algorithms 1 and 2 are displayed in Fig. 2, while the contribution of the generated codes to the timings can be seen in Fig. 3. Comparing the results in Fig. 3 with the times measured without prescreening in Fig. 2 we see that, while for $^3\text{OS}_{\text{Cart,AB}}$, the runtime is almost equivalent to the time spent in the generated codes, and for OS_{Herm} , there is a significant discrepancy increasing with the basis set. This is because of the greater overhead of Algorithm 1 compared to Algorithm 2. Without prescreening, $^3\text{OS}_{\text{Cart,AB}}$ offers speedups of factors of 3.2, 4.7, and 6.6 compared to OS_{Herm} with the DZ, TZ, and QZ bases, respectively. These increase to 4.7, 8.7, and 15.5 if we apply **C–S prim** type prescreening. Screening increases the efficiency of Algorithm 2 more than it does so for Algorithm 1 because it reduces the work performed in the generated codes, which contribute to the total runtime with a higher percentage in the common recursion approach. It is interesting to compare the ratios of the wall times for the two schemes in Fig. 3 with the quotients of the summed FLOP counts of $\text{OS}_{\text{Herm,A}}$, $\text{OS}_{\text{Herm,B}}$, and $\text{OS}_{\text{Herm,C}}$ and the FLOP counts of $^3\text{OS}_{\text{Cart,AB}}$ from Table V. The ratios of the wall times with the DZ, TZ, and QZ basis sets are 2.1, 2.2, and 2.4, respectively. These values for the FLOP counts are 1.9, 2.0, and 2.2.

The good agreement shows that FLOP counting for similar algorithms is a valid tool for predicting their relative runtime performances.

Finally, Table X contains the results of CPU cache simulations for H_2O_2 performed by the VALGRIND⁶¹ program suite. In most aspects, Algorithm 2 produces a smaller percentage of cache misses because of the more local nature of the common recursion route. The exceptions are the level 1 data read and write misses, where Algorithm 1 performs better. This observation is in accordance with our earlier conclusion³⁴ that most data cache misses occur during the contraction of the primitives because of the large size of the arrays involved. Since in Algorithm 2 we calculate derivatives with respect to two centers, there are more integral derivatives to contract, and therefore, there are more cache misses.

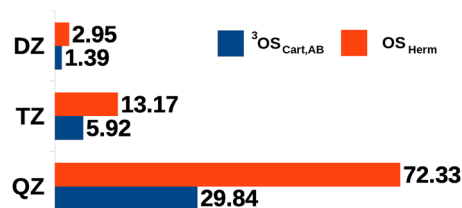


FIG. 3. Contribution (in minutes) of the generated codes to the wall time of Algorithm 1 (red) and Algorithm 2 (blue) without prescreening for acetoacetyl-CoA with the cc-pVXZ ($X = D, T, Q$) basis sets.

TABLE X. Cache simulation results for the evaluation of the ERI derivatives of H_2O_2 with the cc-pVXZ ($X = \text{D, T, Q}$) basis sets. The results indicate the percentage of cache misses relative to all events. Notations: I and D mean instructions and data, respectively, 1 and L mean that the referenced memory address did not have a copy in the first level of the CPU cache memory or in all of it, respectively. r and w refer to data read and data write operations, respectively.

	DZ		TZ		QZ	
	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
II	0.21	0.32	0.76	0.72	1.75	1.23
IL	0.01	0.01	0.04	0.01	0.08	0.03
DrI	0.29	1.32	0.48	2.13	0.87	2.71
DrL	0.01	0.02	0.01	0.01	0.01	0.01
DwI	1.81	3.72	3.97	6.51	6.75	8.37
DwL	0.05	0.05	0.28	0.05	0.65	0.13

