Communications on Applied Mathematics and Computation
https://doi.org/10.1007/s42967-023-00352-z

ORIGINAL PAPER

Hierarchical Interpolative Factorization for Self Green's Function in 3D Modified Poisson-Boltzmann Equations

Yihui Tu[1] · **Zhenli Xu**[2] · **Haizhao Yang**[3]

In Memory of Professor Zhong-Ci Shi.

Received: 15 July 2023 / Revised: 14 November 2023 / Accepted: 20 November 2023

Abstract
The modified Poisson-Boltzmann (MPB) equations are often used to describe the equilibrium particle distribution of ionic systems. In this paper, we propose a fast algorithm to solve the MPB equations with the self Green's function as the self-energy in three dimensions, where the solution of the self Green's function poses a computational bottleneck due to the requirement of solving a high-dimensional partial differential equation. Our algorithm combines the selected inversion with hierarchical interpolative factorization for the self Green's function, building upon our previous result of two dimensions. This approach yields an algorithm with a complexity of $O(N \log N)$ by strategically leveraging the locality and low-rank characteristics of the corresponding operators. Additionally, the theoretical $O(N)$ complexity is obtained by applying cubic edge skeletonization at each level for thorough dimensionality reduction. Extensive numerical results are conducted to demonstrate the accuracy and efficiency of the proposed algorithm for problems in three dimensions.

Keywords Selected inversion · Hierarchical interpolative factorization · Linear scaling · Self Green's function · Modified Poisson-Boltzmann (MPB) equations

Mathematics Subject Classification 65N22 · 65F50 · 82B21

✉ Zhenli Xu
xuzl@sjtu.edu.cn

Yihui Tu
tuyihui22aa@sjtu.edu.cn

Haizhao Yang
hzyang@umd.edu

[1] School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, China

[2] School of Mathematical Sciences, MOE-LSC and CMA-Shanghai, Shanghai Jiao Tong University, Shanghai 200240, China

[3] Department of Mathematics and Department of Computer Science, University of Maryland, College Park, MD 20742, USA

Published online: 06 March 2024

1 Introduction

Electrostatic interactions play a crucial role in various systems at the nano-/micro-scale such as biomolecules, supercapacitors, and charged soft matter [2, 8, 19, 34]. To provide a continuum description of charged systems, the Poisson-Boltzmann (PB) theory [5, 13], based on the mean-field assumption, is a typical implicit solvent model for describing the distribution of ions. However, this theory falls short in capturing many-body characteristics that are essential for describing electrostatic many-body behaviors in various systems, such as ion correlation and dielectric fluctuation.

Various modified theories have been proposed [1, 3, 25] to incorporate many-body effects, along with many numerical methods [23, 24, 40]. The Gaussian variational field theory [30, 32] presents a promising approach to account for long-range Coulomb correlation, including dielectric variation [27, 29, 31]. This theory considers the self-energy of a test ion as a correction to the mean-field potential energy, which is described by the self Green's function. By taking into account the self-energy correction, the effect of dielectric inhomogeneity can be incorporated [7, 17, 28, 36]. The self Green's function used in the field theory satisfies the generalized Debye-Hückel (GDH) equation. However, the numerical solution of the GDH equation is computationally expensive due to its high spatial dimensions. Based on the finite-difference discretization, the self Green's function corresponds to the diagonal of the inverse of the discrete elliptic differential operator in the GDH equation. The aim of our study is to calculate the self-energy in the GDH equation. This procedure serves to accelerate the numerical solution of the modified Poisson-Boltzmann (MPB) equations. To achieve this, an efficient algorithm is necessary for determining the diagonal elements of the matrix inverse.

One straightforward method for extracting the diagonal of the matrix inverse is to initially compute the entire matrix and then simply extract the diagonal. However, this naive inversion approach has a computational complexity of $O(N^3)$, which is equivalent to that of matrix factorization. In the realm of electronic structure and electrostatic correlation, significant efforts are dedicated to devising efficient methods for obtaining the diagonal of the matrix inverse. A promising approach involves the utilization of sparsity and low-rankness of the matrix, leading to the development of fast algorithms. The selected inversion method, introduced by Lin et al. [20–22], offers an algorithm with a computational complexity of $O(N^{3/2})$ for 2D problems and $O(N^2)$ for 3D problems. This method involves a hierarchical decomposition of the computational domain $\mathbf{\Omega}$ and consists of two phases. Constructing the hierarchical Schur complements of the interior points for the blocks of the domain in a bottom-up pass, and then extracting the diagonal entries efficiently in a top-down pass by taking advantage of the hierarchical locality of the inverse matrices. To enhance the efficiency of this method, Lin et al. [21, 22] exploited a supernode left-looking LDL factorization of the matrix, which significantly reduces the prefactor in computational complexity. Additionally, Xia et al. [38] applied structured multifrontal LDL factorizations to achieve $O(N\text{poly}(\log N))$ complexity.

Recently, the hierarchical interpolative factorization (HIF) [15, 16] has been proposed, combining multifrontal [4, 9, 10, 26] with recursive dimensional reduction through frontal skeletonization. This approach aims to generate an approximate generalized LU/LDL decomposition with a linear or quasi-linear estimated computational cost. In contrast to previous methods [11, 12, 14, 33, 37] that utilize fast structured methods to work implicitly with entire fronts while keeping them implicitly, the HIF offers the advantage of explicit front reduction. Consequently, the HIF provides significant

savings in terms of computational resources required for solving 3D problems, making it well-suited for large-scale problems.

A more recent development in this area is the selected inversion with hierarchical interpolative factorization (SelInvHIF) [35]. In the SelInvHIF, the supernode left-looking LDL factorization is replaced with the hierarchical interpolative factorization, and the extraction phase is modified to approximate the diagonal of the matrix inverse with $O(N)$ operations for 2D problems. In this work, we further extend the SelInHIF to 3D problems with the $O(N \log N)$ complexity by face skeletonization and the theoretical $O(N)$ complexity using skeletonizing cubic faces and then edges. We remark that, in 3D problems, the increase in the number of degrees of freedom (DOFs) is significant, resulting in more complex interactions compared to 2D problems. Moreover, the numerical method of the PB equations for 3D problems is more practical for various applications in biophysics and materials science. For convenience, the former algorithm is still referred to as the SelInvHIF, while the latter is named "SelInvHIF with edge skeletonization". To demonstrate the computational complexity of the algorithm, we provide comprehensive theoretical derivations and present various numerical examples. In the following section, we introduce the MPB equation, as it serves as a suitable problem for testing the scaling of the algorithm in the context of 3D problems.

The remaining sections of the paper are organized as follows. Section 2 introduces the concept of skeletonization in matrix factorization and provides a detailed presentation of the SelInvHIF algorithm and the SelInvHIF with edge skeletonization. Section 3 focuses on the MPB equation and the corresponding iterative method. Section 4 performs numerical results to demonstrate the performance of the SelInvHIF algorithm for 3D problems. Finally, we conclude the paper and discuss future work in Sect. 5.

2 The SelInvHIF Algorithm

In this section, we provide a detailed explanation of the SelInvHIF algorithm, followed by the introduction of the SelInvHIF with edge skeletonization in Sect. 2.3. The SelInvHIF algorithm consists of two main steps. In the first step, hierarchical Schur complements are constructed for the diagonal blocks of the matrix A, which is discretized uniformly from the differential operator on a rectangular domain $\mathbf{\Omega}$. In the subsequent step, the diagonal elements of the inverse matrix A^{-1} are extracted from the constructed hierarchy of Schur complements. Before the introduction of the formal description of the SelInvHIF algorithm, we give a brief overview of the skeletonization of matrix factorization.

To establish the foundation for our algorithm, let us introduce some fundamental symbols and present the necessary theorems. Given a matrix A, A_{pq}, or $A(I, J)$ is a submatrix with restricted rows and columns, where the p, q, r, I, and J denote the ordered sets of indices. For the sake of simplicity, the matrix A is assumed to be symmetric and nonsingular, given by

$$A = \begin{bmatrix} A_{pp} & A_{qp}^{\mathrm{T}} & \\ A_{qp} & A_{qq} & A_{rq}^{\mathrm{T}} \\ & A_{rq} & A_{rr} \end{bmatrix}, \tag{1}$$

which is defined over the indices (p, q, r). In this matrix structure, p is related to the DOFs of the interior points on domain $\mathcal{D}$ (which is a subdomain of $\mathbf{\Omega}$), q to the DOFs on the boundary $\partial\mathcal{D}$, and r to the external domain $\mathbf{\Omega}/\overline{\mathcal{D}}$. Typically, the DOFs q separates p from r which is often very large. Let $G = A^{-1}$ and $G_1 = A_1^{-1}$. Here, G_{pp} represents the submatrix

of G corresponding to the row and column index set p, and A_1 is the Schur complement of A_{pp}, i.e.,

$$A_1 = \begin{bmatrix} A_{qq} - A_{qp}A_{pp}^{-1}A_{qp}^{\mathrm{T}} & A_{rq}^{\mathrm{T}} \\ A_{rq} & A_{rr} \end{bmatrix}.$$

It is noted that one can also discretize the operator on an irregular domain using the finite element method. The points in each subdomain correspond to a small matrix and form a stiffness matrix through the relationship of the nodes. The stiffness matrix is similar to (1), and our method can be also employed to solve it.

A crucial observation in the selected inversion method [20], which is employed as a preliminary tool in the SelInvHIF, is based on the fact that to compute G_{pp}, only the values of G_1 involving interactions with the direct matrix A, $(G_1)_{qq}$, are required instead of the entire inverse of the Schur complement. This implies that the determination of G_{pp} relies on $(G_1)_{qq}$. Furthermore, the diagonal entry $(G_1)_{qq}$ can be calculated by utilizing a diagonal block of the inverse of the Schur complement of a submatrix of A_1. By recursively applying this approach, an efficient algorithm for computing G_{pp} can be derived. Essentially, one can compute a diagonal block of A^{-1} by using a diagonal block of the inverse of the Schur complement of a submatrix of A. By repeatedly applying this observation, a recursive algorithm is developed to compute the diagonal entries of A^{-1} efficiently.

The interpolative decomposition (ID) [6] for low-rank matrices, based on Lemma 1 below, is the second frequently used tool in the SelInvHIF. Suppose a disjoint partition of $q = \hat{q} \cup \check{q}$ with $|\hat{q}| = k$ is used. The sets $\hat{q}$ and $\check{q}$ are referred to as the skeleton and redundant indices, respectively.

Lemma 1 *Assume $A \in \mathbb{R}^{m\times n}$ with rank $k \leqslant \min(m, n)$ and q be the set of all column indices of A. Then there exists a matrix $T_q \in \mathbb{R}^{k\times(n-k)}$ such that $A_{:,\check{q}} = A_{:,\hat{q}}T_q$.*

Specifically, the redundant columns of the matrix A can be represented by the skeleton columns and the associated interpolation matrix from Lemma 1, and the following formula holds:

$$A\begin{bmatrix} I \\ -T_q \ I \end{bmatrix} = \begin{bmatrix} A_{:,\check{q}} \ A_{:,\hat{q}} \end{bmatrix}\begin{bmatrix} I \\ -T_q \ I \end{bmatrix} = \begin{bmatrix} \mathbf{0} \ A_{:,\hat{q}} \end{bmatrix}. \tag{2}$$

Equation (2) indicates that the sparsification of the matrix A is feasible by multiplying a triangular matrix formed from the interpolation matrix T_q in Lemma 1.

The utilization of (2) facilitates the elimination of redundant DOFs of a dense matrix featuring low-rank off-diagonal blocks, resulting in a structured matrix of the form (1). This idea is referred to as block inversion with skeletonization and is discussed in Lemma 2, where one uses $A_{:,\hat{q}}T_q$ to approximate $A_{:,\check{q}}$ for the purpose. It is worth noting that the idea of skeletonization was originally introduced in the HIF method [15].

Lemma 2 *Let the symmetric matrix A have the following form*:

$$A = \begin{bmatrix} A_{pp} & A_{qp}^{\mathrm{T}} \\ A_{qp} & A_{qq} \end{bmatrix},$$

where A_{qp} is numerically low-rank. Let the interpolation matrix T_p satisfy $A_{q\check{p}} \approx A_{q\hat{p}} T_p$ with $p = \hat{p} \cup \check{p}$. Without loss of generality, one can approximately rewrite

$$A = \begin{bmatrix} A_{\check{p}\check{p}} & A_{\hat{p}\check{p}}^{\mathrm{T}} & A_{q\check{p}}^{\mathrm{T}} \\ A_{\hat{p}\check{p}} & A_{\hat{p}\hat{p}} & A_{q\hat{p}}^{\mathrm{T}} \\ A_{q\check{p}} & A_{q\hat{p}} & A_{qq} \end{bmatrix}$$

and define

$$Q_p = \begin{bmatrix} I & & \\ -T_p & I & \\ & & I \end{bmatrix}.$$

Let $\bar{A} \triangleq Q_p^{\mathrm{T}} A Q_p$. Then one has

$$\bar{A} = \begin{bmatrix} B_{\check{p}\check{p}} & B_{\hat{p}\check{p}}^{\mathrm{T}} & \\ B_{\hat{p}\check{p}} & A_{\hat{p}\hat{p}} & A_{q\hat{p}}^{\mathrm{T}} \\ & A_{q\hat{p}} & A_{qq} \end{bmatrix} \tag{3}$$

with $B_{\check{p}\check{p}} = A_{\check{p}\check{p}} - T_p^{\mathrm{T}} A_{\hat{p}\check{p}} - A_{\hat{p}\check{p}}^{\mathrm{T}} T_p + T_p^{\mathrm{T}} A_{\hat{p}\hat{p}} T_p$ and $B_{\hat{p}\check{p}} = A_{\hat{p}\check{p}} - A_{\hat{p}\hat{p}} T_p$.

