A linear scaling hierarchical block low-rank representation of the electron repulsion integral tensor

Xin Xing,[1, a)] Hua Huang,[1, b)] and Edmond Chow[1, c)]

School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-4017, USA

(Dated: 4 August 2020)

Efficient representations of the electron repulsion integral (ERI) tensor and fast algorithms for contractions with the ERI tensor often employ a low-rank approximation of the tensor or its subblocks. Such representations include density fitting (DF), the continuous fast multipole method (CFMM) and, more recently, hierarchical matrices. We apply the $\mathcal{H}^2$ hierarchical matrix representation to the ERI tensor with Gaussian basis sets to rapidly calculate the Coulomb matrices in Hartree–Fock and density functional theory calculations. The execution time and storage requirements of the hierarchical matrix approach and the DF approach are compared. The hierarchical matrix approach has very modest storage requirements, allowing large calculations to be performed in memory without recomputing ERIs. We interpret the hierarchical matrix approach as a multilevel, localized DF method, and also discuss the close relationship between the hierarchical matrix approach with CFMM. Like CFMM, the hierarchical matrix approach is asymptotically linear scaling, but the latter requires severalfold less memory (or severalfold less computation, if quantities are computed dynamically) due to being able to efficiently employ low-rank approximations for far more blocks.

[a)]Electronic mail: xxing33@gatech.edu

[b)]Electronic mail: huangh223@gatech.edu

[c)]Electronic mail: echow@cc.gatech.edu

I. INTRODUCTION

Given N_{bf} basis functions $\{\phi_a\}$, the electron repulsion integral (ERI) tensor

$$(ab|cd) = \iint \phi_a(r_1)\phi_b(r_1)\frac{1}{|r_1 - r_2|}\phi_c(r_2)\phi_d(r_2)\mathrm{d}r_1\mathrm{d}r_2$$

contains N_{bf}^4 entries, with the number of nonnegligible entries scaling as $O(N_{\text{bf}}^2)$ with a very large prefactor. Here, we assume the basis functions are Gaussian type functions. The ERI tensor is generally so large that it cannot be stored in memory, so its entries are either stored on disk or are recomputed whenever they are needed. Further, each integral is very costly to compute, depending on irregularly structured recurrence relations. Efficient representations of the ERI tensor exploit properties of the tensor to reduce computational cost as well as reduce storage requirements. Once an efficient representation of the ERI tensor is formed, it may be used to accelerate subsequent calculations, such as the calculation of the Coulomb matrix. Calculating the Coulomb matrix is often the computational bottleneck in density functional theory (DFT) calculations.

Although they have different physical motivations, many existing efficient representations of the ERI tensor can be viewed algebraically as "compressing" the ERI tensor or its subblocks into low-rank form. For example, density fitting[1–9] (DF) constructs a low-rank approximation of the entire ERI tensor. The continuous fast multipole method[10–13] (CFMM) and related methods[14–17] compress specific blocks of the ERI tensor into low-rank form. The clustered low-rank tensor format[18] (CLR), proposed for compressing the blocks in the three-index tensors of DF, can also be used to compress the ERI tensor. The different methods differ in how they exploit the *block low-rank structure* of the ERI tensor, i.e., which blocks to compress and how to compress them.

Previously, the $\mathcal{H}^2$-ERI method[19] was proposed as an efficient method to calculate the Coulomb matrix. This is accomplished by compressing specific ERI blocks and representing the overall ERI tensor in the $\mathcal{H}^2$ matrix representation. The $\mathcal{H}^2$ matrix representation is a type of hierarchical block low-rank representation that has both storage and matrix-vector multiplication costs linear in the matrix dimension. In Ref. 19, a Matlab implementation was used to calculate and verify the accuracy of Coulomb matrices using the $\mathcal{H}^2$-ERI method. In this paper, we use an optimized multithreaded C program to perform Hartree–Fock and DFT calculations with the $\mathcal{H}^2$-ERI method. We also perform calculations with DF for comparison of both accuracy and computational time. In comparison with DF and CFMM,

we show that $\mathcal{H}^2$-ERI better exploits the block low-rank structure of the ERI tensor and thus provides a more efficient representation.

The ERI tensor $(ab|cd)$ can be unfolded into an *ERI matrix* where rows are indexed by ab and columns are indexed by cd. Each entry of the ERI matrix is the Coulomb interaction between two generalized electron densities, $\phi_a\phi_b$ and $\phi_c\phi_d$. It can be justified numerically that (1) the ERI matrix has numerical rank $O(N_{\text{bf}})$; (2) a block of the ERI matrix associated with two "well-separated" (to be defined in Section II) sets of electron densities has numerical rank $O(1)$ independent of the block size. Such rank-$O(1)$ blocks form the majority of the ERI matrix. Thus, DF must have approximation rank at least $O(N_{\text{bf}})$ (equivalent to using $O(N_{\text{bf}})$ auxiliary basis functions), while $\mathcal{H}^2$-ERI and CFMM can compress most blocks of the ERI matrix into rank-$O(1)$ form. As a result, CFMM and $\mathcal{H}^2$-ERI are asymptotically more efficient than DF.

In CFMM, a large number of ERI blocks, though numerically low-rank, are not compressed because multipole expansions cannot be used for determining low-rank expansions when electron densities overlap numerically. Such blocks must be represented in dense form and these dense ERI blocks dominate the storage (if precomputed) or computation (if dynamically computed) in CFMM. In comparison, $\mathcal{H}^2$-ERI uses a compression method that works for more ERI blocks (without needing to compute these ERI blocks explicitly) than multipole expansions. As a result, $\mathcal{H}^2$-ERI has the same asymptotic scalability as CFMM but is far more efficient.

It is worth noting that the CLR tensor format uses a one-level partitioning of a tensor into nonoverlapping blocks as opposed to the hierarchical partitioning used in $\mathcal{H}^2$-ERI and CFMM. This lack of hierarchy in partitioning leads to larger asymptotic computation and storage complexities when using CLR for the ERI tensor than CFMM and $\mathcal{H}^2$-ERI. CLR also does not provide a way to compress blocks without computing them, as opposed to $\mathcal{H}^2$-ERI and CFMM.

II. $\mathcal{H}^2$-ERI METHOD

A. Notation and terminology

In this paper, we refer to a continuous electron density, or a product of two basis functions, $\phi_a\phi_b$, as a *distribution*. Since we assume the basis functions are Gaussian type functions (GTFs), the distributions are also GTFs. Such a distribution decays exponentially and thus has a bounded *numerical support* outside of which the distribution is negligible. We use a ball to compactly characterize the numerical support of a distribution, and will often refer to the *center* of the distribution as the center of this ball.

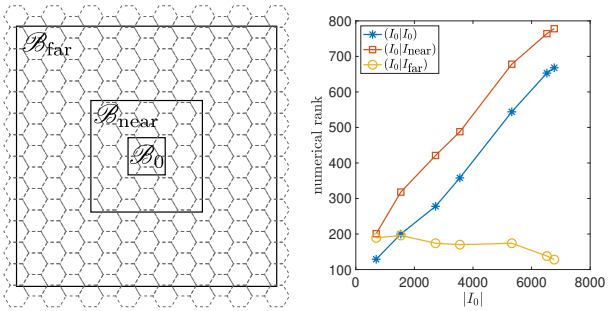
Let I denote the complete set of distributions $\{\phi_a\phi_b\}$ for a molecular system and chosen basis set. We use the notation $(I|I)$ to denote the $N_{\text{bf}}^2 \times N_{\text{bf}}^2$ ERI matrix. Subsets of I can be used in this notation to denote subblocks of the ERI matrix.