VI. CONCLUSIONS

We optimized the evaluation of the first derivatives of three-center Coulomb integrals over contracted spherical harmonic Gaussians in two steps. First, we determined estimations of the FLOP requirements of several schemes, including the ones that apply mixed Gaussian ERIs. The FLOP counts suggest that these integrals containing both Cartesian and Hermite Gaussians are better suited for the task when we compute the derivatives for one center at a time (Algorithm 1), except for the bra-side derivatives with $L_b \geq 3$. The OS method is predicted to be superior for most of the cases; however, for shell triplets up to (*ppl* bra sides, $\text{GHP}_{\text{Mixed}}$ produces the lowest FLOP counts. For Algorithm 2, the most efficient scheme is ${}^3\text{OS}_{\text{Cart,AB}}$, with the exception of ${}^3\text{GHP}_{\text{Mixed,AC1}}$ for (*ssl* and (*psl*), and ${}^3\text{GHP}_{\text{Herm,AB1}}$ for (*ppl*) and a few other shell triplets with $L_b = 0$ and high L_c . In the second step, we compared the wall time performances of certain selected methods. The results for ${}^3\text{OS}_{\text{Cart,AB}}$ and those ${}^3\text{GHP}$ schemes which were predicted to be the best based on the FLOP requirement suggest that FLOP counts alone are insufficient to decide between two methods that are algorithmically very different. On the other hand, the wall times for the generated codes for ${}^3\text{OS}_{\text{Cart,AB}}$ and OS_{Herm} reproduce the ratios of the FLOP counts reasonably well, which suggests that this type of prediction is useful when variants of the same basic algorithm are compared. When we do not only measure the timings for the generated codes but also measure the total time for the execution of Algorithms 1 and 2, we get significantly larger speedups for the common recursion scheme than expected from the FLOP counts, which is the result of the increased overhead in Algorithm 1.

We discussed an efficient approximation of the Cauchy–Schwarz upper bound for ERI derivatives and an inexact screening quantity derived from the Boys function. The test calculations for these approaches reveal that it is important to screen both the primitive integrals and the shell triplets. The **sss prim** type screening is slightly more efficient, but it can introduce errors which are three orders of magnitude greater than the tolerance. This problem does not arise with the **C–S prim** approach. When primitive prescreening is applied, the ${}^3\text{OS}_{\text{Cart,AB}}$ scheme always performs better than the ${}^3\text{GHP}$ algorithms. Based on our results, we recommend the usage of the

Schwarz screening instead of the approximate bound, and the application of ${}^3\text{OS}_{\text{Cart,AB}}$ when it is not necessary to compute the derivatives with respect to one center at a time. For the separate recursion schemes, the OS_{Mixed} approach is predicted to be the most efficient. Our results indicate, however, that in this case the overhead of the loops over the shell triplets of the molecule dominates the wall time rather than the evaluation of the derivatives.

SUPPLEMENTARY MATERIAL

See [supplementary material](#) for the method by which the index ranges of the recursions in Sec. II were determined, for the index ranges of the recursions in Sec. II D, for the detailed FLOP counts of the investigated schemes, and for the geometries of the applied test molecules.

ACKNOWLEDGMENTS

The authors are grateful for the financial support from the National Research, Development, and Innovation Office (NKFIH, Grant No. KKP126451).

APPENDIX: RECURRENCE RELATIONS FOR HERMITE AND MIXED GAUSSIAN INTEGRALS

Here we present the derivations for the new equations introduced in Sec. II: the $\tilde{l}_a \rightarrow l_b$, $l_a \rightarrow \tilde{l}_b$, and $l_a \rightarrow \tilde{l}_c$ translations in OS and the recurrence relations for Hermite and mixed Gaussian integrals in the MD and Rys schemes.

1. Obara–Saika recursions

The validity of Eq. (28) can be easily seen by substituting Eq. (3) into the bra side overlap distribution $\tilde{H}_{i_a} G_{i_b+1}$ and then applying Eqs. (7) and (5),

$$\begin{aligned} \tilde{H}_{i_a} G_{i_b+1} &= x_B \tilde{H}_{i_a} G_{i_b} = x_A \tilde{H}_{i_a} G_{i_b} + X_{AB} \tilde{H}_{i_a} G_{i_b} \\ &= \tilde{H}_{i_a+1} G_{i_b} + \frac{\tilde{l}_a}{2a} \tilde{H}_{i_a-1} G_{i_b} + X_{AB} \tilde{H}_{i_a} G_{i_b}. \end{aligned} \quad (\text{A1})$$

Equation (29) is proven in the same manner by rewriting $G_{i_a} \tilde{H}_{i_b+1}$. For Eq. (30), let us express the incremented integral according to the expansion in the McMurchie–