Further suppose that $B_{\check{p}\check{p}}$ is nonsingular. Let $G = A^{-1}$, $\bar{G} = \bar{A}^{-1}$, $G_1 = G_{\hat{p}\cup q, \hat{p}\cup q}$, and $\bar{A}_1$ be the Schur complement of $B_{\check{p}\check{p}}$, i.e.,

$$\bar{A}_1 = \begin{bmatrix} A_{\hat{p}\hat{p}} - B_{\hat{p}\check{p}} B_{\check{p}\check{p}}^{-1} B_{\hat{p}\check{p}}^{\mathrm{T}} & A_{q\hat{p}}^{\mathrm{T}} \\ A_{q\hat{p}} & A_{qq} \end{bmatrix},$$

and $\bar{G}_1 = \bar{A}_1^{-1}$. Then, by (3) the following formulas hold:

$$G_{\check{p}\check{p}} = \bar{G}_{\check{p}\check{p}} = B_{\check{p}\check{p}}^{-1} + \begin{bmatrix} -B_{\check{p}\check{p}}^{-1} B_{\hat{p}\check{p}}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \bar{G}_1 \begin{bmatrix} -B_{\check{p}\check{p}}^{-1} B_{\hat{p}\check{p}}^{\mathrm{T}} & \mathbf{0} \end{bmatrix}^{\mathrm{T}},$$

$$G_1 = \begin{bmatrix} T_p B_{\check{p}\check{p}}^{-1} T_p^{\mathrm{T}} & \\ & \mathbf{0} \end{bmatrix} + \begin{bmatrix} T_p B_{\check{p}\check{p}}^{-1} B_{\hat{p}\check{p}}^{\mathrm{T}} + I & \\ & I \end{bmatrix} \bar{G}_1 \begin{bmatrix} B_{\hat{p}\check{p}} B_{\check{p}\check{p}}^{-1} T_p^{\mathrm{T}} + I & \\ & I \end{bmatrix}.$$

Lemma 2 demonstrates that computing $G_{\check{p}\check{p}}$ only requires the values of $\bar{G}_1$ associated with row and column indices in $\hat{p}$, rather than the entire inverse of the Schur complement. Consequently, $G_{\check{p}\check{p}}$ is determined by $(\bar{G}_1)_{\hat{p}\hat{p}}$, a diagonal block of $\bar{A}_1^{-1}$, which has a smaller size than the original matrix A. Despite $\bar{A}_1$ may be dense if it has low-rank off-diagonal blocks, then the same approach used in (3) can be applied to compute a diagonal block of $\bar{G}_1$, resulting in a recursive algorithm that efficiently computes the diagonal blocks of G.

This skeletonization technique was proposed by Ho and Ying [15] and is based on the observation that the Schur complements have specific low-rank structures. Specifically, A_{pp}^{-1}, obtained from a local differential operator, often features low-rank off-diagonal blocks. Additionally, numerical experiments illustrate that the Schur complement interaction $A_{qq} - A_{qp} A_{pp}^{-1} A_{qp}^{\mathrm{T}}$ also possesses the same rank structure. This observation may be comprehended through the interpretation of the matrix A_{pp}^{-1} as the discrete Green's function associated with the elliptic PDE. Due to the locality property, it is well established that

such a Green's function tends to exhibit off-diagonal blocks of low rank in numerical analysis. Furthermore, this analogous rank structure effectively extends to the Schur complement B_{qq}. In the following subsection, we employ Lemma 2 to generate hierarchical Schur complements for diagonal blocks of A.

2.1 Hierarchy of Schur Complements

To achieve a hierarchical disjoint partition for the differential operator in domain $\mathbf{\Omega}$, bipartitioning is performed in each dimension, resulting in leaf domains of size $r_0 \times r_0 \times r_0$ and a total integer level L. Domain $\mathbf{\Omega}$ is defined by a grid size of $\sqrt[3]{N} \times \sqrt[3]{N} \times \sqrt[3]{N} = r_0 2^{L-1} \times r_0 2^{L-1} \times r_0 2^{L-1}$ and is associated with a matrix A of size $N \times N$. Furthermore, to take advantage of the low-rankness of the matrix A, $L-1$ fractional levels are introduced between L integer levels. The hierarchy construction of Schur complements is carried out at levels 1, 3/2, 2, 5/2, $\cdots$, and L.

Let us consider the case of $r_0 = 6$ and $L = 3$ to describe the process in detail without loss of generality. Initially, the entire domain is regarded as the top level (Level 3) and is partitioned into eight blocks at the next level (Level 2). Each block is further partitioned into eight sub-blocks at a lower level (Level 1), resulting in a total of $2^{L-1} \times 2^{L-1} \times 2^{L-1} = 64$ blocks at the bottom level, as illustrated in Fig. 1. In addition, one fractional level is considered between two consecutive integer levels, and the low-rank matrices that represent the fronts between domain blocks are reduced into skeletons by this level.

2.1.1 Bottom Level $\ell = 1$

The initial index set J_0 follows the row-major ordering, while domain $\mathbf{\Omega}$ is hierarchically partitioned into disjoint blocks at level $\ell = 1$. Each block has a size of $2^{L-\ell} \times 2^{L-\ell} \times 2^{L-\ell} = 4 \times 4 \times 4$. All points within each block are classified into interior and boundary points. The interior points, denoted as $I_{1;ijk}$ (shown in light blue in Fig. 1), are not related to the points in other blocks. On the other hand, the boundary points, denoted as $J_{1;ijk}$, are connected to neighboring points in other blocks. Here, $i, j, k = 1, 2, 3, 4$ are the indices of the blocks in each dimension. The differential operators have a locality property, which implies that $A(I_{1;ijk}, I_{1;i'j'k'}) = 0$ (or $A(I_{1;ijk}, J_{1;i'j'k'}) = 0$) if $(i, j, k) \neq (i', j', k')$.

The interior points are removed using the block inversion. One can then focuses the problem on the boundary points. To achieve this, one uses the proper row and column permutations to the matrix A defined with the index set J_0 to place all of the interior points in

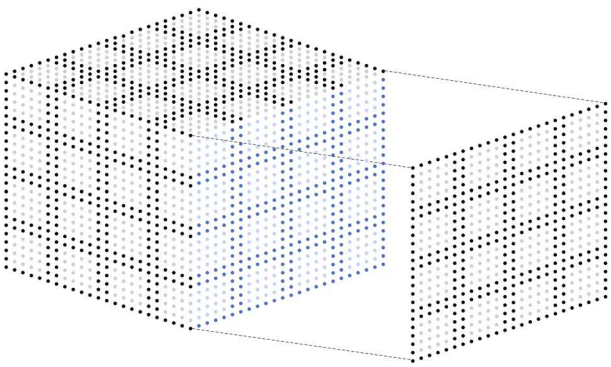

Fig. 1 The DOFs in the bottom level. In this level, the domain is divided into 64 blocks. The interior points are indicated by light blue, while the boundary points are indicated by black (blue or gray). Note that the prefactor is reduced due to the share faces (edges)

front of the boundary points. The matrix A can be permuted into a new matrix by a permutation matrix P_1 as follows:

$$A_1 = P_1^{-1} A P_1 = \begin{bmatrix} U_1 & V_1^{\mathrm{T}} \\ V_1 & W_1 \end{bmatrix} \tag{4}$$

with the index set $(I_1|J_1)$, $U_1 = A_1(I_1, I_1)$, $V_1 = A_1(J_1, I_1)$, and $W_1 = A_1(J_1, J_1)$. Here I_1 represents the indices of all interior points, denoted as $I_1 = I_{1;111} I_{1;121} \cdots I_{1;444}$, and J_1 represents the indices of all boundary points, denoted as $J_1 = J_{1;111} J_{1;121} \cdots J_{1;444}$.

Due to the locality property, both U_1 and V_1 are block diagonal matrices. Figure 1 shows that interior points in different blocks are not connected. Boundary points in each block are only connected to the interior points in the same block. Furthermore, U_1 and V_1 are of the following form:

$$U_1 = \begin{bmatrix} U_{1;111} & & & \\ & U_{1;121} & & \\ & & \ddots & \\ & & & U_{1;444} \end{bmatrix}, V_1 = \begin{bmatrix} V_{1;111} & & & \\ & V_{1;121} & & \\ & & \ddots & \\ & & & V_{1;444} \end{bmatrix}$$

with $U_{1;ijk} = A_1(I_{1;ijk}, I_{1;ijk})$ and $V_{1;ijk} = A_1(J_{1;ijk}, I_{1;ijk})$ for $i, j, k = 1, 2, 3, 4$.

Using the Gaussian elimination, one can obtain

$$A_1^{-1} = L_1^{\mathrm{T}} \begin{bmatrix} U_1^{-1} & \\ & (W_1 - V_1 U_1^{-1} V_1^{\mathrm{T}})^{-1} \end{bmatrix} L_1 \quad \text{with} \quad L_1 = \begin{bmatrix} I & \\ -V_1 U_1^{-1} & I \end{bmatrix}. \tag{5}$$

Since U_1 is a block diagonal matrix with each diagonal block of a size $(r_0 - 2)^3 \times (r_0 - 2)^3$, its inverse can be computed directly.

By using the block diagonal matrices V_1 and U_1^{-1}, $V_1 U_1^{-1}$ is also a block diagonal matrix and can be computed independently within each block,

$$V_1 U_1^{-1} = \begin{bmatrix} V_{1;111} U_{1;111}^{-1} & & & \\ & V_{1;121} U_{1;121}^{-1} & & \\ & & \ddots & \\ & & & V_{1;444} U_{1;444}^{-1} \end{bmatrix}.$$

Similarly, the block diagonal matrix $V_1 U_1^{-1} V_1^{\mathrm{T}}$ is expressed as

$$V_1 U_1^{-1} V_1^{\mathrm{T}} = \begin{bmatrix} V_{1;111} U_{1;111}^{-1} V_{1;111}^{\mathrm{T}} & & & \\ & V_{1;121} U_{1;121}^{-1} V_{1;121}^{\mathrm{T}} & & \\ & & \ddots & \\ & & & V_{1;444} U_{1;444}^{-1} V_{1;444}^{\mathrm{T}} \end{bmatrix}.$$

Combining (4) and (5), one has

$$G = P_1 A_1^{-1} P_1^{-1} = P_1 L_1^{\mathrm{T}} \begin{bmatrix} U_1^{-1} & \\ & G_1 \end{bmatrix} L_1 P_1^{-1}, \tag{6}$$

where $G_1 = (W_1 - V_1 U_1^{-1} V_1^{\mathrm{T}})^{-1}$ is the inverse of the Schur complement of U_1. Consequently, by removing interior points from the matrix A, one is able to simplify the problem.

2.1.2 Fractional Level $\ell = 3/2$

At this level, the objective is to obtain G_1, as defined in (6), for the index set J_1, which corresponds to the boundary points of the domain blocks at the first level. The domain $\mathbf{\Omega}$ is divided into 64 blocks, resulting in a total of 384 faces in this example where $L = 3$ (refer to Fig. 2a). In Fig. 2, one chooses all the skeleton points as separators as a general rule to clearly illustrate the concept of method. Note that in practical applications the distribution of skeleton points depends on the error tolerance of the ID approximation. A greater number of interior points as skeleton points are needed for higher accuracy in the ID. Each face consists of the DOFs within its corresponding area, as well as some DOFs located on its boundary. Moreover, a face not only interacts within its own block but also interacts with faces in neighboring blocks. The resulting matrix exhibits low-rank off-diagonal blocks since the DOFs of a face only interact with a limited number of neighboring blocks. Furthermore, Lemma 2 is applied to skeletonize the DOFs on the faces in each block. An ID can be implemented to approximately select the redundant and skeleton DOFs within each block, and the resulting interpolation matrix is recorded as specified in Lemma 1. Additionally, the redundant DOFs are denoted by $I_{3/2;i}$, the skeleton DOFs are represented by $J_{3/2;i}$, and the associated interpolation matrix is indicated by $T_{3/2;i}$.

