Linear-Scaling Selected Inversion based on Hierarchical Interpolative Factorization for Self Green's Function for Modified Poisson-Boltzmann Equation in Two Dimensions

Yihui Tu[1], Qiyuan Pang[2], Haizhao Yang[2]*, Zhenli Xu[3]
[1] School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, China
[2] Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA
[3] School of Mathematical Sciences, Institute of Natural Sciences and MoE-LSC, Shanghai Jiao Tong University, Shanghai 200240, China

May 20, 2021

Abstract

This paper studies an efficient numerical method for solving modified Poisson-Boltzmann (MPB) equations with the self Green's function as a state equation to describe electrostatic correlations in ionic systems. Previously, the most expensive point of the MPB solver is the evaluation of Green's function. The evaluation of Green's function requires solving high-dimensional partial differential equations, which is the computational bottleneck for solving MPB equations. Numerically, the MPB solver only requires the evaluation of Green's function as the diagonal part of the inverse of the discrete elliptic differential operator of the Debye-Hückel equation. Therefore, we develop a fast algorithm by a coupling of the selected inversion and hierarchical interpolative factorization. By the interpolative factorization, our new selected inverse algorithm achieves linear scaling to compute the diagonal of the inverse of this discrete operator. The accuracy and efficiency of the proposed algorithm will be demonstrated by extensive numerical results for solving MPB equations.

Keywords: Selected Inverse; Hierarchical Interpolative Factorization; Linear Scaling; Elliptic Operator; Self Green's Function; Modified Poisson-Boltzmann Equation.

1 Introduction

Electrostatic interaction plays an important role in many fields of physical and biological sciences [1, 2, 3], as well as materials science such as nanoparticle assembly [4]. The Poisson-Boltzmann (PB) [5, 6] and Poisson-Nernst-Planck (PNP) equations [7, 8] are often used to describe the electrostatic phenomena of equilibrium and dynamical systems, respectively. The PB equation is mean-field theory and fails to capture many-body properties such as dielectric variation and ion correlation, which are essential components of electrostatic behaviors of many systems. Many improved theories have been introduced in the literature to take into account these many-body effects [9, 10, 11] and various numerical methods [12, 13, 14] have been proposed to solve the systems efficiently. Among them, the Gaussian variational field theory [15, 16] is promising to describe the long-range Coulomb correlation including dielectric variation [17, 18, 19]. The theory introduces the self-energy of a test ion as a correction to the mean-field potential energy,

*Corresponding author.

which is described by the self Green's function. Based on the self-energy, the effect due to dielectric inhomogeneity is considered in [20, 21, 22]. The Green's function used in the field theory satisfies the generalized Debye-Hückel (DH) equation for which the numerical solution is expensive due to its high spatial dimensions (including both source and field coordinates). By finite-difference discretization, the self Green's function corresponds to the diagonal of the inverse of the discrete elliptic differential operator of the DH equation.

In this paper, we propose a novel method for solving modified Poisson-Boltzmann (MPB) equations, particularly, a fast algorithm for obtaining the diagonal of the inverse matrix from the discretization of the DH equation. To show the basic idea, here we consider the following elliptic partial differential equation,

$$-\nabla\cdot(a(\boldsymbol{r})\nabla u(\boldsymbol{r}))+b(\boldsymbol{r})u(\boldsymbol{r})=f(\boldsymbol{r}),\quad \boldsymbol{r}\in\boldsymbol{\Omega}\subset\mathbb{R}^d \tag{1.1}$$

with an appropriate boundary condition, where $a(\boldsymbol{r})>0$, $b(\boldsymbol{r})$, and $f(\boldsymbol{r})$ are functions on $\boldsymbol{\Omega}$, and $d=2$. Then Eq. (1.1) leads to a linear system after finite-difference discretization,

$$Au_N=f_N,$$

where $A\in\mathbb{R}^{N\times N}$ is sparse, u_N and f_N are the discrete forms of $u(\boldsymbol{r})$ and $f(\boldsymbol{r})$, respectively. Our goal here is to compute the diagonal of A^{-1} in $O(N)$ operations to obtain the self energy in the DH equation, which accelerates the numerical solver for MPB equations.

Selecting the diagonal of a matrix inverse has been previously studied especially in electronic structure calculation based on sparsity and low-rankness, e.g., [23, 24, 25] with $O(N^{3/2})$ computational complexity for 2D problems, and [26] with $O(N\text{poly}(\log N))$ complexity. The selected inversion method in [23] applies a hierarchical decomposition of the computational domain $\boldsymbol{\Omega}$ and proposes a two-step procedure to form the diagonal of A^{-1} with $O(N^{3/2})$ complexity for 2D problems. First, hierarchical Schur complements of the interior points for the blocks of the domain are constructed in a bottom-up pass. Second, the diagonal entries are extracted efficiently in a top-down pass by exploiting the hierarchical local dependence of the inverse matrices. The method in [24, 25] uses a supernode left-looking LDL factorization of A to improve the efficiency of the selected inversion method in [23] by significantly reducing the prefactor in their complexity. Structured multifrontal LDL factorizations are applied in [26] to obtain $O(N\text{poly}(\log N))$ complexity.

Recently, Hierarchical interpolative factorization (HIF) proposed in [27] can compute a generalized LDL decomposition of A within $O(N)$ complexity in 2D problems. HIF is a fast approximation of Multifrontal Factorization (MF) by introducing additional levels of compression based on skeletonizing separator fronts. Unlike [28, 29, 30, 31, 32] that keep the entire fronts but work with them implicitly using fast structured methods, HIF allows us to reduce the fronts explicitly. Inspired by HIF, we can replace the supernode left-looking LDL factorization with HIF in [24, 25] and revise the extraction procedure to approximate the diagonal of A^{-1} within $O(N)$ operations for 2D problems. We will present the main idea of skeletonizing separator fronts in HIF and its application to the selected inversion method with a visible example and a complexity estimation. For the detailed introduction to the selected inverse and HIF, the reader is referred to [24, 25] and [27], respectively. Our algorithm is called SelInvHIF in this paper.

Organization. The rest of the paper is organized as follows. Section 2 discusses iterative solvers for MPB equations. In Section 3, we introduce some preliminary tools of skeletonization of matrix factorization, then the details of SelInvHIF algorithm are presented. Various numerical results of SelInvHIF are provided in Section 4 for solving the MPB equations. The conclusion and discussion for future work are presented in Section 5.

