SIAM J. SCI. COMPUT.
Vol. 42, No. 1, pp. A162–A186
© 2020 Society for Industrial and Applied Mathematics

FAST COULOMB MATRIX CONSTRUCTION VIA COMPRESSING THE INTERACTIONS BETWEEN CONTINUOUS CHARGE DISTRIBUTIONS*

XIN XING† AND EDMOND CHOW†

Abstract. The continuous fast multipole method (CFMM) is well known for its asymptotically linear complexity for constructing the Coulomb matrix in quantum chemistry. However, in practice, CFMM must evaluate a large number of interactions directly, being unable to utilize multipole expansions for interactions between overlapping continuous charge distributions. Instead of multipole expansions, we propose a technique for compressing the interactions between charge distributions into low-rank form, resulting in far fewer interactions that must be computed directly. The technique is used with an $\mathcal{H}^2$ matrix representation of the electron repulsion integral tensor. Numerical tests on alkane and protein molecules show that our new method requires 5 to 18 times fewer direct interactions to be evaluated than in CFMM, leading to essentially an equal reduction in storage or computation cost.

Key words. continuous fast multipole method, electron repulsion integral tensor, hierarchical matrix representation, block low-rank, proxy point method

AMS subject classifications. 15B99, 65F99, 65Z05

DOI. 10.1137/19M1252855

1. Introduction.

In quantum chemistry, one of the main steps in many methods is constructing the Coulomb matrix, which can be defined as

$$J_{ab} = \sum_{c,d} (\phi_a \phi_b | \phi_c \phi_d) D_{cd}, \tag{1.1}$$

where D is a density matrix and $(\phi_a \phi_b | \phi_c \phi_d)$ denotes an entry of a four-dimensional electron repulsion integral (ERI) tensor. Each entry of the ERI tensor is defined as

$$(\phi_a \phi_b | \phi_c \phi_d) = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \phi_a(r_1) \phi_b(r_1) \frac{1}{|r_1 - r_2|} \phi_c(r_2) \phi_d(r_2) \mathrm{d}r_1 \mathrm{d}r_2, \tag{1.2}$$

where ϕ_a, etc., are known basis functions. In quantum chemical methods where high accuracy is desired, the standard basis functions are Gaussian-type functions (GTFs)

$$\phi_a(r) = (x - x_a)^l (y - y_a)^m (z - z_a)^n e^{-\alpha |r - r_a|^2}, \tag{1.3}$$

where $r_a = (x_a, y_a, z_a)$ is the center of the function, α is an "exponent," and $(l+m+n)$ is the total angular momentum. (In practice, the basis functions are a known linear combination of GTFs that have the same center and are called *contracted* GTFs. This fact does not change the development of this paper, and it will be ignored until section 5 on numerical experiments.)

In self-consistent field iterations, the Coulomb matrix is constructed repeatedly for different density matrices while the ERI tensor is fixed. The computational challenge

*Submitted to the journal's Methods and Algorithms for Scientific Computing section March 27, 2019; accepted for publication (in revised form) October 21, 2019; published electronically January 8, 2020.

https://doi.org/10.1137/19M1252855

Funding: This work was supported NSF under grant ACI-1609842.

†School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA (xxing33@gatech.edu, echow@cc.gatech.edu).

in constructing the Coulomb matrix is the fact that the ERIs are expensive to compute and, for typical numbers of basis functions, the distinct, nonnegligible ERIs are too numerous to store in memory. The ERI tensor is central to many quantum chemical methods, and a variety of techniques have been developed to approximate the ERI tensor to reduce computation and/or storage costs.

From (1.1) and (1.2), constructing the Coulomb matrix involves calculating the Coulomb potential for a system of *continuous charge distributions*. Here, $\phi_a\phi_b$ and $\phi_c\phi_d$ are distributions; the latter multiplied by the corresponding charge weight D_{cd} is a charge distribution. The discrete case, i.e., the Coulomb potential for a system of *point* charges, can be efficiently calculated by the fast multipole method (FMM) [11]. For constructing the Coulomb matrix, the continuous FMM (CFMM) and related methods have been developed for the case of charge distributions [4, 5, 31, 36, 41, 42]. These methods use the multipole expansion technique from FMM to represent the interactions between "well-separated" distributions in "compressed" form. The evaluation of such interactions is thereby accelerated. The remaining interactions are evaluated directly.

CFMM, however, is not as efficient as one would hope. For two distributions to be well-separated, they cannot overlap (to be described precisely in the next section), due to the use of multipole expansions. For typical problems, a large number of distributions overlap, and thus the number of interactions that must be evaluated directly is large [36]. These direct computations dominate the computational cost of CFMM.

In this paper, instead of multipole expansions, we propose a different technique for compressing the interactions between distributions. The new technique allows us to compress far more interactions than could be compressed using multipole expansions, resulting in far fewer interactions that must be computed directly. The technique computes low-rank approximations in the form of an interpolative decomposition [6, 13] (to be explained in subsection 3.2.1). It is known that such algebraic techniques for compressing interactions between point charges can be more efficient (in terms of the rank of the approximation and the range of applicability) than analytic techniques like multipole expansions [12]. However, our technique is not purely algebraic. We also use the knowledge that the interactions are Coulombic to avoid needing to explicitly form a matrix of all actual interactions before compressing them. Here, the technique has similarity to proxy surface methods [19, 28, 29, 44] and the kernel-independent FMM [46, 47].

We use this new compression technique to construct an $\mathcal{H}^2$ matrix representation [14, 15] (to be explained in section 3) of the ERI tensor. We then use the linear-scaling matrix-vector multiplication algorithm [15] available for $\mathcal{H}^2$ matrices to efficiently construct the Coulomb matrix. It is already established that FMM is equivalent to the fast matrix-vector multiplication for a matrix in $\mathcal{H}^2$ format, where the low-rank approximations to certain off-diagonal blocks in this format are from multipole expansions [33, 38, 48]. Similarly, CFMM can be interpreted as a matrix-vector multiplication. The ERI tensor with elements $(\phi_a\phi_b|\phi_c\phi_d)$ can be regarded as a matrix by folding together its first two dimensions and folding together its last two dimensions so that (a, b) denotes a matrix row index and (c, d) denotes a matrix column index. We will refer to this matrix as the *ERI matrix*. At the same time, the density matrix can be regarded as a vector. From this viewpoint, the tensor contraction (1.1) can be regarded as a matrix-vector multiplication. CFMM is equivalent to multiplication by the ERI matrix in $\mathcal{H}^2$ format, where multipole expansions are used to compress the interactions between well-separated distributions. In this context, our compression

method can also be regarded as extending the applicability of $\mathcal{H}^2$ matrix formats to interactions between distributions rather than just between points.

Overview. The overall algorithm for fast Coulomb matrix construction is as follows. The first step is to construct an $\mathcal{H}^2$ matrix representation of the ERI matrix (section 3). To construct the $\mathcal{H}^2$ matrix representation, ERI matrix blocks corresponding to distant Coulomb interactions are compressed into low-rank form. The naive approach to do this is to construct the ERI matrix blocks to be compressed and then apply a rank-revealing algebraic decomposition such as the singular value decomposition. However, forming these matrix blocks is prohibitively expensive and would lead to a construction cost that is quadratic in the number of distributions.

If the Coulomb interactions were between point charges, then the low-rank approximations could be generated via a physically motivated analytic technique called the *proxy surface method* [19, 28]. The proxy surface method is used to efficiently compress the interactions between a box of point charges and all other point charges well-separated from the box. The key feature of the method is that it only needs to form the intermediate interactions between the box and a constant-sized set of "proxy points" on a surface that encloses the box, which is much cheaper than forming all the actual interactions. However, in our case, the Coulomb interactions are between continuous charge distributions that potentially overlap. For this case, we propose a variant of the proxy surface method where the proxy points are chosen in a different manner (section 4). We provide a theoretical justification for this new compression technique. Experimentally, the $\mathcal{H}^2$ matrix construction cost turns out to be nearly linear in the number of distributions.

This completes the first step of constructing the $\mathcal{H}^2$ matrix representation of the ERI matrix. The second step is to simply use the established fast matrix-vector multiplication algorithm for $\mathcal{H}^2$ matrices [15] (where the vector is the vectorized density matrix) to construct the Coulomb matrix. This multiplication algorithm has linear computation cost. This cost is still directly related to the number of direct interactions in the $\mathcal{H}^2$ matrix representation, but we have effectively reduced this number compared to CFMM.

The Coulomb matrix is used in many quantum chemical methods. In the Hartree–Fock method, each self-consistent field iteration requires computing a Coulomb matrix from a density matrix and the ERI tensor. The ERI tensor is fixed, and thus the cost of constructing the $\mathcal{H}^2$ matrix representation of the ERI matrix can be amortized over all the matrix-vector multiplications in the self-consistent field iterations.

In section 5, results of numerical tests of the above procedures are presented. In section 2, to further motivate our approach, we show that many interactions that cannot be compressed by CFMM do indeed have low-rank form.

Related work. Besides CFMM, there have been significant efforts in the past to develop and use compressed representations of the ERI tensor. Density fitting (e.g., [8, 39, 43]) and its variants [2, 9, 22, 21, 30] represent the 4-index ERI tensor as the contraction of two 3-index tensors. Other decompositions of the 4-index ERI tensor, called tensor hypercontraction, have also been recently developed [20].

Block low-rank matrix representations have been used elsewhere in quantum chemistry. Lewis, Calvin, and Valeev [25] use a 1-level matrix representation called "clustered low-rank" for the 2-index and 3-index tensors in density fitting. Lu and Ying [26] use interpolative decompositions to produce approximations of the ERI tensor in tensor hypercontraction form.

2. Limitations of CFMM. In FMM and CFMM, space is partitioned into boxes, and the potential at a point far from a box due to the point charges (FMM) or charge distributions (CFMM) centered in the box is expressed in terms of a multipole expansion. In FMM, if two boxes are *not adjacent*, then the multipole expansion could be used to compactly describe the pairwise interactions between the point charges across the two boxes. In CFMM, it is more complicated to determine whether or not a multipole expansion could be used to approximate the interactions between charge distributions.

To explain the issue with charge distributions, consider the distribution $\phi_a\phi_b$ which is a product of two GTFs. By the Gaussian product rule, $\phi_a\phi_b$ itself is a GTF with center along the line joining the centers of ϕ_a and ϕ_b. For a distribution ϕ, in general, define its *extent* λ with *precision* τ as the radius of the smallest ball centered at the center of the GTF such that $|\phi(r)|$ is less than τ outside the ball [31]. Whether two distributions overlap depends on whether these balls overlap.

Now consider two sets of distributions, $\Phi = \{\varphi_i\}$ being the distributions centered in a given box and $\Theta = \{\theta_j\}$ being the distributions centered in a nonadjacent box. Define V_Φ to be the *numerical support* of Φ, that is, the convex hull of the balls corresponding to the distributions in Φ, and define V_Θ similarly. In this notation, $(\Phi|\Theta)$ is a block of the ERI matrix, and each of its entries is an ERI,

$$(\varphi_i|\theta_j) = \int_{V_\Phi}\int_{V_\Theta} \varphi_i(r_1)\frac{1}{|r_1 - r_2|}\theta_j(r_2)\mathrm{d}r_1\mathrm{d}r_2, \quad \varphi_i \in \Phi,\ \theta_j \in \Theta.$$

If V_Φ and V_Θ do not overlap, then we can approximate $1/|r_1 - r_2|$ by a multipole expansion,

$$(\varphi_i|\theta_j) \approx \int_{V_\Phi}\int_{V_\Theta} \varphi_i(r_1)\left(\sum_{l=0}^{s}\sum_{m=-l}^{l} |r_2|^l Y_l^{-m}(r_2)\frac{Y_l^m(r_1)}{|r_1|^{l+1}}\right)\theta_j(r_2)\mathrm{d}r_1\mathrm{d}r_2,$$

where $Y_l^m(r)$ is the spherical harmonic function of degree l and order m. Here, the multipole expansion is of degree s and is centered at the origin which is assumed to be the center of V_Θ. The above expansion gives an approximation in degenerate form,

$$(\varphi_i|\theta_j) \approx \sum_{l=0}^{s}\sum_{m=-l}^{l}\left(\int_{V_\Phi}\frac{Y_l^m(r_1)}{|r_1|^{l+1}}\varphi_i(r_1)\mathrm{d}r_1\right)\left(\int_{V_\Theta}|r_2|^l Y_l^{-m}(r_2)\theta_j(r_2)\mathrm{d}r_2\right),$$

which is equivalent to a rank-$(s+1)^2$ approximation of $(\Phi|\Theta)$.

