Towards the Minimal FLOP Count Cholesky Decomposition of Electron Repulsion Integrals

Tianyuan Zhang,[†] Xiaolin Liu,[†] Edward F. Valeev,[‡] and Xiaosong Li[*,†]

[†]*Department of Chemistry, University of Washington, Seattle, WA, 98195*

[‡]*Department of Chemistry, Virginia Tech, Blacksburg, VA, 24061*

E-mail: xsli@uw.edu

Abstract

As quantum chemistry calculations deal with molecular systems of increasing size, the memory requirement to store electron-repulsion integrals (ERIs) greatly outpaces the physical memory available in computing hardware. The Cholesky decomposition of ERIs provides a convenient yet accurate technique to reduce the storage requirement of integrals. Recent developments of a two-step algorithm have drastically reduced the memory operation (MOP) count, leaving the floating operation (FLOP) count as the last frontier of cost reduction in the Cholesky ERI algorithm. In this report, we introduce a dynamic integral tracking, reusing, and compression/elimination protocol embedded in the two-step Cholesky ERI method. Benchmark studies suggest that this technique becomes particularly advantageous when the basis set consists of many computationally expensive high-angular-momentum basis functions. With this dynamic ERI improvement, the Cholesky ERI approach proves to be a highly efficient algorithm with minimal FLOP and MOP count.

1 Introduction

Quantum chemistry calculations using localized atomic orbitals require the computation of 4-index electron-repulsion integrals (ERIs) which scale as $\mathcal{M}(N^4)$ in memory storage requirement for in-core tensor product where N is the number of basis functions. The memory requirement for storing ERIs can be reduced by using the Cholesky decomposition (CD)[1–6] or the resolution-of-the-identity (RI)[7–9] techniques to approximate the 4-index ERI tensor as a product of two 3-index tensors.[10–21] The main difference between the RI and CD techniques is that the former requires a pre-optimized auxiliary basis set whereas the latter constructs it on-the-fly. Being able to work with general basis sets and elements gives the Cholesky ERI approach a practical advantage over RI. Additionally, the Cholesky decomposition method features a tunable accuracy, and is generally more accurate than the RI method when their 3-index tensors have similar sizes.[22] However, the computational cost of the conventional algorithm to build the CD 3-index tensor is drastically greater than RI. This is mainly due to the need to determine the Cholesky pivot set, which is the analogue of the auxiliary basis set in RI.

In the conventional CD method, when a pair of basis function is selected as a pivot (Cholesky basis), all associated ERI elements are evaluated for the construction of the Cholesky vectors.[3] However, the conventional algorithm discards many of the evaluated elements that are potentially useful in future iterations. This is because there is no easy way to predict which ERI elements in a shell quartet are needed in future iterations, and storing all of them is not practical. This situation becomes more severe when the system includes many high-angular-momentum basis shells. In this case, construction of each new Cholesky vector requires looping over all previous Cholesky vectors with length N^2. When the number of selected Cholesky pivots is great, the processing speed is limited by the memory bandwidth and memory operation (MOP) count.

To mitigate this issue, Aquilante and co-workers proposed a two-step framework as an alternative to the conventional CD method.[23] The two-step algorithm splits the task into

the prediction and construction steps: (1) determine the Cholesky pivots, and (2) build the 3-index ERI tensors. In this way, the second step is similar to the RI algorithm, where all necessary ERI elements are evaluated only once and the 3-index tensor is built directly by matrix multiplication with an optimal memory bandwidth requirement. Meanwhile, in the first step, since only the diagonal elements need to be evaluated and the exact 3-index vectors are not computed, elements connected to unimportant basis pairs can be discarded. This fact is utilized in the span-factor algorithm proposed along with the two-step framework, which was recently improved by Folkestad and co-workers,[24] which determines the Cholesky pivots in batches. As a result, the overall cost is only a fraction of the conventional algorithm. The span-factor algorithm successfully resolves the memory bottleneck issue in the Cholesky ERI algorithm, but the costly high-angular-momentum shell quartets are still evaluated multiple times. In this work, we continue to improve the efficiency of the two-step Cholesky ERI algorithm by introducing a dynamic ERI tracking, reusing, and compression/elimination protocol when determining the Cholesky pivot set. With this dynamic ERI improvement, the Cholesky ERI approach can be shown to be a highly efficient algorithm with minimal FLOP count as well as an optimal MOP count, supported by benchmark studies.

2 Method

We use the following notations throughout this work:

- $A, B, C, D, ...$ are basis shells;
- $P, Q, R, S, ...$ are basis shell pairs;
- $\mu, \nu, \lambda, \gamma, ...$ are basis functions;
- $p, q, r, s, ...$ are basis function pairs;
- L^p is the Cholesky vector for pivot basis function pair p;

- Calligraphic notations ($\mathcal{B}, \mathcal{L}, \mathcal{P}, \mathcal{Q}$...) are sets/containers;
- Diag(r) is the diagonal element associated with r in the Cholesky decomposition algorithm.
- i is the index for outer iteration, and j is the index for inner iteration.

2.1 Conventional Algorithm

The four-index ERI tensor over basis functions in the chemist (Mulliken) notation $(\mu\nu|\lambda\gamma)$ can be cast as a two-index matrix $M_{rs} = (r|s)$ where $\mu, \nu, \lambda, \gamma$ are basis functions, $r \equiv \mu\nu$ and $s \equiv \lambda\gamma$ are basis function pairs. Due to the nature of electron repulsion, such an ERI matrix is positive semidefinite and can be Cholesky-decomposed to a product of a lower triangular matrix ($\mathbf{L}$) and its transpose:

$$(\mu\nu|\lambda\gamma) \equiv M_{rs} = \sum_p L_r^p L_s^p = (\mathbf{L}\mathbf{L}^{\mathrm{T}})_{rs} \tag{1}$$

where L^p are Cholesky (column) vectors in $\mathbf{L}$. In the exact Cholesky decomposition, since the dimension of $\mathbf{L}$ is $(N^2 \times N^2)$, where N is the number of basis functions, there is no computational saving in either memory storage or floating point operations (FLOPs).

The Cholesky-ERI algorithm aims to directly construct a set of Cholesky vectors with a reduced length that can produce ERIs within a tolerance or accuracy threshold,

$$(\mu\nu|\lambda\gamma) \equiv M_{rs} \approx \sum_{p\in\mathcal{B}} L_r^p L_s^p \tag{2}$$

where $\mathcal{B}$ is the set of basis pairs selected as Cholesky bases (pivots). Usually the length of $\mathcal{B}$ is much smaller than N^2 ($|\mathcal{B}| \ll N^2$), leading to significant savings in the ERI storage.