2 Numerical Method for Modified Poisson-Boltzmann Equation

In this section, we will present mathematical model and numerical scheme for the Poisson-Boltzmann equation to motivate our study of SelInvHIF. We consider an electrolyte. In dimensionless units, the dynamics of the mobile ions can be described by the Nernst-Planck equations[12],

$$\frac{\partial c_i}{\partial t} = \nabla \cdot D\left[\nabla c_i + c_i \nabla\left(z_i \Phi + \Xi u\right)\right], \tag{2.1}$$

where c_i is the ionic concentration of the ith species, $z_i = \pm 1$ is the valence, D is the diffusion constant. Here, Ξ is the coupling parameter that describes the strength of the correlation energy. The Nernst-Planck equations are convection-diffusion equations, where the convection is due to the electrostatic force on each ion, namely the gradient of the electrostatic energy. In the modified PNP equations, the electrostatic energy is composed of the mean potential energy $z_i\Phi$ and the self energy Ξu. The electric potential satisfies the Poisson equation,

$$-\nabla \cdot \varepsilon \nabla \Phi = \rho_f + \sum_i z_i c_i,$$

where ε is the dielectric coefficient and ρ_f is the fixed charge distribution. We suppose $\varepsilon = 1$ in the electrolyte. The self energy is represented by the self Green's function, described by the following DH equation,

$$\begin{cases} -\nabla \cdot \varepsilon \nabla G + \sum_i z_i^2 c_i G = \delta\left(\boldsymbol{r} - \boldsymbol{r}'\right), \\ u = \lim\limits_{\boldsymbol{r}' \to \boldsymbol{r}} \left[G\left(\boldsymbol{r}, \boldsymbol{r}'\right) - G_0\left(\boldsymbol{r}, \boldsymbol{r}'\right)\right], \end{cases}$$

where $G_0 = 1/(\varepsilon|\boldsymbol{r} - \boldsymbol{r}'|)$ is the free-space Green's function. It can be seen that the DH equation is coupled with the PNP equations as the ionic strength $I = \sum_i z_i^2 c_i$ is determined by the Nernst-Planck equations. When the correlation effect can be ignored ($\Xi \to 0$), the whole systems become the classical PNP equations.

At equilibrium, the ionic flux in Eq. (2.1) becomes zero and there is an explicit relation between the ionic concentration and the electrostatic energy,

$$c_\pm = \frac{1}{2}\Lambda e^{\mp\Phi - \Xi u},$$

where Λ is the fugacity determined by the far-field boundary conditions. The MPB equation can be obtained when the Boltzmann distributions are used in the Poisson equation, written as,

$$-\nabla \cdot \varepsilon \nabla \Phi = \rho_f - \Lambda e^{\Xi u} \sinh \Phi.$$

Together with the DH equation, we have the following system of equations[15, 17],

$$\begin{cases} -\nabla \cdot \varepsilon \nabla \Phi = \rho_f - \boldsymbol{\Lambda} e^{\Xi u} \sinh \Phi, \\ -\nabla \cdot \varepsilon \nabla G + \boldsymbol{\Lambda} e^{\Xi u} G = \delta\left(\boldsymbol{r} - \boldsymbol{r}'\right), \\ u = \lim\limits_{\boldsymbol{r}' \to \boldsymbol{r}} \left[G\left(\boldsymbol{r}, \boldsymbol{r}'\right) - G_0\left(\boldsymbol{r}, \boldsymbol{r}'\right)\right], \end{cases} \tag{2.2}$$

where the bold $\boldsymbol{\Lambda}$ indicates that it is Λ in the electrolyte domain and zero outside.

Without loss of generality, we discuss the numerical method for solving Eq. (2.2) in this work. In particular, we focus on the numerical method for the self Green's function. The extension to the modified PNP is straightforward. A self-consistent iterative scheme for the solution of the partial differential equations in Eq. (2.2) was developed in [33]. The iterative scheme is expressed by,

$$\begin{cases} -\nabla \cdot \varepsilon \nabla \Phi^{(k+1)} + \mathbf{\Lambda} e^{\Xi u^{(k)}} \sinh \Phi^{(k+1)} = \rho_f, \\ -\nabla \cdot \varepsilon \nabla G^{(k+1)} + \mathbf{\Lambda} e^{\Xi u^{(k)}} G^{(k+1)} = \delta\left(\boldsymbol{r} - \boldsymbol{r}'\right), \\ u^{(k+1)} = \lim\limits_{\boldsymbol{r}' \to \boldsymbol{r}} \left[G^{(k+1)}\left(\boldsymbol{r}, \boldsymbol{r}'\right) - G_0\left(\boldsymbol{r}, \boldsymbol{r}'\right) \right], \end{cases} \tag{2.3}$$

for $k = 0, 1, \cdots, M$. The stopping criteria is $\max \left| \Phi^{(M)} - \Phi^{(M-1)} \right| < \delta$ with a small error criteria δ.

The iterative scheme consists of two alternating steps. We solve the first equation in Eq. (2.3)) for Φ with the given u. Then for a given u and acquired Φ, we solve the second equation in Eq. (2.3)) to obtain the G and then a new u is computed via the third equation in Eq. (2.3). These two steps are called PB and DH steps respectively. We repeat these two steps until we reach the convergence criteria of the solution. Furthermore, the PB step can be efficiently solved using standard fast direct solvers. The problem at the core of obtaining Green's functions comes from the generalized DH equation. In two dimensions, we can approximate the DH equation by,

$$\boldsymbol{A}\boldsymbol{G} = \boldsymbol{E},$$

where $\boldsymbol{G}$ is a matrix representing the lattice Green's function, $\boldsymbol{E}$ is an identity matrix, and $\boldsymbol{A}$ is a coefficient matrix. Furthermore, we can arrive at the matrix inverse $\boldsymbol{G} = \boldsymbol{A}^{-1}$ directly to achieve the solution of the Green's function with expensive calculation. To reduce the computation cost, let us express U by

$$U = \operatorname{diag}(\boldsymbol{G}) - \operatorname{diag}(\boldsymbol{G_0})$$

where $\boldsymbol{G_0}$ is the lattice Green's function in the free space and $\operatorname{diag}(\cdot)$ is a vector representing the diagonals of the argument matrix. Thus, calculating the whole inverse of the matrix directly is expensive and unnecessary. Our SelInvHIF is used to just obtain the diagonal entries of the operator matrix inverse to solve our target problem efficiently.

3 The SelInvHIF Algorithm

The SelInvHIF consists of two phases. In the first phase, we construct hierarchical Schur complements for the diagonal blocks of a matrix A discretized from the differential operator in (1.1) on a physical domain $\mathbf{\Omega}$. In the second phase, the diagonal of the inverse of A are extracted from the construction of the hierarchy of Schur complements. The total complexity of the proposed algorithm is analyzed at the end of this section. Before the formal introduction to our SelInvHIF algorithm, we first introduce some preliminary background of skeletonization of matrix factorization.