If V_Φ and V_Θ do overlap, then the above approximation is not possible, since the multipole expansion of $1/|r_1 - r_2|$ diverges when r_1 and r_2 are equal. In this situation, computing the interactions between distributions in these two boxes cannot be accelerated by CFMM, and these interactions must be computed directly. The distinguishing feature of CFMM compared to FMM is the need to identify sets of distributions whose numerical supports do not overlap.

An important observation is that even if V_Φ and V_Θ do overlap, the ERI matrix block $(\Phi|\Theta)$ may still be numerically low-rank if Φ and Θ are distributions centered in nonadjacent boxes. We simply do not have the analytical apparatus to find these low-rank approximations. In this paper, a new technique is proposed to find such approximations.

To illustrate the observation that $(\Phi|\Theta)$ may be numerically low-rank although there is no corresponding known degenerate expansion, consider two nonadjacent

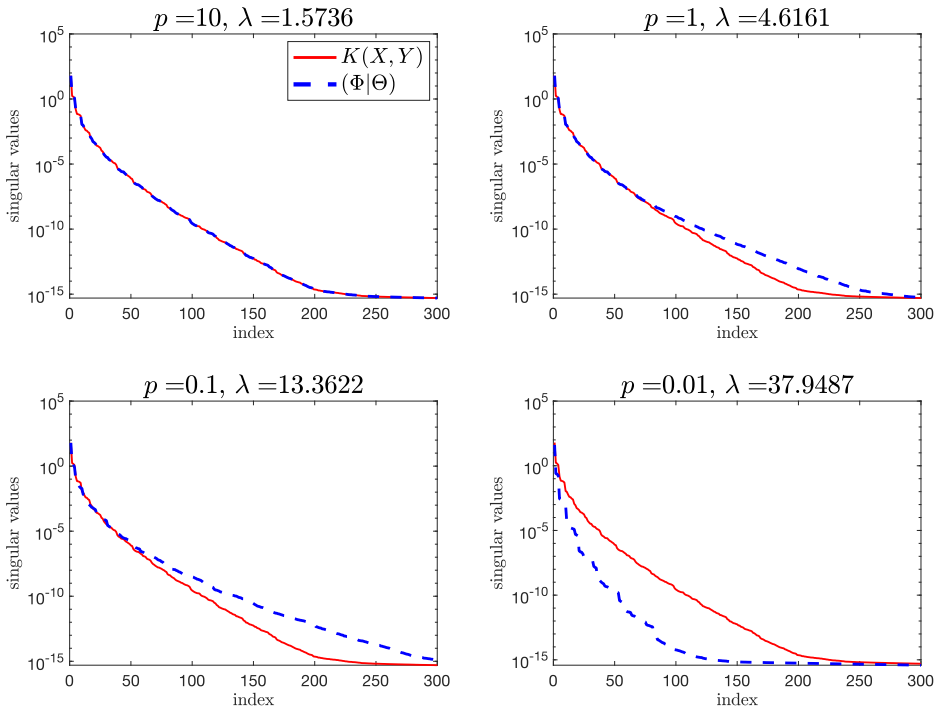


FIG. 2.1. *First* 300 *singular values of* $K(X,Y)$ *and* $(\Phi|\Theta)$ *for GTFs with different exponents* p. *For each* p, *the corresponding value of the extent* λ *is also shown. Only the ERI block* $(\Phi|\Theta)$ *with* $p = 10$ *can be compressed using multipole expansions in CFMM, although* $(\Phi|\Theta)$ *in other cases is also numerically low-rank.*

cubical boxes of edge length $L = 5$ centered at $(0,0,0)$ and $(2L,0,0)$. For each box, select 600 GTF distributions of the form $(p/\pi)^{3/2}e^{-p|r-r_a|^2}$ with the same exponent p and different centers r_a randomly distributed in the box. These GTFs represent very simple "spherical" distributions.

As before, denote the two sets of GTFs as $\Phi = \{\varphi_i\}$ and $\Theta = \{\theta_j\}$. Denote the center of each distribution φ_i as x_i and the center of each distribution θ_j as y_j. Each entry of the ERI matrix block $(\Phi|\Theta)$ can be calculated analytically as

$$(\varphi_i|\theta_j) = \frac{1}{|x_i - y_j|}\operatorname{erf}\left(\sqrt{\frac{p}{2}}|x_i - y_j|\right), \quad i,j = 1,2,\ldots 600. \tag{2.1}$$

Figure 2.1 plots the first 300 singular values of $(\Phi|\Theta)$ for four different cases, corresponding to different values of the exponent p in the GTFs, and thus GTFs with different extents. The extent $\lambda = \sqrt{\frac{1}{p}\left(\frac{3}{2}\ln\frac{p}{\pi} + \ln\frac{1}{\tau}\right)}$ for each value of p is also shown for each subfigure, assuming the extent precision $\tau = 10^{-10}$.

For comparison, we also plot in each subfigure the singular values of the matrix which we denote as $K(X,Y)$, consisting of the entries $K(x_i,y_j)$ for all pairs of centers x_i and y_j, with $K(x,y) = 1/|x-y|$. This is the matrix that describes the Coulomb interactions if we had point charges (instead of distributions) at the location of each center. Since the two boxes under consideration are nonadjacent, the singular values of $K(X,Y)$ decay rapidly, and $K(X,Y)$ is numerically low-rank. FMM considers these two sets of point charges to be well separated.

When $p = 10$, the extent λ of the distributions is small, and $(\Phi|\Theta)$ and $K(X, Y)$ have very similar singular values. When $p = 1$ and $p = 0.1$, the extent is larger, and the distributions from the two boxes can overlap. CFMM would consider the interactions between these two boxes to be near-range in these cases, i.e., interactions based on multipole expansions cannot be used. However, Figure 2.1 shows that the singular value decay of $(\Phi|\Theta)$ and $K(X, Y)$ is similar for the first 8 or more decades of singular values. Thus $(\Phi|\Theta)$ in these cases are also numerically low-rank.

When $p = 0.01$, the distributions are very diffusive, and the singular value decay of $(\Phi|\Theta)$ is even faster than that of $K(X, Y)$. This odd result turns out to be quite natural from the viewpoint of kernel functions. With a sufficiently small p, the formula (2.1), regarded as a kernel function between x_i and y_j, can be flatter than $1/|x_i - y_j|$ for x_i and y_j in the two nonadjacent boxes. Heuristically, this flatness usually indicates that (2.1) can be well approximated by a degenerate expansion with fewer terms than $1/|x_i - y_j|$, leading to the faster singular value decay of $(\Phi|\Theta)$ than that of $K(X, Y)$.

Based on these observations, and unlike CFMM, we will only use the centers of distributions, rather than both centers and extents, to decide whether an interaction can be compressed by a low-rank approximation. A challenge is how to efficiently find such low-rank approximations.

3. $\mathcal{H}^2$ matrix representation of the ERI matrix.

In this section, we establish notation for representing and constructing the ERI matrix in $\mathcal{H}^2$ format.

3.1. Hierarchical partitioning and ERI matrix blocks.

Constructing an $\mathcal{H}^2$ matrix representation of the ERI matrix starts with a hierarchical partitioning of the set of distributions, or basis function products $\{\phi_a\phi_b\}$, for the molecular system and chosen basis set. Like for FMM, the space enclosing all the distributions is partitioned recursively and adaptively into cubic boxes until the number of distributions centered in each finest box is less than a prescribed small constant. This hierarchical partitioning can be represented by an octree whose nodes correspond to the boxes. We number the nodes level-by-level from the root to the leaves of the octree. Figure 3.1 (top part of figure) shows an example of such a partitioning and numbering for 1-dimensional (1D) space and a perfect binary tree.

Let I denote the set of all distributions. Let I_i denote the set of distributions with centers in box i and corresponding to node i in the tree. Using this notation, $(I_i|I_j)$ denotes the block in the ERI matrix corresponding to the Coulomb interactions between distributions with centers in the ith and jth boxes. The entire ERI matrix can be denoted as $(I|I)$.

In $\mathcal{H}^2$ matrix representations, an *admissibility rule* defines whether or not the interactions between two boxes will be approximated in low-rank form. Specifically for the representation of an ERI matrix, we define the pair of boxes i and j at the same level of the tree as an *admissible* pair if they are separated by at least one other box. The pair is *inadmissible* otherwise. We also say that the block $(I_i|I_j)$ is admissible or inadmissible accordingly.

An $\mathcal{H}^2$ matrix representation of $(I|I)$ consists of two parts: (1) *dense* inadmissible blocks defined at the leaf level and (2) *compressed* admissible blocks $(I_i|I_j)$ at any level satisfying the condition that $(I_i|I_j)$ is not contained in a larger admissible block, i.e., i and j are admissible while their parents are not. Figure 3.1 illustrates several of the above ideas.

3.2. Compression of admissible blocks.

Like for FMM and CFMM, a linear scaling algorithm for $\mathcal{H}^2$ matrix-vector multiplication needs more than just low-rank

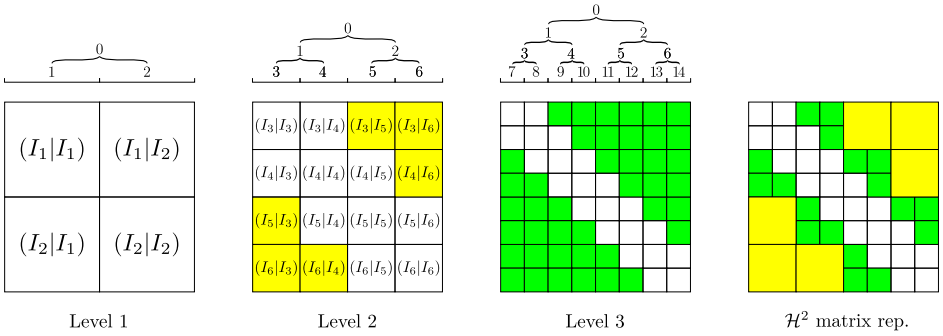


FIG. 3.1. *Illustration of a* 3*-level* $\mathcal{H}^2$ *matrix representation for distributions centered in* 1*D space. Boxes* 7–14 *are the finest level boxes. Inadmissible blocks are white and admissible blocks are colored. The final* $\mathcal{H}^2$ *matrix representation is composed of inadmissible blocks at level* 3 *and admissible blocks at levels* 2 *and* 3*. Some green admissible blocks are not used in the final representation as they are contained in larger yellow admissible blocks.*

representations of admissible blocks. We also need a "uniform basis" property and a "nested basis" property [14].

3.2.1. Uniform compression at each level. For the uniform basis property, the low-rank approximation to an admissible block $(I_i|I_j)$ shares the same column space basis as other admissible blocks that have rows associated with I_i. Similarly, the approximation shares the same row space basis as other admissible blocks that have columns associated with I_j. For example, in Figure 3.1, the low-rank approximations to blocks $(I_7|I_9)$ and $(I_7|I_{10})$ share the same column space basis.

For a node i, recall I_i is the set of distributions centered in box i. Define J_i as the set of distributions centered in all the boxes that are admissible with box i and are at the same level as box i. Then, block $(I_i|J_i)$ is the concatenation of all the admissible blocks that have rows associated with I_i. For example, in Figure 3.1, block $(I_7|J_7)$ is the concatenation of the blocks $(I_7|I_9), \ldots, (I_7|I_{14})$. The low-rank approximations of these latter blocks need to share the same column space basis.