The conventional Cholesky-ERI algorithm iteratively adds new basis pairs into the Cholesky pivot set $\mathcal{B}$ based on a selection criterion which usually corresponds to the largest diagonal ERI remaining in $\mathrm{Diag}(r)|_{r\notin\mathcal{B}}$. The iteration starts with pre-computed diagonal elements of

ERIs, $\mathrm{Diag}(r) = (r|r) \equiv (\mu\nu|\mu\nu)$.

When a new basis pair is added to the Cholesky pivot set, the Cholesky vector length increases for all N^2 Cholesky vectors in $\mathbf{L}$. If a basis pair q that satisfies $\mathrm{Diag}(q) = \max\left[\mathrm{Diag}(r)|_{r\notin\mathcal{B}}\right]$ is selected as the new pivot, Cholesky vectors can be updated as

$$L_s^q = \begin{cases} \frac{(q|s)-\sum_{p\in\mathcal{B}} L_q^p L_s^p}{\sqrt{\mathrm{Diag}(q)}} & \text{for } \mathrm{Diag}(q) > 0, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for } s \notin \mathcal{B}, \tag{3}$$

$$L_s^q = 0, \qquad \text{for } s \in \mathcal{B}. \tag{4}$$

q is then added to the Cholesky pivot set $\mathcal{B}$. The total computational cost for updating Cholesky vectors using Eq. (3) is $\mathcal{O}(|\mathcal{B}|^2 N^2)$ in floating point operation (FLOP) and $\mathcal{M}(|\mathcal{B}|^2 N^2)$ in memory operation (MOP) counts.

When a new basis pair is moved to the Cholesky pivot set, the $\mathrm{Diag}(r)$ values are updated as

$$\mathrm{Diag}(r) = \mathrm{Diag}(r) - (L_r^q)^2. \tag{5}$$

in order to evaluate the importance of the remaining basis pairs. A new iteration starts with the updated $\mathrm{Diag}(r)$. In the conventional Cholesky-ERI approach, when $\mathrm{Diag}(r) < \tau$ for all $r \notin \mathcal{B}$, the remaining basis pairs are considered insignificant and the iteration stops.

If we use $\widetilde{\mathbf{M}}$ to represent the difference between the exact matrix $\mathbf{M}$ and $\mathbf{L}\mathbf{L}^{\mathrm{T}}$,

$$\widetilde{\mathbf{M}} = \mathbf{M} - \mathbf{L}\mathbf{L}^{\mathrm{T}}, \tag{6}$$

the diagonal elements of $\widetilde{\mathbf{M}}$ are

$$\widetilde{M}_{rr} = \mathrm{Diag}(r). \tag{7}$$

According to the Cauchy-Schwarz inequality,

$$\widetilde{M}_{rs}^2 \leq \widetilde{M}_{rr}\widetilde{M}_{ss} \leq \tau^2, \tag{8}$$

the error in approximation

$$(\mu\nu|\lambda\gamma) \equiv (r|s) \approx \sum_{p\in\mathcal{B}} L_r^p L_s^p \leq \tau \tag{9}$$

for all ERI elements.

The conventional Cholesky-ERI approach approximates $\mathcal{O}(N^4)$ ERI elements using 3-index tensors of size $\mathcal{O}(|\mathcal{B}|\, N^2)$, where $|\mathcal{B}|$ is the number of Cholesky basis. The accuracy and $|\mathcal{B}|$ is controlled by a single threshold τ. However, as $\tau \to 0$ and $|\mathcal{B}| \to N^2$, the memory storage requirement approaches that of a full ERI tensor. In addition, the procedure to update Cholesky vectors (Eq. (3)) can be computationally expensive as all vectors of length N^2 have to be updated.

2.2 Two-Step Algorithm

Alternatively, Cholesky vectors can be formed through a two-step algorithm.[23] The idea behind the two-step algorithm is similar to RI using an auxiliary basis, except that in this case the auxiliary basis is the set of Cholesky pivots.

In the first step, a procedure similar to the conventional approach is used to determine the set of Cholesky basis (pivots) $\mathcal{B}$ without computing the complete Cholesky vectors, e.g., the Cholesky vector update in Eq. (3) is avoided.

In the second step, an RI-like algorithm is used. A matrix $\mathbf{J}$ can be computed with elements

$$J_{pp'} = (p|p')\,, \quad p, p' \in \mathcal{B} \tag{10}$$

The ERI approximation in Eq. (9) can be equivalently written in an inner projection form,

$$(\mu\nu|\lambda\gamma) \approx \sum_{p,p'\in\mathcal{B}} (\mu\nu|p')\,(\mathbf{J}^{-1})_{p'p}\,(p|\lambda\gamma)\,. \tag{11}$$

By Cholesky-decomposing $\mathbf{J} = \mathbf{K}\mathbf{K}^{\mathrm{T}}$, the Cholesky vectors in Eq. (9) can be easily formed

$$L^{p}_{\mu\nu} = \sum_{p' \in \mathcal{B}} (\mu\nu|p') \, (\mathbf{K}^{-\mathrm{T}})_{p'p}. \tag{12}$$

There are several clear advantages of the two-step Cholesky-ERI algorithm over the conventional approach. After determining the Cholesky pivot set $\mathcal{B}$ and its associated Coulomb matrix $\mathbf{J}$, the procedures to construct Cholesky vectors can take advantage of high-performance BLAS and LAPACK libraries, including the Cholesky decomposition of $\mathbf{J}$, the inversion of the triangular matrix $\mathbf{K}^{\mathrm{T}}$, and the contraction between the 3-index ERI tensor $(\alpha\beta|p')$ and $\mathbf{K}^{-\mathrm{T}}$. The FLOP count in the linear algebra portion of the conventional algorithm (Eq. (3)) is comparable to the cost of Eq. (12); both scale as $\mathcal{O}(|\mathcal{B}|^2 \, N^2)$. However, the memory operation (MOP) count is higher in the conventional method compared to the two-step approach, i.e. $\mathcal{M}(|\mathcal{B}|^2 \, N^2)$ vs. $\mathcal{M}(|\mathcal{B}| \, N^2)$, respectively. Additionally, Eq. (3) suggests that in the conventional approach all integrals outside the Cholesky pivot set must be computed regardless of their significance. In contrast, in the two-step approach, only significant integrals are computed in Eq. (12).