3.1 Preliminaries

Suppose A is a matrix, p, q, I and J are index sets. A_{pq} (or $A(I, J)$) denotes a submatrix of A corresponding to rows in p (or I) and columns in q (or J). The notation ":" is used to denote the whole row or column index set, e.g., $A_{:,q}$ consists of columns of A corresponding to indices in q. In the discussion below, we will follow the same notation to denote submatrices.

Suppose the differential operator in Eq. (1.1) is defined on a domain $\mathbf{\Omega}$. A typical discretization of a differential operator results in a sparse matrix with special structures. Let A be a symmetric and nonsingular matrix

$$A = \begin{bmatrix} A_{pp} & A_{qp}^T & \\ A_{qp} & A_{qq} & A_{rq}^T \\ & A_{rq} & A_{rr} \end{bmatrix} \tag{3.1}$$

obtained from the discretization of the differential operator in Eq. (1.1), where p, q, and r are index sets of A with a special order. In this matrix structure, we order rows and columns

carefully such that p is related to the degrees of freedom (DOFs) of the interior points of a small given domain $\mathcal{D} \subset \mathbf{\Omega}$, q corresponds to the DOFs on the boundary $\partial\mathcal{D}$, and r is for the DOFs of the external domain $\mathbf{\Omega}/\overline{\mathcal{D}}$. In general, the DOFs q separates p from r, which is often very large.

3.1.1 Block Inversion

The first preliminary tool we are going to use in SelInvHIF comes from the key observation in the selected inversion method [23]: a diagonal block of the inverse of A can be computed via a diagonal block of the inverse of a submatrix of A, the repeated application of which could lead to an efficient recursive algorithm to compute the diagonal of A. The key observation is based on Lemma 3.1 below. The proof of this lemma is based on block Gaussian elimination as discussed in [23].

Lemma 3.1. *Suppose A is given by (3.1) with a nonsingular A_{pp} and $G = A^{-1}$. Let A_1 be the Schur complement of A_{pp}, i.e.,*

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

and let $G_1 = A_1^{-1}$. Then it holds,

$$G_{pp} = A_{pp}^{-1} + \begin{bmatrix} -A_{pp}^{-1}A_{qp}^T & \mathbf{0} \end{bmatrix} G_1 \begin{bmatrix} -A_{pp}^{-1}A_{qp}^T & \mathbf{0} \end{bmatrix}^T,$$

where G_{pp} is the submatrix of G corresponding to the row and column index set p.

According to Lemma 3.1, the calculation of G_{pp} only requires the values of G_1 associated with row and column indices in q, rather than the whole inverse of the Schur complement, i.e., this observation implies that G_{pp} is determined by $(G_1)_{qq} = (A_1^{-1})_{qq}$, a diagonal block of the inverse of the matrix A_1 of a smaller size than the original larger matrix A.

3.1.2 Interpolative Decomposition

The second tool repeatedly applied in SelInvHIF is the interpolative decomposition (ID) [34] for low-rank matrices based on Lemma 3.2 below.

Lemma 3.2. *Let $A \in \mathbb{R}^{m\times n}$ with rank $k \leq \min(m,n)$ and q be the set of all column indices of A. Then there exist a disjoint partition of $q = \hat{q} \cup \check{q}$ with $|\hat{q}| = k$ and a matrix $T_q \in \mathbb{R}^{k\times(n-k)}$ such that $A_{:,\check{q}} = A_{:,\hat{q}}T_q$.*

The sets $\hat{q}$ and $\check{q}$ are called the skeleton and redundant indices, respectively. In particular, the redundant columns of A can be expressed by its skeleton columns and the associated interpolation matrix from Lemma 3.2. The following corollary shows that matrix A can be sparsified by multiplying a triangular matrix constructed from the interpolation matrix T_q in Lemma 3.2.

Corollary 3.3. *With the same assumptions and notations in Lemma 3.2, it holds that*

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

3.1.3 Block Inversion with Skeletonization

The application of Corollary 3.3 can eliminate redundant DOFs of a dense matrix with low-rank off-diagonal blocks to form a structured matrix of the form (3.1) such that we can apply Lemma 3.1. This idea is called block inversion with skeletonization summarized in Lemma 3.4 below. The skeletonization idea was originally proposed in the HIF [27].

Lemma 3.4. *Let*

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

be symmetric with A_{qp} low-rank. Let $p = \hat{p} \cup \check{p}$ and T_p satisfy $A_{q\check{p}} = A_{q\hat{p}} T_p$. Without loss of generality, rewrite

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

and define

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

Then

$$\bar{A} \triangleq Q_p^T A Q_p = \begin{bmatrix} B_{\check{p}\check{p}} & B_{\hat{p}\check{p}}^T & \\ B_{\hat{p}\check{p}} & A_{\hat{p}\hat{p}} & A_{q\hat{p}}^T \\ & A_{q\hat{p}} & A_{qq} \end{bmatrix}, \tag{3.2}$$

where

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

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

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

and $\bar{G}_1 = \bar{A}_1^{-1}$. Then the following formulas hold by Lemma 3.1 and (3.2),

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

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

According to Lemma 3.4, the calculation of $G_{\check{p}\check{p}}$ only requires the values of $\bar{G}_1$ associated with row and column indices in $\hat{p}$, rather than the whole inverse of the Schur complement, i.e., this observation implies that $G_{\check{p}\check{p}}$ is determined by $(\bar{G}_1)_{\hat{p}\hat{p}}$, a diagonal block of the inverse of the matrix $\bar{A}_1$ of a smaller size than the original larger matrix A. Though $\bar{A}_1$ might be dense, as long as it has low-rank off-diagonal blocks, the same idea as in (3.2) can be applied to $\bar{A}_1$ to compute a diagonal block of the inverse of $\bar{A}_1$, which forms a recursive algorithm to compute the diagonal blocks of A efficiently.

This skeletonization is the key contribution of Ho and Ying [27] and the reason why it is applicable is stated as follows. The key conclusion is that the above Schur complements have specific low-rank structures. The matrix A_{pp}^{-1} from a local differential operator often has numerically low-rank off-diagonal blocks. Especially, the Schur complement interaction $A_{qq} - A_{qp} A_{pp}^{-1} A_{qp}^T$ also has the same rank structure, which is verified by numerical experiments. In the next subsection, we apply Lemma 3.1 and Lemma 3.4 to construct the hierarchical Schur complements for the diagonal blocks of a matrix A.

3.2 Hierarchy of Schur Complements

We discuss the differential operator domain with a grid size $\sqrt{N} \times \sqrt{N} = r_0 2^{L-1} \times r_0 2^{L-1}$ and an initial index set J_0 (e.g., row-major ordering). The corresponding matrix A is of size $N \times N$. A hierarchical disjoint partition of the domain $\mathbf{\Omega}$ (bipartition in each dimension) is performed with $r_0 \times r_0$ as the size of leaf domains and L as the total number of integer levels. Between L integer levels, $L-1$ fractional levels are designed to take advantage of the low-rankness of A as

much as possible. The hierarchy construction of Schur complements will be conducted at levels 1, $\frac{3}{2}$, 2, $\frac{5}{2}$, ..., and L.