The uniform basis property can be achieved by constructing the low-rank approximations to all the blocks $(I_i|I_j)$ in $(I_i|J_i)$ from the low-rank approximation to $(I_i|J_i)$. Let U_i denote the matrix of shared column space basis vectors associated with I_i. To find U_i, a common approach is to compute a rank-k interpolative decomposition (ID) [6, 13] of $(I_i|J_i)$,

$$(I_i|J_i) \approx U_i(I_i^{\mathrm{id}}|J_i), \tag{3.1}$$

where U_i has k columns, I_i^{id} denotes a subset of I_i, and $(I_i^{\mathrm{id}}|J_i)$ contains k rows of $(I_i|J_i)$. A purely algebraic way to compute the ID approximation is via the strong rank-revealing QR (SRRQR) decomposition [13], given a target rank k or the desired accuracy of the approximation.

For each admissible block $(I_i|I_j)$, the columns of the ID in (3.1) that correspond to $I_j \subset J_i$ give the approximation $(I_i|I_j) \approx U_i(I_i^{\mathrm{id}}|I_j)$. Similarly, the ID approximation of $(I_j|J_j)$ gives $(I_j|I_i^{\mathrm{id}}) \approx U_j(I_j^{\mathrm{id}}|I_i^{\mathrm{id}})$ based on the fact that $I_i^{\mathrm{id}} \subset I_i \subset J_j$. Combining these two approximations leads to

$$(I_i|I_j) \approx U_i(I_i^{\mathrm{id}}|I_j^{\mathrm{id}})U_j^T, \tag{3.2}$$

which satisfies the uniform basis property. In other words, the low-rank approximation

to $(I_i|I_j)$, which is the intersection block of $(I_i|J_i)$ and $(J_j|I_j)$ in the ERI matrix, is constructed based on ID approximations to $(I_i|J_i)$ and $(I_j|J_j)$ as in (3.1).

An interesting observation is that the approximation (3.2) has the form of a density fitting (DF) approximation [8, 39, 43] where I_i^{id} and I_j^{id} would correspond to the set of "auxiliary functions" for I_i and I_j, respectively. DF, however, is applied to the entire ERI matrix $(I|I)$. Thus, an $\mathcal{H}^2$ matrix representation of the ERI matrix can also be interpreted as a generalization of DF. This generalization locally and hierarchically applies DF to certain pairs of subsets of basis function products.

3.2.2. Nested compression between levels. For the nested basis property, each nonleaf node i with children $\{i_1, i_2, \ldots, i_8\}$ has column space basis matrices satisfying

$$U_i = \begin{pmatrix} U_{i_1} & & \\ & \ddots & \\ & & U_{i_8} \end{pmatrix} R_i \tag{3.3}$$

for some matrix R_i to be computed. Thus, the basis matrices at parent nodes are expressed in terms of the basis matrices of their children nodes. The basis matrices for nonleaf nodes are not formed and can be recovered recursively from quantities at lower levels of the tree.

As a consequence of the nested basis property, the ID approximation of $(I_i|J_i)$ at a parent node i can be constructed efficiently by combining the ID approximations computed at its children nodes as follows. First, partition and approximate $(I_i|J_i)$,

$$(I_i|J_i) = \begin{pmatrix} (I_{i_1}|J_i) \\ \vdots \\ (I_{i_8}|J_i) \end{pmatrix} \approx \begin{pmatrix} U_{i_1}(I_{i_1}^{\mathrm{id}}|J_i) \\ \vdots \\ U_{i_8}(I_{i_8}^{\mathrm{id}}|J_i) \end{pmatrix} = \begin{pmatrix} U_{i_1} & & \\ & \ddots & \\ & & U_{i_8} \end{pmatrix} \begin{pmatrix} (I_{i_1}^{\mathrm{id}}|J_i) \\ \vdots \\ (I_{i_8}^{\mathrm{id}}|J_i) \end{pmatrix}. \tag{3.4}$$

Then we calculate an ID approximation of the last matrix above as

$$\begin{pmatrix} (I_{i_1}^{\mathrm{id}}|J_i) \\ \vdots \\ (I_{i_8}^{\mathrm{id}}|J_i) \end{pmatrix} \approx R_i(I_i^{\mathrm{id}}|J_i), \tag{3.5}$$

where $I_i^{\mathrm{id}} \subset \cup_{s=1}^{8} I_{i_s}^{\mathrm{id}} \subset I_i$. Lastly, combining (3.4) and (3.5) gives the ID approximation of $(I_i|J_i)$ as

$$(I_i|J_i) \approx \begin{pmatrix} U_{i_1} & & \\ & \ddots & \\ & & U_{i_8} \end{pmatrix} R_i(I_i^{\mathrm{id}}|J_i) = U_i(I_i^{\mathrm{id}}|J_i),$$

where U_i is exactly the matrix defined as (3.3) using R_i obtained in (3.5).

The approximated matrix in (3.5) has much fewer rows than $(I_i|J_i)$ and, in fact, the number of rows is $O(1)$ if ID approximations of $(I_{i_1}|J_i), \ldots, (I_{i_8}|J_i)$ all have rank $O(1)$. As a result, this nested approach to compressing $(I_i|J_i)$ for a nonleaf node i is much cheaper than directly compressing $(I_i|J_i)$.

In fact, R_i computed by the ID approximation in (3.5) can be computed even more efficiently. Define

$$\hat{I}_i = \cup_{i_s \in \{\text{children of } i\}} I_{i_s}^{\mathrm{id}} \qquad \text{and} \qquad \hat{J}_i = \cup_{l \in \mathcal{F}_i} \cup_{l_s \in \{\text{children of } l\}} I_{l_s}^{\mathrm{id}},$$

where $\mathcal{F}_i$ is the set of boxes that are admissible with box i and are at the same level as box i. The components U_i and I_i^{id} for the ID approximation of $(I_i|J_i)$ satisfying the nested basis property (3.3) can be calculated from the ID approximation

$$(\hat{I}_i|\hat{J}_i) \approx R_i(I_i^{\mathrm{id}}|\hat{J}_i), \tag{3.6}$$

where U_i is defined as (3.3) using R_i and $I_i^{\mathrm{id}} \subset \hat{I}_i \subset I_i$. Readers can refer to [19, 27] for more details.

As an example, consider $(I_3|J_3) = (I_3|I_5 \cup I_6)$ in Figure 3.1. At level 3, $(I_3|J_3)$ is made up by 8 green blocks, i.e., $(I_7 \cup I_8|I_{11} \cup I_{12} \cup I_{13} \cup I_{14})$. With the components U_i and I_i^{id} for nodes at level 3, these green blocks are approximated as

$$\begin{aligned}(I_3|J_3) &= (I_7 \cup I_8|I_{11} \cup I_{12} \cup I_{13} \cup I_{14}) \\ &\approx \begin{pmatrix} U_7 & \\ & U_8 \end{pmatrix} \left(I_7^{\mathrm{id}} \cup I_8^{\mathrm{id}}|I_{11}^{\mathrm{id}} \cup I_{12}^{\mathrm{id}} \cup I_{13}^{\mathrm{id}} \cup I_{14}^{\mathrm{id}}\right) \begin{pmatrix} U_{11}^T & & & \\ & U_{12}^T & & \\ & & U_{13}^T & \\ & & & U_{14}^T \end{pmatrix}.\end{aligned}$$

To compute the ID approximation of $(I_3|J_3)$, the block $(\hat{I}_3|\hat{J}_3)$ to be actually approximated in (3.6) is exactly $\left(I_7^{\mathrm{id}} \cup I_8^{\mathrm{id}}|I_{11}^{\mathrm{id}} \cup I_{12}^{\mathrm{id}} \cup I_{13}^{\mathrm{id}} \cup I_{14}^{\mathrm{id}}\right)$ in the above equation.

3.3. Fast matrix-vector multiplication. The $\mathcal{H}^2$ matrix representation of the ERI matrix is constructed from its leaves to the root via the ID approximations in (3.1) for leaf nodes and in (3.6) for nonleaf nodes. In the representation, all the admissible blocks $(I_i|I_j)$ are compressed as (3.2) with the basis matrices U_i represented as (3.3), satisfying the uniform basis property and the nested basis property. Details of the construction process are to be discussed in subsection 4.5.

Assuming that the rank of the ID approximation in (3.1) or (3.6) for each node is bounded by a constant r, the fast matrix-vector multiplication algorithm [15] for the $\mathcal{H}^2$ matrix representation can be used to construct the Coulomb matrix with $O(rn)$ complexity, where n is the number of distributions.

3.4. Comparison with CFMM. As already mentioned, CFMM is equivalent to the fast matrix-vector multiplication by the ERI matrix in an $\mathcal{H}^2$ format. The main difference between this equivalent $\mathcal{H}^2$ format and the one we have described in this section is that CFMM has a more strict admissibility rule. In CFMM, an ERI matrix block is admissible (also known as far-field) if the corresponding two sets of distributions have nonoverlapping numerical supports. The block is inadmissible (also known as near-field) otherwise. Additionally, CFMM has the same hierarchical partitioning of I but further splits each I_i into "branches" [41] according to the extents of distributions in I_i. Thus, a leaf-level block $(I_i|I_j)$ that corresponds to two sets of distributions with overlapping numerical supports is further subdivided into smaller blocks, some of which can be defined as admissible blocks. Also, CFMM uses the multipole expansion technique to compress the admissible blocks instead of ID approximations.

4. Accelerated compression via proxy points. For an ERI matrix, the construction of an $\mathcal{H}^2$ matrix representation is dominated by the cost of the ID approximation of $(I_i|J_i)$ for leaf nodes i and of $(\hat{I}_i|\hat{J}_i)$ for nonleaf nodes i. These ERI matrix blocks share the same form $(I_*|J_*)$ where, for some node i, I_* is a set of distributions (I_i or $\hat{I}_i$) centered in box i, and J_* is a set of distributions (J_i or $\hat{J}_i$) centered in boxes that are admissible with box i. In general, the set J_* is much larger than the set I_*.

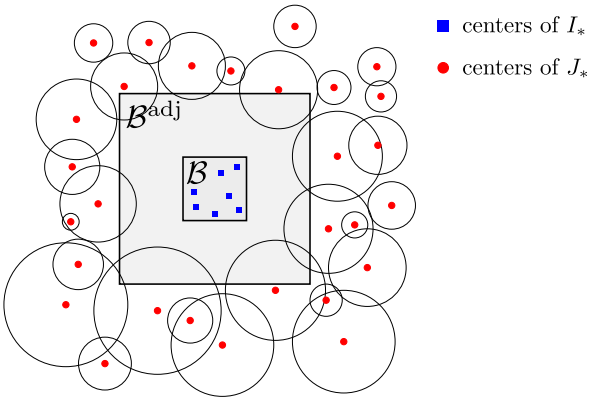


FIG. 4.1. *2D illustration of I_*, J_*, $\mathcal{B}$, and $\mathcal{B}^{adj}$. Each circle around a red point denotes one distribution in J_*. The radius of a circle is the extent of a distribution. Distributions in I_* are not plotted, but the balls associated with these distributions generally can spread outside $\mathcal{B}^{adj}$.*

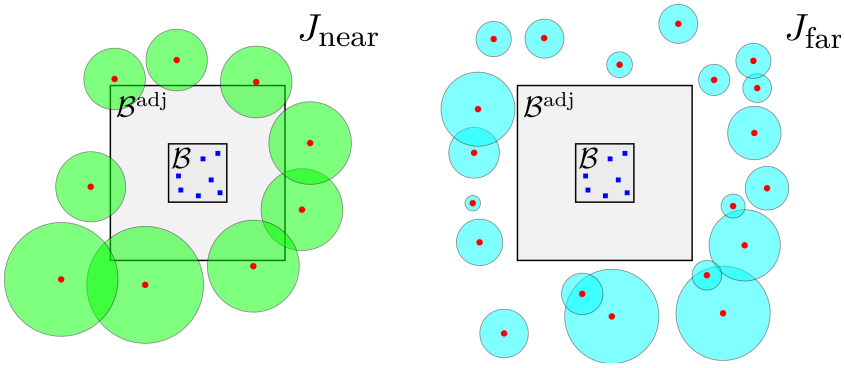


FIG. 4.2. *2D illustration of the splitting of J_* (corresponding to Figure 4.1) into J_{near} and J_{far}, where J_{near} contains all the distributions that overlap with $\mathcal{B}^{adj}$ and $J_{far} = J_* \backslash J_{near}$.*

Using purely algebraic methods such as SRRQR to compress $(I_*|J_*)$ leads to quadratic $\mathcal{H}^2$ construction cost, due to needing at least to form and examine every element in $(I_*|J_*)$. At the same time, the multipole expansion technique used in CFMM cannot generally be applied here, since I_* and J_* can have overlapping numerical supports.