B. Block low-rank structure of the ERI matrix

To illustrate the block low-rank structure of the ERI matrix, consider one ideal graphene layer with the STO-3G basis set. Place a cubic box $\mathcal{B}_0 = [-L/2,\ L/2]^3$ centered on the graphene layer (see Figure 1a). Let $\mathcal{B}_{\text{near}} = [-3L/2,\ 3L/2]^3 \setminus \mathcal{B}_0$ be the union of 26 adjacent boxes and $\mathcal{B}_{\text{far}} = [-7L/2,\ 7L/2]^3 \setminus (\mathcal{B}_{\text{near}} \cup \mathcal{B}_0)$ be the union of 316 nonadjacent boxes. Denote the sets of distributions with center in these three domains, $\mathcal{B}_0$, $\mathcal{B}_{\text{near}}$, $\mathcal{B}_{\text{far}}$, as I_0, I_{near}, I_{far}, respectively. Increasing the edge length L from 3 Bohr to 9 Bohr, we obtain a series of sets I_0, I_{near}, and I_{far} with increasing numbers of distributions. Figure 1b plots the numerical ranks of the ERI blocks $(I_0|I_0)$, $(I_0|I_{\text{near}})$, and $(I_0|I_{\text{far}})$ vs. the size of I_0. The numerical rank of a matrix block is estimated by its singular values with relative threshold 10^{-10}.

We observe that the numerical ranks of the ERI blocks $(I_0|I_0)$ and $(I_0|I_{\text{near}})$ both increase with the size of I_0, while the numerical rank of $(I_0|I_{\text{far}})$ tends to be small and independent of the sizes of I_0 and I_{far}. This low-rank property of blocks of the form $(I_0|I_{\text{far}})$ is the basis of the $\mathcal{H}^2$-ERI method.

We say that two boxes of the same size are *well-separated* if they are separated by at least one box of the same size. As an example, box $\mathcal{B}_0$ and any box of the same size in $\mathcal{B}_{\text{far}}$ are said to be well-separated.



(a) (b)

FIG. 1. (a) 2D illustration of the graphene layer and $\mathcal{B}_0$, $\mathcal{B}_{\text{near}}$, and $\mathcal{B}_{\text{far}}$; (b) numerical ranks of the three ERI blocks vs. $|I_0|$.

C. Low rank approximation for well-separated boxes

Let the domain containing the centers of all distributions be partitioned into boxes of equal size. Let I_i denote the set of distributions in box i. Let J_i denote the set of distributions with centers in boxes that are well-separated from box i. The ERI block $(I_i|J_i)$ is low-rank like the block $(I_0|I_{\text{far}})$ from Section II B. This block can be compressed into a low-rank form

$$\underbrace{(I_i|J_i)}_{|I_i|\times|J_i|} \approx \underbrace{U_i}_{|I_i|\times r_0} \underbrace{(I_i^{\text{id}}|J_i)}_{r_0\times|J_i|}, \tag{1}$$

where I_i^{id} is a subset of I_i, and thus $(I_i^{\text{id}}|J_i)$ is a subset of the rows of $(I_i|J_i)$, U_i is a tall-and-skinny matrix with bounded values, and r_0 is the approximation rank. Such a low-rank form is called an interpolative decomposition[20] (ID).

One could compute an ID algebraically via the pivoted QR decomposition given a rank or an absolute/relative error threshold. However, such a procedure requires that the ERI block $(I_i|J_i)$ is formed explicitly, i.e., requiring computation of all the ERIs in $(I_i|J_i)$. Both CFMM and $\mathcal{H}^2$-ERI avoid needing $(I_i|J_i)$ in explicit form. In CFMM, the low-rank approximation is computed via multipole expansions if the distributions in I_i and J_i do not overlap numerically. This limitation of CFMM means that the low-rank form of many ERI blocks cannot be computed and are treated as full-rank, dense matrices. In $\mathcal{H}^2$-ERI, the low-rank approximation is computed without this limitation via a method called the proxy point method.[19] The key that allows the ERI matrix to be efficiently represented in the $\mathcal{H}^2$ matrix representation is this proxy point method.

To briefly explain the proxy point method, let $(I_*|J_*)$ denote any above ERI block to be compressed into ID form. Referring to Figure 2 (left), the centers of the distributions in I_* are contained in a box $\mathcal{B}$ and the centers of the distributions in J_* are within the union of well-separated boxes $\mathcal{F}$. In the proxy point method, the distributions in J_* are split into two subsets, J_{near} and J_{far}. The set J_{near} contains the distributions that numerically overlap with $\mathcal{B}$ or its 26 adjacent boxes. These distributions are shown in green in Figure 2. The set $J_{\text{far}} = J_* \setminus J_{\text{near}}$. The set J_{near} is usually a small fraction of J_* since GTFs decay exponentially, while J_{far} is a large fraction.

The challenge is how to efficiently compress the interactions between I_* and J_{far}. Here, by virtue of Green's theorem, we replace J_{far} by a small set of point charges Y_p located in $\mathcal{F}$ as illustrated in Figure 2 (right), i.e., the column space of $(I_*|J_{\text{far}})$ is replaced by the column space of the much smaller matrix $(I_*|Y_p)$. The elements of $(I_*|Y_p)$ are much cheaper to compute than ERIs; the elements are interactions between distributions and point charges and are thus analogous to electron-nuclear attraction integrals.

Finally, the ID approximation $(I_*|J_*) \approx U(I_*^{\text{id}}|J_*)$ is computed via the "mixed" ID approximations of $(I_*|J_{\text{near}})$ and $(I_*|Y_p)$ as shown in Algorithm 1. The proxy points in Y_p are uniformly sampled on several layers of cubic surfaces that enclose the interior boundary of domain $\mathcal{F}$ as shown in Figure 2. The size of Y_p is independent of the overall problem size $|I|$ or the sizes of I_* and J_*. In most cases, a few hundreds to one thousand proxy points for Y_p are sufficient to guarantee the accuracy of the proxy point method. More details on the selection of Y_p can be found in Ref. 19.

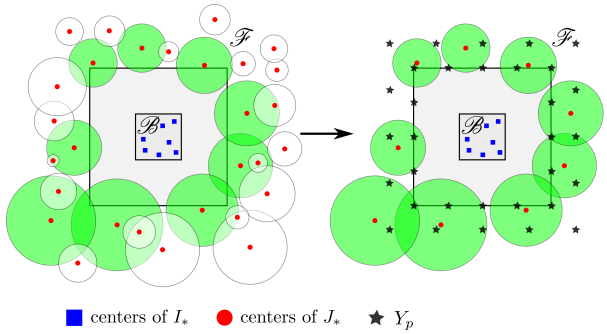


FIG. 2. 2D illustration of the proxy point method for an ERI block $(I_*|J_*)$. A circle represents the numerical support of a distribution in J_*. In the proxy point method, J_{far} is replaced by a set of point charges Y_p in $\mathcal{F}$.