For simplicity, consider the case of $r_0 = 5$ and $L = 3$. The whole domain is considered as the top level (Level 3) and is divided into four blocks at the next level (Level 2). Each block is again partitioned into four sub-blocks at a lower level (Level 1). Between two adjacent integral levels, one fractional level will be added to skeletonize low-rank matrices corresponding to the fronts between domain blocks. Hence, the whole domain is divided into $2^{L-1} \times 2^{L-1} = 16$ blocks at the bottom level (Level 1) as shown in Figure 1.

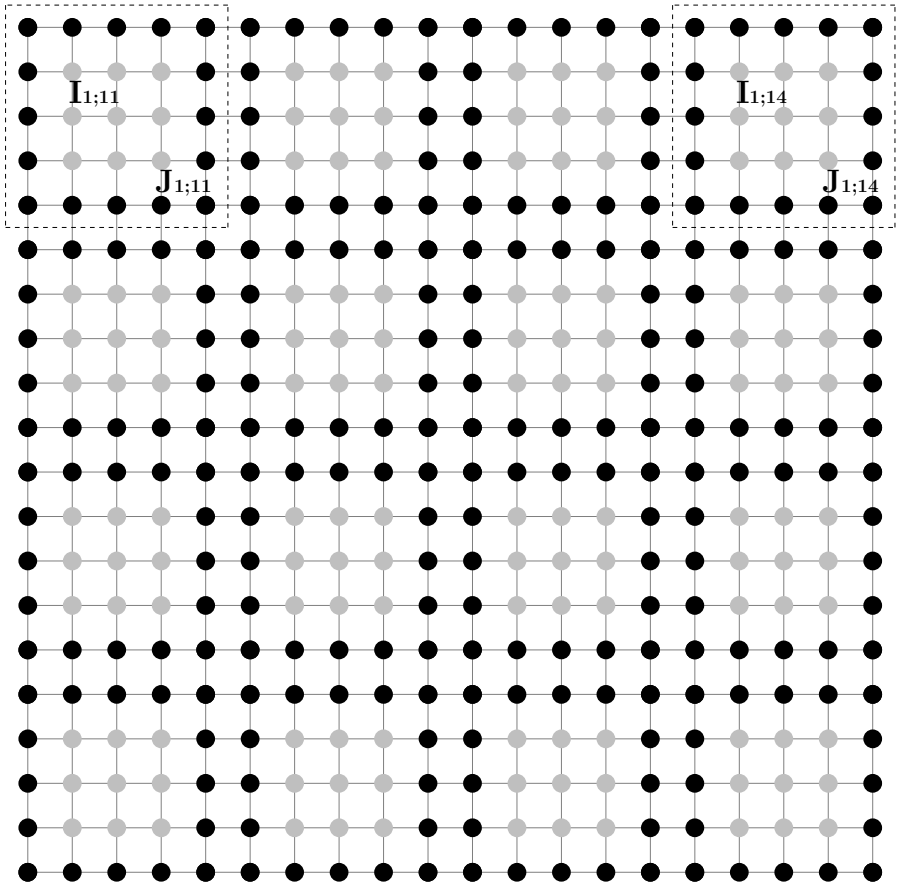

Figure 1: The DOFs in the first level. The domain is partitioned into 16 blocks in the Level 1 and the dash lines show two of blocks in the decomposition. In this and following integral levels, the interior points are marked in gray and the boundary points are marked in black. It's noted that the blocks share edge in practice to reduce the prefactor.

3.2.1 Level $\ell = 1$

The domain $\mathbf{\Omega}$ is partitioned into $2^{L-\ell} \times 2^{L-\ell} = 4 \times 4$ disjoint blocks at Level $\ell = 1$. All points are separated into interior points and boundary points in each block. The interior points indicate that they are not related to the points of other blocks, and the boundary points indicate that they are related to the points of other blocks, i.e., having neighboring points in other blocks. The index set of the interior points are denoted as $I_{1;ij}$ for each block (gray points in Figure 1), and the index set of the boundary points are denoted as $J_{1;ij}$ of each block (black points in Figure 1), where $i, j = 1, 2, 3, 4$ are the indices of blocks in each dimension. The locality of differential operators leads to $A(I_{1;ij}, I_{1;i'j'}) = 0$ (or $A(I_{1;ij}, J_{1;i'j'}) = 0$) if $(i, j) \neq (i', j')$.

Firstly, Gauss elimination is used to eliminate the interior points, thereby shifting the focus of the problem to the boundary points. To do this, we need to apply necessary row and column permutations to the matrix A defined with the index set J_0 such that all the interior points are in front of the boundary points, that is, the indices J_0 is changed to

$$J_0 \xrightarrow{P_1} (I_{1;11}I_{1;12}...I_{1;44}|J_{1;11}J_{1;12}...J_{1;44})$$

by a permutation matrix P_1. Then all interior points and boundary points are separated by notation $|$. In fact, the permutation matrix P_1 can permute the matrix A into a new matrix

$$A_1 = P_1^{-1}AP_1$$

with index set $(I_1|J_1)$, where $I_1 = I_{1;11}I_{1;12}...I_{1;44}$ gathering all the interior points, and $J_1 = J_{1;11}J_{1;12}...J_{1;44}$ for all the boundary points. Represent A_1 via

$$A_1 = P_1^{-1}AP_1 = \begin{bmatrix} U_1 & V_1^T \\ V_1 & W_1 \end{bmatrix}, \tag{3.3}$$

where U_1 is a block diagonal matrix as follows because the interior points of different blocks in Figure 1 are not related to the points of other blocks:

$$U_1 = A_1(I_1, I_1) = \begin{bmatrix} U_{1;11} & & & \\ & U_{1;12} & & \\ & & \ddots & \\ & & & U_{1;44} \end{bmatrix}$$

with $U_{1;ij} = A_1(I_{1;ij}, I_{1;ij})$. Moreover, V_1 is also a block diagonal matrix as follows because the interior points of each block just is related to the boundary points of the same block:

$$V_1 = A_1(J_1, I_1) = \begin{bmatrix} V_{1;11} & & & \\ & V_{1;12} & & \\ & & \ddots & \\ & & & V_{1;44} \end{bmatrix}$$

with $V_{1;ij} = A_1(J_{1;ij}, I_{1;ij})$. As for W_1, we just write it with index set,

$$W_1 = A_1(J_1, J_1).$$

The inverse of U_1 can be computed directly as follows since it is a block diagonal matrix with diagonal blocks of a small size $(r_0 - 2)^2 \times (r_0 - 2)^2$,

$$U_1^{-1} = \begin{bmatrix} U_{1;11}^{-1} & & & \\ & U_{1;12}^{-1} & & \\ & & \ddots & \\ & & & U_{1;44}^{-1} \end{bmatrix}.$$