This section introduces a new hybrid analytic-algebraic method to efficiently calculate an ID approximation of $(I_*|J_*)$ while avoiding the evaluation of all the elements in $(I_*|J_*)$.

4.1. Splitting of J_*. Consider two sets of distributions, I_* and J_*, as described above. Let $\mathcal{B}$ denote the box that encloses the centers of distributions in I_*, and let $\mathcal{B}^{\text{adj}}$ denote the union of $\mathcal{B}$ and its 26 adjacent boxes of the same size. By the definition of admissible blocks, all the distributions in J_* have their centers outside $\mathcal{B}^{\text{adj}}$. A 2-dimensional (2D) example of I_*, J_*, $\mathcal{B}$, and $\mathcal{B}^{\text{adj}}$ is illustrated in Figure 4.1.

We split J_* into two subsets, J_{near} and J_{far}, where J_{near} contains all the distributions in J_* that overlap with $\mathcal{B}^{\text{adj}}$ and $J_{\text{far}} = J_* \setminus J_{\text{near}}$. Figure 4.2 illustrates an example of this splitting. We note that this splitting of J_* is not related to the numerical support of I_*, and distributions in both J_{near} and J_{far} may overlap with distributions in I_*. Since all the distributions in J_* have bounded extents, J_{near} gen-

erally has $O(1)$ number of distributions, and J_{far} can be of size on the order of the total number of distributions, i.e., $O(|I|)$. To efficiently compute an ID approximation of $(I_*|J_*)$, we will consider two parts: $(I_*|J_{\text{near}})$ and $(I_*|J_{\text{far}})$. The former has a relatively small size; the latter is the critical part that we discuss now.

For the case of point charges rather than charge distributions, methods already exist for computing low-rank approximations to $(I_*|J_{\text{far}})$ without needing to evaluate the entire matrix itself (note that $J_{\text{far}} = J_*$ for the case of point charges). In the proxy surface and related methods [19, 28, 44, 47], the points in J_{far} are replaced by a smaller set of *proxy points* on the surface $\partial\mathcal{B}^{\text{adj}}$ between the points in I_* and J_{far}. This does not work for charge distributions because the distributions in I_* and J_{far} may overlap. One could redefine J_{far} as the set of distributions that do not overlap with those of I_*, but this would result in a very large set of distributions J_{near} which we are trying to avoid in the first place.

The proxy surface method also may not work when interactions between point charges are defined by general kernel functions. In this case, a remedy that has been proposed [29, 46] is to replace the points in J_{far} by a set of proxy points on multiple layers of surfaces between the points in I_* and J_{far} instead of just one layer. However, the selection of these surfaces and proxy points is completely heuristic, and there is no guarantee for the effectiveness of this remedy.

For the case of charge distributions, our approach to the low-rank approximation of $(I_*|J_{\text{far}})$ resembles the above remedy to the proxy surface method using multiple layers of proxy points but has a solid theoretical foundation. We begin below with a theoretical motivation for this approach.

4.2. Theoretical motivation. If we imagine each distribution φ_i in I_* is a unit charge distribution, then its induced potential $p_i(y)$ in $\mathbb{R}^3\backslash\mathcal{B}^{\text{adj}}$ is

$$p_i(y) = \int_{\mathbb{R}^3} \varphi_i(r)\frac{1}{|r-y|}\mathrm{d}r, \quad y \in \mathbb{R}^3\backslash\mathcal{B}^{\text{adj}}. \tag{4.1}$$

For any $\varphi_i \in I_*$ and $\theta_j \in J_{\text{far}}$, the entry $(\varphi_i|\theta_j)$ of $(I_*|J_{\text{far}})$ can be written as

$$(\varphi_i|\theta_j) = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \varphi_i(r_1)\frac{1}{|r_1-r_2|}\theta_j(r_2)\mathrm{d}r_1\mathrm{d}r_2 = \int_{\mathbb{R}^3\backslash\mathcal{B}^{\text{adj}}} p_i(r_2)\theta_j(r_2)\mathrm{d}r_2, \tag{4.2}$$

where the numerical support of θ_j is entirely within $\mathbb{R}^3\backslash\mathcal{B}^{\text{adj}}$ by the definition of J_{far}.

For analysis purposes, let $U(I_*^{\text{id}}|J_{\text{far}})$ be an ID approximation of $(I_*|J_{\text{far}})$. Each entry of $(I_*|J_{\text{far}})$ is then approximated as

$$(\varphi_i|\theta_j) \approx u_i^T(I_*^{\text{id}}|\theta_j), \quad \varphi_i \in I_*,\ \theta_j \in J_{\text{far}},$$

where u_i^T denotes the ith row of U. Substituting (4.2) into the above equation gives

$$\int_{\mathbb{R}^3\backslash\mathcal{B}^{\text{adj}}} p_i(r_2)\theta_j(r_2)\mathrm{d}r_2 \approx \int_{\mathbb{R}^3\backslash\mathcal{B}^{\text{adj}}} \big(u_i^T P^{\text{id}}(r_2)\big)\theta_j(r_2)\mathrm{d}r_2, \tag{4.3}$$

where $P^{\text{id}}(y)$ denotes the vector of potentials $p_j(y)$ for all $\varphi_j \in I_*^{\text{id}}$. This rewriting shows that the ID approximation actually approximates each potential $p_i(y)$ in the domain $\mathbb{R}^3\backslash\mathcal{B}^{\text{adj}}$ by $u_i^T P^{\text{id}}(y)$ which is a linear combination of the potentials due to the distributions in I_*^{id}. Define the error of each approximation as

$$e_i(y) = p_i(y) - u_i^T P^{\text{id}}(y), \quad y \in \mathbb{R}^3\backslash\mathcal{B}^{\text{adj}},\ \varphi_i \in I_*. \tag{4.4}$$

Using Hölder's inequality, the elementwise error of the ID approximation in (4.3) can be bounded as

$$\left|(\varphi_i|\theta_j) - u_i^T(I_*^{\mathrm{id}}|\theta_j)\right| \leqslant \max_{y\in\mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}} |e_i(y)| \int_{\mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}} |\theta_j(r_2)|\mathrm{d}r_2. \tag{4.5}$$

From this analysis, a good ID approximation $U(I_*^{\mathrm{id}}|J_{\mathrm{far}})$ to $(I_*|J_{\mathrm{far}})$ can be found by seeking U and I_*^{id} such that every $e_i(y)$ defined above is small in the domain $\mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}$. In other words, we seek a subset of the potentials $\{p_j(y)\}_{\varphi_j\in I_*}$ whose linear combination can well approximate each $p_i(y)$ in the domain $\mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}$.

To make the problem tractable, instead of considering the approximation to $p_i(y)$ at every $y \in \mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}$, we consider it at a finite set of proxy points Y_p that lie in $\mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}$. An approximation to $p_i(y)$ can be accurate in $\mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}$ as long as it is accurate at a small set of properly selected points Y_p. Such a choice of Y_p will be discussed in the next subsection.

Let $P(y)$ denote the vector of potentials $p_i(y)$ for all $\varphi_i \in I_*$. Assuming we have a set of proxy points Y_p, then the approximation to $P(y)$ can be computed by using SRRQR to compute the ID approximation,

$$P(Y_p) = \begin{pmatrix} p_1(Y_p)^T \\ p_2(Y_p)^T \\ \vdots \\ p_{|I_*|}(Y_p)^T \end{pmatrix} \approx U \begin{pmatrix} p_{i_1}(Y_p)^T \\ p_{i_2}(Y_p)^T \\ \vdots \\ p_{i_k}(Y_p)^T \end{pmatrix} = UP^{\mathrm{id}}(Y_p). \tag{4.6}$$

The error in the ith row of this approximation is $p_i(Y_p)^T - u_i^T P^{\mathrm{id}}(Y_p)$ and has its norm bounded by the error threshold specified for SRRQR [13]. Thus, the ID approximation (4.6) defines an approximation $u_i^T P^{\mathrm{id}}(y)$ to each $p_i(y)$ with error $e_i(y)$ bounded at Y_p. Based on the previous analysis, the resulting U and I_*^{id} from (4.6) can then be used for the ID approximation of $(I_*|J_{\mathrm{far}})$. The remaining problem becomes how to select an effective but small set of proxy points Y_p.

4.3. Proxy point selection. We define $\mathcal{X}$ to be the smallest cubical domain that encloses the numerical support of I_*. In particular, $\mathcal{X}$ encloses $\mathcal{B}$ and shares the same center, as illustrated in Figure 4.3. Each potential $p_i(y)$ defined in (4.1) can be further written as

$$p_i(y) = \int_{\mathcal{X}} \varphi_i(r)\frac{1}{|r-y|}\mathrm{d}r, \quad y \in \mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}.$$

From this formula, it can be noted that $p_i(y)$ is a harmonic function outside $\mathcal{X}$. As a linear combination of potentials $p_j(y)$ for all $\varphi_j(y) \in I_*$, $e_i(y)$ defined in (4.4) with any u_i^T and I_*^{id} is harmonic outside $\mathcal{X}$. By the maximum principle of harmonic functions, $e_i(y)$ satisfies $\max_{y\in\mathbb{R}^3\backslash\mathcal{X}} |e_i(y)| = \max_{y\in\partial\mathcal{X}} |e_i(y)|$, and thus

$$\max_{y\in\mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}} |e_i(y)| = \begin{cases} \max_{y\in\mathcal{X}\backslash\mathcal{B}^{\mathrm{adj}}} |e_i(y)| & \text{if } \mathcal{X} \supset \mathcal{B}^{\mathrm{adj}} \\ \max_{y\in\partial\mathcal{B}^{\mathrm{adj}}} |e_i(y)| & \text{if } \mathcal{X} \subset \mathcal{B}^{\mathrm{adj}} \end{cases}. \tag{4.7}$$

As a result, it is sufficient to make $e_i(y)$ small in $\mathcal{X}\backslash\mathcal{B}^{\mathrm{adj}}$ (or on $\partial\mathcal{B}^{\mathrm{adj}}$) in order to make $e_i(y)$ small in $\mathbb{R}^3\backslash\mathcal{B}^{\mathrm{adj}}$. This indicates that we only need to select the proxy points Y_p in $\mathcal{X}\backslash\mathcal{B}^{\mathrm{adj}}$ (or on $\partial\mathcal{B}^{\mathrm{adj}}$) for the ID approximation (4.6).

For the case of point charges, $\mathcal{X}$ is within $\mathcal{B}^{\mathrm{adj}}$, and the proxy points are selected on $\partial\mathcal{B}^{\mathrm{adj}}$. The calculation of U and I_*^{id} for the ID approximation of $(I_*|J_{\mathrm{far}})$ via (4.6)

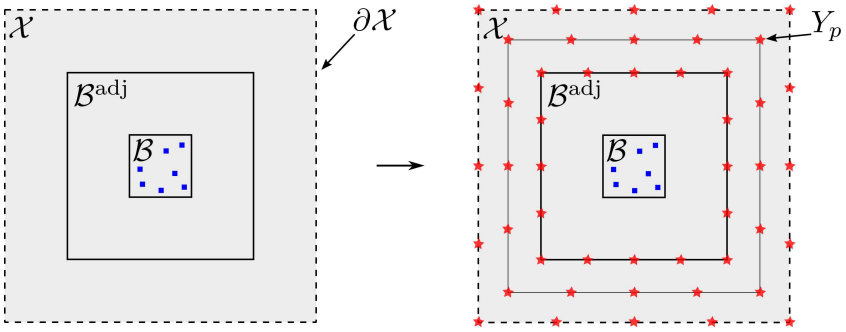


FIG. 4.3. *2D illustration of the selection of proxy points Y_p in $\mathcal{X}\backslash\mathcal{B}^{adj}$.*

is exactly the proxy surface method [19, 28]. In this case, [45] shows that the number of proxy points needed only depends on the ratio of the radius of $\mathcal{B}$ to that of $\mathcal{B}^{\text{adj}}$ and is not related to the absolute size of $\partial\mathcal{B}^{\text{adj}}$.