Davidson scheme (see Sec. II C for the definition of the quantities) as

$$(\mathbf{I}_a \mathbf{0} | [\tilde{\mathbf{I}}_c + \tilde{\mathbf{I}}_x])^{(n)} = \left(\frac{1}{2c}\right)^{\tilde{l}_c + \tilde{l}_x} \sum_{\tilde{i}_p, \tilde{j}_p, \tilde{k}_p} E_{\tilde{i}_p}^{i_a, 0} E_{\tilde{j}_p}^{j_a, 0} E_{\tilde{k}_p}^{k_a, 0} (-1)^{\tilde{l}_c + \tilde{l}_x} \times (\tilde{\mathbf{I}}_c + \tilde{\mathbf{I}}_p + \tilde{\mathbf{I}}_x)^{(n)}, \quad (\text{A2})$$

where we expanded the Cartesian overlap distribution in the bra in terms of unscaled Hermite Gaussians defined by Eq. (8), and the scaling with the exponent present in Eq. (2) has been accounted for the ket side function. The sums over i_p, j_p , and k_p run from 0 to i_a, j_a , and k_a , respectively. Inserting Eq. (32) into Eq. (A2), we get

$$\begin{aligned} (\mathbf{I}_a \mathbf{0} | [\tilde{\mathbf{I}}_c + \tilde{\mathbf{I}}_x])^{(n)} &= X_{\text{PC}} \left(\frac{1}{2c}\right)^{\tilde{l}_c + \tilde{l}_x} \sum_{\tilde{i}_p, \tilde{j}_p, \tilde{k}_p} E_{\tilde{i}_p}^{i_a, 0} E_{\tilde{j}_p}^{j_a, 0} E_{\tilde{k}_p}^{k_a, 0} (-1)^{\tilde{l}_c + \tilde{l}_x} (\tilde{\mathbf{I}}_c + \tilde{\mathbf{I}}_p)^{(n+1)} \\ &\quad + \left(\frac{1}{2c}\right)^{\tilde{l}_c + \tilde{l}_x} \sum_{\tilde{i}_p, \tilde{j}_p, \tilde{k}_p} E_{\tilde{i}_p}^{i_a, 0} E_{\tilde{j}_p}^{j_a, 0} E_{\tilde{k}_p}^{k_a, 0} (-1)^{\tilde{l}_c + \tilde{l}_x} (\tilde{i}_p + \tilde{i}_c) (\tilde{\mathbf{I}}_c + \tilde{\mathbf{I}}_p - \tilde{\mathbf{I}}_x)^{(n+1)} \\ &= -\frac{1}{2c} X_{\text{PC}} (\mathbf{I}_a \mathbf{0} | \tilde{\mathbf{I}}_c)^{(n+1)} + \frac{\tilde{i}_c}{4c^2} (\mathbf{I}_a \mathbf{0} | [\tilde{\mathbf{I}}_c - \tilde{\mathbf{I}}_x])^{(n+1)} \\ &\quad + \left(\frac{1}{2c}\right)^{\tilde{l}_c + \tilde{l}_x} \sum_{\tilde{i}_p, \tilde{j}_p, \tilde{k}_p} \tilde{i}_p E_{\tilde{i}_p}^{i_a, 0} E_{\tilde{j}_p}^{j_a, 0} E_{\tilde{k}_p}^{k_a, 0} (-1)^{\tilde{l}_c + \tilde{l}_x} (\tilde{\mathbf{I}}_c + \tilde{\mathbf{I}}_p - \tilde{\mathbf{I}}_x)^{(n+1)}, \end{aligned} \quad (\text{A3})$$

where the first two terms in the last equation arise from substituting back Eq. (A2). By factoring out $1/(2p)$ in the last term, we can use the recurrence⁵⁵

$$2p \tilde{i}_p E_{\tilde{i}_p}^{i_a, 0} = i_a E_{\tilde{i}_p - 1}^{i_a - 1, 0}, \quad (\text{A4})$$

and noting that $E_{\tilde{i}_p}^{i_a, i_b} = 0$ if $\tilde{i}_p < 0$ or $\tilde{i}_p > i_a + i_b$, we can substitute Eq. (A2) for the last term. It becomes $-i_a/(4pc) (\mathbf{I}_a - \mathbf{1}_x | \mathbf{0} | \tilde{\mathbf{I}}_c)^{(n+1)}$, and we arrive at Eq. (30).