Unlike the relative straightforward implementation of the conventional Cholesky-ERI approach, in order for the two-step method to reach its full potential, several important cost reduction techniques and considerations must be employed. The key lies in an efficient way to determine the Cholesky pivot set without computing all Cholesky vectors in the first step.

2.3 Efficient Cholesky Pivot Determination

Recently, Folkestad and co-workers proposed the so-called "span-factor" algorithm to efficiently determine the Cholesky pivot set.[24] Here, we introduce an additional step that features a reduced FLOP count by avoiding redundant evaluations of the basis shell quartet. Due to the dynamic nature of ERI tracking and compression in this algorithm, we term it the "dynamic-ERI" algorithm. Both "span-factor" and "dynamic-ERI" algorithms can be

organized into Algorithm 1.

The idea behind the "span-factor" approach is to determine new Cholesky pivots in batches (Lines 12-14 in Algorithm 1) so as to take advantage of the fast linear algebra libraries (Line 22 in Algorithm 1). In the "dynamic-ERI" algorithm, in addition to batching evaluations of Cholesky pivots, all ERIs computed inside the loop are saved (Lines 8 and 20 in Algorithm 1) and reused (Line 22 in Algorithm 1). Due to the fast growing number of ERIs, all integrals are tracked and removed when the associated Cholesky pivots fall below the threshold (Line 39 in Algorithm 1). Note that although Cholesky pivots are selected based on basis pairs, efficient ERI evaluations are batched by shell quartet. As a result, ERI tracking and removal should be carried out in shell pairs.

Algorithm 1: Cholesky Pivot Determination

Initialize a list $\mathcal{B}$ to store selected Cholesky pivots;
Initialize a container $\mathcal{L}$ to store computed Cholesky vectors;
Initialize a container $\mathcal{E}$ to store computed ERIs;
Initialize a container $\mathcal{P}$ to store Cholesky pivot candidates;
for $R = AB, A \leq B$ **do**
    Compute diagonal shell quartet $(R|R)$;
    for $r \in R$ *&* $\mathrm{Diag}(r) \equiv (r|r) \geq \tau_0$ **do** $r \to \mathcal{P}_{\mathcal{R}}$;
    for $r, s \in \mathcal{P}_{\mathcal{R}}$ **do** $(r|s) \to \mathcal{E}$;
    $\mathcal{P} = \cup_R \mathcal{P}_{\mathcal{R}}$;
end
while $|\mathcal{P}| \neq 0$ **do**
    $D_{\max} = \max_{r \in \mathcal{P}} \mathrm{Diag}(r)$;
    Select $\mathrm{Diag}(r)_{r \in \mathcal{P}} \geq \sigma D_{\max}$ where σ is the "span factor";
    if $|\{r\}| > m_\sigma$ **then** Only select the largest m_σ number of $\mathrm{Diag}(r)_{r \in \mathcal{P}}$.;
    Initialize an empty container $\mathcal{Q}$, store selected $\{r\}$ in $\mathcal{Q}$;
    for $s \in \mathcal{Q}$, $s \in S$ **do**
        for $r \in \mathcal{P}$, $r \in R$ **do**
            if $S \neq R$ *&* $(R|S) \notin \mathcal{E}$ **then**
                Compute shell quartet $(R|S)$;
                if $s \in \mathcal{P}$ **then** $(r|s) \to \mathcal{E}$;
            end
            $\widetilde{M}_{rs} = (r|s) - \sum_{L^p \in \mathcal{L}} L^p_r L^p_s$;
        end
    end
    if $\mathcal{P} \setminus \mathcal{Q} \neq 0$ **then** $\tau = \max_{r \in \mathcal{P} \setminus \mathcal{Q}} \mathrm{Diag}(r)$;
    else $\tau = \tau_0$;
    while $\mathcal{Q} \neq 0$ *&* $\mathrm{Diag}(q) = \max_{r \in \mathcal{Q}} \mathrm{Diag}(r)$ *&* $\mathrm{Diag}(q) \geq \tau$ **do**
        Initialize a container $\mathcal{L}^{(i)}$ to store new Cholesky vectors;
        for $r \in \mathcal{P}$ **do**
            $L^q_r = \frac{\widetilde{M}_{rq} - \sum_{L^p \in \mathcal{L}^{(i)}} L^p_q L^p_r}{\sqrt{\mathrm{Diag}(q)}}$;
            $\mathcal{Q} = \mathcal{Q} \setminus \{q\}$, $\mathcal{B} = \mathcal{B} \cup \{q\}$, $\mathcal{L}^{(i)} = \mathcal{L}^{(i)} \cup \{L^q\}$;
            $\mathrm{Diag}(r) = \mathrm{Diag}(r) - (L^q_r)^2$;
        end
        $\mathcal{L} = \mathcal{L} \cup \mathcal{L}^{(i)}$;
    end
    for $R \in \mathcal{P}$ **do**
        if $\mathrm{Diag}(r) < \tau$ **then**
            for $s \in \mathcal{P}$ **do** $\mathcal{E} = \mathcal{E} \setminus (r|s)$, $\mathcal{E} = \mathcal{E} \setminus (s|r)$;
            Compress the length of $L \in \mathcal{L}$ by removing the element r;
        end
    end
end

3 Benchmarks and Discussion

The conventional, span-factor, and dynamic-ERI are implemented in the open-source Chronus Quantum package,[25] which uses the LIBINT library[26] for ERI evaluations. Tables 1 to 3 report timing comparisons of the three Cholesky-ERI approaches discussed here. Unless otherwise noted, all computations were performed with 28 threads on a computation node with two Intel® Xeon® E5-2680 v4 CPUs. Additionally, Chronus Quantum was compiled using the Intel® C++ Compiler version 19.0.0.117. For the two-step algorithms, we report in wall-clock times the time spent to determine the Cholesky pivot set as T_1 and the time to build 3-index RI-ERI tensors as T_2. The most time-consuming procedures in T_1 include the evaluation of ERI vectors ($T_{1,\mathrm{ERI}}$), the computation of Cholesky vectors ($T_{1,\mathrm{CD}}$), and the compression of Cholesky vectors ($T_{1,\mathrm{COM}}$). For the dynamic-ERI algorithm, we also include the integral tracking and removal time as part of the compression step ($T_{1,\mathrm{COM}}$). T_2 primarily consists of the time to compute 3-index ERI elements ($T_{2,\mathrm{ERI}}$) and the matrix multiplication in Eq. (12) ($T_{2,\mathrm{MM}}$). We also report the total number of shell quartets ($N_{1,\mathrm{SQ}}$ and $N_{2,\mathrm{SQ}}$) computed during T_1 and T_2. The conventional algorithm is a single step approach, which requires the evaluation of ERI vectors (T_{ERI}) and Cholesky decomposition (T_{CD}). For convenience, these values are listed together with those in the first step of two-step algorithms.