For GTF distributions in I_* (which have exponentially decaying tails), we continue to expect that only a constant number of proxy points is needed on $\partial\mathcal{X}$ (or on $\partial\mathcal{B}^{\text{adj}}$ when $\mathcal{X} \subset \mathcal{B}^{\text{adj}}$). With this idea, the proxy points are chosen heuristically as follows. If $\mathcal{X} \subset \mathcal{B}^{\text{adj}}$, we select a fixed number of points uniformly distributed on $\partial\mathcal{B}^{\text{adj}}$. Otherwise, we select multiple layers of evenly spaced cubic surfaces between and including $\partial\mathcal{B}^{\text{adj}}$ and $\partial\mathcal{X}$, with a fixed number of proxy points distributed uniformly on each cubic surface. The number of surfaces is proportional to the ratio of the distance between $\partial\mathcal{X}$ and $\partial\mathcal{B}^{\text{adj}}$ to the edge length of $\mathcal{B}$. Figure 4.3 gives a 2D example of the selected proxy points. Such a selection gives $O(1)$ number of proxy points, and thus the approximated matrix $P(Y_p)$ in (4.6) is also of $O(1)$ size.

To be consistent with ERI notation $(\cdot|\cdot)$, denote $P(Y_p)$ from (4.6) as $(I_*|Y_p)$ where $y_j \in Y_p$ stands for a point charge at y_j, and thus

$$(\varphi_i|y_j) = (\varphi_i|\delta_{y_j}) = \int_{\mathbb{R}^3} \varphi_i(r)\frac{1}{|r-y_j|}\mathrm{d}r = p_i(y_j), \quad \varphi_i \in I_*,\ y_j \in Y_p.$$

It is important to note that each entry of $(I_*|Y_p)$ above is not an ERI—it is a nuclear attraction integral and is much cheaper to evaluate than an ERI [18].

4.4. Algorithm for computing the ID of $(I_*|J_*)$. From the above discussion, to construct the ID approximation

$$(I_*|J_*) \approx U(I_*^{\text{id}}|J_*), \tag{4.8}$$

it is sufficient to compute the components U and I_*^{id} such that

$$(I_*|J_{\text{near}}) \approx U(I_*^{\text{id}}|J_{\text{near}}) \qquad \text{and} \qquad (I_*|Y_p) \approx U(I_*^{\text{id}}|Y_p).$$

Using the idea of the randomized ID approximation method [16], the components U and I_*^{id} are computed as follows. First, generate two random matrices Ω_1 and Ω_2 of dimension $|J_{\text{near}}| \times |I_*|$ and $|Y_p| \times |I_*|$, respectively, whose entries follow the standard normal distribution. Multiply $(I_*|J_{\text{near}})$ with Ω_1 and $(I_*|Y_p)$ with Ω_2:

$$A_1 = (I_*|J_{\text{near}})\Omega_1 \quad \text{and} \quad A_1 = (I_*|Y_p)\Omega_2.$$

Algorithm 4.1 Efficient ID approximation of $(I_*|J_*)$

Input: I_*, J_*, $\mathcal{B}^{\text{adj}}$, $\mathcal{X}$.
Output: U and I_*^{id} for an ID approximation $U(I_*^{\text{id}}|J_*)$ to $(I_*|J_*)$.

- Split J_* into J_{near} and J_{far}.
- Select proxy points Y_p in $\mathcal{X}\backslash\mathcal{B}^{\text{adj}}$ (or on $\partial\mathcal{B}^{\text{adj}}$ when $\mathcal{X} \subset \mathcal{B}^{\text{adj}}$).
- Generate random matrices $\Omega_1 \in \mathbb{R}^{|J_{\text{near}}|\times|I_*|}$ and $\Omega_1 \in \mathbb{R}^{|Y_p|\times|I_*|}$.
- Calculate $A_1 = (I_*|J_{\text{near}})\Omega_1$ and $A_2 = (I_*|Y_p)\Omega_2$.
- Normalize the columns of A_1 and A_2 to obtain $\tilde{A}_1$ and $\tilde{A}_2$.
- Compute U and I_*^{id} from an ID approximation of $[\tilde{A}_1,\ \tilde{A}_2]$ using SRRQR.

Then, normalize each column of A_1 and A_2 to have unit norm, and denote the normalized matrices as $\tilde{A}_1$ and $\tilde{A}_2$. Lastly, compute U and I_*^{id} from the ID approximation,

$$[\tilde{A}_1,\ \tilde{A}_2] \approx U[\tilde{A}_1,\ \tilde{A}_2]_{I_*^{\text{id}},:}, \tag{4.9}$$

using SRRQR, where $[\tilde{A}_1,\ \tilde{A}_2]_{I_*^{\text{id}},:}$ denotes the subset of rows in $[\tilde{A}_1,\ \tilde{A}_2]$ computed by this ID and $I_*^{\text{id}} \subset I_*$ is associated with the indices of this subset.

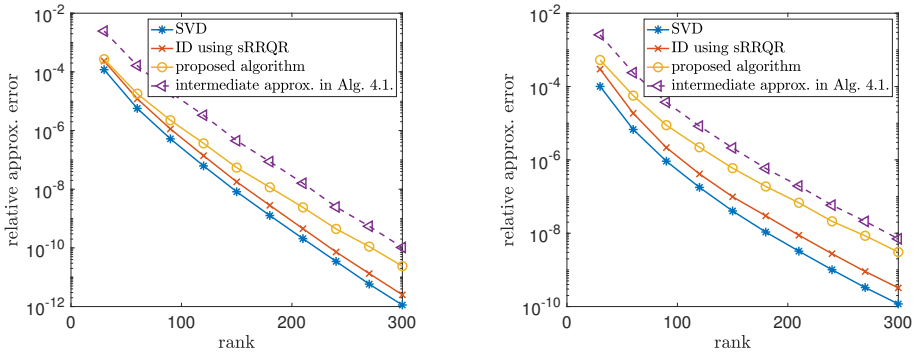
The reason for the normalization step is that $(I_*|J_{\text{near}})$ and $(I_*|Y_p)$ can have different number of columns and also their entries can be of different magnitudes. As a result, A_1 and A_2 can have their entries of different magnitudes. If directly computing an ID approximation of $[A_1,\ A_2]$, the obtained U and I_*^{id} could be biased and define a better ID approximation to the one of A_1 and A_2 that has larger entries.

This accelerated ID approximation of $(I_*|J_*)$ is summarized in Algorithm 4.1. Noting that $(I_*|J_{\text{near}})$ and $(I_*|Y_p)$ only have $O(1)$ number of columns, Algorithm 4.1 can be much faster than the purely algebraic ID approximation using SRRQR alone. More importantly, applying this compression method in $\mathcal{H}^2$ matrix construction can reduce the construction cost to nearly linear in the number of distributions.

We numerically demonstrate Algorithm 4.1 as follows. Consider a cube $\mathcal{B} = [-\frac{1}{2}L, \frac{1}{2}L]^3$ of edge length $L = 5$ and $\mathcal{B}^{\text{adj}} = [-\frac{3}{2}L, \frac{3}{2}L]^3$. Select 600 and 20000 GTF distributions of the form $\{(p/\pi)^{3/2}e^{-p|r-r_a|^2}\}$ with the same exponent p and different centers r_a randomly distributed in $\mathcal{B}$ and $[-\frac{11}{2}L, \frac{11}{2}L]^3\backslash\mathcal{B}^{\text{adj}}$, respectively. Denote the two sets of GTFs as I_* and J_*. To define $\mathcal{X}$, let the extent precision be $\tau = 10^{-10}$. Two exponents $p = 1$ and $p = 0.1$ are tested. Figure 4.4 shows the relative error of the low-rank approximation of $(I_*|J_*)$ calculated by Algorithm 4.1 for different choices of the rank. For both values of p, Algorithm 4.1 gives relative errors close to those of SVD and ID using SRRQR. Meanwhile, the intermediate approximation of $[\tilde{A}_1,\ \tilde{A}_2]$ in Algorithm 4.1 has slightly larger relative errors than the obtained ID approximation of $(I_*|J_*)$. The accuracy of the final approximation can be controlled by controlling the accuracy of the ID approximation (4.9) computed by SRRQR.

4.5. Summary of the $\mathcal{H}^2$ method. We refer to the proposed Coulomb matrix construction method (Algorithm 4.2) as the $\mathcal{H}^2$ *method*. The method consists of two phases: (1) use the new compression technique to construct an $\mathcal{H}^2$ matrix representation of the ERI matrix $(I|I)$, and then (2) use the fast $\mathcal{H}^2$ matrix-vector multiplication algorithm to construct the Coulomb matrix.

Assuming that the ranks of the ID approximations at lines 4 and 10 of Algorithm 4.2 are bounded by a constant r (to be experimentally justified in section 5), the first phase has $O(|I|r^2)$ computation cost, and the second phase has $O(|I|r)$ computation cost. As mentioned in the Introduction, in self-consistent field iterations, a



(a) $p = 1$ (b) $p = 0.1$

FIG. 4.4. *Relative error of the low-rank approximations of* $(I_*|J_*)$ *in the Frobenius norm. Three methods are used: SVD, ID using SRRQR, and Algorithm* 4.1. *In addition, the dashed lines show the relative error of the intermediate approximation* (4.9) *for* $[\tilde{A}_1,\ \tilde{A}_2]$. *The test problem parameters are* (a) $p = 1$, $\lambda = 4.6$, $|J_{near}| = 1354$, $|J_{far}| = 18646$, *and* 1 *layer of proxy points in* Y_p *with* 384 *points;* (b) $p = 0.1$, $\lambda = 13.4$, $|J_{near}| = 8409$, $|J_{far}| = 11591$, *and* 3 *layers of proxy points in* Y_p *with* 1152 *points.*

Coulomb matrix is constructed with different density matrices in each iteration while the ERI matrix is fixed. The relatively expensive cost for constructing the $\mathcal{H}^2$ matrix representation can be amortized over many matrix-vector multiplications.

The constructed $\mathcal{H}^2$ matrix representation has $O(|I|r)$ storage cost. The representation stores the following "necessary" components for each node i with a nonempty J_i: (1) I_i^{id}, (2) U_i if i is a leaf node, and (3) R_i if i is a nonleaf node. Further, the representation can either store the following components if line 13 of Algorithm 4.2 is applied, or compute them when they are needed in the second phase of Algorithm 4.2: (4) *inadmissible blocks* $(I_i|I_j)$ for each inadmissible pair of nodes i and j at the leaf level and (5) *skeleton blocks* $(I_i^{\mathrm{id}}|I_j^{\mathrm{id}})$ for each admissible pair of nodes i and j at the same level whose parent nodes are inadmissible. These latter blocks are associated with the low-rank approximations (3.2) to the admissible blocks used in the final $\mathcal{H}^2$ matrix representation as exemplified in Figure 3.1.

As will be shown in the numerical tests, the storage cost for the inadmissible blocks and the skeleton blocks is much larger than the storage required for the other components in the $\mathcal{H}^2$ matrix representation. If these blocks are to be stored, they are best stored in dense matrix format, since they do not have enough sparsity to warrant storage in a sparse matrix format. In addition, storage of these blocks should be nonredundant, utilizing the 8-way symmetry present in the ERI tensor.