2. McMurchie–Davidson method

For brevity, let us only deal with the x -dependent part of the auxiliary functions defined by Eqs. (37) and (40). We only show the derivation of Eq. (38) since the remaining recursions for Hermite and mixed Gaussian ERIs can be acquired in an analogous way. We use the auxiliary function

$$\Omega_{\tilde{i}_a, \tilde{i}_b}^{\tilde{i}_p} = \left(\frac{\partial}{2a\partial A_x}\right)^{\tilde{i}_a} \left(\frac{\partial}{2b\partial B_x}\right)^{\tilde{i}_b} \left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p} \Lambda_{ab}, \quad (\text{A5})$$

where $\Lambda_{ab} = \kappa_{ab} \exp(-px_p^2) = \exp(-ax_A^2) \exp(-bx_B^2)$. We will also apply the following relations:

$$\left[\left(\frac{\partial}{\partial A_x}\right)^{\tilde{i}_a}, X_{\text{PA}}\right] = -\tilde{i}_a \frac{b}{p} \left(\frac{\partial}{\partial A_x}\right)^{\tilde{i}_a - 1}, \quad (\text{A6})$$

$$\left[\left(\frac{\partial}{\partial B_x}\right)^{\tilde{i}_b}, X_{\text{PA}}\right] = \tilde{i}_b \frac{b}{p} \left(\frac{\partial}{\partial B_x}\right)^{\tilde{i}_b - 1}, \quad (\text{A7})$$

$$\left[\left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p}, X_{\text{PA}}\right] = \left[\left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p}, X_{\text{PB}}\right] = 0, \quad (\text{A8})$$

$$\left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p + 1} \Lambda_{ab} = \left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p} 2p x_p \Lambda_{ab}. \quad (\text{A9})$$

The commutators can be proven by induction. For Eq. (A8), we also need $\partial/\partial P_x = \partial/\partial A_x + \partial/\partial B_x$, $X_{\text{PA}} = -b/p X_{\text{AB}}$, and $X_{\text{PB}} = a/p X_{\text{AB}}$. Incrementing i_a , we can write

$$\begin{aligned} \Omega_{\tilde{i}_a + \tilde{l}_x, \tilde{i}_b}^{\tilde{i}_p} &= \left(\frac{\partial}{2a\partial A_x}\right)^{\tilde{i}_a + \tilde{l}_x} \left(\frac{\partial}{2b\partial B_x}\right)^{\tilde{i}_b} \left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p} \Lambda_{ab} \\ &= \left(\frac{1}{2a}\right)^{\tilde{i}_a + \tilde{l}_x} \left(\frac{\partial}{\partial A_x}\right)^{\tilde{i}_a} \left(\frac{\partial}{2b\partial B_x}\right)^{\tilde{i}_b} \left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p} 2ax_A \Lambda_{ab} \\ &= \left(\frac{\partial}{2a\partial A_x}\right)^{\tilde{i}_a} \left(\frac{\partial}{2b\partial B_x}\right)^{\tilde{i}_b} \left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p} (x_p + X_{\text{PA}}) \Lambda_{ab} \\ &= \left(\frac{\partial}{2a\partial A_x}\right)^{\tilde{i}_a} \left(\frac{\partial}{2b\partial B_x}\right)^{\tilde{i}_b} \left[\frac{1}{2p} \left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p + 1} + X_{\text{PA}} \left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p}\right] \Lambda_{ab} \\ &= \frac{1}{2p} \Omega_{\tilde{i}_a, \tilde{i}_b}^{\tilde{i}_p + 1} + \left(\frac{1}{2b}\right)^{\tilde{i}_b} \left(\frac{\partial}{2a\partial A_x}\right)^{\tilde{i}_a} \left[\tilde{i}_b \frac{b}{p} \left(\frac{\partial}{\partial B_x}\right)^{\tilde{i}_b - 1} + X_{\text{PA}} \left(\frac{\partial}{\partial B_x}\right)^{\tilde{i}_b}\right] \left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p} \Lambda_{ab} \\ &= \frac{1}{2p} \Omega_{\tilde{i}_a, \tilde{i}_b}^{\tilde{i}_p + 1} + \frac{\tilde{i}_b}{2p} \Omega_{\tilde{i}_a, \tilde{i}_b - 1}^{\tilde{i}_p} + \left(\frac{1}{2a}\right)^{\tilde{i}_a} \left[-\tilde{i}_a \frac{b}{p} \left(\frac{\partial}{\partial A_x}\right)^{\tilde{i}_a - 1} + X_{\text{PA}} \left(\frac{\partial}{\partial A_x}\right)^{\tilde{i}_a}\right] \left(\frac{\partial}{2b\partial B_x}\right)^{\tilde{i}_b} \left(\frac{\partial}{\partial P_x}\right)^{\tilde{i}_p} \Lambda_{ab} \\ &= \frac{1}{2p} \Omega_{\tilde{i}_a, \tilde{i}_b}^{\tilde{i}_p + 1} + \frac{\tilde{i}_b}{2p} \Omega_{\tilde{i}_a, \tilde{i}_b - 1}^{\tilde{i}_p} - \frac{\tilde{i}_a}{2p} \frac{b}{a} \Omega_{\tilde{i}_a - 1, \tilde{i}_b}^{\tilde{i}_p} + X_{\text{PA}} \Omega_{\tilde{i}_a, \tilde{i}_b}^{\tilde{i}_p}. \end{aligned} \quad (\text{A10})$$