Therefore, we can obtain the following inverse by Gaussian elimination,

$$A_1^{-1} = \begin{bmatrix} U_1 & V_1^T \\ V_1 & W_1 \end{bmatrix}^{-1} = L_1^T \begin{bmatrix} U_1^{-1} & \\ & (W_1 - V_1U_1^{-1}V_1^T)^{-1} \end{bmatrix} L_1 \tag{3.4}$$

with $L_1 = \begin{bmatrix} I & \\ -V_1U_1^{-1} & I \end{bmatrix}$. Furthermore, $V_1U_1^{-1}$ can be computed independently within each block with block diagonal matrices V_1 and U_1^{-1},

$$V_1U_1^{-1} = \begin{bmatrix} V_{1;11}U_{1;11}^{-1} & & & \\ & V_{1;12}U_{1;12}^{-1} & & \\ & & \ddots & \\ & & & V_{1;44}U_{1;44}^{-1} \end{bmatrix}.$$

Moreover, the block diagonal matrix $V_1 U_1^{-1} V_1^T$ also can be represented as

$$V_1 U_1^{-1} V_1^T = \begin{bmatrix} V_{1;11} U_{1;11}^{-1} V_{1;11}^T & & & \\ & V_{1;12} U_{1;12}^{-1} V_{1;12}^T & & \\ & & \ddots & \\ & & & V_{1;44} U_{1;44}^{-1} V_{1;44}^T \end{bmatrix}.$$

Combining (3.3) and (3.4), we have

$$G = A^{-1} = P_1 A_1^{-1} P_1^{-1} = P_1 L_1^T \begin{bmatrix} U_1^{-1} & \\ & G_1 \end{bmatrix} L_1 P_1^{-1}, \tag{3.5}$$

where $G_1 = (W_1 - V_1 U_1^{-1} V_1^T)^{-1}$ is the inverse of the Schur complement of U_1. Therefore, we reduce the problem to a smaller matrix $W_1 - V_1 U_1^{-1} V_1^T$ by eliminating interior points, which is essentially the block inversion idea in Lemma 3.1.

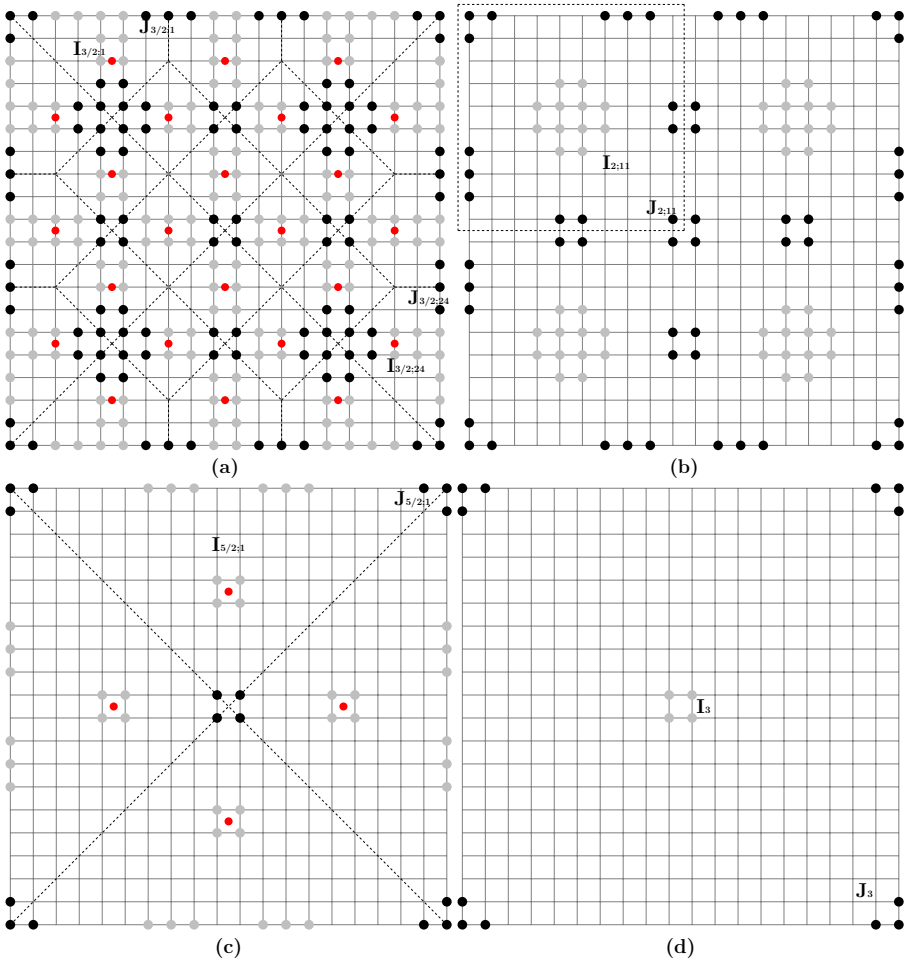

Figure 2: (a): The DOFs in the Level 3/2. The domain is partitioned into 24 Voronoi cells about the edge centers. In this and following fractional level, the redundant DOFs are marked in gray and the skeleton DOFs are marked in black. (b): The DOFs in the Level 2. The domain is partitioned into 4 blocks in Level 2 and the dash lines show one of blocks in the decomposition. (c): The DOFs in the Level 5/2. The domain is partitioned into 4 Voronoi cells about the edge centers. (d): The DOFs in the Level 3. This is the top level.