In Tables 1 to 3, the computational costs of computing Cholesky vectors are presented for a cubic H_{1000} system, a C_{60} molecule, and a Au_{14} cluster. All three methods result in the same set of Cholesky vectors and identically converged energies for a given threshold τ_0.

Both the span-factor and dynamic-ERI algorithms are significantly faster than the conventional Cholesky-ERI approach. For the case of H_{1000}, since the majority of the basis pairs are selected in the Cholesky pivot set, the only computational saving in the span-factor and dynamic-ERI algorithms comes from the utilization of the fast linear algebra library to construct the Cholesky vectors ($T_{2,\mathrm{MM}}$ in span-factor and dynamic-ERI methods vs. $T_{1,\mathrm{CD}}$ in the conventional method). As a result, we only observed a 2.5× speed-up when a loose threshold ($\tau_0 = 1 \times 10^{-4}$) was used. As the threshold is decreased and the size of the Cholesky vectors

Table 1. Computational costs (in seconds) of computing Cholesky vectors using the conventional, span-factor, and dynamic-ERI approaches. This system consists of $10 \times 10 \times 10$ H atoms in a cube. The distance between adjacent H atoms is 1.0 Å. The STO-3G[27] (1000 basis functions) is used in this test. Note that the storage requirement for the full 4-index ERI tensor is 8 TB. The reference Hartree–Fock energy using AO-direct 4-index ERI is −388.14611071 a.u.

Method	T_1				T_2			Total
	$N_{1,\mathrm{SQ}}$	$T_{1,\mathrm{ERI}}$	$T_{1,\mathrm{CD}}$	$T_{1,\mathrm{COM}}$	$N_{2,\mathrm{SQ}}$	$T_{2,\mathrm{ERI}}$	$T_{2,\mathrm{MM}}$	
$\tau_0 = 1 \times 10^{-4}$ [a]								
Conventional	1.85×10^9	285.3	504.7					801.0
Span-Factor	1.12×10^8	16.0	3.8	0.2	1.85×10^9	286.0	14.4	327.2
Dynamic-ERI	7.00×10^7	9.7	3.8	0.2	1.85×10^9	287.2	14.1	322.1
$\tau_0 = 1 \times 10^{-6}$ [b]								
Conventional	4.59×10^9	701.5	3086.4					3823.5
Span-Factor	5.61×10^8	81.0	12.1	1.3	4.55×10^9	721.9	86.1	918.5
Dynamic-ERI	2.67×10^8	38.0	12.7	1.9	4.55×10^9	721.2	85.5	876.0
$\tau_0 = 1 \times 10^{-8}$ [c]								
Conventional	7.11×10^9	1092.3	7112.2					8259.5
Span-Factor	1.71×10^9	248.4	36.9	5.1	7.01×10^9	1119.8	206.3	1649.0
Dynamic-ERI	6.72×10^8	96.1	39.2	8.3	7.01×10^9	1113.8	202.7	1492.1

[a] 3-index tensor size = 29.6 GB, $|\mathcal{B}| = 3700$, initial $|\mathcal{P}| = 43144$, $\Delta E = 4.4 \times 10^{-1}$ a.u.
[b] 3-index tensor size = 73.3 GB, $|\mathcal{B}| = 9168$, initial $|\mathcal{P}| = 80172$, $\Delta E = 1.9 \times 10^{-3}$ a.u.
[c] 3-index tensor size = 113.6GB, $|\mathcal{B}| = 14205$, initial $|\mathcal{P}| = 126924$, $\Delta E = 4.0 \times 10^{-6}$ a.u.

Table 2. Computational costs (in seconds) of computing Cholesky vectors using the conventional, span-factor, and dynamic-ERI approaches. A C_{60} molecule with the cc-pVDZ[28] basis set (840 basis functions) is used in this test. Note that the storage requirement for the full 4-index ERI tensor is 4 TB. The reference B3LYP[29–31] energy using AO-direct 4-index ERI is −2286.08361884 a.u.

Method	T_1				T_2			Total
	$N_{1,\mathrm{SQ}}$	$T_{1,\mathrm{ERI}}$	$T_{1,\mathrm{CD}}$	$T_{1,\mathrm{COM}}$	$N_{2,\mathrm{SQ}}$	$T_{2,\mathrm{ERI}}$	$T_{2,\mathrm{MM}}$	
$\tau = 1 \times 10^{-4}$ [a]								
Conventional	2.47×10^8	104.3	354.5					468.4
Span-Factor	2.31×10^7	8.6	2.5	0.9	4.60×10^7	30.5	9.9	57.1
Dynamic-ERI	1.06×10^7	4.2	2.2	0.6	4.60×10^7	30.7	10.2	53.0
$\tau = 1 \times 10^{-6}$ [b]								
Traditional	4.33×10^8	155.0	1114.9					1286.5
Span-Factor	6.91×10^7	19.5	19.8	4.8	8.56×10^7	43.9	31.3	129.5
Dynamic-ERI	1.97×10^7	7.4	11.4	3.3	8.56×10^7	44.2	31.0	107.1
$\tau = 1 \times 10^{-8}$ [c]								
Conventional	7.22×10^8	263.2	3222.1					3517.7
Span-Factor	1.64×10^8	39.8	95.9	23.9	1.58×10^8	68.7	86.9	331.5
Dynamic-ERI	3.67×10^7	11.3	33.4	9.9	1.58×10^8	68.1	89.6	230.1

[a] 3-index tensor size = 21GB, $|\mathcal{B}| = 3802$, initial $|\mathcal{P}| = 58923$, $\Delta E = 1.0 \times 10^{-3}$ a.u.
[b] 3-index tensor size = 38GB, $|\mathcal{B}| = 6663$, initial $|\mathcal{P}| = 93636$, $\Delta E = 4.8 \times 10^{-5}$ a.u.
[c] 3-index tensor size = 63GB, $|\mathcal{B}| = 11108$, initial $|\mathcal{P}| = 123200$, $\Delta E = 1.7 \times 10^{-7}$ a.u.