5. Numerical experiments. We test the $\mathcal{H}^2$ method and compare it to CFMM using two sets of molecular systems. The first is a set of linear alkanes of different sizes. The second is a set of truncated protein-ligand systems derived from the 1hsg system in the protein data bank. In this second set, each system consists of a ligand with its protein environment within a certain radius. Different radii give different sized systems. Such truncated systems are used in order to make protein-ligand simulations tractable. See [7] for more information on these systems.

These two sets of systems span an important determinant of CFMM and $\mathcal{H}^2$ method performance. The alkane systems are long and narrow while the 1hsg protein-

Algorithm 4.2 Construct the Coulomb matrix by the $\mathcal{H}^2$ method

Input: distribution set I, density matrix D.
Output: $J = (I|I)D$.

Phase 1: Construct an $\mathcal{H}^2$ matrix representation of $(I|I)$

1: • Hierarchically partition I into subsets $\{I_i\}$ with L levels.
2: **for** node i at level L (the leaf level) **do**
3: • Compute U_i and I_i^{id} from the ID approximation of $(I_i|J_i)$ in (3.1) using
4: Algorithm 4.1.
5: **end for**
6: **for** $k = L-1, L-2, \ldots, 3$ **do**
7: **for** node i at level k **do**
8: • Construct $\hat{I}_i$ and $\hat{J}_i$ according to subsection 3.2.2.
9: • Compute R_i and I_i^{id} from the ID approximation of $(\hat{I}_i|\hat{J}_i)$ in (3.6) using
10: Algorithm 4.1.
11: **end for**
12: **end for**
13: • (optional, see line 15) Construct inadmissible blocks $(I_i|I_j)$ for each inadmissible pair of nodes i and j at level L, and skeleton blocks $(I_i^{\mathrm{id}}|I_j^{\mathrm{id}})$ for each admissible pair of nodes i and j at the same level whose parent nodes are inadmissible.

Phase 2: Construct the Coulomb matrix

14: • Unfold the density matrix D as a vector.
15: • Apply the $\mathcal{H}^2$ matrix-vector multiplication algorithm to construct $J = (I|I)D$. If line 13 is not applied, the inadmissible blocks and skeleton blocks are constructed when needed in the matrix-vector multiplication.
16: • Fold the vector J as the computed Coulomb matrix.

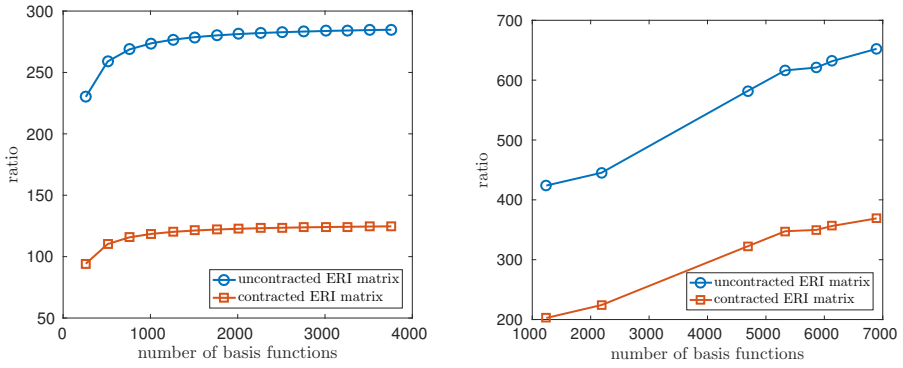
ligand systems are globular. One may say that they have 1D and 3-dimensional (3D) "shapes," respectively. We thus expect a larger proportion of interactions that can be compressed in CFMM and the $\mathcal{H}^2$ method for the alkanes than for the 1hsg systems.

Prescreening. In practice, many rows and columns of the ERI matrix are numerically zero. Specifically, the row and column associated with a product $\phi_a\phi_b$ can be neglected if $|(\phi_a\phi_b|\phi_c\phi_d)| \leqslant \delta$ for any $\phi_c\phi_d$. A threshold of $\delta = 10^{-10}$ is used in our tests. Such numerically zero rows and columns can be identified efficiently as follows. From the Schwarz inequality $|(\phi_a\phi_b|\phi_c\phi_d)| \leqslant \sqrt{(\phi_a\phi_b|\phi_a\phi_b)(\phi_c\phi_d|\phi_c\phi_d)}$, a product $\phi_a\phi_b$ and its corresponding row and column can be neglected if

$$\sqrt{(\phi_a\phi_b|\phi_a\phi_b)} \leqslant \frac{\delta}{\max_{c,d}\sqrt{(\phi_c\phi_d|\phi_c\phi_d)}}, \tag{5.1}$$

which only requires evaluating $(\phi_a\phi_b|\phi_a\phi_b)$ for each pair of basis functions. This process is called *prescreening* of basis function products [17]. Prescreening effectively reduces the dimension of the ERI matrix. We refer to this reduced dimension as the "effective dimension."

Basis set and contracted basis functions. The cc-pVDZ basis set is used for both sets of molecular systems. Like almost all Gaussian basis sets, the basis functions in this basis set are *contracted* GTFs (known linear combinations of GTFs),



(a) 1D alkane molecules (b) 3D 1hsg molecules

FIG. 5.1. *Ratio of the effective ERI matrix dimension to the number of contracted basis functions for two types of molecules of different sizes. This ratio is plotted against the size of the molecular systems in terms of the number of contracted basis functions. Results for both uncontracted and contracted ERI matrices are shown.*

as mentioned in the Introduction. The product of two contracted GTF basis functions can be written as (neglecting contraction coefficients)

$$\phi_a \phi_b = \sum_{\chi_e \in [\phi_a]} \sum_{\chi_f \in [\phi_b]} \chi_e \chi_f,$$

where $[\phi_a]$ denotes the set of "primitive" GTFs that make up ϕ_a. Each ERI matrix entry $(\phi_a \phi_b | \phi_c \phi_d)$ can thus be written as the sum of ERIs with primitive GTFs as

$$(\phi_a \phi_b | \phi_c \phi_d) = \sum_{\chi_e \in [\phi_a]} \sum_{\chi_f \in [\phi_b]} \sum_{\chi_g \in [\phi_c]} \sum_{\chi_h \in [\phi_d]} (\chi_e \chi_f | \chi_g \chi_h). \tag{5.2}$$

CFMM and the $\mathcal{H}^2$ method can be applied to either the original contracted ERI matrix $(\phi_a \phi_b | \phi_c \phi_d)$ or the uncontracted ERI matrix $(\chi_e \chi_f | \chi_g \chi_h)$. Compared to the contracted ERI matrix, the uncontracted ERI matrix has larger dimensions, i.e., more products in $\{\chi_e \chi_f\}$. However, there are also more products in $\{\chi_e \chi_f\}$ that can be prescreened. A more important advantage of using the uncontracted ERI matrix is that each $\chi_e \chi_f$ is a primitive GTF, and thus its numerical support can be more precisely described by a ball than contracted GTFs, which improves the identification of well-separated interactions in CFMM and the identification of J_{near}, J_{far}, and $\mathcal{X}$ for Algorithm 4.1 in $\mathcal{H}^2$ matrix construction.

Figure 5.1 plots the ratio of the effective ERI matrix dimension to the number of basis functions for molecular systems of different sizes. The x-axis in this and other figures is the size of the molecular system in terms of the number of contracted basis functions $\{\phi_a\}$ (roughly 10 basis functions per atom). The figure shows that for our choice of $\delta = 10^{-10}$, the uncontracted ERI matrix is only about twice the dimension of the corresponding contracted ERI matrix. For increasing molecular system size, the effective ERI matrix dimension is expected to be asymptotically linear in the number of basis functions [18, 37]. This can be observed for the tested alkane molecules and is expected to be observed for larger 1hsg molecules.

In the following numerical tests, we apply CFMM and the $\mathcal{H}^2$ method to uncontracted and prescreened ERI matrices, i.e., the set of distributions I in Algorithm 4.2

contains the primitive basis function products obtained by prescreening and uncontraction. In practice, especially for basis sets with highly contracted basis functions, it may be advantageous to work with contracted rather than uncontracted ERI matrices, which we intend to investigate in future work.

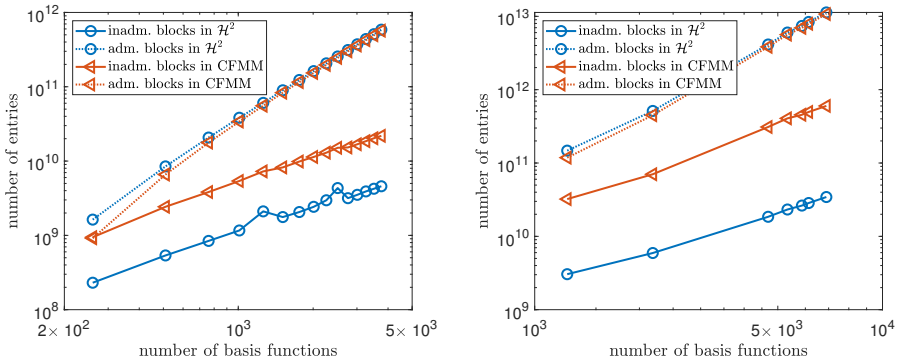
Method settings. In both the $\mathcal{H}^2$ method and CFMM, the extent precision is set to $\tau = 10^{-10}$. The hierarchical partitioning of the set of distributions is stopped when each finest box has less than 300 distributions or has edge length less than 1 Bohr.

For the selection of proxy points Y_p described in subsection 4.3, when $\mathcal{X}$ is within $\mathcal{B}^{\text{adj}}$, only one cubical surface $\partial\mathcal{B}^{\text{adj}}$ is selected. Otherwise, we select cubical surfaces evenly spaced between and including $\partial\mathcal{B}^{\text{adj}}$ and $\partial\mathcal{X}$. The total number of these cubical surfaces is 3, 5, 7,... when the ratio of the distance between $\partial\mathcal{X}$ and $\partial\mathcal{B}^{\text{adj}}$ to the edge length of $\mathcal{B}$ (when rounded up) equals 1, 2, 3,..., respectively. Figure 4.3 gives an example of three selected cubical surfaces when the ratio equals 1. The number of proxy points selected on each cubical surface is 384, i.e., 8×8 uniform grid points on each face of the cubical surface.

5.1. Total number of direct interactions and rank of the low-rank approximations. In CFMM, the computation of the interactions that cannot be accelerated by multipole expansions dominates the total computation time. Similarly, in the $\mathcal{H}^2$ method, the computation of the interactions associated with inadmissible blocks dominates the computation time. In both cases, these interactions are evaluated directly. In this section, we compare the two methods in terms of the total number of these interactions. For convenience, we also refer to the interactions between two sets of distributions that are directly evaluated in CFMM as entries of an inadmissible block, and the interactions between two sets of distributions accelerated using multipole expansions in CFMM as entries of an admissible block. We follow [41] in defining admissible and inadmissible blocks in CFMM.

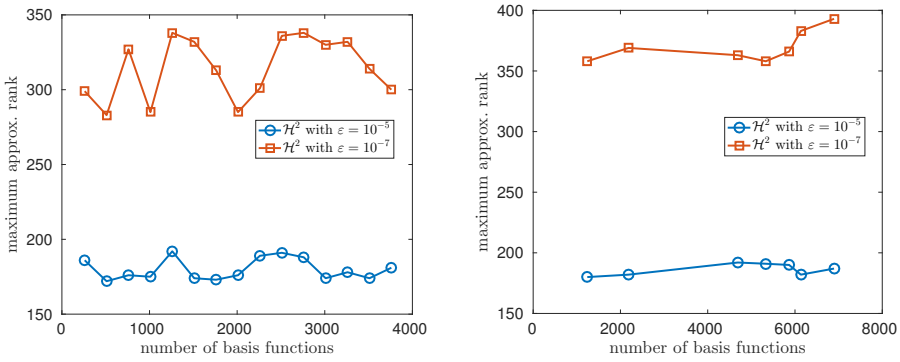
Figure 5.2 plots the total number of entries in the inadmissible and admissible blocks in the two methods. The main experimental result of this paper is that CFMM has approximately 5 times more inadmissible block entries (direct interactions) than the $\mathcal{H}^2$ method for the alkane molecules, and approximately 18 times more for the 1hsg molecules. Thus, the evaluation, multiplication, and storage of inadmissible blocks in CFMM are expected to be 5 and 18 times more expensive than in the $\mathcal{H}^2$ method for the two types of molecules, respectively. The result shows that the $\mathcal{H}^2$ method has even more advantage over CFMM on globular molecules like 1hsg. The number of admissible block entries is large, but these interactions are computed very efficiently (they are not computed explicitly in either method).