Similar to the bottom level, an appropriate permutation matrix $P_{3/2}$ is designed to move all of the redundant points in front of the skeleton points and reindex J_1 by the permutation matrix. Furthermore, the matrix $W_1 - V_1 U_1^{-1} V_1^{\mathrm{T}}$ can be permuted into a new matrix by the permutation matrix $P_{3/2}$ as follows:

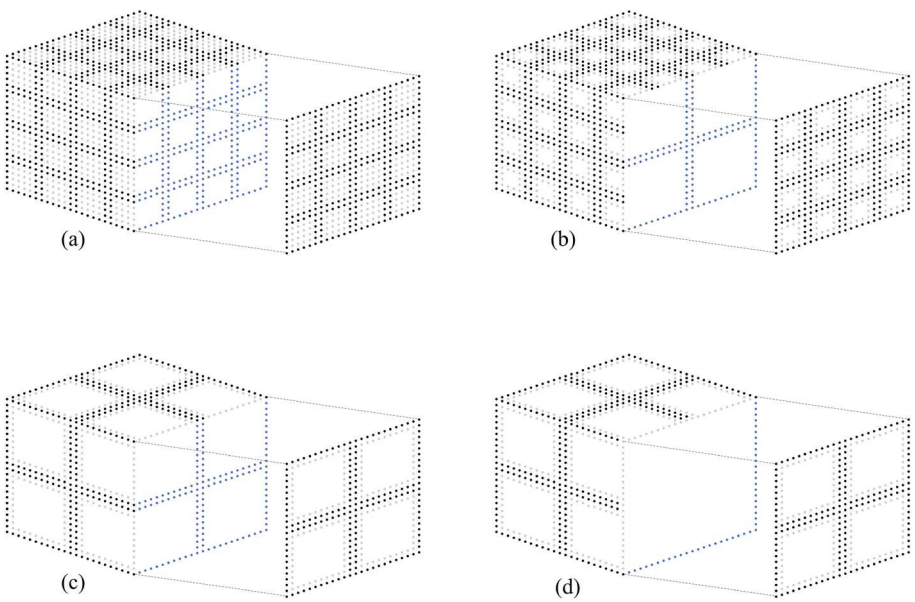

Fig. 2 **a** The DOFs in the level $\ell = 3/2$. The domain is divided into 384 faces of 64 blocks. **b** The DOFs in the level $\ell = 2$ after elimination. The domain is divided into 8 blocks. **c** The DOFs in the level $\ell = 5/2$ after skeletonization. **d** The DOFs in the level $\ell = 3$. This is the top level

$$A_{\frac{3}{2}} = P_{\frac{3}{2}}^{-1}(W_1 - V_1 U_1^{-1} V_1^{\mathrm{T}})P_{\frac{3}{2}} = \begin{bmatrix} U_{\frac{3}{2}} & V_{\frac{3}{2}}^{\mathrm{T}} \\ V_{\frac{3}{2}} & W_{\frac{3}{2}} \end{bmatrix}$$

with the index set $(I_{3/2}|J_{3/2})$, $U_{3/2} = A_{3/2}(I_{3/2}, I_{3/2})$, $V_{3/2} = A_{3/2}(J_{3/2}, I_{3/2})$, and the dense matrix $W_{3/2} = A_{3/2}(J_{3/2}, J_{3/2})$. Here $I_{3/2}$ represents the indices of all redundant points, denoted as $I_{3/2;1} I_{3/2;2} \cdots I_{3/2;384}$, and $J_{3/2}$ represents the indices of all skeleton points, such that it can be denoted as $J_{3/2;1} J_{3/2;2} \cdots J_{3/2;384}$.

Denote $T_{3/2}$ by a block diagonal matrix

$$T_{\frac{3}{2}} = \begin{bmatrix} -T_{\frac{3}{2};1} & & \\ & \ddots & \\ & & -T_{\frac{3}{2};384} \end{bmatrix}$$

and arrange a $|J_1| \times |J_1|$ matrix

$$Q_{\frac{3}{2}} = \begin{bmatrix} I & \\ T_{\frac{3}{2}} & I \end{bmatrix}.$$

Thus, the new matrix is updated

$$\bar{A}_{\frac{3}{2}} = Q_{\frac{3}{2}}^{\mathrm{T}} A_{\frac{3}{2}} Q_{\frac{3}{2}} = \begin{bmatrix} \bar{U}_{\frac{3}{2}} & \bar{V}_{\frac{3}{2}}^{\mathrm{T}} \\ \bar{V}_{\frac{3}{2}} & W_{\frac{3}{2}} \end{bmatrix},$$

where $\bar{U}_{3/2}$ and $\bar{V}_{3/2}$ are block diagonal matrices with

$$\bar{U}_{\frac{3}{2}}\left(I_{\frac{3}{2};i}, I_{\frac{3}{2};j}\right) = 0, \quad \bar{V}_{\frac{3}{2}}\left(J_{\frac{3}{2};i}, I_{\frac{3}{2};j}\right) = 0, \quad \forall i \neq j.$$

Similarly, one can obtain the following inverse by Gaussian elimination:

$$\bar{A}_{\frac{3}{2}}^{-1} = L_{\frac{3}{2}}^{\mathrm{T}} \begin{bmatrix} \bar{U}_{\frac{3}{2}}^{-1} & \\ & G_{\frac{3}{2}} \end{bmatrix} L_{\frac{3}{2}}$$

with

$$L_{\frac{3}{2}} = \begin{bmatrix} I & \\ -\bar{V}_{\frac{3}{2}} \bar{U}_{\frac{3}{2}}^{-1} & I \end{bmatrix}, \quad G_{\frac{3}{2}} = \left(W_{\frac{3}{2}} - \bar{V}_{\frac{3}{2}} \bar{U}_{\frac{3}{2}}^{-1} \bar{V}_{\frac{3}{2}}^{\mathrm{T}}\right)^{-1}$$

as in Lemma 2. Since $-\bar{V}_{3/2}\bar{U}_{3/2}^{-1}$ and $\bar{V}_{3/2}\bar{U}_{3/2}^{-1}\bar{V}_{3/2}^{\mathrm{T}}$ are block diagonal matrices, they can be computed independently within each block. Thus,

$$G_1 \approx P_{\frac{3}{2}} Q_{\frac{3}{2}} L_{\frac{3}{2}}^{\mathrm{T}} \begin{bmatrix} \bar{U}_{\frac{3}{2}}^{-1} & \\ & G_{\frac{3}{2}} \end{bmatrix} L_{\frac{3}{2}} Q_{\frac{3}{2}}^{\mathrm{T}} P_{\frac{3}{2}}^{-1}.$$

Therefore, the inversion problem is reduced to a smaller matrix $W_{3/2} - \bar{V}_{3/2}\bar{U}_{3/2}^{-1}\bar{V}_{3/2}^{\mathrm{T}}$ by eliminating the redundant DOFs as in Lemma 2.

2.1.3 Middle Level $\ell = 2$

At Level $\ell = 2$, the domain $\mathbf{\Omega}$ is divided into $2^{L-\ell} \times 2^{L-\ell} \times 2^{L-\ell} = 2 \times 2 \times 2$ blocks, each consisting of interior and boundary points. Similar to the previous integer level, a permutation matrix P_2 is used to reindex the points in $J_{3/2}$ into I_2 and J_2,

$$J_{\frac{3}{2}} \xrightarrow{P_2} (I_{2;11} I_{2;12} I_{2;21} I_{2;22} | J_{2;11} J_{2;12} J_{2;21} J_{2;22}) := (I_2 | J_2).$$

Use the same strategy as at Level 1 and denote

$$A_2 = P_2^{-1} \left(W_{\frac{3}{2}} - \bar{V}_{\frac{3}{2}} \bar{U}_{\frac{3}{2}}^{-1} \bar{V}_{\frac{3}{2}}^{\mathrm{T}} \right) P_2 = \begin{bmatrix} U_2 & V_2^{\mathrm{T}} \\ V_2 & W_2 \end{bmatrix}$$

with

$$U_2 = A_2(I_2, I_2), \quad V_2 = A_2(J_2, I_2), \quad \text{and} \quad W_2 = A_2(J_2, J_2).$$

It can be observed that matrices U_2 and V_2 possess a block diagonal structure. Thus,

$$G_{\frac{3}{2}} = P_2 L_2^{\mathrm{T}} \begin{bmatrix} U_2^{-1} & \\ & G_2 \end{bmatrix} L_2 P_2^{-1},$$

$$L_2 = \begin{bmatrix} I & \\ -V_2 U_2^{-1} & I \end{bmatrix}, \quad \text{and} \quad G_2 = (W_2 - V_2 U_2^{-1} V_2^{\mathrm{T}})^{-1}.$$

Finally, by removing the interior points, the problem can be simplified to a smaller matrix $W_2 - V_2 U_2^{-1} V_2^{\mathrm{T}}$. The DOFs remaining after elimination at Level $\ell = 2$ are depicted in Fig. 2b.

2.1.4 Fractional $\ell = 5/2$

At Level $\ell = 5/2$, the objective is to find G_2 indexed by J_2. Similarly to Level $\ell = 3/2$, the domain $\mathbf{\Omega}$ is divided into 8 blocks with 48 faces. By employing the ID, one can distinguish the redundant DOFs $I_{5/2;i}$ and the skeleton DOFs $J_{5/2;i}$ in the ith face, and record the interpolation matrix $T_{5/2;i}$. Furthermore, a permutation matrix $P_{5/2}$ is used to reindex J_2 such that

$$J_2 \xrightarrow{P_{\frac{5}{2}}} \left(I_{\frac{5}{2};1} I_{\frac{5}{2};2} I_{\frac{5}{2};3} I_{\frac{5}{2};48} | J_{\frac{5}{2};1} J_{\frac{5}{2};2} J_{\frac{5}{2};3} J_{\frac{5}{2};48} \right) := \left(I_{\frac{5}{2}} | J_{\frac{5}{2}} \right).$$

Denote

$$T_{\frac{5}{2}} = \begin{bmatrix} -T_{\frac{5}{2};1} & & \\ & \ddots & \\ & & -T_{\frac{5}{2};48} \end{bmatrix}$$

and a $|J_2| \times |J_2|$ matrix

$$Q_{\frac{5}{2}} = \begin{bmatrix} I \\ T_{\frac{5}{2}} & I \end{bmatrix}.$$

Then

$$\bar{A}_{\frac{5}{2}} = Q_{\frac{5}{2}}^{\mathrm{T}} P_{\frac{5}{2}}^{-1} (W_2 - V_2 U_2^{-1} V_2^{\mathrm{T}}) P_{\frac{5}{2}} Q_{\frac{5}{2}} = \begin{bmatrix} \bar{U}_{\frac{5}{2}} & \bar{V}_{\frac{5}{2}}^{\mathrm{T}} \\ \bar{V}_{\frac{5}{2}} & W_{\frac{5}{2}} \end{bmatrix}$$

with

$$\bar{U}_{\frac{5}{2}}\left(I_{\frac{5}{2};i}, I_{\frac{5}{2};j}\right) = 0, \quad \bar{V}_{\frac{5}{2}}\left(J_{\frac{5}{2};i}, I_{\frac{5}{2};j}\right) = 0, \quad \forall i \neq j.$$

Therefore,

$$G_2 \approx P_{\frac{5}{2}} Q_{\frac{5}{2}} L_{\frac{5}{2}}^{\mathrm{T}} \begin{bmatrix} \bar{U}_{\frac{5}{2}}^{-1} & \\ & G_{\frac{5}{2}} \end{bmatrix} L_{\frac{5}{2}} Q_{\frac{5}{2}}^{\mathrm{T}} P_{\frac{5}{2}}^{-1}$$

with

$$L_{\frac{5}{2}} = \begin{bmatrix} I \\ -\bar{V}_{\frac{5}{2}} \bar{U}_{\frac{5}{2}}^{-1} & I \end{bmatrix} \quad \text{and} \quad G_{\frac{5}{2}} = \left(W_{\frac{5}{2}} - \bar{V}_{\frac{5}{2}} \bar{U}_{\frac{5}{2}}^{-1} \bar{V}_{\frac{5}{2}}^{\mathrm{T}}\right)^{-1}.$$

It is worth noting that $\bar{U}_{5/2}$ and $\bar{V}_{5/2}$ are block diagonal matrices. As a result, the matrix inversion problem has now been reduced to $W_{5/2} - \bar{V}_{5/2} \bar{U}_{5/2}^{-1} \bar{V}_{5/2}^{\mathrm{T}}$. The DOFs remaining after skeletonization at Level $\ell = 5/2$ are shown in Fig. 2c.

2.1.5 Top Level $\ell = 3$

At this level, the domain $\mathbf{\Omega}$ is partitioned into a single block. This means there is no partition at this level, as shown in Fig. 2d. Similarly, the index set $J_{5/2}$ is reindexed by partitioning it into the union of an interior index set I_3 and a boundary index set J_3. This reindexing is accomplished using a permutation matrix P_3 as follows:

$$J_{\frac{5}{2}} \xrightarrow{P_3} (I_3 | J_3).$$

Thus, one has

$$G_{\frac{5}{2}} = P_3 L_3^{\mathrm{T}} \begin{bmatrix} U_3^{-1} & \\ & G_3 \end{bmatrix} L_3 P_3^{-1}$$

with

$$L_3 = \begin{bmatrix} I \\ -V_3 U_3^{-1} & I \end{bmatrix} \text{and} \quad G_3 = (W_3 - V_3 U_3^{-1} V_3^{\mathrm{T}})^{-1}.$$

In this top level, the inverse of G_3 can be computed directly because of its small size.

2.1.6 The Algorithm for the Hierarchy of Schur Complements

In this section, one aims to construct a hierarchical structure of Schur complements for the matrix A is constructed on an $\sqrt[3]{N} \times \sqrt[3]{N} \times \sqrt[3]{N}$ grid. The construction process involves dividing the points in each block at each integer level into interior and boundary points. Specifically, the interior points are only involved in interactions with other points within the same block, prompting a reindexing and subsequent elimination of the interior points. At each fractional level, face skeletonization is considered, and an ID approach is applied to distinguish redundant and skeleton points. Here, the redundant points only interact with other points in the same cell, leading to reindexing and elimination of the redundant points.