Here we first differentiate with respect to A_x and then rewrite the result according to Eq. (7). For $x_P \Lambda_{ab}$, we substitute Eq. (A9), and for $X_{PA} \Lambda_{ab}$, we substitute Eq. (A8). For the first term of the resulting expression, we substitute Eq. (A5) to get $1/2p\Omega_{i_a, \tilde{i}_b}^{\tilde{i}_p+1}$, and for the second term Eq. (A7). After the fifth equation sign, we substitute Eq. (A5) again for the first term in the bracket to get $\tilde{i}_b/2p\Omega_{i_a, \tilde{i}_b}^{\tilde{i}_p}$, and for the second we use Eq. (A6). After the sixth equation sign, we can apply Eq. (A5) for both terms in the bracket to gain the remaining two terms in the recursion.

3. Rys quadrature

To derive the 2D recurrence relations for Hermite and mixed Gaussian integrals, we mostly follow the methods applied in Refs. 18 and 55. Here we outline the process for the Hermite 2D ERIs. We exploit that the 2D integrals can be expressed by the modified Hermite polynomials⁵⁵ \mathcal{H} according to the McMurchie–Davidson scheme as

$$\Theta_x^{\tilde{i}_a, \tilde{i}_b, \tilde{i}_c}(t_r^2) = \left(-\frac{1}{2c}\right)^{\tilde{i}_c} \sum_{\tilde{i}_p=0}^{\tilde{i}_a+\tilde{i}_b} \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b} \mathcal{H}_{\tilde{i}_p+\tilde{i}_c}(t_r^2). \quad (\text{A11})$$

From this, after incrementing indices and using the recurrence relation⁵⁵ for \mathcal{H} ,

$$\mathcal{H}_{\tilde{i}_p+\tilde{i}}(t_r^2) = -2\alpha t_r^2 [X_{PC} \mathcal{H}_{\tilde{i}_p}(t_r^2) + \tilde{i}_p \mathcal{H}_{\tilde{i}_p-\tilde{i}}(t_r^2)], \quad (\text{A12})$$

and the ones for the $\mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b}$ expansion coefficients for the transformation of the unscaled Hermite Gaussians centered on \mathbf{P} to a Hermite overlap distribution, Eqs. (65), (67), and (68) can be proven. The required recurrences for the \mathcal{E} coefficients can be derived from

$$\tilde{H}_{\tilde{i}_a} \tilde{H}_{\tilde{i}_b} = \sum_{\tilde{i}_p=0}^{\tilde{i}_a+\tilde{i}_b} \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b} H_{\tilde{i}_p} \quad (\text{A13})$$

in an analogous manner as it is done for Eqs. (101) and (103) of Ref. 18. We obtain

$$\tilde{i}_a \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a-1, \tilde{i}_b} + \tilde{i}_b \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b-1} = 2p(\tilde{i}_p+1) \mathcal{E}_{\tilde{i}_p+1}^{\tilde{i}_a, \tilde{i}_b} \quad (\text{A14})$$

and

$$\mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a+1, \tilde{i}_b} = X_{PA} \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b} + \frac{1}{2p} \mathcal{E}_{\tilde{i}_p-1}^{\tilde{i}_a, \tilde{i}_b} + (\tilde{i}_p+1) \mathcal{E}_{\tilde{i}_p+1}^{\tilde{i}_a, \tilde{i}_b} - \frac{\tilde{i}_a}{2a} \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a-1, \tilde{i}_b}. \quad (\text{A15})$$