Algorithm 1 The proxy point method for $(I_*|J_*)$

Input: I_*, J_*, $\mathcal{B}$, $\mathcal{F}$

Output: U and I_*^{id} for the ID approximation $(I_*|J_*) \approx U(I_*^{\mathrm{id}}|J_*)$

1: Split J_* into J_{near} and J_{far}

2: Select proxy point charges Y_p in $\mathcal{F}$

3: Generate random matrices $\Omega_1 \in \mathbb{R}^{|J_{\mathrm{near}}| \times |I_*|}$ and $\Omega_2 \in \mathbb{R}^{|Y_p| \times |I_*|}$

4: Calculate $A_1 = (I_*|J_{\mathrm{near}})\Omega_1$ and $A_2 = (I_*|Y_p)\Omega_2$

5: Normalize the columns of A_1 and A_2 to obtain $\tilde{A}_1$ and $\tilde{A}_2$

6: Compute U and I_*^{id} by an algebraic ID approximation of $[\tilde{A}_1,\ \tilde{A}_2]$

In Algorithm 1, a randomized linear algebra technique[21] is used in steps 3 and 4 to approximate the column spaces of $(I_*|J_{\mathrm{near}})$ and $(I_*|Y_p)$ by the column spaces of the smaller matrices A_1 and A_2, respectively. Typically, Ω_1 and Ω_2 are chosen to be dense with elements from a standard normal distribution. In this paper, to reduce the cost of the multiplications in step 4, we choose Ω_1 and Ω_2 as sparse random matrices, with 16 nonzero entries per column with locations selected randomly and nonzero values following a standard normal distribution.

D. The $\mathcal{H}^2$ matrix representation

We can now briefly describe the $\mathcal{H}^2$ matrix representation and establish additional terminology for this paper. See Ref. 22 for additional details.

An $\mathcal{H}^2$ matrix representation is composed of *far-field* (FF) blocks stored in low-rank form and *near-field* (NF) blocks stored in dense matrix form. FF blocks represent the interaction between distributions centered in two boxes that are well-separated. NF blocks represent the remaining interactions. See Figure 3 (right) for an example.

In the example, the distributions I in a 1D domain are hierarchically partitioned into boxes. The structure is represented by a *partition tree*; see Figure 3 (left). Let $I_i \subset I$ denote the set of distributions centered in box i and corresponding to node i in the partition tree. The distributions centered in the finest level boxes are labeled $I_7, \ldots, I_{14}$ in the example. The union of the distributions I_7 and I_8 is I_3, etc. Larger FF blocks are formed from merging smaller FF blocks, if possible. In the example, the large FF block $(I_3|I_5)$ is due to

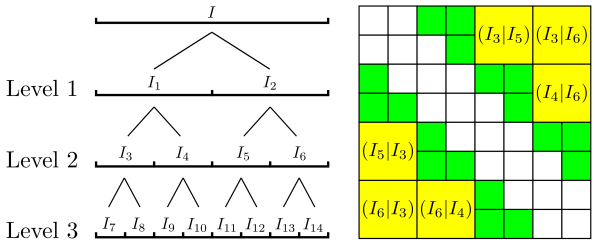


FIG. 3. 1D illustration of a recursive partitioning of the distributions in I and the associated structure of the ERI matrix $(I|I)$. NF blocks are white, and FF blocks are colored. Yellow FF blocks are defined by nodes in level 2 and green FF blocks are defined by nodes in level 3 of the partition tree.

the interactions between I_3 and I_5, which are in well-separated boxes.

For each node i in the partition tree, the ID approximation of $(I_i|J_i)$ (see Equation (1)) is computed. Each FF block $(I_i|I_j)$ is the intersection between the low-rank ERI blocks $(I_i|J_i)$ and $(J_j|I_j)$. (The latter is the transpose of $(I_j|J_j)$.) Thus, the low-rank approximation of $(I_i|I_j)$ can be constructed directly based on the ID approximations of the two blocks as

$$(I_i|I_j) \approx U_i(I_i^{\mathrm{id}}|I_j^{\mathrm{id}})U_j^T. \tag{2}$$

We refer to $(I_i^{\mathrm{id}}|I_j^{\mathrm{id}})$ as an *intermediate* block.

For a nonleaf node i in the partition tree, the ID approximation of $(I_i|J_i)$ is constructed in terms of the ID approximations associated with its children nodes in order to reduce computation and storage cost.[19]

E. Low-rank approximation of NF and intermediate blocks.

In standard $\mathcal{H}^2$ matrix representations, the NF and intermediate blocks are stored in dense matrix format and storage of these dense blocks typically constitutes a large majority of the total storage cost. For ERI matrices, however, we note from Figure 1b that the numerical ranks of $(I_0|I_0)$ and $(I_0|I_{\mathrm{near}})$, although not independent of $|I_0|$, are small relative to $|I_0|$. Thus, in $\mathcal{H}^2$-ERI, we also compress each NF and intermediate block into an ID form using the pivoted QR decomposition when this is beneficial. Since the total size of NF and intermediate blocks is $O(|I|)$, this additional compression does not change the asymptotic computation cost of $\mathcal{H}^2$-ERI.

F. Complexity and accuracy

Let r denote an upper bound on the rank of all the ID approximations. Then the $\mathcal{H}^2$ matrix representation can be formed with $O(|I|r^2)$ computation cost and requires $O(|I|r)$ storage cost.

$\mathcal{H}^2$-ERI uses a relative error threshold τ to control the accuracy of each computed ID approximation. The same τ is used in the compression of the NF and intermediate blocks. The constructed representation of $(I|I)$ in the end has relative error $O(\tau)$ in the Frobenius norm.

G. Using the $\mathcal{H}^2$ matrix representation

The $\mathcal{H}^2$ matrix representation, once computed, can then be used to rapidly calculate the Coulomb matrix,

$$J_{ab} = \sum_{c,d}(ab|cd)D_{cd}.$$

This operation is equivalent to the matrix-vector multiplication $J = (I|I)D$ with Coulomb matrix J and density matrix D unfolded into vectors. With $(I|I)$ represented in $\mathcal{H}^2$ format, the established fast $\mathcal{H}^2$ matrix-vector multiplication algorithm[22,23] can calculate the Coulomb matrix with $O(|I|r)$ computation cost.

III. RELATIONSHIP TO DF

In DF, a set of N_{aux} auxiliary basis functions $\{\psi_\alpha\}$ is used to fit each distribution $\phi_a\phi_b$ by a linear combination, i.e., $\phi_a\phi_b \approx \sum_\alpha C^\alpha_{a,b}\psi_\alpha$. This leads to the general DF approximation

$$(ab|cd) \approx \sum_{\alpha,\beta} C^\alpha_{a,b}(\alpha|\beta)C^\beta_{c,d}.$$

In classical DF,[3] the fitting coefficients $C^\alpha_{a,b}$ are computed as $C^\alpha_{a,b} = \sum_\beta (ab|\beta)(\beta|\alpha)^{-1}$, and the DF approximation becomes

$$(ab|cd) \approx \sum_{\alpha,\beta}(ab|\alpha)(\alpha|\beta)^{-1}(\beta|cd).$$

This can be viewed as a rank-N_{aux} approximation of the ERI matrix $(I|I)$. Typically, the number of auxiliary basis functions is about 5 times the number of basis functions, i.e., $N_{\text{aux}} \sim O(N_{\text{bf}})$.