3.2.2 Level $\ell = 3/2$

At this level, our goal is to find G_1 in (3.5), which is defined on the index set J_1 corresponding to boundary points of domain blocks in the first level (i.e., the black points in Figure 1). We will apply Lemma 3.4 to skeletonize the fronts. Partition the domain $\mathbf{\Omega}$ into $2^{L-\ell+\frac{3}{2}}(2^{L-\ell+\frac{1}{2}}-1)$ Voronoi cells [35] about the edge centers (red points in Figure 2 (a)). In this example, there are 24 Voronoi cells in total since $L = 3$. In Figure 2 (a), a Voronoi cell is an area centered at a redpoint with dashed lined compassed. Each DOF on the boundary between two adjacent Voronoi cells is randomly assigned to one and only one of these two cells. Thus a Voronoi cell includes the DOFs inside the corresponding area and some of the DOFs on its boundary. Since the DOFs of a Voronoi cell only interact with the DOFs of a few other cells nearby, that is, the matrix allows low-rank off-diagonal blocks. We can apply an ID to select the redundant and skeleton DOFs approximately in each cell and record the interpolation matrix T_q as in Lemma 3.2. In the ith cell (see Figure 2 (a)), the redundant DOFs (gray points) are denoted by $I_{\frac{3}{2};i}$, the skeleton DOFs (black points) are denoted by $J_{\frac{3}{2};i}$, and the corresponding interpolation matrix is denoted by $T_{\frac{3}{2};i}$. As in the former level, we reindex J_1 by a permutation matrix $P_{\frac{3}{2}}$ such that

$$J_1 \xrightarrow{P_{\frac{3}{2}}} (I_{\frac{3}{2};1}I_{\frac{3}{2};2}...I_{\frac{3}{2};24}|J_{\frac{3}{2};1}J_{\frac{3}{2};2}...J_{\frac{3}{2};24}) = (I_{\frac{3}{2}}|J_{\frac{3}{2}}).$$

Denote

$$A_{\frac{3}{2}} = P_{\frac{3}{2}}^{-1}(W_1 - V_1U_1^{-1}V_1^T)P_{\frac{3}{2}} = \begin{bmatrix} U_{\frac{3}{2}} & V_{\frac{3}{2}}^T \\ V_{\frac{3}{2}} & W_{\frac{3}{2}} \end{bmatrix}$$

with

$$U_{\frac{3}{2}} = A_{\frac{3}{2}}(I_{\frac{3}{2}}, I_{\frac{3}{2}}), \quad V_{\frac{3}{2}} = A_{\frac{3}{2}}(J_{\frac{3}{2}}, I_{\frac{3}{2}}), \quad \text{and} \quad W_{\frac{3}{2}} = A_{\frac{3}{2}}(J_{\frac{3}{2}}, J_{\frac{3}{2}}).$$

Arrange $T_{1;1}, \ldots, T_{1;24}$ in a block diagonal matrix

$$T_{\frac{3}{2}} = \begin{bmatrix} -T_{\frac{3}{2};1} & & \\ & \ddots & \\ & & -T_{\frac{3}{2};24} \end{bmatrix},$$

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

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

Then we have

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

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

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

Then

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

with

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

as in Lemma 3.4. Note that $-\bar{V}_{\frac{3}{2}}\bar{U}_{\frac{3}{2}}^{-1}$ and $\bar{V}_{\frac{3}{2}}\bar{U}_{\frac{3}{2}}^{-1}\bar{V}_{\frac{3}{2}}^T$ are block diagonal. Therefore,

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

Therefore, we reduce the inversion problem to a smaller matrix $W_{\frac{3}{2}} - \bar{V}_{\frac{3}{2}}\bar{U}_{\frac{3}{2}}^{-1}\bar{V}_{\frac{3}{2}}^T$ by eliminating the redundant DOFs as in Lemma 3.4.

3.2.3 Level $\ell = 2$

At Level $\ell = 2$, the domain $\mathbf{\Omega}$ is partitioned into $2^{L-\ell} \times 2^{L-\ell} = 2 \times 2$ blocks with interior and boundary points as shown in Figure 2 (b). Similarly, we reindex the points in $J_{\frac{3}{2}}$ into I_2 and J_2, by a permutation matrix P_2 such that

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

Apply a similar procedure as at Level 1 and denote

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

with

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

Note that U_2 and V_2 are block diagonal. Analogously,

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

where

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

Note that the update $V_2 U_2^{-1} V_2^T$ is block diagonal. Now we have eliminated the interior points and the inversion problem is reduced to a smaller matrix $W_2 - V_2 U_2^{-1} V_2^T$ as in Lemma 3.1.

3.2.4 Level $\ell = 5/2$

Just as in Section 3.2.2, at this level, we want to find G_2 indexed by J_2. Again, we divide the domain $\mathbf{\Omega}$ into $2^{L-\ell+\frac{3}{2}}(2^{L-\ell+\frac{1}{2}} - 1) = 4$ Voronoi cells (see Figure 2 (c)). Again, the DOFs on the boundary between two cells are randomly assigned to one of cells. Through ID, we distinguish the redundant DOFs $I_{\frac{5}{2};i}$ and the skeleton DOFs $J_{\frac{5}{2};i}$ in the ith cell, and record the interpolation matrix $T_{\frac{5}{2};i}$. Reindexing J_2 with a permutation matrix $P_{\frac{5}{2}}$ such that

$$J_2 \xrightarrow{P_{\frac{5}{2}}} (I_{\frac{5}{2};1} I_{\frac{5}{2};2} I_{\frac{5}{2};3} I_{\frac{5}{2};4} | J_{\frac{5}{2};1} J_{\frac{5}{2};2} J_{\frac{5}{2};3} J_{\frac{5}{2};4}) := (I_{\frac{5}{2}} | J_{\frac{5}{2}}).$$

Denote

$$T_{\frac{5}{2}} = \begin{bmatrix} -T_{\frac{5}{2};1} & & \\ & \ddots & \\ & & -T_{\frac{5}{2};4} \end{bmatrix}$$

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

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

Then

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

with

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

Therefore,

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

where

$$L_{\frac{5}{2}} = \begin{bmatrix} I & \\ -\bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1} & I \end{bmatrix}, \quad G_{\frac{5}{2}} = (\bar{W}_{\frac{5}{2}} - \bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1}\bar{V}_{\frac{5}{2}}^{T})^{-1}.$$

Note that $U_{\frac{5}{2}}$ and $V_{\frac{5}{2}}$ are block diagonal. The matrix inversion problem now has been reduced to $\bar{W}_{\frac{5}{2}} - \bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1}\bar{V}_{\frac{5}{2}}^{T}$.

3.2.5 Level $\ell = 3$

The domain $\mathbf{\Omega}$ is partitioned into $2^{L-\ell} \times 2^{L-\ell} = 1 \times 1$ block, i.e., no partition at this level. The interior and boundary points are shown in Figure 2 (d). Similarly to previous integer levels, reindexing $J_{\frac{5}{2}}$ into the union of an interiors index set I_3 and a boundary index set J_3 with a permutation matrix P_3 such that

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

Finally,

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

where

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

We calculate the inverse of G_3 directly at this point.

3.2.6 Summary of Construction

We start with the hierarchical domain decomposition scheme and the skeletonization technique. To construct the hierarchical structure of Schur complements of the matrix A on the grid of size $N \times N$, at each integral level, the points in each block are divided into interior points and boundary points. So the interior points only interact with the points within the same block. We reindex the points and eliminate the interior points accordingly. At each fractional level, the domain is divided into Voronoi cells, and ID is applied to each unit to distinguish redundant points and skeleton points such that the redundant points only interact with the points within the same cell. We will reindex these points accordingly and eliminate the redundant points.