Table 3. Computational costs (in seconds) of computing Cholesky vectors using the conventional, span-factor, and dynamic-ERI approaches. An Au_{14} cluster with the Jorge-TZP-DKH[32] basis set (1148 basis functions) is used in this test. Note that the storage requirement for the full 4-index ERI tensor is 14 TB. The reference ALH-X2C-PBE0[33–44] energy using AO-direct 4-index ERI is −265731.11806264 a.u.

Method	T_1				T_2			Total
	$N_{1,\mathrm{SQ}}$	$T_{1,\mathrm{ERI}}$	$T_{1,\mathrm{CD}}$	$T_{1,\mathrm{COM}}$	$N_{2,\mathrm{SQ}}$	$T_{2,\mathrm{ERI}}$	$T_{2,\mathrm{MM}}$	
$\tau = 1 \times 10^{-4}$ [a]								
Conventional	3.64×10^8	1515.1	1869.0					3415.5
Span-Factor	1.87×10^7	155.4	16.2	3.7	4.51×10^7	115.4	55.7	361.1
Dynamic-ERI	6.15×10^6	13.0	4.6	2.6	4.51×10^7	114.5	54.5	204.6
$\tau = 1 \times 10^{-6}$ [b]								
Conventional	5.05×10^8	1977.2	3608.1					5630.2
Span-Factor	4.43×10^7	277.6	51.5	10.7	6.56×10^7	133.6	111.8	609.5
Dynamic-ERI	1.16×10^7	17.1	16.3	6.5	6.56×10^7	133.8	112.2	311.4
$\tau = 1 \times 10^{-8}$ [c]								
Conventional	7.32×10^8	2681.7	7705.5					10447.0
Span-Factor	8.46×10^7	459.6	152.4	28.4	9.49×10^7	171.4	228.6	1082.8
Dynamic-ERI	1.83×10^7	21.6	41.3	14.2	9.49×10^7	170.3	233.4	525.9

[a] 3-index tensor size = 68GB, $|\mathcal{B}| = 6436$, initial $|\mathcal{P}| = 58377$, $\Delta E = 3.4 \times 10^{-3}$ a.u.
[b] 3-index tensor size = 94GB, $|\mathcal{B}| = 8920$, initial $|\mathcal{P}| = 95009$, $\Delta E = 2.8 \times 10^{-4}$ a.u.
[c] 3-index tensor size = 136GB, $|\mathcal{B}| = 12925$, initial $|\mathcal{P}| = 125694$, $\Delta E = 2.7 \times 10^{-6}$ a.u.

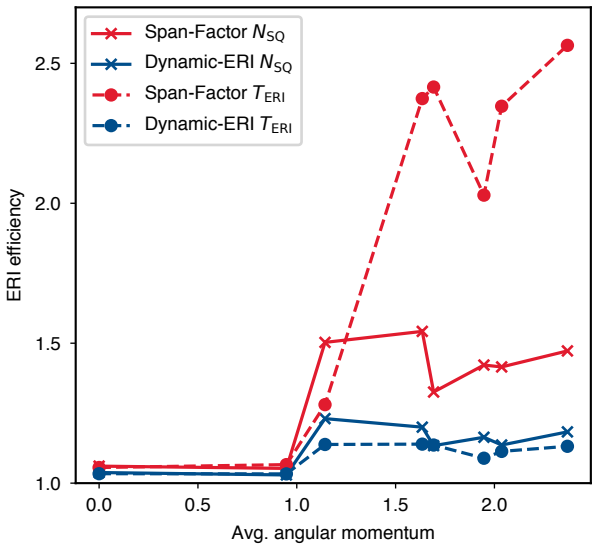

Figure 1. ERI efficiency as a function of the average angular momentum of basis. Systems used in this plot are H_{1000} with STO-3G,[27] $(H_2O)_{50}$ with 6-31G(d),[45–47] C_{60} with cc-pVDZ,[28] and Au_{14} with Jorge-DZP-DKH,[48] Jorge-TZP-DKH,[32] Sapporo-DKH3-XZP (X=D,T,Q)[49] at $\tau_0 = 1 \times 10^{-4}$. The baseline N_{SQ} and T_{ERI} is defined to be $N_{2,\mathrm{SQ}}$ and $T_{2,\mathrm{ERI}}$ of the two-step algorithm. This plot computes the ratio of $N_{1,\mathrm{SQ}} + N_{2,\mathrm{SQ}}$ and $T_{1,\mathrm{ERI}} + T_{2,\mathrm{ERI}}$ to the baseline.

increases, the advantage of using fast linear algebra library to construct the Cholesky vector becomes more prominent. At $\tau_0 = 1 \times 10^{-8}$, the span-factor and dynamic-ERI method show a factor of ~ 5.5 speed-up compared to the conventional method. This behavior is consistent throughout the tests carried out here.

For the C_{60} and Au_{14} test cases, additional computational savings arise from the reduction of the number of shell quartets evaluated in the Cholesky-ERI procedures ($N_{2,\mathrm{SQ}}$ in span-factor and dynamic-ERI methods vs. $N_{1,\mathrm{SQ}}$ in the conventional approach). The reduced computation in ERI evaluation and the utilization of the fast linear algebra in the span-factor and dynamic-ERI algorithms give rise to a $10 \sim 20$-fold speed-up for Au_{14} with $\tau_0 = 1 \times 10^{-8}$.