The relationships between levels are defined as follows:

$$G_\ell = \begin{cases} A^{-1}, & \ell = 0; \\ (W_\ell - V_\ell U_\ell^{-1} V_\ell^{\mathrm{T}})^{-1}, & \ell \text{ is an integer}; \\ (W_\ell - \bar{V}_\ell \bar{U}_\ell^{-1} \bar{V}_\ell^{\mathrm{T}})^{-1}, & \ell \text{ is a fractional}. \end{cases} \tag{7}$$

Based on (7), it follows the recursive relation with the integer ℓ,

$$G_{\ell-1} \approx P_{\ell-\frac{1}{2}} Q_{\ell-\frac{1}{2}} L^{\mathrm{T}}_{\ell-\frac{1}{2}} \begin{bmatrix} \bar{U}^{-1}_{\ell-\frac{1}{2}} & \\ & G_{\ell-\frac{1}{2}} \end{bmatrix} L_{\ell-\frac{1}{2}} Q^{\mathrm{T}}_{\ell-\frac{1}{2}} P^{-1}_{\ell-\frac{1}{2}},$$

$$G_{\ell-\frac{1}{2}} = P_\ell L_\ell^{\mathrm{T}} \begin{bmatrix} U_\ell^{-1} & \\ & G_\ell \end{bmatrix} L_\ell P_\ell^{-1}.$$

Therefore, the hierarchy of Schur complements can be constructed from Level 1. One describes the steps in Algorithm 1. Note that the reindexing is implicitly included in Algorithm 1, when one uses the index sets $I_{\ell;ijk}$ and $J_{\ell;ijk}$ or $I_{\ell;i}$ and $J_{\ell;i}$ for A_ℓ.

2.2 Extracting the Diagonals of the Matrix Inverse

Once the hierarchy of Schur complements is constructed, the next step is to extract the diagonals of the matrix G. It is important to note that computing the entire Schur complement G_ℓ is not required. This is based on the following observations:

$$\begin{cases} G_{\ell-1}\left(I_{\ell;ijk}J_{\ell;ijk}, I_{\ell;ijk}J_{\ell;ijk}\right) \text{ is determined by } G_{\ell-\frac{1}{2}}\left(J_{\ell;ijk}, J_{\ell;ijk}\right), \\ G_{\ell-\frac{1}{2}}\left(I_{\ell-\frac{1}{2};i}J_{\ell-\frac{1}{2};i}, I_{\ell-\frac{1}{2};i}J_{\ell-\frac{1}{2};i}\right) \text{ is determined by } G_\ell\left(J_{\ell-\frac{1}{2};i}, J_{\ell-\frac{1}{2};i}\right). \end{cases} \tag{8}$$

For the purpose of extracting relevant information, we begin with considering the top level $\ell = L = 3$. $G_{5/2}$ can be calculated using the following formula with given G_3:

$$G_{\frac{5}{2}} = P_3 \begin{bmatrix} U_3^{-1} + U_3^{-1} V_3^{\mathrm{T}} G_3 V_3 U_3^{-1} & -U_3^{-1} V_3^{\mathrm{T}} G_3 \\ -G_3 V_3 U_3^{-1} & G_3 \end{bmatrix} P_3^{-1}.$$

The submatrices enclosed in the bracket are indexed by $(I_3|J_3)$, while $G_{5/2}$ is indexed by $J_{5/2} = J_{5/2;1}J_{5/2;2}\cdots J_{5/2;48}$, as a result of the permutation matrix P_3. However, it suffices

Algorithm 1: Constructing the hierarchy of Schur complements of A

Determine $\ell_{\max}$ and decompose the domain hierarchically
Generate index sets $I_{1;ijk}$ and $J_{1;ijk}$
$A_1 \leftarrow A$
for $\ell = 1$ to $\ell_{\max}$ **do**
  $A_{\ell+\frac{1}{2}} \leftarrow A_\ell(J_\ell, J_\ell)$
  for *(i, j, k)*$\in$ {*block index at level* ℓ} **do**
    $U_{\ell;ijk} \leftarrow A_\ell(I_{\ell;ijk}, I_{\ell;ijk})$
    $V_{\ell;ijk} \leftarrow A_\ell(J_{\ell;ijk}, I_{\ell;ijk})$
    Calculate $U^{-1}_{\ell;ijk}$
    Calculate $K_{\ell;ijk} \leftarrow -V_{\ell;ijk}U^{-1}_{\ell;ijk}$
    Calculate $A_{\ell+\frac{1}{2}}(J_{\ell;ijk}, J_{\ell;ijk}) \leftarrow A_{\ell+\frac{1}{2}}(J_{\ell;ijk}, J_{\ell;ijk}) + K_{\ell;ijk}V^T_{\ell;ijk}$
  end
  if $\ell < \ell_{\max}$ **then**
    Skeletonize cubic faces at level $\ell + \frac{1}{2}$
    for $s \in$ {*block index at level* $\ell + \frac{1}{2}$} **do**
      Use ID to compute $T_{\ell+\frac{1}{2};s}$, $I_{\ell+\frac{1}{2};s}$ and $J_{\ell+\frac{1}{2};s}$
      $U_{\ell+\frac{1}{2};s} \leftarrow A_{\ell+\frac{1}{2}}(I_{\ell+\frac{1}{2};s}, I_{\ell+\frac{1}{2};s})$
      $V_{\ell+\frac{1}{2};s} \leftarrow A_{\ell+\frac{1}{2}}(J_{\ell+\frac{1}{2};s}, I_{\ell+\frac{1}{2};s})$
      Calculate $\bar{\ell}_{\ell+\frac{1}{2};s} \leftarrow V^T_{\ell+\frac{1}{2};s}T_{\ell+\frac{1}{2};s}$
      Calculate $\bar{V}_{\ell+\frac{1}{2};s} \leftarrow V_{\ell+\frac{1}{2};s} - A_{\ell+\frac{1}{2}}(J_{\ell+\frac{1}{2};s}, J_{\ell+\frac{1}{2};s})T_{\ell+\frac{1}{2};s}$
      Calculate $\bar{U}_{\ell+\frac{1}{2};s} \leftarrow U_{\ell+\frac{1}{2};s} - \bar{\ell}_{\ell+\frac{1}{2};s} - T^T_{\ell+\frac{1}{2};s}\bar{V}_{\ell+\frac{1}{2};s}$
    end
    $A_{\ell+1} \leftarrow A_{\ell+\frac{1}{2}}(J_{\ell+\frac{1}{2}}, J_{\ell+\frac{1}{2}})$
    for $s \in$ {*block index at level* $\ell + \frac{1}{2}$} **do**
      Calculate $\bar{U}^{-1}_{\ell+\frac{1}{2};s}$
      Calculate $\bar{K}_{\ell+\frac{1}{2};s} \leftarrow -\bar{V}_{\ell+\frac{1}{2};s}\bar{U}^{-1}_{\ell+\frac{1}{2};s}$
      Calculate $A_{\ell+1}(J_{\ell+\frac{1}{2};s}, J_{\ell+\frac{1}{2};s}) \leftarrow A_{\ell+1}(J_{\ell+\frac{1}{2};s}, J_{\ell+\frac{1}{2};s}) + \bar{K}_{\ell+\frac{1}{2};s}\bar{V}^T_{\ell+\frac{1}{2};s}$
    end
    Construct $I_{\ell+1}$ and $J_{\ell+1}$
  end
end
Calculate $G_{\ell_{\max}} \leftarrow A^{-1}_{\ell_{\max}+\frac{1}{2}}$

Output:

$$I_\ell, J_\ell, I_{\ell+\frac{1}{2}}, J_{\ell+\frac{1}{2}}, U^{-1}_{\ell;ijk}, \bar{U}^{-1}_{\ell+\frac{1}{2};s}, K_{\ell;ijk}, \bar{K}_{\ell+\frac{1}{2};s}, G_{\ell_{\max}}, \text{ for each } \ell, i, j, k, s$$

to focus on $G_{5/2}(J_{5/2;i}, J_{5/2;i})$ instead of the off-diagonal blocks to extract the diagonal elements of $G_{5/2}$. As a consequence, we can represent $G_{5/2}$ as

$$G_{\frac{5}{2}} = \begin{bmatrix} G_{\frac{5}{2};1} & * & * & * & * & * \\ * & G_{\frac{5}{2};2} & * & * & * & * \\ * & * & G_{\frac{5}{2};3} & * & * & * \\ * & * & * & G_{\frac{5}{2};4} & * & * \\ * & * & * & * & G_{\frac{5}{2};5} & * \\ * & * & * & * & * & G_{\frac{5}{2};48} \end{bmatrix}$$

with $G_{5/2;i} = G_{5/2}(J_{5/2;i}, J_{5/2;i}), i = 1, 2, \cdots, 48$.

One recovers the elements of diagonal blocks for the matrix $G_{5/2}$ in the previous layer. Furthermore, the diagonal blocks of the following G_2 are acquired based on the observation of (8):

$$G_2 \approx P_{\frac{5}{2}} \begin{bmatrix} \mathcal{H}_2 & -\bar{U}_{\frac{5}{2}}^{-1}\bar{V}_{\frac{5}{2}}^{\mathrm{T}} G_{\frac{5}{2}} + \mathcal{H}_2 T_{\frac{5}{2}}^{\mathrm{T}} \\ -G_{\frac{5}{2}}\bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1} + T_{\frac{5}{2}}\mathcal{H}_2 & \mathfrak{H}_2 \end{bmatrix} P_{\frac{5}{2}}^{-1} \tag{9}$$

with

$$\mathcal{H}_2 = \bar{U}_{5/2}^{-1} + \bar{U}_{5/2}^{-1}\bar{V}_{5/2}^{\mathrm{T}} G_{5/2}\bar{V}_{5/2}\bar{U}_{5/2}^{-1}$$

and

$$\mathfrak{H}_2 = T_{\frac{5}{2}}\mathcal{H}_2 T_{\frac{5}{2}}^{\mathrm{T}} - G_{\frac{5}{2}}\bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1} T_{\frac{5}{2}}^{\mathrm{T}} - T_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1}\bar{V}_{\frac{5}{2}}^{\mathrm{T}} G_{\frac{5}{2}} + G_{\frac{5}{2}}.$$

All matrices in the bracket of (9) are indexed by $(I_{5/2}|J_{5/2})$. G_2 is indexed by

$$J_2 = J_{2;111}J_{2;121}J_{2;211}J_{2;221}J_{2;112}J_{2;122}J_{2;212}J_{2;222}$$

due to the permutation matrix $P_{5/2}$.

Recalling the construction process, one can assert that $T_{5/2}, U_{5/2}^{-1}$, and $V_{5/2}$ are block diagonal matrices and the diagonal blocks of the following matrices are both obtained:

$$\bar{U}_{\frac{5}{2}}^{-1}\bar{V}_{\frac{5}{2}}^{\mathrm{T}} G_{\frac{5}{2}}\bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1} = \begin{bmatrix} \bar{U}_{\frac{5}{2};1}^{-1}\bar{V}_{\frac{5}{2};1}^{\mathrm{T}} G_{\frac{5}{2};1}\bar{V}_{\frac{5}{2};1}\bar{U}_{\frac{5}{2};1}^{-1} & \cdots & * \\ \vdots & & \vdots \\ * & \cdots & \bar{U}_{\frac{5}{2};48}^{-1}\bar{V}_{\frac{5}{2};48}^{\mathrm{T}} G_{\frac{5}{2};48}\bar{V}_{\frac{5}{2};48}\bar{U}_{\frac{5}{2};48}^{-1} \end{bmatrix},$$

$$G_{\frac{5}{2}}\bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1} T_{\frac{5}{2}}^{\mathrm{T}} = \begin{bmatrix} G_{\frac{5}{2};1}\bar{V}_{\frac{5}{2};1}\bar{U}_{\frac{5}{2};1}^{-1} T_{\frac{5}{2};1}^{\mathrm{T}} & \cdots & * \\ \vdots & & \vdots \\ * & \cdots & G_{\frac{5}{2};48}\bar{V}_{\frac{5}{2};48}\bar{U}_{\frac{5}{2};48}^{-1} T_{\frac{5}{2};48}^{\mathrm{T}} \end{bmatrix}.$$

This means that only block-block multiplication is needed to get the elements of the diagonal blocks $\mathcal{H}_2$ and $\mathfrak{H}_2$. The computational complexity is then greatly reduced. Additionally, the diagonal blocks $G_2(J_{2;ijk}, J_{2;ijk})$ are obtained directly.

At Level 2, one has

$$G_{\frac{3}{2}} = P_2 \begin{bmatrix} U_2^{-1} + U_2^{-1}V_2^{\mathrm{T}} G_2 V_2 U_2^{-1} & -U_2^{-1}V_2^{\mathrm{T}} G_2 \\ -G_2 V_2 U_2^{-1} & G_2 \end{bmatrix} P_2^{-1}. \tag{10}$$

Similarly, the submatrices in the bracket of (10) are indexed by $(I_2|J_2)$. $G_{3/2}$ is indexed by $J_{3/2} = J_{3/2;1} \cdots J_{3/2;384}$ with the permutation matrix P_2. Moreover, one just needs to compute $G_{3/2}(J_{3/2;i}, J_{3/2;i})$ in this step.