The following relationship is defined for each level

$$G_\ell = \begin{cases} G = A^{-1}, & \ell = 0; \\ G_\ell = (W_\ell - V_\ell U_\ell^{-1} V_\ell^T)^{-1}, & \ell \text{ is integral}; \\ G_\ell = (W_\ell - \bar{V}_\ell \bar{U}_\ell^{-1} \bar{V}_\ell^T)^{-1}, & \ell \text{ is fractional}. \end{cases} \tag{3.6}$$

Based on (3.6), it follows the recursive relation with integral ℓ,

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

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

Therefore, we can construct the hierarchy of Schur complements from the bottom. We organize this algorithm in Algorithm 1. Note that the reindexing is implicitly included in Algorithm 1, when we use the index sets $I_{\ell;ij}$ and $J_{\ell;ij}$ or $I_{\ell;i}$ and $J_{\ell;i}$ for A_ℓ.

Algorithm 1: Constructing the hierarchy of Schur complements of A

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

Output:

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

3.3 Extracting the Diagonal of the Inverse of Matrix

After obtaining the hierarchical structure of Schur complements, we now apply the observation in Lemma 3.1 to extract the diagonal of the inverse matrix G. The point is that it is not necessary to compute the whole Schur complement G_ℓ. More precisely, our observations show that:

$$G_{\ell-1}(I_{\ell;ij}J_{\ell;ij}, I_{\ell;ij}J_{\ell;ij}) \text{ is determined by } G_{\ell-\frac{1}{2}}(J_{\ell;ij}, J_{\ell;ij}),$$

$$G_{\ell-\frac{1}{2}}(I_{\ell-\frac{1}{2};i}J_{\ell-\frac{1}{2};i}, I_{\ell-\frac{1}{2};i}J_{\ell-\frac{1}{2};i}) \text{ is determined by } G_{\ell}(J_{\ell-\frac{1}{2};i}, J_{\ell-\frac{1}{2};i}).$$

Therefore, we can develop a linear scaling algorithm to exact the diagonal elements of G recursively. We organize this algorithm in Algorithm 2. Note that the reindexing is implicitly included in Algorithm 2, when we use the index sets $J_{\ell;ij}$ or $J_{\ell;i}$ for G_ℓ.

3.3.1 Level $\ell = 3$

We start from the top level $\ell = L = 3$ to extract information of interest. Given G_3, $G_{\frac{5}{2}}$ is obtained by the following formula:

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

Submatrices in the bracket are indexed by $(I_3|J_3)$. $G_{\frac{5}{2}}$ is indexed by $J_{\frac{5}{2}} = J_{\frac{5}{2};1}J_{\frac{5}{2};2}J_{\frac{5}{2};3}J_{\frac{5}{2};4}$ due to the permutation matrix P_3. In fact, we only need to focus on $G_{\frac{5}{2}}(J_{\frac{5}{2};i}, J_{\frac{5}{2};i})$ instead of off-diagonal blocks in order to extract the diagonal entries of $G_{\frac{5}{2}}$. Hence, represent $G_{\frac{5}{2}}$ as

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

with

$$G_{\frac{5}{2};i} = G_{\frac{5}{2}}(J_{\frac{5}{2};i}, J_{\frac{5}{2};i}).$$

3.3.2 Level $\ell = 5/2$

At Level $\ell = 5/2$, we now have

$$G_2 \approx P_{\frac{5}{2}} \begin{bmatrix} \mathcal{G}_2 & -\bar{U}_{\frac{5}{2}}^{-1}\bar{V}_{\frac{5}{2}}^T G_{\frac{5}{2}} + \mathcal{G}_2 T_{\frac{5}{2}}^T \\ -G_{\frac{5}{2}}\bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1} + T_{\frac{5}{2}}\mathcal{G}_2 & \mathfrak{G}_2 \end{bmatrix} P_{\frac{5}{2}}^{-1} \tag{3.7}$$

where

$$\mathcal{G}_2 = \bar{U}_{\frac{5}{2}}^{-1} + \bar{U}_{\frac{5}{2}}^{-1}\bar{V}_{\frac{5}{2}}^T G_{\frac{5}{2}}\bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1},$$

and

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

Note that $T_{\frac{5}{2}}$, $U_{\frac{5}{2}}^{-1}$, and $V_{\frac{5}{2}}$ are block diagonal. We have

$$\bar{U}_{\frac{5}{2}}^{-1}\bar{V}_{\frac{5}{2}}^T G_{\frac{5}{2}}\bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1} = \begin{bmatrix} \bar{U}_{\frac{5}{2};1}^{-1}\bar{V}_{\frac{5}{2};1}^T G_{\frac{5}{2};1}\bar{V}_{\frac{5}{2};1}\bar{U}_{\frac{5}{2};1}^{-1} & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & \bar{U}_{\frac{5}{2};4}^{-1}\bar{V}_{\frac{5}{2};4}^T G_{\frac{5}{2};4}\bar{V}_{\frac{5}{2};4}\bar{U}_{\frac{5}{2};4}^{-1} \end{bmatrix},$$

as well as

$$G_{\frac{5}{2}}\bar{V}_{\frac{5}{2}}\bar{U}_{\frac{5}{2}}^{-1}T_{\frac{5}{2}}^{T}=\begin{bmatrix} G_{\frac{5}{2};1}\bar{V}_{\frac{5}{2};1}\bar{U}_{\frac{5}{2};1}^{-1}T_{\frac{5}{2};1}^{T} & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & G_{\frac{5}{2};4}\bar{V}_{\frac{5}{2};4}\bar{U}_{\frac{5}{2};4}^{-1}T_{\frac{5}{2};4}^{T} \end{bmatrix}.$$

Therefore, the corresponding diagonal blocks of $\mathcal{G}_2$ can be computed just using block-block multiplication accordingly. Furthermore, similar operations can be applied to $\mathfrak{G}_2$.

All matrices in the bracket of (3.7) are indexed by $(I_{\frac{5}{2}}|J_{\frac{5}{2}})$. $G_{\frac{5}{2}}$ is indexed by $J_2 = J_{2;11}J_{2;12}J_{2;21}J_{2;22}$ due to the permutation matrix $P_{\frac{5}{2}}$. Similarly to the previous level, we only need to seek the diagonal blocks $G_2(J_{2;ij}, J_{2;ij})$.

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

Input:

Output of Algorithm 1.