In $\mathcal{H}^2$-ERI, each ID approximation $(I_i|J_i) \approx U_i(I_i^{\mathrm{id}}|J_i)$ can be viewed as a *localized DF*. More precisely, each row of $(I_i|J_i)$ is approximated in the ID approximation as

$$(\varphi_k|J_i) \approx u_{k,:}(I_i^{\mathrm{id}}|J_i) = \textstyle\sum_{\varphi_\alpha \in I_i^{\mathrm{id}}} u_{k,\alpha}(\varphi_\alpha|J_i), \quad \varphi_k \in I_i$$

where $u_{k,:}$ is the kth row of U_i and $u_{k,\alpha}$ is the (k,α)th entry of U_i. Each distribution $\varphi_k \in I_i$ is thus approximated by a linear combination of the distributions in I_i^{id} as

$$\varphi_k \approx \textstyle\sum_{\varphi_\alpha \in I_i^{\mathrm{id}}} u_{k,\alpha}\varphi_\alpha.$$

From the viewpoint of DF, I_i^{id} is exactly a set of auxiliary basis functions used to *approximate I_i for the interactions between I_i and J_i*, and U_i contains the associated fitting coefficients. Conversely, computing an ID approximation of an ERI block can be viewed as a numerical way to construct a DF approximation for the associated Coulomb interactions.

Most existing localized DF methods[4–9,24] limit the number of auxiliary basis functions used to fit a distribution by heuristically applying a local fitting domain or a local fitting metric. In contrast, $\mathcal{H}^2$-ERI can be considered as a distinct localized DF approach that restricts DF to only approximate the Coulomb interactions between well-separated sets of distributions.

There are two main differences between $\mathcal{H}^2$-ERI and DF. The first is how accuracies of the constructed representations are controlled. In DF, the accuracy depends on the choice of a precomputed auxiliary basis set. In $\mathcal{H}^2$-ERI, the relative error threshold τ is used to directly control the accuracy of the $\mathcal{H}^2$ matrix representation.

The second difference is in the resulting approximation ranks, i.e., N_{aux} and r. The maximum approximation rank r in $\mathcal{H}^2$-ERI is experimentally $O(1)$ with increasing problem sizes, while N_{aux} scales as $O(N_{\mathrm{bf}})$. This results in $\mathcal{H}^2$-ERI requiring much less storage cost than DF.

IV. TEST CALCULATIONS

In this section, we first verify the accuracy of Hartree–Fock (HF) and Kohn–Sham DFT calculations when $\mathcal{H}^2$-ERI is used to calculate the Coulomb matrices. We then compare the performance of the $\mathcal{H}^2$-ERI approach with that of DF in terms of computation time and storage.

Both HF and DFT use self-consistent field (SCF) iterations with direct inversion of the iterative subspace. Superposition of atomic densities is used for the initial density matrix. The SCF iterations are stopped when the energy difference between two consecutive iterations is less than 10^{-11} Hartrees. For DFT, we use the hybrid exchange-correlation functional B3LYP.

Previously, only the accuracy of the Coulomb matrices calculated by $\mathcal{H}^2$-ERI had been verified.[19] Also, $\mathcal{H}^2$-ERI was only used for ERI matrices with primitive basis functions. In this paper, we use contracted basis functions. Two basis sets are used: cc-pVDZ and aug-cc-pVDZ. The latter is important for comparison because it contains diffuse functions. For classical DF, the corresponding auxiliary basis sets cc-pVDZ-jkfit and aug-cc-pVDZ-jkfit are used.

$\mathcal{H}^2$-ERI and DF both start with prescreening negligible distributions in $I = \{\phi_a\phi_b\}$. If

$$\sqrt{(\phi_a\phi_b|\phi_a\phi_b)} \leqslant 10^{-10}/\max_{c,d}\sqrt{(\phi_c\phi_d|\phi_c\phi_d)},$$

then the distribution $\phi_a\phi_b$ is relatively very small and the corresponding rows and columns of the ERI matrix are omitted from computations. We now use I to denote the set of distributions that survive prescreening. For large enough chemical systems, the scaling $|I| \sim O(N_{\text{bf}})$ has been justified previously.[2,10,28]

In $\mathcal{H}^2$-ERI, we partition the distributions in I adaptively according to their centers. The hierarchical partitioning is stopped when each finest box has fewer than 400 unique distribution centers. However, since many distributions share the same center, they cannot be further split into subsets. These distributions result in a finest box that may contain more than 400 distributions.

The test calculations were carried out on a dual Intel Xeon Gold 6226 CPU computer with a total of 24 cores and 1.5 TB of memory. One hyperthread per core was used. Our software was developed in the GTFock framework[26,29] which contains very efficient C language parallel implementations of HF and DF. Our software uses H2Pack[23] for storing and applying the $\mathcal{H}^2$ matrix representation, and the Simint package[30] for computing ERIs and other integrals.

A. Ground state energy calculations

In this section, we report on testing the accuracy of using $\mathcal{H}^2$-ERI in HF and DFT ground state energy calculations. We assume the exact energies are computed from our

baseline HF and DFT calculations, which use "direct" calculation of the Coulomb matrices. In direct calculation, the ERIs that survive Cauchy-Schwarz screening are recomputed at every SCF iteration, and the Coulomb and exchange matrices are computed via the usual tensor contractions.

We first consider 8 sets of test molecules from the "large system" group of the GMTKN55 benchmark database[25]. Table I shows the average, maximum, and standard deviation of the absolute errors (difference from the baseline) in HF and DFT ground state energy per electron for the molecules in each of the test sets when the Coulomb matrices are calculated by $\mathcal{H}^2$-ERI and by DF. In these calculations, the cc-pVDZ basis set was used. $\mathcal{H}^2$-ERI used a relative error threshold of $\tau = 10^{-7}$.

We observe that the average error for $\mathcal{H}^2$-ERI is always smaller than 1.5×10^{-3} Hartrees (chemical accuracy). For a small number of molecules in the RSE43 test set, the maximum error is larger than chemical accuracy. In these cases, DF also shows error larger than chemical accuracy. These cases are also associated with slow convergence of the SCF iterations compared to the other cases. We note that the averages in Table I do not include cases where SCF did not converge in the baseline calculations, and did not include the molecule "i12p" in the ISOL24 test set, where the HF SCF iterations in the DF case did not converge to the baseline energy. The statistics consider 301 molecules for HF and 292 molecules for DFT, out of 379 total molecules in the 8 test sets.

As the molecules in the GMTKN55 benchmark database are rather small for demonstrat-

TABLE I. Average, maximum, and standard deviation of the absolute errors in ground state energy per electron (in Hartrees) for test sets from GMTKN55. Results for HF and DFT calculations are shown, each using DF and $\mathcal{H}^2$-ERI for calculating Coulomb matrices.