Substituting Eq. (A14) into Eq. (A15), we get

$$\mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a+1, \tilde{i}_b} = X_{PA} \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b} + \frac{1}{2p} \left(\mathcal{E}_{\tilde{i}_p-1}^{\tilde{i}_a, \tilde{i}_b} - \tilde{i}_a \frac{b}{a} \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a-1, \tilde{i}_b} + \tilde{i}_b \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b-1} \right), \quad (\text{A16})$$

and for the other bra index,

$$\mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b+1} = X_{PB} \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b} + \frac{1}{2p} \left(\mathcal{E}_{\tilde{i}_p-1}^{\tilde{i}_a, \tilde{i}_b} + \tilde{i}_a \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a-1, \tilde{i}_b} - \tilde{i}_b \frac{a}{b} \mathcal{E}_{\tilde{i}_p}^{\tilde{i}_a, \tilde{i}_b-1} \right). \quad (\text{A17})$$

Equations (A16) and (A17) differ from the corresponding recursions of the E coefficients for Cartesian ERIs⁵⁵ in the $-b/a$ and $-a/b$ factors, respectively. Incrementing \tilde{i}_a in

Eq. (A11) and then using Eqs. (A16), (A12), and (A14), we get

$$\begin{aligned} \Theta_x^{\tilde{i}_a+1, \tilde{i}_b, \tilde{i}_c}(t_r^2) &= \left(X_{PA} - \frac{\alpha}{p} X_{PC} t_r^2\right) \Theta_x^{\tilde{i}_a, \tilde{i}_b, \tilde{i}_c}(t_r^2) \\ &+ \frac{\tilde{i}_a}{2p} \left(-\frac{b}{a} - \frac{\alpha}{p} t_r^2\right) \Theta_x^{\tilde{i}_a-1, \tilde{i}_b, \tilde{i}_c}(t_r^2) \\ &+ \frac{\tilde{i}_b}{2p} \left(1 - \frac{\alpha}{p} t_r^2\right) \Theta_x^{\tilde{i}_a, \tilde{i}_b-1, \tilde{i}_c}(t_r^2) \\ &+ \frac{\tilde{i}_c t_r^2}{2(p+c)} \Theta_x^{\tilde{i}_a, \tilde{i}_b, \tilde{i}_c-1}(t_r^2) \end{aligned} \quad (\text{A18})$$

and the similar equation for \tilde{i}_b ,

$$\begin{aligned} \Theta_x^{\tilde{i}_a, \tilde{i}_b+1, \tilde{i}_c}(t_r^2) &= \left(X_{PB} - \frac{\alpha}{p} X_{PC} t_r^2\right) \Theta_x^{\tilde{i}_a, \tilde{i}_b, \tilde{i}_c}(t_r^2) \\ &+ \frac{\tilde{i}_a}{2p} \left(1 - \frac{\alpha}{p} t_r^2\right) \Theta_x^{\tilde{i}_a-1, \tilde{i}_b, \tilde{i}_c}(t_r^2) \\ &+ \frac{\tilde{i}_b}{2p} \left(-\frac{a}{b} - \frac{\alpha}{p} t_r^2\right) \Theta_x^{\tilde{i}_a, \tilde{i}_b-1, \tilde{i}_c}(t_r^2) \\ &+ \frac{\tilde{i}_c t_r^2}{2(p+c)} \Theta_x^{\tilde{i}_a, \tilde{i}_b, \tilde{i}_c-1}(t_r^2). \end{aligned} \quad (\text{A19})$$

If we increment \tilde{i}_c in Eq. (A11) and then use Eqs. (A12) and (A14), we obtain the recursion for \tilde{i}_c , which is

$$\begin{aligned} \Theta_x^{\tilde{i}_a, \tilde{i}_b, \tilde{i}_c+1}(t_r^2) &= \frac{\alpha}{c} X_{PC} t_r^2 \Theta_x^{\tilde{i}_a, \tilde{i}_b, \tilde{i}_c}(t_r^2) + \frac{\tilde{i}_c}{2c} \left(-\frac{\alpha t_r^2}{c}\right) \Theta_x^{\tilde{i}_a, \tilde{i}_b, \tilde{i}_c-1}(t_r^2) \\ &+ \frac{t_r^2}{2(p+c)} [\tilde{i}_a \Theta_x^{\tilde{i}_a-1, \tilde{i}_b, \tilde{i}_c}(t_r^2) + \tilde{i}_b \Theta_x^{\tilde{i}_a, \tilde{i}_b-1, \tilde{i}_c}(t_r^2)]. \end{aligned} \quad (\text{A20})$$