The maximum ranks of the low-rank approximations of all the admissible blocks in each constructed $\mathcal{H}^2$ matrix representation are shown in Figure 5.3. Here, the results are shown for two values of the relative error threshold, $\varepsilon = 10^{-5}$ and $\varepsilon = 10^{-7}$, which is required for SRRQR in Algorithm 4.1. This threshold affects the approximation rank and storage required for the admissible blocks in the $\mathcal{H}^2$ matrix representation. The figure shows that the maximum rank is bounded for problems of different sizes. This justifies the observation in section 2 that we can simply use the centers of distributions to decide whether an interaction can be compressed by a low-rank approximation. With bounded maximum rank, the $\mathcal{H}^2$ method (both phases in Algorithm 4.2) has computation cost and storage cost that are linear in the effective dimension of the ERI matrix, as explained in subsection 4.5. The numerical results below in subsections 5.2 and 5.3 also confirm this linear scaling property.



(a) 1D alkane molecules (b) 3D 1hsg molecules

FIG. 5.2. *Total number of entries in the admissible and inadmissible blocks defined in CFMM and in the $\mathcal{H}^2$ method for two types of molecules of different sizes. Redundant interactions due to 8-way symmetry in the ERI tensor are not counted.*

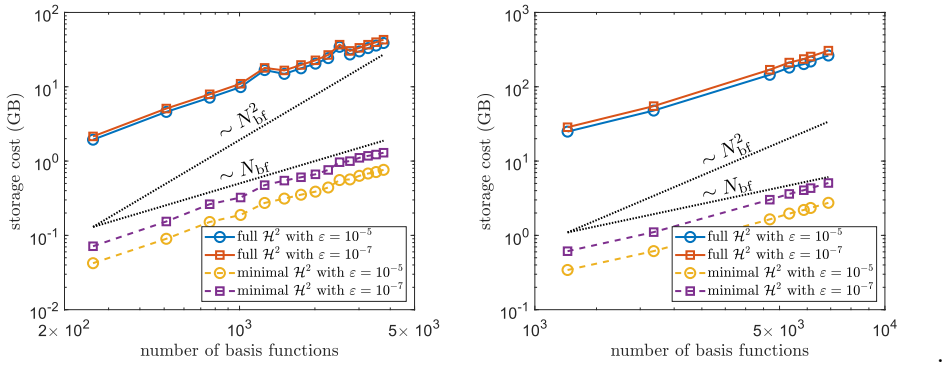


(a) 1D alkane molecules (b) 3D 1hsg molecules

FIG. 5.3. *Maximum rank of the low-rank approximations of all the admissible blocks in the constructed $\mathcal{H}^2$ matrix representation for two types of molecules of different sizes.*

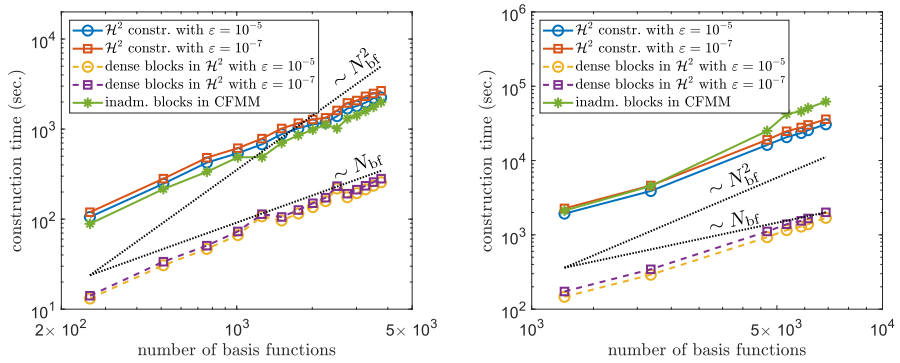
5.2. $\mathcal{H}^2$ matrix construction. The first phase of Algorithm 4.2 is the construction of the $\mathcal{H}^2$ matrix representation of an ERI matrix. In this subsection, the aim is to demonstrate this construction and show how the $\mathcal{H}^2$ matrix storage and construction execution time vary with increasing problem size. Again, we use two values of the relative error threshold ε for the ID approximations.

The storage cost for the $\mathcal{H}^2$ matrix representations is shown in Figure 5.4. Results are shown for both the case when line 13 in Algorithm 4.2 is applied and the inadmissible blocks and skeleton blocks are stored (full $\mathcal{H}^2$), and the case when line 13 is not applied and these blocks are not stored (minimal $\mathcal{H}^2$). The high cost of storing the inadmissible and skeleton blocks is evident (although storage for the skeleton blocks is much less than the storage for the inadmissible blocks). The results show that the storage cost is almost linear in the number of basis functions for alkane molecules in either storage mode. The slightly superlinear cost for the 1hsg molecules is due to



(a) 1D alkane molecules (b) 3D 1hsg molecules

FIG. 5.4. *Storage cost for $\mathcal{H}^2$ matrix representations of ERI matrices for two types of molecules of different sizes. "Full $\mathcal{H}^2$" refers to storing both the necessary components and the inadmissible and skeleton blocks according to subsection* 4.5. *"Minimal $\mathcal{H}^2$" refers to storing only the necessary components. Reference lines for linear and quadratic scaling with the number of basis functions N_{bf} are also shown.*



(a) 1D alkane molecules (b) 3D 1hsg molecules

FIG. 5.5. *Timings for constructing $\mathcal{H}^2$ matrix representations of ERI matrices for two types of molecules of different sizes. "$\mathcal{H}^2$ constr." refers to the timings for constructing the $\mathcal{H}^2$ matrix without evaluating the inadmissible and skeleton blocks, i.e., the first phase of Algorithm* 4.2 *without line* 13. *"Dense blocks in $\mathcal{H}^2$" refers to the timings for evaluating the inadmissible and skeleton blocks, i.e., line* 13 *of Algorithm* 4.2. *"Inadm. blocks in CFMM" refers to the timings for evaluating the inadmissible blocks in CFMM.*

the slightly superlinear growth of the effective dimension of the ERI matrix with the number of basis functions, as shown earlier in Figure 5.1.

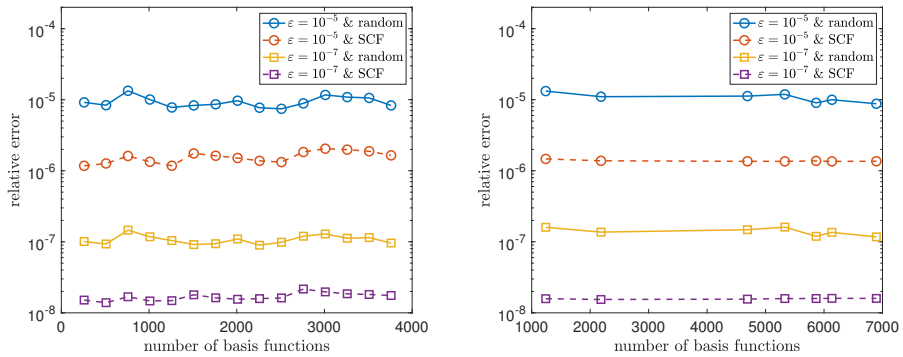
The timings for constructing the $\mathcal{H}^2$ matrix representations are shown in Figure 5.5. These timings should only be regarded as an indication of relative trends, as our codes are implemented in MATLAB. (ERIs and nuclear attraction integrals were computed analytically using recurrence relations implemented in the Simint package [32] using the C programming language.) For alkane molecules, the construction time is linear in the number of basis functions. For 1hsg molecules, the construction time is slightly superlinear, again because of the slightly superlinear growth of the effective dimension of the ERI matrix with the number of basis functions.

The timings for evaluating the inadmissible blocks in CFMM are also shown. As expected, these timings are much larger than for the $\mathcal{H}^2$ method, since there are far more entries in these blocks for CFMM as shown earlier in Figure 5.2. Meanwhile, $\mathcal{H}^2$ matrix construction has similar execution time as evaluating the inadmissible blocks in CFMM. Since the cost for constructing the $\mathcal{H}^2$ matrix representations can be amortized by many matrix-vector multiplications (whose cost is to be shown next), the $\mathcal{H}^2$ method has better overall performance compared to CFMM.

5.3. Coulomb matrix construction. In the second phase of Algorithm 4.2, the Coulomb matrix for a given density matrix is constructed based on the $\mathcal{H}^2$ matrix representation of the ERI matrix constructed in the first phase. This second phase simply involves the fast $\mathcal{H}^2$ matrix-vector multiplication algorithm. The aim of this subsection is to demonstrate how the execution time of this $\mathcal{H}^2$ matrix-vector multiplication algorithm in different settings (storing the inadmissible and skeleton blocks or computing them dynamically) varies with increasing problem size. In comparison to CFMM, the improvement in execution time is directly related to the number of entries in the inadmissible blocks, as shown earlier in Figure 5.2. In this subsection, we also show the accuracy of the computed Coulomb matrix and demonstrate that this accuracy can be controlled by the SRRQR threshold, ε. For each molecule, we test Coulomb matrix construction with two types of density matrices: (a) randomly generated symmetric matrices whose entries follow the standard normal distribution and (b) a density matrix obtained after 10 self-consistent field (SCF) iterations of the Hartree–Fock method.

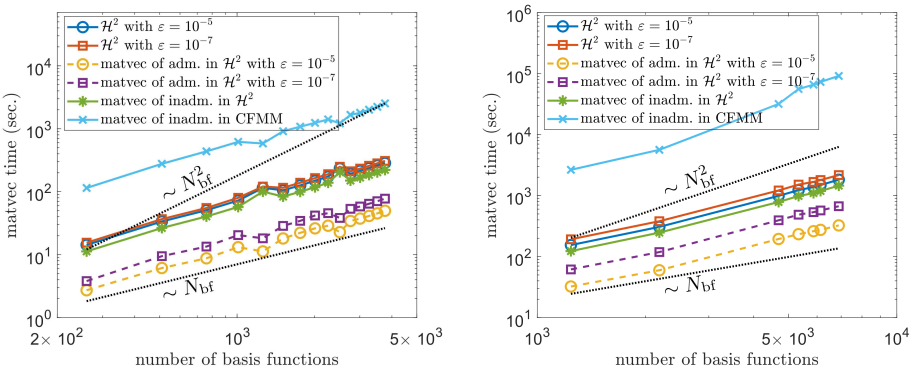
Figure 5.6 plots the relative errors in the constructed Coulomb matrices, where the "exact" Coulomb matrices are calculated directly. As before, we test two values of the relative error threshold ε used for the ID approximations. The results show that the relative error in the Coulomb matrices is consistent across the different types of molecules and molecule sizes. More specifically, the relative error is close to the value of the threshold ε for random density matrices and is one order of magnitude smaller than the value of the threshold ε for density matrices generated by SCF iterations.