Comparing the two two-step Cholesky-ERI methods, the dynamic-ERI algorithm consistently outperforms the span-factor approach with a speed-up ranging from a factor of 1.02 (H_{1000}, $\tau_0 = 1 \times 10^{-4}$) to 2.06 ($Au_{14}$, $\tau_0 = 1 \times 10^{-8}$). The computational savings in the dynamic-ERI algorithm mainly comes from the elimination of redundant ERI evaluations through a tracking and removal process without exhausting the memory resources. For basis sets mostly consisting of low angular momentum functions (i.e., the STO-3G basis set used with H_{1000}) the speed-up in the dynamic-ERI algorithm is only marginal (about 10% faster) compared to the span-factor method. As more high-angular-momentum bases are included in the basis set, the computational saving in the dynamic-ERI algorithm becomes more significant. Since both the dynamic-ERI and span-factor algorithms have identical procedures in the second step, the difference in computational savings comes exclusively from the first step. Figure 1 plots ERI efficiencies as a function of average angular momentum of the basis set, including the number of evaluated shell quartets and the time spent on the ERI evaluation. Since the computational cost of ERI evaluation in T_2 can be considered as the theoretical limit when all Cholesky pivots have been determined, the ERI efficiency in each algorithm is defined as the total number of ERI computed and total ERI time in relative to T_2, e.g., $(N_{1,\mathrm{SQ}} + N_{2,\mathrm{SQ}})/N_{2,\mathrm{SQ}}$ and $(T_{1,\mathrm{ERI}} + T_{2,\mathrm{ERI}})/T_{2,\mathrm{ERI}}$. Figure 1 shows that

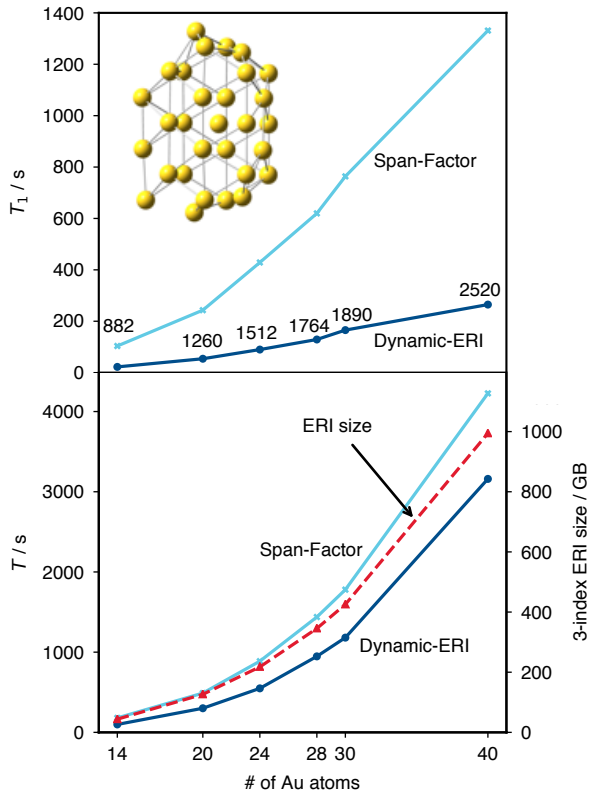

Figure 2. (**Top**): The computational times of Cholesky basis determination (first step). The number of basis functions in each system is labelled along the curve. (**Bottom**)The computational times of ERI Cholesky decomposition and the sizes of result 3-index tensors for gold cluster series of 14, 20, 24, 28, 30, and 40 gold atoms in Jorge-DZP-DKH[48] basis set with $\tau_0 = 1 \times 10^{-6}$. Computations were performed with 48 threads on an computation node with two Intel® Xeon® Platinum 8160M CPUs. Additionally, Chronus Quantum was compiled using the Intel® C++ Compiler version 2021.1 Beta 20201112.

the dynamic-ERI algorithm recomputes only 10%∼20% more ERIs than what needed for constructing Cholesky vectors. In contrast, the span-factor approach needs to recompute 70%∼80% more ERIs than those needed in T_2. If the extra ERIs are associated with high angular momentum bases with high computational cost, the computational saving in the dynamic-ERI algorithm can be significant, as shown for the case of Au_{14}.

Figure 2 plot wall-clock times for a series of Au_n clusters. Across the series, the dynamic-ERI algorithm constantly shows an 80% savings in the computational cost to determine Cholesky pivots. However, as the size of the system increases, the second step to build

3-index Cholesky tensors dominates the Cholesky-ERI procedure. As a result, the dynamic-ERI algorithm shows a 45% computational saving in the overall wall-clock time at Au_{14}, but only 25% at Au_{40}. Note that storing all 3-index Cholesky vectors requires ∼1 TB memory for 2520 basis functions (Au_{40}). As the size of the system increases, AO-direct algorithm will be a better choice.

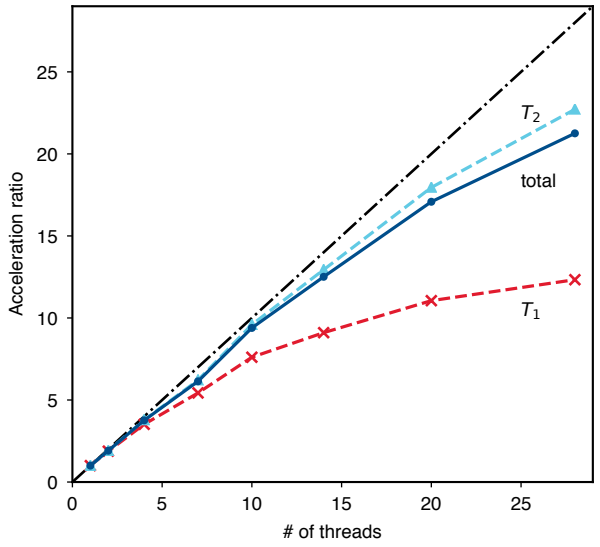

Figure 3. The shared-memory parallel acceleration of the dynamic-ERI algorithm in computing the Cholesky vectors of the Au_{14} cluster using the Jorge-TZP-DKH[32] basis set (1148 basis functions) with $\tau = 1 \times 10^{-6}$. We show the acceleration ratio, T(1-core)/T(n-core), in the first step, second step, and total time.

Figure 3 shows the parallel performance of the dynamic-ERI algorithm. As expected, the matrix product step (T_2), by taking advantage of linear algebra libraries, nicely scales with respect to the number of computing cores. Since not all procedures in the first step (T_1) can be vectorized, its parallel performance is not ideal. However, since the computational cost of T_1 is only a small fraction of that of T_2, the overall parallel performance of the dynamic-ERI algorithm is still near optimal.

4 Conclusion

In this work, we introduced an efficiency-improved two-step Cholesky-ERI method. The algorithm focuses on minimizing the floating point operation (FLOP) count in the ERI evaluation by employing an ERI tracking, reusing, and eliminating protocol without exhausting the memory resource. Benchmark tests show that the dynamic-ERI algorithm consistently outperforms the span-factor approach, with both methods being significantly faster than the conventional Cholesky-ERI method. The advantage of the dynamic-ERI algorithm becomes more prominent as more high angular momentum bases are used. We also demonstrated the excellent parallel scaling of the dynamic-ERI method as the underlying algorithm is designed to take the full advantage of linear algebra libraries in both steps.