At Level 3/2, one has

$$G_1 \approx P_{\frac{3}{2}} \begin{bmatrix} \mathcal{H}_1 & -\bar{U}_{\frac{3}{2}}^{-1}\bar{V}_{\frac{3}{2}}^{\mathrm{T}} G_{\frac{3}{2}} + \mathcal{H}_1 T_{\frac{3}{2}}^{\mathrm{T}} \\ -G_{\frac{3}{2}}\bar{V}_{\frac{3}{2}}\bar{U}_{\frac{3}{2}}^{-1} + T_{\frac{3}{2}}\mathcal{H}_1 & \mathfrak{H}_1 \end{bmatrix} P_{\frac{3}{2}}^{-1} \tag{11}$$

with

$$\mathcal{H}_1 = \bar{U}_{\frac{3}{2}}^{-1} + \bar{U}_{\frac{3}{2}}^{-1} \bar{V}_{\frac{3}{2}}^{\mathrm{T}} G_{\frac{3}{2}} \bar{V}_{\frac{3}{2}} \bar{U}_{\frac{3}{2}}^{-1}$$

and

$$\mathfrak{H}_1 = T_{\frac{3}{2}} \mathcal{H}_1 T_{\frac{3}{2}}^{\mathrm{T}} - G_{\frac{3}{2}} \bar{V}_{\frac{3}{2}} \bar{U}_{\frac{3}{2}}^{-1} T_{\frac{3}{2}}^{\mathrm{T}} - T_{\frac{3}{2}} \bar{U}_{\frac{3}{2}}^{-1} \bar{V}_{\frac{3}{2}}^{\mathrm{T}} G_{\frac{3}{2}} + G_{\frac{3}{2}}.$$

The diagonal blocks of $\mathcal{H}_1$ and $\mathfrak{H}_1$ can be efficiently computed using block-block multiplication, just like Level $\ell = 5/2$. The index of the submatrices in the bracket of (11) is $(I_{3/2}|J_{3/2})$. Due to the permutation the matrix $P_{3/2}$, the matrix G_1 is indexed by $J_1 = J_{1;111} J_{1;121} \cdots J_{1;444}$ and all elements needed are the diagonal blocks $G_1(J_{1;ijk}, J_{1;ijk})$.

At the bottom level $\ell = 1$, one applies the same procedure as at Level 2 and Level 3. Specifically, one obtains $G_1(J_{1;ijk}, J_{1;ijk})$ from Level 3/2, while $G(J_{0;ijk}, J_{0;ijk})$ is computed directly. Consequently, the diagonal elements of G can be obtained by combining the diagonal elements of each level.

Finally, a quasilinear scaling algorithm can be implemented to recursively extract the diagonal elements of G. Algorithm 2 outlines the organized procedure for this purpose. It is important to note that the reindexing process is implicitly incorporated within Algorithm 2 through the use of index sets $J_{\ell;ijk}$ or $J_{\ell;i}$ for G_ℓ.

2.3 The SelInvHIF with Edge Skeletonization

In Sect. 2.1, the construction step of the SelInvHIF is characterized by its incorporation of interior points elimination and face skeletonization. The SelInvHIF with edge skeletonization further advances this approach by introducing additional edge skeletonization, enabling complete dimensionality reduction. To recall the construction step in the SelInvHIF, the hierarchy construction of Schur complements is systematically performed at levels 1, 3/2, 2, 5/2, $\cdots$, and L. In contrast, the construction step of the SelInvHIF with edge skeletonization is carried out at levels 1, 4/3, 5/3, 2, 7/3, $\cdots$, and L. Specifically, at each integer level, the points are reindexed and the interior points are eliminated accordingly. Additionally, the face skeletonization is performed at level $\ell + 1/3$, and the edge skeletonization is performed at level $\ell + 2/3$. Precisely, the relationships between levels are defined the same as (7). Furthermore, one obtains the following recursive relation with the integer ℓ:

$$\begin{aligned}
G_{\ell-1} &\approx P_{\ell-\frac{2}{3}} Q_{\ell-\frac{2}{3}} L_{\ell-\frac{2}{3}}^{\mathrm{T}} \begin{bmatrix} \bar{U}_{\ell-\frac{2}{3}}^{-1} & \\ & G_{\ell-\frac{2}{3}} \end{bmatrix} L_{\ell-\frac{2}{3}} Q_{\ell-\frac{2}{3}}^{\mathrm{T}} P_{\ell-\frac{2}{3}}^{-1}, \\
G_{\ell-\frac{2}{3}} &\approx P_{\ell-\frac{1}{3}} Q_{\ell-\frac{1}{3}} L_{\ell-\frac{1}{3}}^{\mathrm{T}} \begin{bmatrix} \bar{U}_{\ell-\frac{1}{3}}^{-1} & \\ & G_{\ell-\frac{1}{3}} \end{bmatrix} L_{\ell-\frac{1}{3}} Q_{\ell-\frac{1}{3}}^{\mathrm{T}} P_{\ell-\frac{1}{3}}^{-1}, \\
G_{\ell-\frac{1}{3}} &= P_\ell L_\ell^{\mathrm{T}} \begin{bmatrix} U_\ell^{-1} & \\ & G_\ell \end{bmatrix} L_\ell P_\ell^{-1}.
\end{aligned}$$

Similar to Sect. 2.2, one can extract the diagonals of the matrix G based on the hierarchy of Schur complements. The following observations shows that computing the entire G_ℓ is not required:

Algorithm 2: Extracting the diagonals of A^{-1}

Input:
Output of Algorithm 1

for $\ell = \ell_{\max}$ to 1 **do**
  for $(i, j, k) \in$ {*block index at level* ℓ} **do**
    Calculate $G_{\ell-\frac{1}{2}}(I_{\ell;ijk}, I_{\ell;ijk}) \leftarrow U^{-1}_{\ell;ijk} + K^T_{\ell;ijk} G_\ell(J_{\ell;ijk}, J_{\ell;ijk}) K_{\ell;ijk}$
    Calculate $G_{\ell-\frac{1}{2}}(J_{\ell;ijk}, I_{\ell;ijk}) \leftarrow G_\ell(J_{\ell;ijk}, J_{\ell;ijk}) K_{\ell;ijk}$
    $G_{\ell-\frac{1}{2}}(I_{\ell;ijk}, J_{\ell;ijk}) \leftarrow G_{\ell-\frac{1}{2}}(J_{\ell;ijk}, I_{\ell;ijk})^T$
    $G_{\ell-\frac{1}{2}}(J_{\ell;ijk}, J_{\ell;ijk}) \leftarrow G_\ell(J_{\ell;ijk}, J_{\ell;ijk})$
  end
  if $\ell > 1$ **then**
    for $s \in \{$*block index at level* $\ell - \frac{1}{2}\}$ **do**
      Calculate $G_{\ell-1}(I_{\ell-\frac{1}{2};s}, I_{\ell-\frac{1}{2};s}) \leftarrow \bar{U}^{-1}_{\ell-\frac{1}{2};s} + \bar{K}^T_{\ell-\frac{1}{2};s} G_{\ell-\frac{1}{2}}(J_{\ell-\frac{1}{2};s}, J_{\ell-\frac{1}{2};s}) \bar{K}_{\ell-\frac{1}{2};s}$
      Calculate $\bar{W}_{\ell-1}(J_{\ell-\frac{1}{2};s}, I_{\ell-\frac{1}{2};s}) \leftarrow G_{\ell-\frac{1}{2}} \bar{K}_{\ell-\frac{1}{2};s}$
      $G_{\ell-1}(J_{\ell-\frac{1}{2};s}, I_{\ell-\frac{1}{2};s}) \leftarrow \bar{W}_{\ell-1}(J_{\ell-\frac{1}{2};s}, I_{\ell-\frac{1}{2};s}) + T_{\ell-\frac{1}{2};s} G_{\ell-1}(I_{\ell-\frac{1}{2};s}, I_{\ell-\frac{1}{2};s})$
      $G_{\ell-1}(I_{\ell-\frac{1}{2};s}, J_{\ell-\frac{1}{2};s}) \leftarrow G_{\ell-1}(J_{\ell-\frac{1}{2};s}, I_{\ell-\frac{1}{2};s})^T$
      $G_{\ell-1}(J_{\ell-\frac{1}{2};s}, J_{\ell-\frac{1}{2};s}) \leftarrow$
        $G_{\ell-1}(J_{\ell-\frac{1}{2};s}, I_{\ell-\frac{1}{2};s}) T^T_{\ell-\frac{1}{2};s} + T_{\ell-\frac{1}{2};s} \bar{W}_{\ell-1}(J_{\ell-\frac{1}{2};s}, I_{\ell-\frac{1}{2};s}) + G_{\ell-\frac{1}{2}}(J_{\ell-\frac{1}{2};s}, J_{\ell-\frac{1}{2};s})$
    end
  end
end

$$\begin{cases} G_{\ell-1}\left(I_{\ell;ijk} J_{\ell;ijk}, I_{\ell;ijk} J_{\ell;ijk}\right) \text{ is determined by } G_{\ell-\frac{2}{3}}\left(J_{\ell;ijk}, J_{\ell;ijk}\right), \\ G_{\ell-\frac{2}{3}}\left(I_{\ell-\frac{2}{3};i} J_{\ell-\frac{2}{3};i}, I_{\ell-\frac{2}{3};i} J_{\ell-\frac{2}{3};i}\right) \text{ is determined by } G_{\ell-\frac{1}{3}}\left(J_{\ell-\frac{2}{3};i}, J_{\ell-\frac{2}{3};i}\right), \\ G_{\ell-\frac{1}{3}}\left(I_{\ell-\frac{1}{3};i} J_{\ell-\frac{1}{3};i}, I_{\ell-\frac{1}{3};i} J_{\ell-\frac{1}{3};i}\right) \text{ is determined by } G_{\ell}\left(J_{\ell-\frac{1}{3};i}, J_{\ell-\frac{1}{3};i}\right). \end{cases} \tag{12}$$

Based on (12), the recovery process for G can be carried out as in Sect. 2.2. Hence, the construction step and extracting step of the SelInvHIF with edge skeletonization can be described in Algorithms 3 and 4.

2.4 Computational Complexity

In this section, we analyze the computational complexity of the SelInvHIF algorithm. Assume that the domain consists of $N = \sqrt[3]{N} \times \sqrt[3]{N} \times \sqrt[3]{N}$ points and set $\sqrt[3]{N} = 2^L$ with $\ell_{\max} < L$. The number of blocks at level ℓ is defined as $n_B(\ell)$, and the following formula holds:

$$n_B(\ell) = O(2^{3(\ell_{\max} - \ell)}).$$

The number of points in each block, whether it is a cubic face or a cubic edge, is denoted as $n_P(\ell)$. It is important to note that the interior or redundant points from the previous level are not included in $n_P(\ell)$ because they have already been eliminated in previous levels. To estimate $n_P(\ell)$, we rely on the assumption made in [15] regarding the skeletonization. According to this assumption, the typical size of the skeleton is given by