Equation (68) is obtained by subtracting $\Theta_x^{\tilde{i}_a+1, \tilde{i}_b, \tilde{i}_c}(t_r^2)$ from $\Theta_x^{\tilde{i}_a, \tilde{i}_b+1, \tilde{i}_c}(t_r^2)$. Concerning mixed Gaussian 2D ERIs, Eq. (A14) retains the same form. The recursion for incrementing the Hermite part of the mixed bra side is either Eq. (A16) or (A17), while the index referring to the Cartesian component is incremented by the same equations as for the Cartesian E coefficients.⁵⁵

¹Y. Yamaguchi and H. F. Schaefer III, “Analytic derivative methods in molecular electronic structure theory: A new dimension to quantum chemistry and its applications to spectroscopy,” in *Handbook of High-Resolution Spectroscopy*, edited by M. Quack and F. Merkt (Wiley, Chichester, 2011), pp. 325–362.

²For a general review of analytic derivative techniques see, P. Pulay, in *Modern Electronic Structure Theory*, edited by D. R. Yarkony (World Scientific, Singapore, 1995), Vol. 2.

³R. D. Amos and J. E. Rice, *Comput. Phys. Rep.* **10**, 147 (1989).

⁴H. F. King and M. Dupuis, *J. Comput. Phys.* **21**, 144 (1976).

⁵M. Dupuis, J. Rys, and H. F. King, *J. Chem. Phys.* **65**, 111 (1976).

⁶J. Rys, M. Dupuis, and H. F. King, *J. Comput. Chem.* **4**, 154 (1983).

⁷R. Lindh, U. Ryu, and B. Liu, *J. Chem. Phys.* **95**, 5889 (1991).

⁸H. B. Schlegel, J. S. Binkley, and J. A. Pople, *J. Chem. Phys.* **80**, 1976 (1984).

⁹N. Flocke and V. Lotrich, *J. Comput. Chem.* **29**, 2722 (2008).

¹⁰L. E. McMurchie and E. R. Davidson, *J. Comput. Phys.* **26**, 218 (1978).

¹¹T. Helgaker and P. R. Taylor, *Theor. Chim. Acc.* **83**, 177 (1992).

¹²P. M. W. Gill, M. Head-Gordon, and J. A. Pople, *Int. J. Quantum Chem.* **36**, 269 (1989).

¹³P. M. W. Gill and J. A. Pople, *Int. J. Quantum Chem.* **40**, 753 (1991).

¹⁴P. M. W. Gill, M. Head-Gordon, and J. A. Pople, *J. Phys. Chem.* **94**, 5564 (1990).

¹⁵I. Panas, *Chem. Phys. Lett.* **184**, 86 (1991).

¹⁶M. Hayami, J. Seino, and H. Nakai, *J. Chem. Phys.* **142**, 204110 (2015).