	HF						DFT (B3LYP)					
	DF			$\mathcal{H}^2$-ERI			DF			$\mathcal{H}^2$-ERI		
	ave	max	std	ave	max	std	ave	max	std	ave	max	std
BSR36	2.0e-6	2.4e-6	2.0e-7	6.7e-9	1.6e-8	4.5e-9	1.7e-6	2.0e-6	1.7e-7	7.7e-9	1.6e-8	4.6e-9
CDIE20	1.8e-6	2.2e-6	2.1e-7	1.1e-8	1.8e-8	2.7e-9	1.8e-6	1.4e-5	2.2e-6	1.6e-8	1.4e-7	2.2e-8
DARC	1.9e-6	2.6e-6	4.8e-7	1.0e-8	1.6e-8	4.6e-9	1.6e-6	2.2e-6	4.3e-7	1.1e-8	2.1e-8	5.6e-9
PArel	2.4e-6	9.4e-6	2.2e-6	2.0e-7	3.3e-6	7.1e-7	1.5e-6	2.8e-6	7.4e-7	9.1e-9	1.6e-8	3.9e-9
RSE43	8.9e-5	1.6e-3	2.7e-4	1.7e-5	2.1e-4	4.9e-5	8.4e-5	1.5e-3	2.5e-4	2.4e-5	4.8e-4	7.1e-5
ISO34	2.0e-6	4.9e-6	9.6e-7	1.0e-8	1.5e-8	3.2e-9	1.7e-6	4.3e-6	8.6e-7	1.2e-8	1.9e-8	3.6e-9
ISOL24	1.7e-6	3.2e-6	6.4e-7	2.5e-7	9.5e-6	1.5e-6	2.1e-5	5.9e-4	1.1e-4	1.4e-6	4.2e-5	7.7e-6
C60ISO	1.9e-6	1.9e-6	2.2e-8	1.1e-8	3.8e-8	1.3e-8	1.6e-6	1.6e-6	1.7e-8	2.1e-8	5.5e-8	1.6e-8

ing the efficiency of $\mathcal{H}^2$-ERI, we now consider three types of larger test molecules: alkanes, graphenes, and truncated protein-ligand systems. The latter are derived from the protein-ligand system "1hsg" from the protein data bank. These systems consist of a ligand and a portion of its protein environment within a given distance of the ligand (see Refs 26 and 27 for more information).

Table II shows the ground state energy error for the larger test molecules. The results show that HF and DFT calculations using $\mathcal{H}^2$-ERI, which uses error tolerance $\tau = 10^{-7}$, achieves better than chemical accuracy for this set of test molecules. The number of SCF iterations (see the Appendix) remained essentially the same. We also observe that the energy errors for the aug-cc-pVDZ basis set are larger than for the cc-pVDZ basis set.

Table II also shows the average rank r_{avg} of all the ERI blocks $(I_i|J_i)$ in the $\mathcal{H}^2$ matrix representation. The results show that the average rank is very small relative to the number of auxiliary basis functions N_{aux} used in DF. Further, this rank appears to be bounded and may even decrease when the molecular system size increases (for the same molecular system

TABLE II. Signed errors (in Hartrees) of the ground state energies computed in HF and DFT (B3LYP) with cc-pVDZ and aug-cc-pVDZ basis sets. The dashed entries indicate that the baseline calculation did not converge. N_{bf} is the number of basis functions; N_{aux} is the number of auxiliary basis functions in DF; r_{avg} is the average rank of all the ERI blocks $(I_i|J_i)$ in $\mathcal{H}^2$-ERI; $|I|$ is the number of distributions after prescreening.

	N_{bf}	N_{aux}	r_{avg}	$\lvert I\rvert$	HF		DFT (B3LYP)	
					DF	$\mathcal{H}^2$-ERI	DF	$\mathcal{H}^2$-ERI
cc-pVDZ								
alkane $C_{60}H_{122}$	1510	7910	163	183148	-1.0e-03	-7.2e-06	-8.7e-04	-4.8e-06
alkane $C_{100}H_{202}$	2510	13150	135	310068	-1.7e-03	-1.8e-05	-1.5e-03	-1.0e-05
graphene $C_{96}H_{24}$	1560	8376	205	443319	-4.3e-04	1.3e-05	-3.7e-04	4.5e-05
graphene $C_{150}H_{30}$	2400	12900	227	744909	-6.6e-04	4.1e-05	-5.7e-04	9.8e-05
1hsg30	1240	6572	109	251261	-7.9e-04	-2.1e-05	-6.8e-04	-3.4e-05
1hsg32	1560	8268	105	356063	-9.7e-04	-7.5e-06	-8.4e-04	4.1e-06
aug-cc-pVDZ								
alkane $C_{60}H_{122}$	2598	10330	161	801403	-9.6e-04	8.9e-06	-8.9e-04	9.1e-05
alkane $C_{100}H_{202}$	4318	17170	176	1373643	-1.6e-03	5.5e-05	-1.5e-03	2.6e-04
graphene $C_{54}H_{18}$	1512	6084	317	859638	-2.2e-04	2.2e-05	-1.9e-04	-3.5e-05
graphene $C_{96}H_{24}$	2616	10536	256	2029611	-3.9e-04	2.8e-04	-3.4e-04	6.0e-04
1hsg30	2108	8432	166	1301608	-7.0e-04	1.8e-04	–	–
1hsg32	2652	10608	162	1902022	-8.1e-04	4.6e-04	–	–

type). In contrast, in DF, N_{aux} grows with the system size. Thus we expect that the total storage cost for $\mathcal{H}^2$-ERI will be low compared to DF and only grow linearly with system size, rather than superlinearly for DF.

B. Energy differences and error cancellation

To study possible error cancellation in quantum chemical computation using $\mathcal{H}^2$-ERI, we consider perturbations of the truncated protein-ligand system 1hsg32. This system consists of a ligand and the portion of its protein environment within 3.2 Å of the ligand. Perturbed systems were produced by shifting the ligand toward the protein pocket along the vector joining the pair of ligand and protein atoms that are the closest. Two perturbed systems, corresponding to shifts of 0.25 and 0.5 Å were used. HF calculations were performed using the cc-pVDZ basis set.

Table III shows the ground state energies computed by direct, DF, and $\mathcal{H}^2$-ERI methods, for the original and the two perturbed systems. As previously observed, if the direct calculation is assumed to be exact, then $\mathcal{H}^2$-ERI appears to have smaller errors than DF. However, the main feature in Table III is the error differences between the original and perturbed systems. Again assuming that the direct calculation is exact, we observe that the error in the energy differences for DF are now very small and comparable to the error in the energy differences for $\mathcal{H}^2$-ERI.

DF has significant error cancellation in computing the energy differences (around two digits of accuracy improvement from ground state energies) but $\mathcal{H}^2$-ERI does not. This lack of error cancellation in $\mathcal{H}^2$-ERI is expected because $\mathcal{H}^2$-ERI only focuses on accurately approximating the ERI matrix blocks algebraically. Errors in the computed energies generally would not be biased towards any specific direction, e.g., $\mathcal{H}^2$-ERI has both positive and negative errors in Table II, but DF only has negative errors. Note that even without significant error cancellation, $\mathcal{H}^2$-ERI still has at least comparable accuracy with DF in computing energy differences, as shown in Table III.

TABLE III. Energy (in Hartrees) of three truncated protein-ligand configurations calculated by three methods. Assuming the direct calculation to be exact, the error in energy is greater for DF than for $\mathcal{H}^2$-ERI. Energy differences between the shift=0 configuration and the other two configurations are also shown. Again assuming the direct calculation to be exact, the error in the energy differences is now comparable, due to beneficial error cancellation in DF.

Ligand shift (in Å)	0	0.25	0.5
Energy			
Direct	-3756.9223225	-3756.9079609	-3756.8582130
DF	-3756.9232927	-3756.9089286	-3756.8591746
$\mathcal{H}^2$-ERI	-3756.9223299	-3756.9079674	-3756.8582221
Error in energy			
DF	-9.70e-04	-9.68e-04	-9.62e-04
$\mathcal{H}^2$-ERI	-7.49e-06	-6.53e-06	-9.09e-06
Energy difference			
Direct		0.0143616	0.0641095
DF		0.0143641	0.0641181
$\mathcal{H}^2$-ERI		0.0143625	0.0641079
Error in energy difference			
DF		2.53e-06	8.62e-06
$\mathcal{H}^2$-ERI		9.57e-07	-1.60e-06

C. Computational and memory storage costs

The use of $\mathcal{H}^2$-ERI and DF both require a precomputation step. For $\mathcal{H}^2$-ERI, this is the construction of the $\mathcal{H}^2$ matrix representation. For DF, this is the construction of $(ab|\alpha)$ and $(\alpha|\beta)^{-1}$. After the precomputation step, the SCF iterations calculate the Coulomb matrix once per iteration.