Algorithm 3 The hierarchy of Schur complements of A with edge skeletonization

Determine $\ell_{\max}$, the hierarchical structure of domain, and index sets $I_{1;ijk}$ and $J_{1;ijk}$
for $\ell = 1$ to $\ell_{\max}$ **do**
  $A_{\ell+\frac{1}{3}} \leftarrow A_\ell(J_\ell, J_\ell)$
  for $(i, j, k) \in$ {*block index at level* ℓ} **do**
    $U_{\ell;ijk} \leftarrow A_\ell(I_{\ell;ijk}, I_{\ell;ijk})$, $V_{\ell;ijk} \leftarrow A_\ell(J_{\ell;ijk}, I_{\ell;ijk})$
    Calculate $U^{-1}_{\ell;ijk}$, $K_{\ell;ijk} \leftarrow -V_{\ell;ijk}U^{-1}_{\ell;ijk}$
    Calculate $A_{\ell+\frac{1}{3}}(J_{\ell;ijk}, J_{\ell;ijk}) \leftarrow A_{\ell+\frac{1}{3}}(J_{\ell;ijk}, J_{\ell;ijk}) + K_{\ell;ijk}V^{\mathrm{T}}_{\ell;ijk}$
  end
  if $\ell < \ell_{\max}$ **then**
    Skeletonize cubic faces at level $\ell + \frac{1}{3}$
    for $s \in$ {*block index at level* $\ell + \frac{1}{3}$} **do**
      Use ID to compute $T_{\ell+\frac{1}{3};s}$, $I_{\ell+\frac{1}{3};s}$, and $J_{\ell+\frac{1}{3};s}$
      $U_{\ell+\frac{1}{3};s} \leftarrow A_{\ell+\frac{1}{3}}(I_{\ell+\frac{1}{3};s}, I_{\ell+\frac{1}{3};s})$, $V_{\ell+\frac{1}{3};s} \leftarrow A_{\ell+\frac{1}{3}}(J_{\ell+\frac{1}{3};s}, I_{\ell+\frac{1}{3};s})$
      Calculate $\bar{\ell}_{\ell+\frac{1}{3};s} \leftarrow V^{\mathrm{T}}_{\ell+\frac{1}{3};s}T_{\ell+\frac{1}{3};s}$
      Calculate $\bar{V}_{\ell+\frac{1}{3};s} \leftarrow V_{\ell+\frac{1}{3};s} - A_{\ell+\frac{1}{3}}(J_{\ell+\frac{1}{3};s}, J_{\ell+\frac{1}{3};s})T_{\ell+\frac{1}{3};s}$
      Calculate $\bar{U}_{\ell+\frac{1}{3};s} \leftarrow U_{\ell+\frac{1}{3};s} - \bar{\ell}_{\ell+\frac{1}{3};s} - T^T_{\ell+\frac{1}{3};s}\bar{V}_{\ell+\frac{1}{3};s}$
    end
    $A_{\ell+\frac{2}{3}} \leftarrow A_{\ell+\frac{1}{3}}(J_{\ell+\frac{1}{3}}, J_{\ell+\frac{1}{3}})$
    for $s \in$ {*block index at level* $\ell + \frac{1}{3}$} **do**
      Calculate $\bar{U}^{-1}_{\ell+\frac{1}{3};s}$, $\bar{K}_{\ell+\frac{1}{3};s} \leftarrow -\bar{V}_{\ell+\frac{1}{3};s}\bar{U}^{-1}_{\ell+\frac{1}{3};s}$
      Calculate $A_{\ell+1}(J_{\ell+\frac{1}{3};s}, J_{\ell+\frac{1}{3};s}) \leftarrow A_{\ell+1}(J_{\ell+\frac{1}{3};s}, J_{\ell+\frac{1}{3};s}) + \bar{K}_{\ell+\frac{1}{3};s}\bar{V}^{\mathrm{T}}_{\ell+\frac{1}{3};s}$
    end
  end
  if $\ell < \ell_{\max}$ **then**
    Skeletonize cubic edges at level $\ell + \frac{2}{3}$
    for $s \in$ {*block index at level* $\ell + \frac{2}{3}$} **do**
      Use ID to compute $T_{\ell+\frac{2}{3};s}$, $I_{\ell+\frac{2}{3};s}$, and $J_{\ell+\frac{2}{3};s}$
      $U_{\ell+\frac{2}{3};s} \leftarrow A_{\ell+\frac{2}{3}}(I_{\ell+\frac{2}{3};s}, I_{\ell+\frac{2}{3};s})$, $V_{\ell+\frac{2}{3};s} \leftarrow A_{\ell+\frac{2}{3}}(J_{\ell+\frac{2}{3};s}, I_{\ell+\frac{2}{3};s})$
      Calculate $\bar{\ell}_{\ell+\frac{2}{3};s} \leftarrow V^{\mathrm{T}}_{\ell+\frac{2}{3};s}T_{\ell+\frac{2}{3};s}$
      Calculate $\bar{V}_{\ell+\frac{2}{3};s} \leftarrow V_{\ell+\frac{2}{3};s} - A_{\ell+\frac{2}{3}}(J_{\ell+\frac{2}{3};s}, J_{\ell+\frac{2}{3};s})T_{\ell+\frac{2}{3};s}$
      Calculate $\bar{U}_{\ell+\frac{2}{3};s} \leftarrow U_{\ell+\frac{2}{3};s} - \bar{\ell}_{\ell+\frac{2}{3};s} - T^T_{\ell+\frac{2}{3};s}\bar{V}_{\ell+\frac{2}{3};s}$
    end
    $A_{\ell+1} \leftarrow A_{\ell+\frac{2}{3}}(J_{\ell+\frac{2}{3}}, J_{\ell+\frac{2}{3}})$
    for $s \in$ {*block index at level* $\ell + \frac{2}{3}$} **do**
      Calculate $\bar{U}^{-1}_{\ell+\frac{2}{3};s}$, $\bar{K}_{\ell+\frac{2}{3};s} \leftarrow -\bar{V}_{\ell+\frac{2}{3};s}\bar{U}^{-1}_{\ell+\frac{2}{3};s}$
      Calculate $A_{\ell+1}(J_{\ell+\frac{2}{3};s}, J_{\ell+\frac{2}{3};s}) \leftarrow A_{\ell+1}(J_{\ell+\frac{2}{3};s}, J_{\ell+\frac{2}{3};s}) + \bar{K}_{\ell+\frac{2}{3};s}\bar{V}^{\mathrm{T}}_{\ell+\frac{2}{3};s}$
    end
  end
end
Output: $I_\ell, J_\ell, I_{\ell+\frac{1}{3}}, J_{\ell+\frac{1}{3}}, I_{\ell+\frac{2}{3}}, J_{\ell+\frac{2}{3}}, U^{-1}_{\ell;ijk}, \bar{U}^{-1}_{\ell+\frac{1}{3};s}, \bar{U}^{-1}_{\ell+\frac{2}{3};s}, K_{\ell;ijk}, \bar{K}_{\ell+\frac{1}{3};s}, \bar{K}^{-1}_{\ell+\frac{2}{3};s}, G_{\ell_{\max}}$

$$n_P(\ell) = \begin{cases} O(\ell) & \text{for edges;} \\ O(2^\ell) & \text{for faces.} \end{cases}$$

Algorithm 4 Extracting the diagonals of A^{-1} with edge skeletonization

Input:

Output of Algorithm 3

for $\ell = \ell_{\max}$ to 1 **do**
  for $(i, j, k) \in \{$*block index at level* $\ell\}$ **do**
    Calculate $G_{\ell-\frac{1}{3}}(I_{\ell;ijk}, I_{\ell;ijk}) \leftarrow U^{-1}_{\ell;ijk} + K^{\mathrm{T}}_{\ell;ijk} G_\ell(J_{\ell;ijk}, J_{\ell;ijk}) K_{\ell;ijk}$
    Calculate $G_{\ell-\frac{1}{3}}(J_{\ell;ijk}, I_{\ell;ijk}) \leftarrow G_\ell(J_{\ell;ijk}, J_{\ell;ijk}) K_{\ell;ijk}$
    $G_{\ell-\frac{1}{3}}(I_{\ell;ijk}, J_{\ell;ijk}) \leftarrow G_{\ell-\frac{1}{3}}(J_{\ell;ijk}, I_{\ell;ijk})^{\mathrm{T}}$
    $G_{\ell-\frac{1}{3}}(J_{\ell;ijk}, J_{\ell;ijk}) \leftarrow G_\ell(J_{\ell;ijk}, J_{\ell;ijk})$
  end
  if $\ell > 1$ **then**
    for $s \in \{$*block index at level* $\ell - \frac{1}{3}\}$ **do**
      Calculate $G_{\ell-\frac{2}{3}}(I_{\ell-\frac{1}{3};s}, I_{\ell-\frac{1}{3};s}) \leftarrow \bar{U}^{-1}_{\ell-\frac{1}{3};s} + \bar{K}^{\mathrm{T}}_{\ell-\frac{1}{3};s} G_{\ell-\frac{1}{3}}(J_{\ell-\frac{1}{3};s}, J_{\ell-\frac{1}{3};s}) \bar{K}_{\ell-\frac{1}{3};s}$
      Calculate $\bar{W}_{\ell-\frac{2}{3}}(J_{\ell-\frac{1}{3};s}, I_{\ell-\frac{1}{3};s}) \leftarrow G_{\ell-\frac{1}{3}} \bar{K}_{\ell-\frac{1}{3};s}$
      $G_{\ell-\frac{2}{3}}(J_{\ell-\frac{1}{3};s}, I_{\ell-\frac{1}{3};s}) \leftarrow \bar{W}_{\ell-1}(J_{\ell-\frac{1}{3};s}, I_{\ell-\frac{1}{3};s}) + T_{\ell-\frac{1}{3};s} G_{\ell-\frac{2}{3}}(I_{\ell-\frac{1}{3};s}, I_{\ell-\frac{1}{3};s})$
      $G_{\ell-\frac{2}{3}}(I_{\ell-\frac{1}{3};s}, J_{\ell-\frac{1}{3};s}) \leftarrow G_{\ell-\frac{2}{3}}(J_{\ell-\frac{1}{3};s}, I_{\ell-\frac{1}{3};s})^{\mathrm{T}}$
      $G_{\ell-\frac{2}{3}}(J_{\ell-\frac{1}{3};s}, J_{\ell-\frac{1}{3};s}) \leftarrow$
        $G_{\ell-\frac{2}{3}}(J_{\ell-\frac{1}{3};s}, I_{\ell-\frac{1}{3};s}) T^{\mathrm{T}}_{\ell-\frac{1}{3};s} + T_{\ell-\frac{1}{3};s} \bar{W}_{\ell-1}(J_{\ell-\frac{1}{3};s}, I_{\ell-\frac{1}{3};s}) + G_{\ell-\frac{1}{3}}(J_{\ell-\frac{1}{3};s}, J_{\ell-\frac{1}{3};s})$
    end
  end
  if $\ell > 1$ **then**
    for $s \in \{$*block index at level* $\ell - \frac{2}{3}\}$ **do**
      Calculate $G_{\ell-1}(I_{\ell-\frac{2}{3};s}, I_{\ell-\frac{2}{3};s}) \leftarrow \bar{U}^{-1}_{\ell-\frac{2}{3};s} + \bar{K}^{\mathrm{T}}_{\ell-\frac{2}{3};s} G_{\ell-\frac{2}{3}}(J_{\ell-\frac{2}{3};s}, J_{\ell-\frac{2}{3};s}) \bar{K}_{\ell-\frac{2}{3};s}$
      Calculate $\bar{W}_{\ell-1}(J_{\ell-\frac{2}{3};s}, I_{\ell-\frac{2}{3};s}) \leftarrow G_{\ell-\frac{2}{3}} \bar{K}_{\ell-\frac{2}{3};s}$
      $G_{\ell-1}(J_{\ell-\frac{2}{3};s}, I_{\ell-\frac{2}{3};s}) \leftarrow \bar{W}_{\ell-1}(J_{\ell-\frac{2}{3};s}, I_{\ell-\frac{2}{3};s}) + T_{\ell-\frac{2}{3};s} G_{\ell-1}(I_{\ell-\frac{2}{3};s}, I_{\ell-\frac{2}{3};s})$
      $G_{\ell-1}(I_{\ell-\frac{2}{3};s}, J_{\ell-\frac{2}{3};s}) \leftarrow G_{\ell-1}(J_{\ell-\frac{2}{3};s}, I_{\ell-\frac{2}{3};s})^{\mathrm{T}}$
      $G_{\ell-1}(J_{\ell-\frac{2}{3};s}, J_{\ell-\frac{2}{3};s}) \leftarrow$
        $G_{\ell-1}(J_{\ell-\frac{2}{3};s}, I_{\ell-\frac{2}{3};s}) T^{\mathrm{T}}_{\ell-\frac{2}{3};s} + T_{\ell-\frac{2}{3};s} \bar{W}_{\ell-1}(J_{\ell-\frac{2}{3};s}, I_{\ell-\frac{2}{3};s}) + G_{\ell-\frac{2}{3}}(J_{\ell-\frac{2}{3};s}, J_{\ell-\frac{2}{3};s})$
    end
  end
end

Remark 1 From the complexity analysis, the size of the skeleton is independent of the tolerance of the ID approximation. In fact, the rank of the ID step should not be excessively large relative to the matrix size N. A large rank is equivalent to retaining a large number of redundant points rather than skeleton points. In such case, only the operations of integer levels are effective and the operations of fraction levels are skipped, leading to a computational complexity of $O(N^2)$. That is, the SelInvHIF tends to the original selected inversion [20] for which the solution is exact.

Remark 2 In 3D problems, the DOFs naturally have more interactions than those in 2D problems. Dimensionality affects the typical skeleton size in each level and thus the overall computational complexity. The typical skeleton size of 2D problems is $O(\ell)$, resulting in a linear computational complexity. In 3D problems, the typical skeleton size is $O(2^\ell)$ with face skeletonization, leading to the quasi-linear computational complexity.

Let us examine the construction step of the SelInvHIF, which involves the following steps in Algorithm 1. At the integer level ℓ, one computes $U^{-1}_{\ell;ijk}$ (Line 9) for each block. One then multiplies the inverse with $V_{\ell;ijk}$ to obtain $K_{\ell;ijk}$ (Line 10) and update the new $A_{\ell+1/2}(J_{\ell;ijk}, J_{\ell;ijk})$ (Line 11). At the fractional level $\ell + 1/2$, the $T_{\ell+1/2;k}$ is recorded by ID for each cell (Line 16). The cost for this step is $O(n_P(\ell)^3)$ since each cell only interacts with a constant number of cells. Then, one applies it (Lines 19, 20, and 21) and multiplies the inverse of $\bar{U}_{\ell+1/2;k}$ (Line 25) with $\bar{V}_{\ell+1/2;k}$ to obtain $\bar{K}_{\ell+1/2;k}$ (Line 26). Finally, one update $A_{\ell+1}(J_{\ell+1/2;k}, J_{\ell+1/2;k})$ (Line 27). Thus, the computational cost for these steps at each level is $O(n_P(\ell)^3)$. The total computational complexity is

$$\sum_{\ell=1}^{\ell_{\max}} n_B(\ell) n_P(\ell)^3 \leqslant C \sum_{\ell=1}^{\ell_{\max}} 2^{3\ell_{\max}-3\ell} 2^{3\ell} \leqslant C_0 N \log N,$$

where C and C_0 are constant. The total computational cost for the construction step is $O(N \log N)$ with $\ell_{\max} = O(L)$.