- ¹⁷M. Hayami, J. Seino, and H. Nakai, *Int. J. Quantum Chem.* **118**, e25640 (2018).
- ¹⁸S. Reine, E. Tellgren, and T. Helgaker, *Phys. Chem. Chem. Phys.* **9**, 4771 (2007).
- ¹⁹S. Obara and A. Saika, *J. Chem. Phys.* **84**, 3963 (1986).
- ²⁰M. Head-Gordon and J. A. Pople, *J. Chem. Phys.* **89**, 5777 (1988).
- ²¹A. Banerjee, J. O. Jensen, and J. Simons, *J. Chem. Phys.* **82**, 4566 (1985).
- ²²A. Komornicki, K. Ishida, K. Morokuma, R. Ditchfield, and M. Conrad, *Chem. Phys. Lett.* **45**, 595 (1977).
- ²³M. Dupuis and H. King, *J. Chem. Phys.* **68**, 3998 (1978).
- ²⁴L. R. Kahn, *J. Chem. Phys.* **75**, 3962 (1981).
- ²⁵M. A. Vincent, P. Saxe, and H. F. Schaefer III, *Chem. Phys. Lett.* **94**, 351 (1983).
- ²⁶M. A. Vincent and H. F. Schaefer III, *Theor. Chim. Acc.* **64**, 21 (1983).
- ²⁷M. Page, P. Saxe, G. F. Adams, and B. H. Lengsfeld, *Chem. Phys. Lett.* **104**, 587 (1984).
- ²⁸S. F. Boys and I. Shavitt, Report No. WIS-AF-13, University of Wisconsin Naval Research Laboratory, 1959.
- ²⁹E. J. Baerends, D. E. Ellis, and P. Ros, *Chem. Phys.* **2**, 41 (1973).
- ³⁰J. L. Whitten, *J. Chem. Phys.* **58**, 4496 (1973).
- ³¹B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).
- ³²B. I. Dunlap, *Phys. Chem. Chem. Phys.* **2**, 2113 (2000).
- ³³R. Ahlrichs, *Phys. Chem. Chem. Phys.* **6**, 5119 (2004).
- ³⁴G. Samu and M. Kállay, *J. Chem. Phys.* **146**, 204101 (2017).
- ³⁵A. M. Köster, *J. Chem. Phys.* **104**, 4114 (1996).
- ³⁶A. M. Köster, *J. Chem. Phys.* **118**, 9943 (2003).
- ³⁷P. Calaminici, V. D. Domínguez-Soria, G. Geudtner, E. Hernández-Marín, and A. M. Köster, *Theor. Chem. Acc.* **115**, 221 (2006).
- ³⁸E. F. Valeev, Libint: A library for the evaluation of molecular integrals of many-body operators over Gaussian functions, <http://libint.valeyev.net/>.
- ³⁹E. F. Valeev and C. L. Janssen, *J. Chem. Phys.* **121**, 1214 (2004).
- ⁴⁰H.-J. Werner, G. Knizia, and F. R. Manby, *Mol. Phys.* **109**, 407 (2011).
- ⁴¹D. S. Hollman, H. F. Schaefer III, and E. F. Valeev, *J. Chem. Phys.* **142**, 154106 (2015).
- ⁴²P. M. W. Gill, B. G. Johnson, and J. A. Pople, *Chem. Phys. Lett.* **217**, 65 (1994).
- ⁴³H. Horn, H. Weiss, M. Häser, M. Ehrig, and R. Ahlrichs, *J. Comput. Chem.* **12**, 1058 (1991).
- ⁴⁴Y. Shao and M. Head-Gordon, *Chem. Phys. Lett.* **323**, 425 (2000).
- ⁴⁵R. Polly, H.-J. Werner, F. R. Manby, and P. J. Knowles, *Mol. Phys.* **102**, 2311 (2004).
- ⁴⁶S. Reine, E. Tellgren, A. Krapp, T. Kjærgaard, T. Helgaker, B. Jansik, S. Høst, and P. Salek, *J. Chem. Phys.* **129**, 104101 (2008).
- ⁴⁷S. F. Manzer, E. Epifanovsky, and M. Head-Gordon, *J. Chem. Theory Comput.* **11**, 518 (2014).
- ⁴⁸D. Mejía-Rodríguez and A. M. Köster, *J. Chem. Phys.* **141**, 124114 (2014).
- ⁴⁹D. Mejía-Rodríguez, X. Huang, J. M. del Campo, and A. M. Köster, *Adv. Quantum Chem.* **71**, 41 (2015).
- ⁵⁰C. Köppl and H.-J. Werner, *J. Chem. Theory Comput.* **12**, 3122 (2016).
- ⁵¹P. R. Nagy, G. Samu, and M. Kállay, *J. Chem. Theory Comput.* **12**, 4897 (2016).
- ⁵²M. Sierka, A. Hogekamp, and R. Ahlrichs, *J. Chem. Phys.* **118**, 9136 (2003).
- ⁵³A. Alvarez-Ibarra and A. M. Köster, *J. Chem. Phys.* **139**, 024102 (2013).
- ⁵⁴A. Alvarez-Ibarra and A. M. Köster, *Mol. Phys.* **113**, 3128 (2015).
- ⁵⁵T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic Structure Theory* (Wiley, Chichester, 2000).
- ⁵⁶T. P. Hamilton and H. F. Schaefer III, *Chem. Phys.* **150**, 163 (1991).
- ⁵⁷H. F. King, *J. Phys. Chem. A* **120**, 9348 (2016).
- ⁵⁸T. H. Dunning, Jr., *J. Chem. Phys.* **90**, 1007 (1989).
- ⁵⁹F. Weigend, A. Köhn, and C. Hättig, *J. Chem. Phys.* **116**, 3175 (2002).
- ⁶⁰U. Ryu, Y. S. Lee, and R. Lindh, *Chem. Phys. Lett.* **185**, 562 (1991).
- ⁶¹J. Weidendorfer, M. Kowarschik, and C. Trinitis, in Proceedings of the 4th International Conference on Computational Science (ICCS 2004), Krakow, Poland, 2004.