For $\mathcal{H}^2$-ERI and DF, Table IV lists the execution timings for the precomputation and for one calculation of the Coulomb matrix. The memory storage costs for $\mathcal{H}^2$-ERI and DF are also listed. For reference, the timings for direct calculation of the Coulomb matrix are also

shown. For direct calculation, 8-way symmetry and Schwarz screening of the ERI tensor are exploited. Precomputation and storage are not needed in direct calculation. Timings for calculating the exact exchange term and exchange-correlation term in DFT term are provided in the Appendix.

Although the execution time for calculating the Coulomb matrix is lower for $\mathcal{H}^2$-ERI than for DF, the results show that the main advantage of $\mathcal{H}^2$-ERI over DF is that of memory storage requirements. DF requires storing $(ab|\alpha)$ and $(\alpha|\beta)^{-1}$, which can be very large. Thus $\mathcal{H}^2$-ERI extends the size of the molecular systems that can be processed in memory. In particular, if the DF memory requirement exceeds that of a given machine, then $\mathcal{H}^2$-ERI

TABLE IV. Execution time (in seconds) and storage cost (in GB) for one Coulomb matrix calculation using direct calculation, DF, and $\mathcal{H}^2$-ERI. For DF and $\mathcal{H}^2$-ERI, the execution time is given separately for precomputation ("precomp.") and the calculation of Coulomb matrices ("J"). The timings for calculating Coulomb matrices are averaged over 5 runs.

	Direct	DF			$\mathcal{H}^2$-ERI		
	J	storage	precomp.	J	storage	precomp.	J
cc-pVDZ							
alkane $C_{60}H_{122}$	25.52	17.3	3.20	0.19	2.8	19.89	0.05
alkane $C_{100}H_{202}$	74.62	48.6	9.07	0.54	4.8	25.64	0.07
graphene $C_{96}H_{24}$	101.63	41.1	5.53	0.72	5.4	67.82	0.11
graphene $C_{150}H_{30}$	273.56	104.5	13.73	1.11	9.3	136.33	0.15
1hsg30	32.75	17.7	2.70	0.19	4.7	34.08	0.06
1hsg32	60.85	30.6	4.76	0.37	6.4	45.93	0.08
aug-cc-pVDZ							
alkane $C_{60}H_{122}$	350.29	101.3	12.39	1.43	21.7	130.60	0.30
alkane $C_{100}H_{202}$	1025.82	287.0	39.00	3.27	41.5	241.71	0.61
graphene $C_{54}H_{18}$	338.48	67.6	9.76	0.76	16.1	349.22	0.25
graphene $C_{96}H_{24}$	1597.58	260.2	33.98	2.73	34.4	664.28	0.53
1hsg30	725.67	130.6	13.90	2.26	47.8	478.04	0.61
1hsg32	1481.97	236.1	25.54	2.65	70.8	683.31	0.88

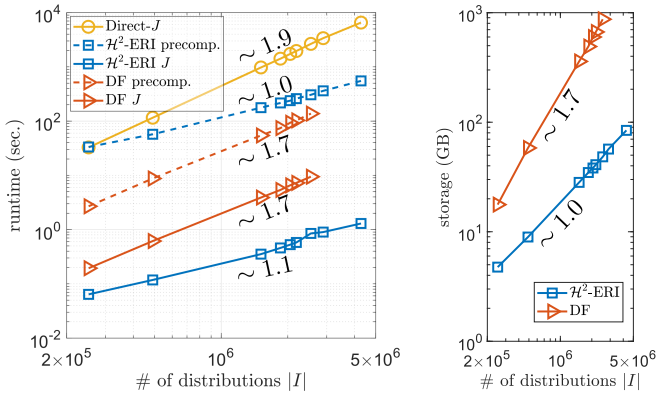
could be used instead.

On the other hand, the precomputation cost of $\mathcal{H}^2$-ERI is higher than that of DF. High precomputation cost is a common issue with the use of $\mathcal{H}^2$ matrix representations in general, although such precomputation costs are amortized over each time the $\mathcal{H}^2$ matrix representation is used (e.g., to form the Coulomb matrix).

D. Linear scaling

The execution time of the $\mathcal{H}^2$-ERI method (both precomputation and calculating the Coulomb matrix) scales linearly with the number of distributions $|I|$ (after prescreening), as verified in Figure 4 for a sequence of 1hsg systems ranging from 124 to 1208 atoms with the cc-pVDZ basis set. The storage cost also scales linearly.

In contrast, DF computation and storage scales with an exponent of ~ 1.7 over this range. Due to better scaling of $\mathcal{H}^2$-ERI, the precomputation time for $\mathcal{H}^2$-ERI would be smaller than that of DF for large enough systems. However, if the three-index tensors of DF are stored in memory, the bottleneck for DF is memory usage. Calculations for DF were not carried out if the memory storage exceeded 1 TB.

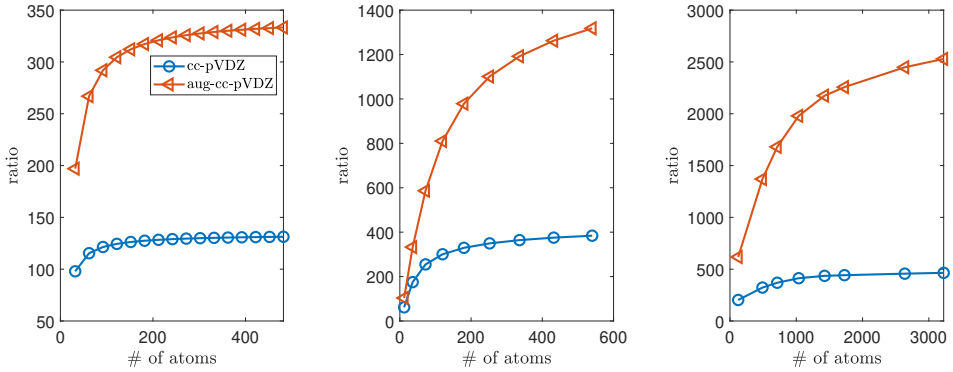


(a) Execution time (b) Storage

FIG. 4. Execution time and storage cost vs. $|I|$ for calculating the Coulomb matrix via direct calculation, $\mathcal{H}^2$-ERI, and DF. The molecular systems are 1hsg systems of different sizes. The estimated slope of each curve in these log-log plots are marked along the curve.

Since the scaling of $\mathcal{H}^2$-ERI is linear with respect to $|I|$, the scaling with respect to the number of basis functions N_{bf} (equivalently, the number of atoms) is also linear if $|I|$ is linear

in N_{bf}. This holds true for large molecular system sizes. Figure 5 plots the ratio of $|I|$ to N_{bf} for a range of molecular system sizes. A ratio curve turning flat indicates that $|I|$ becomes linear in N_{bf}. As can be observed, for systems that are more globular and for basis sets with more diffuse functions, the point at which the ratio curve turns flat is larger.