Figure 5.7 plots the timings for the $\mathcal{H}^2$ matrix-vector multiplication used to construct the Coulomb matrix for the two types of molecules of different sizes. Here,



(a) 1D alkane molecules (b) 3D 1hsg molecules

FIG. 5.6. *Relative error (in the Frobenius norm) of the Coulomb matrix constructed by the $\mathcal{H}^2$ method for two types of molecules of different sizes. For random density matrices, the results are the average of 5 independent tests.*



(a) 1D alkane molecules (b) 3D 1hsg molecules

FIG. 5.7. *Timings for constructing Coulomb matrices by the $\mathcal{H}^2$ method where inadmissible and skeleton blocks are dynamically calculated when needed (the second phase of Algorithm* 4.2 *with line* 13 *not applied). The timing is also broken down into the portion for multiplying by admissible blocks and by inadmissible blocks. For comparison, the timings for the multiplications with inadmissible blocks in CFMM are also shown. The timings are the average of* 5 *independent tests.*

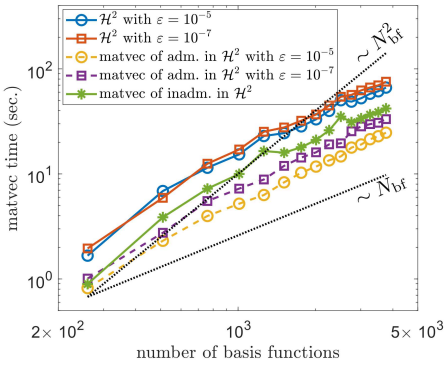


FIG. 5.8. *Timings for constructing Coulomb matrices for alkane molecules by the $\mathcal{H}^2$ method where inadmissible and skeleton blocks have been precomputed and stored (the second phase of Algorithm* 4.2 *with line* 13 *applied). The timings are also broken down into the portion for multiplying by admissible blocks and by inadmissible blocks. The timings are the average of* 5 *independent tests.*

the ERIs in the inadmissible and skeleton blocks are dynamically calculated when needed during the matrix-vector multiplication. Just like for $\mathcal{H}^2$ matrix construction, the matrix-vector multiplication for a matrix in $\mathcal{H}^2$ format is almost linear in the effective dimension of the ERI matrix (which is slightly superlinear in the number of basis function in the case of 1hsg).

The figure also shows the timings broken down into the portion for multiplying by admissible blocks and by inadmissible blocks. It is evident that forming and multiplying by the inadmissible blocks, i.e., computing the direct interactions, is the bottleneck, even after the reduction in the total size of these blocks due to the $\mathcal{H}^2$ method compared to CFMM.

Figure 5.8 again plots the timings for Coulomb matrix construction for molecules of different sizes, but this time we assume that the inadmissible and skeleton blocks

have been precomputed and stored. Due to memory limitations, only the alkane molecules are tested. In this case, the multiplication of admissible blocks and that of inadmissible blocks require a similar amount of time. With the $\mathcal{H}^2$ method, multiplying by the inadmissible blocks when these blocks have been precomputed is no longer a clear bottleneck.

6. Conclusion. In this paper, a new technique is proposed to efficiently compress the interactions between continuous charge distributions. Using this technique, an $\mathcal{H}^2$ matrix representation of the ERI matrix is constructed, which is then used to construct the Coulomb matrix. The new technique can also be viewed as extending the capability of $\mathcal{H}^2$ matrices to represent the interactions between continuous charge distributions, at least for charge distributions from Gaussian basis sets.

Our approach to constructing the Coulomb matrix has cost that appears to be nearly linear in the effective ERI matrix dimension. The effective ERI matrix dimension has been argued to be asymptotically linear (rather than quadratic) with the number of basis functions [18].

More importantly, compared to CFMM, far fewer interactions need to be directly computed. The promise of this approach is demonstrated using a common Gaussian basis set on alkane and globular molecules of different sizes. In general, basis sets using compactly supported or fast-decaying basis functions could be used.

The new compression technique and the $\mathcal{H}^2$ matrix approach can be extended to accelerate the tensor contractions in DF [8, 35, 39, 43] and, in general, quantum chemical methods that already use CFMM. In particular, the approach could be extended to calculate Coulomb energy gradients [3, 34, 40] and potentials for periodic systems [23, 24].

To further improve the proposed compression technique and reduce the $\mathcal{H}^2$ matrix construction cost, it is possible to apply heuristic algebraic compression methods, such as sampling-based methods [1, 10], to accelerate the intermediate ID approximation in Algorithm 4.1. Finally, it is also possible to use an even weaker admissibility rule than that used in this paper for $\mathcal{H}^2$ matrix representations to try to compress even more interactions in the ERI matrix.

REFERENCES

[1] M. Bebendorf, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.

[2] N. H. F. Beebe and J. Linderberg, *Simplifications in the generation and transformation of two-electron integrals in molecular calculations*, Int. J. Quantum Chem., 12 (1977), pp. 683–705.

[3] J. C. Burant, M. C. Strain, G. E. Scuseria, and M. J. Frisch, *Analytic energy gradients for the Gaussian very fast multipole method (GvFMM)*, Chem. Phys. Lett., 248 (1996), pp. 43–49.

[4] M. Challacombe and E. Schwegler, *Linear scaling computation of the Fock matrix*, J. Chem. Phys., 106 (1997), pp. 5526–5536.

[5] M. Challacombe, E. Schwegler, and J. Almlöf, *Fast assembly of the Coulomb matrix: A quantum chemical tree code*, J. Chem. Phys., 104 (1996), pp. 4685–4698.

[6] H. Cheng, Z. Gimbutas, P. G. Martinsson, and V. Rokhlin, *On the compression of low rank matrices*, SIAM J. Sci. Comput., 26 (2005), pp. 1389–1404.

[7] E. Chow, X. Liu, S. Misra, M. Dukhan, M. Smelyanskiy, J. R. Hammond, Y. Du, X. Liao, and P. Dubey, *Scaling up Hartree-Fock calculations on Tianhe-2*, Int. J. High Performance Comput. Appl., 30 (2016), pp. 85–102.

[8] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, *On some approximations in applications of $X\alpha$ theory*, J. Chem. Phys., 71 (1979), pp. 3396–3402.

[9] R. A. Friesner, *Solution of self-consistent field electronic structure equations by a pseudospectral method*, Chem. Phys. Lett., 116 (1985), pp. 39–43.

[10] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21.
[11] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
[12] L. Greengard and V. Rokhlin, *A new version of the fast multipole method for the Laplace equation in three dimensions*, Acta Numer., 6 (1997), pp. 229–269.
[13] M. Gu and S. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.
[14] W. Hackbusch and S. Börm, *Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices*, Computing, 69 (2002), pp. 1–35.
[15] W. Hackbusch, B. Khoromskij, and S. A. Sauter, *On $\mathcal{H}^2$-matrices*, in Lectures on Applied Mathematics, H.-J. Bungartz, R. H. W. Hoppe, and C. Zenger, eds., Springer-Verlag, Berlin, 2000, pp. 9–29.
[16] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.
[17] M. Häser and R. Ahlrichs, *Improvements on the direct SCF method*, J. Comput. Chem., 10 (1989), pp. 104–111.
[18] T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory*, John Wiley & Sons, Hoboken, NJ, 2014.
[19] K. Ho and L. Greengard, *A fast direct solver for structured linear systems by recursive skeletonization*, SIAM J. Sci. Comput., 34 (2012), pp. A2507–A2532.
[20] E. G. Hohenstein, R. M. Parrish, and T. J. Martínez, *Tensor hypercontraction density fitting.* I. *Quartic scaling second- and third-order Møller-Plesset perturbation theory*, J. Chem. Phys., 137 (2012), 044103.
[21] V. Khoromskaia, B. N. Khoromskij, and R. Schneider, *Tensor-structured factorized calculation of two-electron integrals in a general basis*, SIAM J. Sci. Comput., 35 (2013), pp. A987–A1010.
[22] H. Koch, A. Sánchez de Merás, and T. B. Pedersen, *Reduced scaling in electronic structure calculations using Cholesky decompositions*, J. Chem. Phys., 118 (2003), pp. 9481–9484.
[23] R. Łazarski, A. M. Burow, L. Grajciar, and M. Sierka, *Density functional theory for molecular and periodic systems using density fitting and continuous fast multipole method: Analytical gradients*, J. Comput. Chem., 37 (2016), pp. 2518–2526.
[24] R. Łazarski, A. M. Burow, and M. Sierka, *Density functional theory for molecular and periodic systems using density fitting and continuous fast multipole methods*, J. Chem. Theory Comput., 11 (2015), pp. 3029–3041.
[25] C. A. Lewis, J. A. Calvin, and E. F. Valeev, *Clustered low-rank tensor format: Introduction and application to fast construction of Hartree–Fock exchange*, J. Chem. Theory Comput., 12 (2016), pp. 5868–5880.
[26] J. Lu and L. Ying, *Compression of the electron repulsion integral tensor in tensor hypercontraction format with cubic scaling cost*, J. Comput. Phys., 302 (2015), pp. 329–335.
[27] P. G. Martinsson, *A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1251–1274.
[28] P. G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comput. Phys., 205 (2005), pp. 1–23.
[29] V. Minden, A. Damle, K. L. Ho, and L. Ying, *Fast spatial Gaussian process maximum likelihood estimation via skeletonization factorizations*, Multiscale Model. Simul., 15 (2017), pp. 1584–1611.
[30] B. Peng and K. Kowalski, *Highly efficient and scalable compound decomposition of two-electron integral tensor and its application in coupled cluster calculations*, J. Chem. Theory Comput., 13 (2017), pp. 4179–4192.
[31] J. M. Pérez-Jordá and W. Yang, *Fast evaluation of the Coulomb energy for electron densities*, J. Chem. Phys., 107 (1997), pp. 1218–1226.
[32] B. P. Pritchard and E. Chow, *Horizontal vectorization of electron repulsion integrals*, J. Comput. Chem., 37 (2016), pp. 2537–2546.
[33] E. Rudberg and P. Sałek, *Efficient implementation of the fast multipole method*, J. Chem. Phys., 125 (2006), 084106.
[34] Y. Shao, C. A. White, and M. Head-Gordon, *Efficient evaluation of the Coulomb force in density-functional theory calculations*, J. Chem. Phys., 114 (2001), pp. 6572–6577.
[35] M. Sierka, A. Hogekamp, and R. Ahlrichs, *Fast evaluation of the Coulomb potential for electron densities using multipole accelerated resolution of identity approximation*, J. Chem. Phys., 118 (2003), pp. 9136–9148.

[36] M. C. Strain, G. E. Scuseria, and M. J. Frisch, *Achieving linear scaling for the electronic quantum Coulomb problem*, Science, 271 (1996), pp. 51–53.
[37] D. L. Strout and G. E. Scuseria, *A quantitative study of the scaling properties of the Hartree-Fock method*, J. Chem. Phys., 102 (1995), pp. 8448–8452.
[38] X. Sun and N. P. Pitsianis, *A matrix version of the fast multipole method*, SIAM Rev., 43 (2001), pp. 289–300.
[39] F. Weigend, *A fully direct RI-HF algorithm: Implementation, optimised auxiliary basis sets, demonstration of accuracy and efficiency*, Phys. Chem. Chem. Phys., 4 (2002), pp. 4285–4291.
[40] F. Weigend and M. Häser, *RI-MP2: first derivatives and global consistency*, Theor. Chem. Acc., 97 (1997), pp. 331–340.
[41] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, *The continuous fast multipole method*, Chem. Phys. Lett., 230 (1994), pp. 8–16.
[42] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, *Linear scaling density functional calculations via the continuous fast multipole method*, Chem. Phys. Lett., 253 (1996), pp. 268–278.
[43] J. L. Whitten, *Coulombic potential energy integrals and approximations*, J. Chem. Phys., 58 (1973), pp. 4496–4501.
[44] X. Xing and E. Chow, *Interpolative decomposition via proxy points for kernel matrices*, SIAM J. Matrix Anal. Appl., to appear.
[45] X. Xing and E. Chow, *Error analysis of an accelerated interpolative decomposition for 3D Laplace problems*, Appl. Comput. Harmon. Anal., in press.
[46] L. Ying, *A kernel independent fast multipole algorithm for radial basis functions*, J. Comput. Phys., 213 (2006), pp. 451–457.
[47] L. Ying, G. Biros, and D. Zorin, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, J. Comput. Phys., 196 (2004), pp. 591–626.
[48] R. Yokota, H. Ibeid, and D. Keyes, *Fast Multipole Method as a Matrix-Free Hierarchical Low-Rank Approximation*, https://arxiv.org/abs/1602.02244, 2016.