Furthermore, the extraction phase is considered, and the following steps are outlined in Algorithm 2. At the integer level ℓ, one can calculate $G_{\ell-1/2}(I_{\ell;ijk}, I_{\ell;ijk})$ (Line 3) and $G_{\ell-1/2}(J_{\ell;ijk}, I_{\ell;ijk})$ (Line 4) for each block. At the fractional level $\ell - 1/2$, $G_{\ell-1}(I_{\ell-1/2;k}, I_{\ell-1/2;k})$ (Line 10), $G_{\ell-1}(J_{\ell-1/2;k}, I_{\ell-1/2;k})$ (Line 12), and $G_{\ell-1}(J_{\ell-1/2;k}, J_{\ell-1/2;k})$ (Line 14) are calculated for each cell. The computational cost for these steps at each level is $O(n_P(\ell)^3)$. Remarkably, it turns out that the complexity for the extraction phase is also $O(N \log N)$.

As for the computational complexity of the SelInvHIF with edge skeletonization, the cost for each step at each level remains $n_B(\ell) n_P(\ell)^3$. Consequently, the total computational complexity is determined by summing the cost over all levels:

$$\sum_{\ell=1}^{\ell_{\max}} n_B(\ell) n_P(\ell)^3 \leqslant C \sum_{\ell=1}^{\ell_{\max}} 2^{3\ell_{\max}-3\ell} \ell^3 \leqslant C_0 N,$$

where C and C_0 are constant. That is, the total computational cost is $O(N)$ with $\ell_{\max} = O(L)$.

Finally, the quasi-linear scaling of the SelInvHIF and the linear scaling of the SelInvHIF with edge skeletonization are demonstrated. While the SelInvHIF with edge skeletonization can achieve the theoretical $O(N)$ complexity, there is some fill-in generated after edge skeletonization, leading to additional computational cost. This indicates that the optimal complexity can be achieved only if N is sufficiently large, which presents challenge in directly applying it to the MPB equation of interest. As a result, only the SelInvHIF without edge skeletonization is employed in all subsequent numerical examples.

3 Numerical Method for MPB Equations

In this section, the iterative solver is proposed to solve the MPB equations with the effect of Coulomb correlation [39]. The governing equations for the whole space are obtained by the Gaussian variational field theory [30, 32] and expressed by

$$\begin{cases} \nabla \cdot \eta(\boldsymbol{r}) \nabla \phi - \chi \Lambda \mathrm{e}^{-\Xi c(\boldsymbol{r})/2} \sinh \phi = -2\rho_f(\boldsymbol{r}), \\ \left[\nabla \cdot \eta(\boldsymbol{r}) \nabla - \chi \Lambda \mathrm{e}^{-\Xi c(\boldsymbol{r})/2} \cosh \phi\right] G(\boldsymbol{r}, \boldsymbol{r}') = -4\pi \delta(\boldsymbol{r} - \boldsymbol{r}') \end{cases} \tag{13}$$

with the potential ϕ, the relative dielectric function $\eta(\boldsymbol{r})$, the density of fixed charge $\rho_f(\boldsymbol{r})$, and the Green's function $G(\boldsymbol{r},\boldsymbol{r}')$. The coupling parameter Ξ and the rescaled fugacity Λ are given for specific problems. The function $\chi(\boldsymbol{r})$ is defined as 1 to represent the region that is accessible for ions, while it is defined as 0 elsewhere. The correlation function $c(\boldsymbol{r})$ reads

$$c(\boldsymbol{r}) = \lim_{\boldsymbol{r}'\to\boldsymbol{r}} \left[G(\boldsymbol{r},\boldsymbol{r}') - 1/\eta(\boldsymbol{r})|\boldsymbol{r}-\boldsymbol{r}'|\right].$$

To solve the partial differential equations (13), a self-consistent iterative scheme is employed, following [39]. This scheme consists of two alternating steps: first, given a $c(\boldsymbol{r})$, the MPB equation (the first equation) is solved to obtain the potential ϕ with given boundary conditions. Second, for a given $c(\boldsymbol{r})$ and ϕ, the GDH equation (the second equation) is solved to determine the G and obtain a new $c(\boldsymbol{r})$. These two steps are iterated until the solution reaches the desired convergence criteria. The iterative scheme is mathematically expressed [30, 31]:

$$\begin{cases} \nabla\cdot\eta(\boldsymbol{r})\nabla\phi^{(k+1)} - \Lambda e^{-\frac{\Xi c^{(k)}}{2}}\sinh\phi^{(k+1)} = -2\rho_f(\boldsymbol{r}), \\ \left[\nabla\cdot\eta(\boldsymbol{r})\nabla - \Lambda e^{-\frac{\Xi c^{(k)}}{2}}\cosh\phi^{(k+1)}\right]G^{(k+1)} = -4\pi\delta(\boldsymbol{r}-\boldsymbol{r}'), \\ c^{(k+1)}(\boldsymbol{r}) = \lim\limits_{\boldsymbol{r}\to\boldsymbol{r}'}\left[G^{(k+1)}(\boldsymbol{r},\boldsymbol{r}') - 1/\eta(\boldsymbol{r})|\boldsymbol{r}-\boldsymbol{r}'|\right], \end{cases} \tag{14}$$

where the superscript $k = 0, 1, \cdots, K$ indicates the kth iteration step.

In addition, the PB steps can be efficiently solved using standard direct solvers. However, the key to solving a self-consistent equation lies in the GDH equation, which can be reformulated during a self-consistent iteration. In three dimensions, a finite difference approximation of the equation is employed to obtain the following algebraic equation:

$$\boldsymbol{AG} = \boldsymbol{I}, \tag{15}$$

where the matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \frac{h^3}{4\pi}\left[\boldsymbol{DHD}^{\mathrm{T}} + \mathrm{diag}\{P\}\right], \tag{16}$$

with P being the vector of function $p(\boldsymbol{r})$, $\boldsymbol{G}$ representing the lattice Green's function, $\boldsymbol{D}$ being the difference matrices of operators ∇, and $\boldsymbol{I}$ being the unit matrix. It can be observed that solving for the Green's function is equivalent to performing matrix inversion, $\boldsymbol{G} = \boldsymbol{A}^{-1}$. However, directly computing the inverse of the matrix is computationally complex and unnecessary. In this case, only diag($\boldsymbol{G}$) is needed to obtain the self-energy $c(\boldsymbol{r})$. Thus, the SelInvHIF is employed to efficiently extract the diagonals of the inverse.

4 Numerical Results

To assess the effectiveness of the SelInvHIF, we present numerical results for the MPB equations in three dimensions. The scaling of the computational time is of particular interest. We set the coupling parameter $\Xi = 1$ and the uniform fugacity parameter $\Lambda = 0.05$. For both the PB and the self-consistent iterations, the error criteria are set at 10^{-8}. The initial value for the potential in the iteration is always constant in our instances with $\phi^{(0)} = 0$. It is important to note that the choice regarding the ID step depends on the problem specified.

Table 1 The CPU time and accuracy as function of the matrix size. The SelInvHIF time means the execution time spent for one step SelInvHIF. The rank in the ID step is 37

$\sqrt[3]{N}$	SelInvHIF time (s)	E_a	E_r
48	2.0E+3	6.5E−3	2.7E−2
64	4.8E+3	8.1E−3	3.4E−2
80	1.0E+4	9.2E−3	3.8E−2
96	2.5E+4	9.8E−3	4.0E−2
128	5.8E+4	−	−

Both the PB and the GDH stages make use of Dirichlet boundary conditions. The calculation is executed on a machine with Intel Xeon 2.2 GHz and 2.2 TB memory. All experiments are performed in Matlab with the FLAM package [18] for hierarchical matrices. Prior to resolving the MPB equation, we begin by demonstrating an illustrative case of evaluating the diagonal elements of an elliptic differential operator. The statistical computation time is computed as the average of five measurements.

Example 1 (The discrete elliptic differential operator) In our examination of 3D problems, we first explore the diagonals of the inverse of a discrete elliptic differential operator. This is achieved through the implementation of a seven-point stencil discretization. Subsequently, one computes the diagonals of the inverse matrix utilizing both the SelInvHIF method and the exact approach as described in [20]. The diagonals of the inverse of the discrete operator are set as d_s and d_e, respectively. Table 1 presents the absolute L^2 error $E_a = \sqrt{\sum (d_s - d_e)^2 / N}$ between the numerical results and the reference solution, which corresponds to the matrix size as determined by the exact method. Additionally, Table 1 highlights the relative L^2 errors $E_r = \|d_s - d_e\|_2 / \|d_e\|_2$, serving to validate the accuracy of the SelInvHIF. Furthermore, the computational time of the algorithm is displayed in Table 1, while Fig. 3 confirms the quasi-linear scaling of the SelInvHIF. Table 2 illustrates the relationship between the rank of the ID step and the numerical error. This relationship demonstrates that the error can be effectively controlled as the rank of the ID step increases.

Example 2 (The charge density with a delta function) In this example, we consider discontinuous charged distribution in a region $[0, L]^3$ with $L = 32$. Let the fixed charge density be a face charge:

$$\rho_f(x) = \delta(x - L/2).$$

The results of the MPB equations are calculated using the SelInvHIF algorithm. Figure 4 visualizes the distribution of the potential in this system at $z = L/2$ with different matrix sizes $N = 16^3, 32^3, 48^3$, and 64^3. The potential with respect to $x = L/2$ and $y = L/2$ remains symmetric due to the symmetry of the fixed charge and is most pronounced at $x = L/2$ due to the presence of the charge. Table 3 shows the accuracy of the whole algorithm to compute the potential ϕ compared to a reference potential computed with a

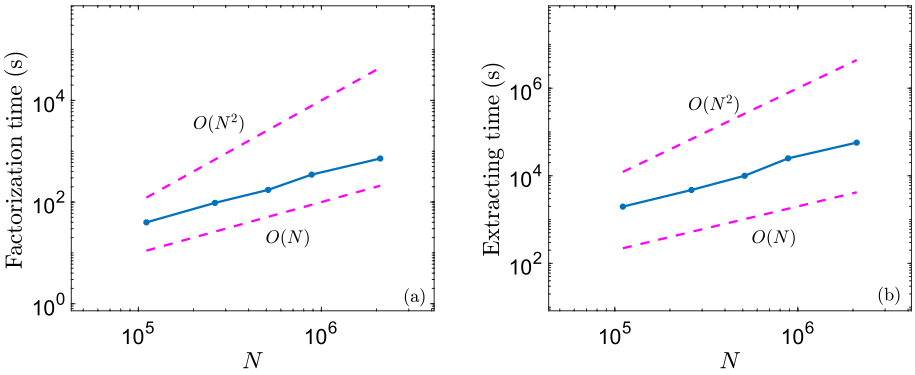

Fig. 3 One step time for the SelInvHIF. The solid lines represent the computational time for the factorization step (**a**) and the extracting step (**b**). The dash lines represent the reference scalings

Table 2 Numerical errors for different ranks of the ID step

The rank of ID step	E_a	E_r
32	3.5E−3	9.5E−2
128	6.5E−4	8.1E−3
256	7.2E−8	9.2E−7
512	5.3E−16	9.8E−15

sufficiently large grid size $N = 64^3$, which verifies the approximate accuracy of first order due to the discontinuous of the derivative of the potential at $y = z = L/2$. Additionally, the table includes the computational time of the algorithm, demonstrating the quasi-linear scaling of the SelInvHIF.

Example 3 (The charge density with continuous function) In this example, we consider discontinuous charged distribution in a region $[0, L]^3$ with $L = 32$. Let the fixed charges density be

$$\rho_f(x) = \sin(\pi x/L).$$

The results of the MPB equations are then computed using the SelInvHIF, where the precision of the ID step is set to be 10^{-8}. Figure 5 visualizes the distribution of the potential in this system at $x = L/2$ with different matrix sizes $N = 16^3, 32^3, 48^3$, and 64^3, which displays the convergence. The potential is most evident at $x = L/2$ due to the dense charge. Table 4 shows the accuracy of the whole algorithm to compute the potential ϕ compared to a reference potential computed with a sufficiently large grid size $N = 64^3$, which verifies the convergence of our algorithm. Furthermore, Table 4 also shows the computational time of the algorithm to verify the quasilinear scaling of the SelInvHIF.