Acknowledgement

T. Z. thank Dr. Folkestad and Dr. Kjønstad for valuable discussions on the span-factor algorithm. The development of efficient relativistic method is supported by the U.S. Department of Energy in the Heavy-Element Chemistry program (Grant No. DE-SC0021100 to X.L.). The development of the Chronus Quantum open source software package is supported by the National Science Foundation (OAC-1663636). X. L. also acknowledges support from the Computational Chemical Sciences (CCS) Program of the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division in the Center for Scalable and Predictive methods for Excitations and Correlated phenomena (SPEC) at the Pacific Northwest National Laboratory.

References

(1) Beebe, N. H. F.; Linderberg, J. Simplifications in the Generation and Transformation of Two-Electron Integrals in Molecular Calculations. *Int. J. Quant. Chem.* **1977**, *12*,

683–705.

(2) Røeggen, I.; Wisløff-Nilssen, E. On the Beebe-Linderberg Two-Electron Integral Approximation. *Chem. Phys. Lett.* **1986**, *132*, 154–160.

(3) Koch, H.; de Merás, A. S.; Pedersen, T. B. Reduced Scaling in Electronic Structure Calculations Using Cholesky Decompositions. *J. Chem. Phys.* **2003**, *118*, 9481–9484.

(4) Boman, L.; Koch, H.; de Merás, A. S. Method Specific Cholesky Decomposition: Coulomb and Exchange energies. *J. Chem. Phys.* **2008**, *129*, 134107.

(5) Aquilante, F.; Lindh, R.; Pedersen, T. B. Unbiased Auxiliary Basis Sets for Accurate Two-Electron Integral approximations. *J. Chem. Phys.* **2007**, *127*, 114107.

(6) Aquilante, F.; Gagliardi, L.; Pedersen, T. B.; Lindh, R. Atomic Cholesky Decompositions: A Route to Unbiased Auxiliary Basis Sets for Density Fitting Approximation with Tunable Accuracy and Efficiency. *J. Chem. Phys.* **2009**, *130*, 154107.

(7) Whitten, J. L. Coulombic Potential Energy Integrals and Approximations. *J. Chem. Phys.* **1973**, *58*, 4496–4501.

(8) Dunlap, B. I.; Connolly, J. W. D.; Sabin, J. R. On Some Approximations in Applications of Xα Theory. *J. Chem. Phys.* **1979**, *71*, 3396–3402.

(9) Kendall, R. A.; Früchtl, H. A. The Impact of the Resolution of the Identity Approximate Integral Method on Modern Ab Initio Algorithm Development. *Theor. Chem. Acc.* **1997**, *97*, 158–163.

(10) Aquilante, F.; Pedersen, T. B.; Lindh, R.; Roos, B. O.; Sánchez de Merás, A.; Koch, H. Accurate *Ab Initio* Density Fitting for Multiconfigurational Self-Consistent Field Methods. *J. Chem. Phys.* **2008**, *129*, 024113.

(11) Sherrill, C. D. Frontiers in Electronic Structure Theory. *J. Chem. Phys.* **2010**, *132*, 110902.

(12) Hohenstein, E. G.; Sherrill, C. D. Density Fitting and Cholesky Decomposition Approximations in Symmetry-Adapted Perturbation Theory: Implementation and Application to Probe the Nature of π-π Interactions in Linear Acenes. *J. Chem. Phys.* **2010**, *132*, 184111.

(13) Bozkaya, U.; Sherrill, C. D. Analytic Energy Gradients for the Coupled-Cluster Singles and Doubles Method with the Density-Fitting Approximation. *J. Chem. Phys.* **2016**, *144*, 174103.

(14) Epifanovsky, E.; Zuev, D.; Feng, X.; Khistyaev, K.; Shao, Y.; Krylov, A. I. General Implementation of the Resolution-of-the-Identity and Cholesky Representations of Electron Repulsion Integrals within Coupled-Cluster and Equation-of-Motion Methods: Theory and Benchmarks. *J. Chem. Phys.* **2013**, *139*, 134105.

(15) Fosso-Tande, J.; Nguyen, T.-S.; Gidofalvi, G.; DePrince, A. E. Large-Scale Variational Two-Electron Reduced-Density-Matrix-Driven Complete Active Space Self-Consistent Field Methods. *J. Chem. Theory Comput.* **2016**, *12*, 2260–2271.

(16) Freitag, L.; Knecht, S.; Angeli, C.; Reiher, M. Multireference Perturbation Theory with Cholesky Decomposition for the Density Matrix Renormalization Group. *J. Chem. Theory Comput.* **2017**, *13*, 451–459.

(17) Motta, M.; Shee, J.; Zhang, S.; Chan, G. K.-L. Efficient Ab Initio Auxiliary-Field Quantum Monte Carlo Calculations in Gaussian Bases via Low-Rank Tensor Decomposition. *J. Chem. Theory Comput.* **2019**, *15*, 3510–3521.

(18) Hannon, K. P.; Li, C.; Evangelista, F. A. An Integral-Factorized Implementation of the Driven Similarity Renormalization Group Second-Order Multireference Perturbation Theory. *J. Chem. Phys.* **2016**, *144*, 204111.

(19) Zhang, T.; Li, C.; Evangelista, F. A. Improving the Efficiency of the Multireference Driven Similarity Renormalization Group via Sequential Transformation, Density Fit-

ting, and the Noninteracting Virtual Orbital Approximation. *J. Chem. Theory Comput.* **2019**, *15*, 4399–4414.

(20) Bozkaya, U. Efficient Implementation of the Second-Order Quasidegenerate Perturbation Theory with Density-Fitting and Cholesky Decomposition Approximations: Is It Possible To Use Hartree–Fock Orbitals for a Multiconfigurational Perturbation Theory? *J. Chem. Theory Comput.* **2019**, *15*, 4415–4429.

(21) Lesiuk, M. Implementation of the Coupled-Cluster Method with Single, Double, and Triple Excitations using Tensor Decompositions. *J. Chem. Theory Comput.* **2020**, *16*, 453–467.

(22) Weigend, F.; Kattannek, M.; Ahlrichs, R. Approximated Electron Repulsion Integrals: Cholesky Decomposition Versus Resolution of the Identity Methods. *J. Chem. Phys.* **2009**, *130*, 164106.