(a) alkane (b) graphene (c) 1hsg

FIG. 5. Ratio of $|I|$ (after prescreening) to N_{bf} for alkanes, graphenes, and 1hsg systems of different sizes. Results for cc-pVDZ and aug-cc-pVDZ basis sets are given.

V. RELATIONSHIP BETWEEN $\mathcal{H}^2$-ERI AND CFMM

$\mathcal{H}^2$-ERI and CFMM share the same hierarchical partitioning of I. From an algebraic viewpoint, CFMM is equivalent to multiplication by the ERI matrix in a specific $\mathcal{H}^2$ matrix representation. This $\mathcal{H}^2$ matrix representation in CFMM differs from the $\mathcal{H}^2$ matrix representation constructed for $\mathcal{H}^2$-ERI in two important ways: the definition of FF blocks and the compression of FF blocks.

In CFMM, $(I_i|I_j)$ is a FF block if the following two conditions hold: boxes i and j are well-separated, and the numerical supports of distributions in box i do not overlap with those in box j. The numerical support of a GTF distribution is characterized by a ball in CFMM and the radius of this ball is referred to as the "extent" of the distribution. In comparison, $\mathcal{H}^2$-ERI does not require the second condition. As a result of this difference, CFMM defines far more NF blocks than $\mathcal{H}^2$-ERI since, for typical problems, the numerical support of a distribution usually spreads over several boxes at the leaf level of the partition tree.

Consider the 1D hierarchical partitioning in Figure 3 as an example. The FF and NF blocks in $\mathcal{H}^2$-ERI plotted in Figure 3 are independent of the actual distribution extents. Assuming that all the distributions have their extents being 0.9× (and 1.4×) the edge length of a leaf box, the corresponding FF and NF blocks defined in CFMM are plotted in Figure 6a (and Figure 6b). In addition, Table V lists the total number of entries of the NF and FF blocks defined in both CFMM and $\mathcal{H}^2$-ERI for several examples. As can be observed from both the abstract and practical examples, far more NF blocks are defined in CFMM than in $\mathcal{H}^2$-ERI, especially when basis sets with diffuse functions are used and when the molecular structure is globular.

It is worth noting that, in CFMM, distributions in each leaf-level set I_i are further grouped into "branches" according to their extents and each NF block is then subdivided into smaller blocks, some of which can also be characterized as FF blocks and compressed. Figure 6 does not illustrate the "branch" idea for simplicity, but the number of NF block entries counted in Table V have taken this approach into account.

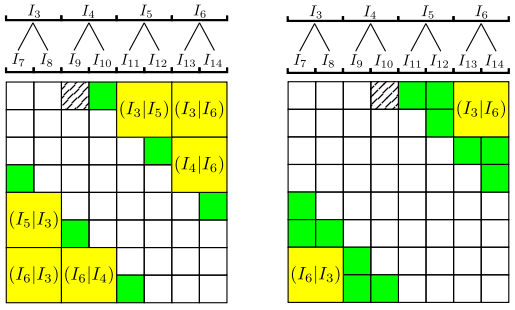


FIG. 6. The NF (white) and FF (yellow and green) blocks defined in CFMM associated with the 1D example in Figure 3 when the distribution extent equals (a) 0.9× and (b) 1.4× the edge length of a leaf box. In (a), the hatched block $(I_7|I_9)$ is a NF block because I_7 and I_9 have overlapping distributions. Similarly for the hatched block $(I_7|I_{10})$ in (b). Also, $(I_3|I_5)$ is a FF block in (a) but is not in (b) due to the assumption of a larger distribution extent in (b).

In both $\mathcal{H}^2$-ERI and CFMM, the NF blocks can be precomputed, stored in memory, and recalled when needed. Alternatively, the NF blocks can be computed when they are needed. The bottleneck in CFMM is usually the storage or computation of the NF blocks[17]. Thus the reduction in the number of NF blocks in $\mathcal{H}^2$-ERI compared to CFMM alleviates this bottleneck.

TABLE V. Total number of entries in the FF and NF blocks defined in $\mathcal{H}^2$-ERI and in CFMM. "Ratio" refers to the ratio of the number of NF block entries in CFMM to that in $\mathcal{H}^2$-ERI. Symmetry of the NF and FF blocks in the ERI matrix is considered.

	CFMM		$\mathcal{H}^2$-ERI		ratio
	NF	FF	NF	FF	
cc-pVDZ					
alkane $C_{60}H_{122}$	2.2e09	1.5e10	1.1e09	1.6e10	2.0
alkane $C_{100}H_{202}$	3.6e09	4.5e10	1.6e09	4.7e10	2.3
graphene $C_{96}H_{24}$	1.5e10	8.3e10	1.8e09	9.7e10	8.4
graphene $C_{150}H_{30}$	3.2e10	2.5e11	4.4e09	2.7e11	7.4
1hsg30	8.0e09	2.4e10	1.4e09	3.0e10	5.8
1hsg32	1.3e10	5.1e10	2.1e09	6.1e10	5.9
aug-cc-pVDZ					
alkane $C_{60}H_{122}$	3.5e10	2.9e11	4.8e09	3.2e11	7.3
alkane $C_{100}H_{202}$	6.2e10	8.8e11	7.9e09	9.4e11	7.8
graphene $C_{54}H_{18}$	1.5e11	2.2e11	6.1e09	3.6e11	24.3
graphene $C_{96}H_{24}$	4.2e11	1.6e12	1.1e10	2.0e12	38.0
1hsg30	2.3e11	6.2e11	1.1e10	8.4e11	21.6
1hsg32	4.0e11	1.4e12	1.6e10	1.8e12	25.3

The reason for this restricted definition of FF blocks in CFMM is that the application of the multipole expansion technique used in CFMM is restrictive, i.e., requiring two distributions to be nonoverlapping. Using multipole expansions, a FF block $(I_i|I_j)$ in CFMM is compressed into the low-rank form

$$(I_i|I_j) \approx T_i B_{i,j} S_j,$$

where S_j corresponds to the source-to-multipole linear operator for box j, $B_{i,j}$ corresponds to the multipole-to-local linear operator from box j to box i, and T_i corresponds to the local-to-target linear operator for box i. Comparing this approximation with $(I_i|I_j) \approx U_i (I_i^{\mathrm{id}}|I_j^{\mathrm{id}}) U_j^T$ in $\mathcal{H}^2$-ERI, we can note that T_i, S_j, and $B_{i,j}$ in CFMM correspond to U_i, U_j^T, and $(I_i^{\mathrm{id}}|I_j^{\mathrm{id}})$ in $\mathcal{H}^2$-ERI, respectively.

In CFMM, all operators T_i, S_j, and $B_{i,j}$ can be analytically computed using I_i, I_j, and geometric information for boxes i and j. These operators can be dynamically computed when needed. In $\mathcal{H}^2$-ERI, however, the components U_i and I_i^{id} must be precomputed via ID approximation of ERI blocks. Experimentally, the computation cost in $\mathcal{H}^2$-ERI for constructing U_i and I_i^{id} is usually multiple times the computation cost for evaluating and compressing the NF blocks, and the storage cost for U_i and I_i^{id} is very small compared to the storage cost of the NF blocks. Thus, compared to CFMM, the advantage of $\mathcal{H}^2$-ERI having far fewer NF blocks could easily offset the additional precomputation and storage cost required for U_i and I_i^{id}.