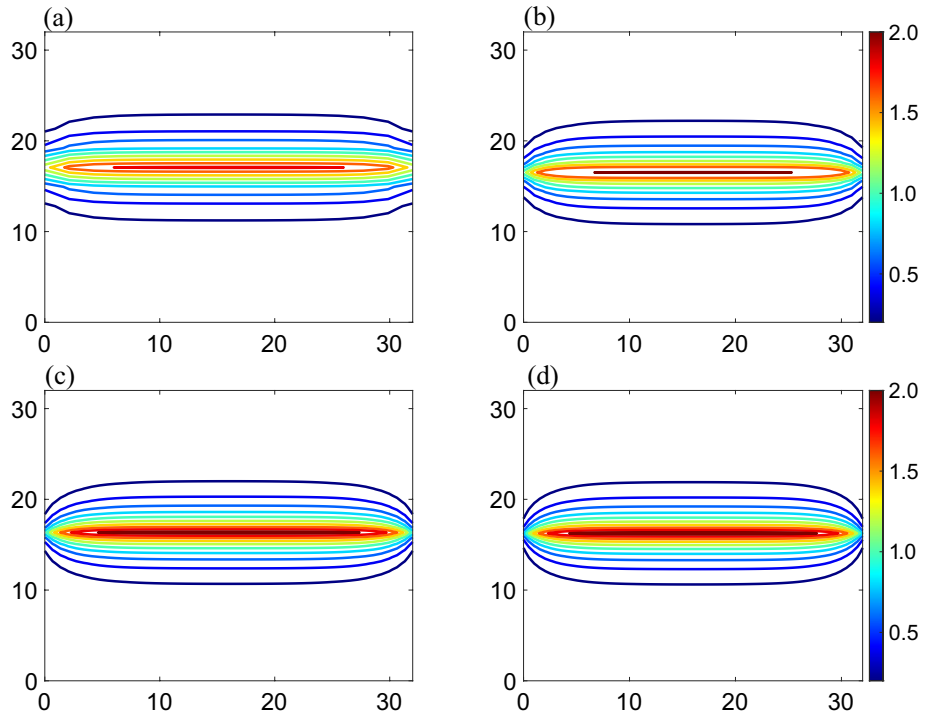

Fig. 4 Potential distributions with different matrix sizes at $z = L/2$. **a** $N = 16$; **b** $N = 32$; **c** $N = 48$; **d** $N = 64$

Table 3 The CPU time and accuracy as function of the matrix size. The total time and the SelInvHIF time mean the execution time spent for one step iteration in the whole program and the time for one step SelInvHIF, respectively

$\sqrt[3]{N}$	Total time (s)	SelInvHIF time (s)	L_∞
16	4.02E+0	2.83E+0	5.56E−1
32	2.12E+2	1.89E+2	3.23E−1
48	2.08E+3	1.95E+3	1.33E−1
64	7.74E+3	6.92E+3	–

5 Conclusions

This paper develops the SelInvHIF, a fast algorithm to solve the MPB equations with the effect of Coulomb correlation. The SelInvHIF effectively integrates hierarchical interpolative factorization and selected inverse techniques to achieve the $O(N \log N)$ computational

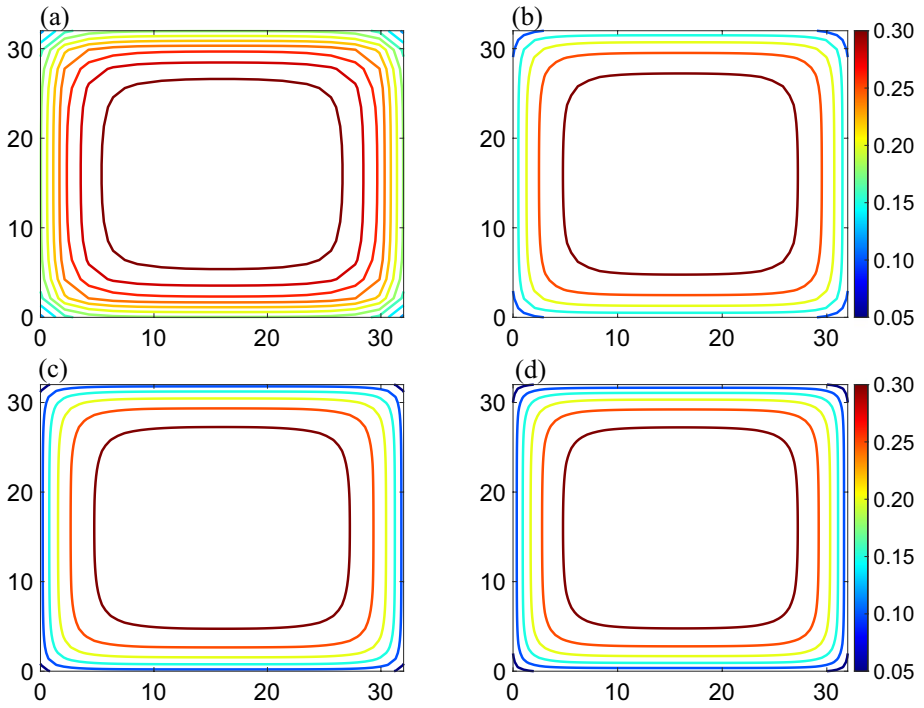

Fig. 5 Potential distributions with different matrix sizes at $z = L/2$. **a** $N = 16$; **b** $N = 32$; **c** $N = 48$; **d** $N = 64$

Table 4 The CPU time and accuracy as function of the matrix size. The total time and the SelInvHIF time mean the execution time spent for one step iteration in the whole program and the time for one step SelInvHIF, respectively

$\sqrt[3]{N}$	Total time (s)	SelInvHIF time (s)	L_∞
16	2.75E+1	2.35E+1	2.12E−2
32	3.06E+2	2.47E+2	3.84E−3
48	2.80E+3	2.40E+3	1.92E−3
64	1.06E+4	8.37E+3	−

complexity and the theoretical $O(N)$ computational complexity with edge skeletonization, in terms of operations and memory, necessary for computing the diagonal of the inverse of a sparse matrix discretized from an elliptic differential operator. The proposed algorithm was applied to 3D MPB problems, demonstrating impressive performance in terms of both accuracy and efficiency. We remark that the proposed algorithm also permits its application to discretized integral operators. More precisely, since the discretization matrix from an integral operator is dense, one can skeletonize the interior points on integer levels and apply the skeletonization similar to the SelInvHIF on fractional levels. This process leads to the corresponding factorization form of a matrix, followed by the utilization of an extraction method to obtain the diagonal part of the matrix inverse. This will be our future work. Moreover, we plan to develop parallel implementations of the SelInvHIF algorithm

to tackle larger-scale problems. This development will open up possibilities for simulating and analyzing complex systems at a larger scale, further advancing our understanding of electrostatic interactions in various applications.

Acknowledgements Y. T. and Z. X. acknowledge the financial support from the National Natural Science Foundation of China (Grant Nos. 12071288 and 12325113), the Science and Technology Commission of Shanghai Municipality of China (Grant No. 21JC1403700), and Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA25010403). H. Y. thanks the support of the US National Science Foundation under awards DMS-2244988 and DMS-2206333.

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

Declarations

Conflict of Interest The authors state that there is no conflict of interest.

Ethical Statement This manuscript complies with all ethical standards for scientific publishing.

References

1. Bazant, M.Z., Storey, B.D., Kornyshev, A.A.: Double layer in ionic liquids: overscreening versus crowding. Phys. Rev. Lett. **106**(4), 046102 (2011)
2. Boroudjerdi, H., Kim, Y.-W., Naji, A., Netz, R.R., Schlagberger, X., Serr, A.: Statics and dynamics of strongly charged soft matter. Phys. Rep. **416**, 129–199 (2005)
3. Borukhov, I., Andelman, D., Orland, H.: Steric effects in electrolytes: a modified Poisson-Boltzmann equation. Phys. Rev. Lett. **79**(3), 435–438 (1998)
4. Brandt, D.: Multi-level adaptive solutions to boundary-value problems. Math. Comput. **138**, 333–390 (1977)
5. Chapman, D.L.: A contribution to the theory of electrocapillarity. Phil. Mag. **25**, 475–481 (1913)
6. Cheng, H., Gimbutas, Z., Martinsson, P., Rokhlin, V.: On the compression of low rank matrices. SIAM J. Sci. Comput. **26**(4), 1389–1404 (2005)
7. Corry, B., Kuyucak, S., Chung, S.H.: Dielectric self-energy in Poisson-Boltzmann and Poisson-Nernst-Planck models of ion channels. Biophys. J. **84**(6), 3594–3606 (2003)
8. Daiguji, H., Yang, P., Majumdar, A.: Ion transport in nanofluidic channels. Nano Lett. **4**(1), 137–142 (2004)
9. Duff, I.S., Reid, J.K.: The multifrontal solution of indefinite sparse symmetric linear. ACM Trans. Math. Softw. **9**(3), 302–325 (1983)
10. George, A.: Nested dissection of a regular finite element mesh. SIAM J. Numer. Anal. **10**(2), 345–363 (1973)
11. Gillman, A., Martinsson, P.G.: A direct solver with $O(N)$ complexity for variable coefficient elliptic PDEs discretized via a high-order composite spectral collocation method. SIAM J. Sci. Comput. **36**(4), 2023–2046 (2013)
12. Gillman, A., Martinsson, P.G.: An $O(N)$ algorithm for constructing the solution operator to 2D elliptic boundary value problems in the absence of body loads. Adv. Comput. Math. **40**(4), 773–796 (2014)
13. Gouy, G.: Constitution of the electric charge at the surface of an electrolyte. J. Phys. **9**, 457–468 (1910)
14. Grasedyck, L., Kriemann, R., Borne, S.L.: Domain-decomposition based $\mathcal{H}$-LU preconditioners. Numer. Math. **112**(4), 565–600 (2009)
15. Ho, K.L., Ying, L.: Hierarchical interpolative factorization for elliptic operators: differential equations. Comm. Pure Appl. Math. **69**(8), 1415–1451 (2016)
16. Ho, K.L., Ying, L.: Hierarchical interpolative factorization for elliptic operators: integral equations. Comm. Pure Appl. Math. **69**(7), 1314–1353 (2016)
17. Ji, L., Liu, P., Xu, Z., Zhou, S.: Asymptotic analysis on dielectric boundary effects of modified Poisson-Nernst-Planck equations. SIAM J. Appl. Math. **78**, 1802–1822 (2018)

18. Kenneth, L.H.: FLAM: fast linear algebra in MATLAB—algorithms for hierarchical matrices. J. Open Source Softw. **5**, 1906 (2020)
19. Liljeström, V., Seitsonen, J., Kostiainen, M.: Electrostatic self-assembly of soft matter nanoparticle cocrystals with tunable lattice parameters. ACS Nano **9**(11), 11278–11285 (2015)
20. Lin, L., Lu, J., Ying, L., Car, R., E, W.: Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems. Commun. Math. Sci. **7**(3), 755–777 (2009)
21. Lin, L., Yang, C., Lu, J., Ying, L., E, W.: A fast parallel algorithm for selected inversion of structured sparse matrices with application to 2D electronic structure calculations. SIAM J. Sci. Comput. **33**(3), 1329–1351 (2011)
22. Lin, L., Yang, C., Meza, J.C., Lu, J., Y, L., E, W.: SelInv–an algorithm for selected inversion of a sparse symmetric matrix. ACM Trans. Math. Softw. **37**(4), 1–19 (2011)
23. Liu, C., Wang, C., Wise, S., Yue, X., Zhou, S.: A positivity-preserving, energy stable and convergent numerical scheme for the Poisson-Nernst-Planck system. Math. Comput. **90**, 2071–2106 (2021)
24. Liu, H., Wang, Z.: A free energy satisfying finite difference method for Poisson-Nernst-Planck equations. J. Comput. Phys. **268**(2), 363–376 (2014)
25. Liu, J.-L., Eisenberg, R.S.: Molecular mean-field theory of ionic solutions: a Poisson-Nernst-Planck-Bikerman model. Entropy **22**(5), 550 (2020)
26. Liu, J.W.H.: The multifrontal method for sparse matrix solution: theory and practice. SIAM Rev. **34**(1), 82–109 (1992)
27. Liu, P., Ji, X., Xu, Z.: Modified Poisson-Nernst-Planck model with accurate Coulomb correlation in variable media. SIAM J. Appl. Math. **78**, 226–245 (2018)
28. Ma, M., Xu, Z.: Self-consistent field model for strong electrostatic correlations and inhomogeneous dielectric media. J. Chem. Phys. **141**(24), 244903 (2014)
29. Ma, M., Xu, Z., Zhang, L.: Modified Poisson-Nernst-Planck model with Coulomb and hard-sphere correlations. SIAM J. Appl. Math. **81**, 1645–1667 (2021)
30. Netz, R.R., Orland, H.: Beyond Poisson-Boltzmann: fluctuation effects and correlation functions. Eur. Phys. J. E **1**(2), 203–214 (2000)
31. Netz, R.R., Orland, H.: Variational charge renormalization in charged systems. Eur. Phys. J. E **11**(3), 301–311 (2003)
32. Podgornik, R.: Electrostatic correlation forces between surfaces with surface specific ionic interactions. J. Chem. Phys. **91**, 5840–5849 (1989)
33. Schmitz, P.G., Ying, L.: A fast direct solver for elliptic problems on general meshes in 2D. J. Comput. Phys. **231**(4), 1314–1338 (2012)
34. Schoch, R.B., Han, J., Renaud, P.: Transport phenomena in nanofluidics. Rev. Mod. Phys. **80**, 839–883 (2008)
35. Tu, Y., Pang, Q., Yang, H., Xu, Z.: Linear-scaling selected inversion based on hierarchical interpolative factorization for self Green's function for modified Poisson-Boltzmann equation in two dimensions. J. Comput. Phys. **461**, 110893 (2022)
36. Wang, Z.-G.: Fluctuation in electrolyte solutions: the self energy. Phys. Rev. E **81**, 021501 (2010)
37. Xia, J., Chandrasekaran, S., Gu, M., Li, X.: Superfast multifrontal method for large structured linear systems of equations. SIAM J. Matrix Anal. Appl. **31**(3), 1382–1411 (2009)
38. Xia, J., Xi, Y., Cauley, S., Balakrishnan, V.: Fast sparse selected inversion. SIAM J. Matrix Anal. Appl. **36**(3), 1283–1314 (2015)
39. Xu, Z., Maggs, A.C.: Solving fluctuation-enhanced Poisson-Boltzmann equations. J. Comput. Phys. **36**(3), 310–322 (2014)
40. Xu, Z., Ma, M., Liu, P.: Self-energy-modified Poisson-Nernst-Planck equations: WKB approximation and finite-difference approaches. Phys. Rev. E **90**(1), 013307 (2014)