(23) Aquilante, F.; Boman, L.; Boström, J.; Koch, H.; Lindh, R.; de Merás, A. S.; Pedersen, T. B. In *Linear-Scaling Techniques in Computational Chemistry and Physics: Methods and Applications*; Zalesny, R., Papadopoulos, M. G., Mezey, P. G., Leszczynski, J., Eds.; Springer Netherlands: Dordrecht, 2011; Chapter Cholesky Decomposition Techniques in Electronic Structure Theory, pp 301–343.

(24) Folkestad, S. D.; Kjønstad, E. F.; Koch, H. An Efficient Algorithm for Cholesky Decomposition of Electron Repulsion Integrals. *J. Chem. Phys.* **2019**, *150*, 194112.

(25) Williams-Young, D. B.; Petrone, A.; Sun, S.; Stetina, T. F.; Lestrange, P.; Hoyer, C. E.; Nascimento, D. R.; Koulias, L.; Wildman, A.; Kasper, J.; Goings, J. J.; Ding, F.; DePrince III, A. E.; Valeev, E. F.; Li, X. The Chronus Quantum (ChronusQ) Software Package. *WIREs Comput. Mol. Sci.* **2020**, *10*, e1436.

(26) Valeev, E. F. Libint: A Library for the Evaluation of Molecular Integrals of Many-body

Operators over Gaussian Functions. http://libint.valeyev.net/, 2020; version 2.7.0-beta.6.

(27) Hehre, W. J.; Stewart, R. F.; Pople, J. A. Self-Consistent Molecular-Orbital Methods. I. Use of Gaussian Expansions of Slater-Type Atomic Orbitals. *J. Chem. Phys.* **1969**, *51*, 2657–2664.

(28) Dunning, T. H. Gaussian Basis Sets for Use in Correlated Molecular Calculations. I. The Atoms Boron Through Neon and Hydrogen. *J. Chem. Phys.* **1989**, *90*, 1007–1018.

(29) Becke, Axel D, Density-functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* **1993**, *98*, 5648–5652.

(30) Lee, Chengteh,; Yang, Weitao,; Parr, Robert G, Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density. *Phys. Rev. B* **1988**, *37*, 785–789.

(31) Miehlich, Burkhard,; Savin, Andreas,; Stoll, Hermann,; Preuss, Heinzwerner, Results Obtained With the Correlation Energy Density Functionals of Becke and Lee, Yang and Parr. *Chem. Phys. Lett.* **1989**, *157*, 200–206.

(32) Martins, L. S. C.; Jorge, F. E.; Machado, S. F. All-Electron Segmented Contraction Basis Sets of Triple Zeta Valence Quality for the Fifth-Row Elements. *Mol. Phys.* **2015**, *113*, 3578–3586.

(33) Hess, B. A. Relativistic Electronic-Structure Calculations Employing a Two-Component No-Pair Formalism with External-Field Projection Operators. *Phys. Rev. A* **1986**, *33*, 3742–3748.

(34) Liu, W.; Peng, D. Infinite-Order Quasirelativistic Density Functional Method Based on the Exact Matrix Quasirelativistic Theory. *J. Chem. Phys.* **2006**, *125*, 044102.

(35) Liu, W.; Peng, D. Exact Two-component Hamiltonians Revisited. *J. Chem. Phys.* **2009**, *131*, 031104.

(36) Kutzlenigg, W.; Liu, W. Quasirelativistic Theory Equivalent to Fully Relativistic Theory. *J. Chem. Phys.* **2005**, *123*, 241102.

(37) Liu, W. Ideas of Relativistic Quantum Chemistry. *Mol. Phys.* **2010**, *108*, 1679–1706.

(38) Ilias, M.; Saue, T. An Infinite-Order Relativistic Hamiltonian by a Simple One-Step Transformation. *J. Chem. Phys.* **2007**, *126*, 064102.

(39) Saue, T. Relativistic Hamiltonians for Chemistry: A Primer. *ChemPhysChem* **2011**, *12*, 3077–3094.

(40) Kasper, J. M.; Lestrange, P. J.; Stetina, T. F.; Li, X. Modeling $L_{2,3}$-Edge X-ray Absorption Spectroscopy with Real-Time Exact Two-Component Relativistic Time-Dependent Density Functional Theory. *J. Chem. Theory Comput.* **2018**, *14*, 1998–2006.

(41) Zhang, T.; Kasper, J. M.; Li, X. In *Annual Reports in Computational Chemistry*; Dixon, D. A., Ed.; Elsevier, 2020; Vol. 16; Chapter Chapter Two - Localized Relativistic Two-Component Methods for Ground and Excited State Calculations, pp 17–37.

(42) Perdew, John P,; Burke, Kieron,; Ernzerhof, Matthias, Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865–3868.

(43) Perdew, J. P.; Burke, K.; Ernzerhof, M. Errata: Generalized Gradient Approximation Made Simple [Phys. Rev. Lett. 77, 3865 (1996)]. *Phys. Rev. Lett.* **1997**, *78*, 1396–1396.

(44) Adamo, Carlo,; Barone, Vincenzo, Toward Reliable Density Functional Methods Without Adjustable Parameters: The PBE0 Model. *J. Chem. Phys.* **1999**, *110*, 6158–6170.

(45) Ditchfield, R.; Hehre, W. J.; Pople, J. A. Self-Consistent Molecular-Orbital Methods. IX. An Extended Gaussian-Type Basis for Molecular-Orbital Studies of Organic Molecules. *J. Chem. Phys.* **1971**, *54*, 724–728.

(46) Hehre, W. J.; Ditchfield, R.; Pople, J. A. Self-Consistent Molecular Orbital Methods. XII. Further Extensions of Gaussian-Type Basis Sets for Use in Molecular Orbital Studies of Organic Molecules. *J. Chem. Phys.* **1972**, *56*, 2257–2261.

(47) Hariharan, P. C.; Pople, J. A. The Influence of Polarization Functions on Molecular Orbital Hydrogenation Energies. *Theor. Chem. Acc.* **1973**, *28*, 213–222.

(48) Canal Neto, A.; Jorge, F. E. All-electron Double Zeta Basis Sets for the Most Fifth-Row Atoms: Application in DFT Spectroscopic Constant Calculations. *Chem. Phys. Lett.* **2013**, *582*, 158–162.

(49) Noro, T.; Sekiya, M.; Koga, T. Sapporo-(DKH3)-nZP (n = D, T, Q) Sets for the Sixth Period s-, d-, and p-Block Atoms. *Theor. Chem. Acc.* **2013**, *132*, 1363.