VI. CONCLUSION

The main advantage of $\mathcal{H}^2$-ERI over DF for calculating the Coulomb matrix is the dramatically reduced storage needed for an accurate representation of the ERI matrix, allowing large-scale quantum chemical computations in memory without recomputing ERIs. On the other hand, $\mathcal{H}^2$-ERI has a relatively expensive precomputation step compared to DF. The cost of this precomputation, however, is usually much less than the cost of forming the Coulomb matrix directly (Table IV), and can be amortized over multiple SCF iterations where the Coulomb matrix is calculated each time.

Compared to CFMM, $\mathcal{H}^2$-ERI reduces the number of NF blocks. This reduces the storage required for the compressed representation of the ERI tensor (if the NF blocks are precomputed and stored), or the computation required when computing the Coulomb matrix (if the NF blocks are computed dynamically when needed). We note that the precomputation time we have reported for $\mathcal{H}^2$-ERI includes the time for computing the required NF blocks.

Once the ERI tensor is represented in $\mathcal{H}^2$ matrix form, the possibility exists for using this representation to accelerate other computations involving the ERI tensor. The $\mathcal{H}^2$ matrix form could also be used to efficiently represent the three-index tensor $(ab|\alpha)$ in DF.

ACKNOWLEDGMENTS

The authors would like to thank C. David Sherrill for helpful discussions. The authors gratefully acknowledge funding from the National Science Foundation grant ACI-1609842.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]J. L. Whitten, The Journal of Chemical Physics **58**, 4496 (1973).

[2]K. Eichkorn, O. Treutler, H. Öhm, M. Häser, and R. Ahlrichs, Chemical Physics Letters **240**, 283 (1995).

[3]F. Weigend, Physical Chemistry Chemical Physics **4**, 4285 (2002).

[4]A. Sodt, J. E. Subotnik, and M. Head-Gordon, The Journal of Chemical Physics **125**, 194109 (2006).

[5]F. Aquilante, R. Lindh, and T. Bondo Pedersen, The Journal of Chemical Physics **127**, 114107 (2007).

[6]F. Aquilante, L. Gagliardi, T. B. Pedersen, and R. Lindh, The Journal of Chemical Physics **130**, 154107 (2009).

[7]S. Reine, E. Tellgren, A. Krapp, T. Kjærgaard, T. Helgaker, B. Jansik, S. Høst, and P. Salek, The Journal of Chemical Physics **129**, 104101 (2008).

[8]P. Merlot, T. Kjærgaard, T. Helgaker, R. Lindh, F. Aquilante, S. Reine, and T. B. Pedersen, Journal of Computational Chemistry **34**, 1486 (2013).

[9]A. C. Ihrig, J. Wieferink, I. Y. Zhang, M. Ropo, X. Ren, P. Rinke, M. Scheffler, and V. Blum, New Journal of Physics **17**, 093020 (2015).

[10]C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chemical Physics Letters **230**, 8 (1994).

[11]C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chemical Physics Letters **253**, 268 (1996).

[12]R. Łazarski, A. M. Burow, and M. Sierka, Journal of Chemical Theory and Computation **11**, 3029 (2015).

[13]R. Łazarski, A. M. Burow, L. Grajciar, and M. Sierka, Journal of Computational Chemistry **37**, 2518 (2016).

[14]M. Challacombe, E. Schwegler, and J. Almlöf, The Journal of Chemical Physics **104**, 4685 (1996).

[15]M. Challacombe and E. Schwegler, The Journal of Chemical Physics **106**, 5526 (1997).

[16]J. M. Pérez-Jordá and W. Yang, The Journal of Chemical Physics **107**, 1218 (1997).

[17]M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271**, 51 (1996).

[18]C. A. Lewis, J. A. Calvin, and E. F. Valeev, Journal of Chemical Theory and Computation **12**, 5868 (2016).

[19]X. Xing and E. Chow, SIAM Journal on Scientific Computing **42**, A162 (2020).

[20]H. Cheng, Z. Gimbutas, P. G. Martinsson, and V. Rokhlin, SIAM Journal on Scientific Computing **26**, 1389 (2005).

[21]Y. Aizenbud and A. Averbuch, Information and Inference: A Journal of the IMA **8**, 445 (2019).

[22]W. Hackbusch, B. Khoromskij, and S. A. Sauter, in *Lectures on Applied Mathematics* (Springer-Verlag, Berlin, 2000) pp. 9–29.

[23]H. Huang, X. Xing, and E. Chow, ACM Transactions on Mathematical Software (2020), to appear (doi:10.1145/3412850).

[24]P. Sałek, S. Høst, L. Thøgersen, P. Jørgensen, P. Manninen, J. Olsen, B. Jansík, S. Reine, F. Pawłowski, E. Tellgren, *et al.*, The Journal of Chemical Physics **126**, 114110 (2007).

[25]L. Goerigk, A. Hansen, C. Bauer, S. Ehrlich, A. Najibi, and S. Grimme, Physical Chemistry Chemical Physics **19**, 32184 (2017).

[26]E. Chow, X. Liu, M. Smelyanskiy, and J. R. Hammond, The Journal of Chemical Physics **142**, 104103 (2015).

[27]E. Chow, X. Liu, S. Misra, M. Dukhan, M. Smelyanskiy, J. R. Hammond, Y. Du, X. Liao, and P. Dubey, The International Journal of High Performance Computing Applications **30**, 85 (2016).

[28]T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (John Wiley & Sons, 2014).

[29]H. Huang, C. D. Sherrill, and E. Chow, The Journal of Chemical Physics **152**, 024122 (2020).

[30]B. P. Pritchard and E. Chow, Journal of Computational Chemistry **37**, 2537 (2016).

APPENDIX

Corresponding to the data in Table II and Table IV, the table below shows the number of SCF iterations and the average time per iteration for direct calculation of the Coulomb matrix, direct calculation of the the exchange matrix, and calculation of the DFT exchange-correlation matrix (not including time for the exchange matrix). We note that our implementation of the DFT exchange-correlation matrix calculation is not optimized.

	HF #Iter			DFT (B3LYP) #Iter			Time (sec.)		
	Direct	DF	$\mathcal{H}^2$-ERI	Direct	DF	$\mathcal{H}^2$-ERI	Direct-J	Direct-K	XC
cc-pVDZ									
alkane $C_{60}H_{122}$	10	10	10	12	12	12	25.5	23.2	44.3
alkane $C_{100}H_{202}$	10	10	10	12	12	12	74.6	67.5	166.1
graphene $C_{96}H_{24}$	15	15	15	13	13	14	101.6	94.1	35.9
graphene $C_{150}H_{30}$	17	17	17	14	14	14	273.6	253.9	103.2
1hsg30	15	16	15	22	21	19	32.8	29.9	26.9
1hsg32	16	16	16	17	19	19	60.9	55.5	44.3
aug-cc-pVDZ									
alkane $C_{60}H_{122}$	11	11	11	19	19	17	350.3	305.6	44.3
alkane $C_{100}H_{202}$	11	11	11	16	16	16	1025.8	892.8	166.1
graphene $C_{54}H_{18}$	14	14	14	13	11	11	338.5	304.3	35.9
graphene $C_{96}H_{24}$	15	15	15	14	13	12	1597.6	1432.5	103.2
1hsg30	16	16	16	-	-	-	725.7	634.4	26.9
1hsg32	19	19	19	-	-	-	1482.0	1330.3	101.6