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

3.3.3 Level $\ell = 2$

At Level $\ell = 2$, we have

$$G_{\frac{3}{2}} = P_2\begin{bmatrix} U_2^{-1} + U_2^{-1}V_2^{T}G_2V_2U_2^{-1} & -U_2^{-1}V_2^{T}G_2 \\ -G_2V_2U_2^{-1} & G_2 \end{bmatrix}P_2^{-1}. \tag{3.8}$$

Analogously to Level 3, submatrices in the bracket of (3.8) are indexed by $(I_2|J_2)$. $G_{\frac{3}{2}}$ is indexed by $J_{\frac{3}{2}} = J_{\frac{3}{2};1}\cdots J_{\frac{3}{2};24}$ due to the permutation matrix P_2. Again, only $G_{\frac{3}{2}}(J_{\frac{3}{2};i}, J_{\frac{3}{2};i})$ needs to be computed.

3.3.4 Level $\ell = 3/2$

Proceeding to Level 3/2, now

$$G_1 \approx P_{\frac{3}{2}} \begin{bmatrix} \mathcal{G}_1 & -\bar{U}_{\frac{3}{2}}^{-1}\bar{V}_{\frac{3}{2}}^T G_{\frac{3}{2}} + \mathcal{G}_1 T_{\frac{3}{2}}^T \\ -G_{\frac{3}{2}}\bar{V}_{\frac{3}{2}}\bar{U}_{\frac{3}{2}}^{-1} + T_{\frac{3}{2}}\mathcal{G}_1 & \mathfrak{G}_1 \end{bmatrix} P_{\frac{3}{2}}^{-1}, \tag{3.9}$$

where

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

and

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

Similarly to Level $\ell = 5/2$, diagonal blocks of $\mathcal{G}_2$ and $\mathfrak{G}_2$ can be computed quickly using block-block multiplication accordingly. Submatrices in the bracket of (3.9) are indexed by $(I_{\frac{3}{2}}|J_{\frac{3}{2}})$. G_1 is indexed by $J_1 = J_{1;11}J_{1;12}\cdots J_{1;44}$ due to the permutation matrix $P_{\frac{3}{2}}$. Again, only the diagonal blocks $G_1(J_{1;ij}, J_{1;ij})$ are needed.

3.3.5 Level $\ell = 1$

At Level 1, the same procedure is done as at Level 2 and Level 3. We can obtain $G_1(J_{1;ij}, J_{1;ij})$ from Level $\frac{3}{2}$ and $G(J_{0;ij}, J_{0;ij})$ is computed directly. Finally, the diagonal elements in G can be obtained by combining the diagonal elements of each level.

3.4 Complexity Estimates

We next investigate the computation complexity of SelInvHIF. Let us assume the domain contains $N = \sqrt{N} \times \sqrt{N}$ points and set $\sqrt{N} = 2^L$ with $\ell_{\max} < L$.
We denote the number of blocks at level ℓ as $n_B(\ell)$, and have following fomurla

$$n_B(\ell) = \begin{cases} 2^{2(\ell_{\max}-\ell)}, & \ell \text{ is integral;} \\ 2^{2\ell_{\max}-2(\ell-1)} - 2^{\ell_{\max}-\ell+\frac{3}{2}}, & \ell \text{ is fractional.} \end{cases}$$

The number of points of each block or cell is denoted as $n_P(\ell)$. Note that interior or redundant points of the previous level are not counted because they have been eliminated in previous levels. To approximate $n_P(\ell)$, we use the assumption about the skeletonization in [27]. Then it can be shown that the typical skeleton size of a cell is

$$k_\ell = O(\ell).$$

Then we have

$$n_P(\ell) = \begin{cases} 2^{2(L-\ell_{\max}+1)}, & \ell = 1; \\ O(2^{L-\ell_{\max}}), & \ell = \frac{3}{2}; \\ O(\ell), & \ell > \frac{3}{2}. \end{cases}$$

Firstly, the construction step is considered and the following steps are shown in Algorithm 1. At an integral level ℓ, we need to compute the inverse of $U_{\ell;ij}$ (Step 9) for each block. Then multiply the inverse with $V_{\ell;ij}$ to obtain $K_{\ell;ij}$ (Step 10) and finally update the new $A_{\ell+\frac{1}{2}}(J_{\ell;ij}, J_{\ell;ij})$ (Step 11). At a fractional level $\ell + \frac{1}{2}$, for each cell, we need to compute $T_{\ell+\frac{1}{2};k}$ using ID (Step 16, since each cell only interact with $O(1)$ cells, then the cost for this step is $O(n_P(\ell)^3)$). Then apply it (Step 19, 20, and 21) and multiply the inverse of $\bar{U}_{\ell+\frac{1}{2};k}$ (Step 25) with $\bar{V}_{\ell+\frac{1}{2};k}$ to get $\bar{K}_{\ell+\frac{1}{2};k}$ (Step 26). Finally, update $A_{\ell+1}(J_{\ell+\frac{1}{2};k}, J_{\ell+\frac{1}{2};k})$ (Step 27). The computational cost for these steps at each level is $O(n_P(\ell)^3)$. Furthermore, the total cost for level ℓ is $O(n_B(\ell)n_P(\ell)^3)$ for $\ell > \frac{3}{2}$, since there are $n_B(\ell)$ blocks at Level ℓ.

Since

$$
\begin{aligned}
&2^{2(\ell_{\max}-1)}2^{6(L-\ell_{\max}+1)} + 2^{2\ell_{\max}-1}2^{3L-3\ell_{\max}} + \sum_{\ell=2,\frac{5}{2}}^{\ell_{\max}} (n_B(\ell)n_P(\ell)^3) \\
&\leq C(2^{2(\ell_{\max}-1)2^{6(L-\ell_{\max}+1)}} + 2^{2\ell_{\max}-1}2^{3L-3\ell_{\max}} + \sum_{\ell=2,\frac{5}{2}}^{\ell_{\max}} (2^{2\ell_{\max}-2\ell}\ell^3)) \\
&\leq C_0(2^{6L-4\ell_{\max}} + 2^{3L-\ell_{\max}} + 2^{2\ell_{\max}}),
\end{aligned}
$$

where C and C_0 are constant. Let $\ell_{\max} = O(L)$, the total computational cost for the construction step is $O(N)$ (the cost for Step 32 is $O(n_P(\ell)^3)$).

Furthermore, the extraction step is analyzed now and the following steps are considered in Algorithm 2. At an integral level ℓ, $G_{\ell-\frac{1}{2}}(I_{\ell;ij}, I_{\ell;ij})$ (Step 3) and $G_{\ell-\frac{1}{2}}(J_{\ell;ij}, I_{\ell;ij})$ (Step 4) are calculated for each block. At a fractional level $\ell - \frac{1}{2}$, for each cell, we need to calculate $G_{\ell-1}(I_{\ell-\frac{1}{2};k}, I_{\ell-\frac{1}{2};k})$ (Step 10), $G_{\ell-1}(J_{\ell-\frac{1}{2};k}, I_{\ell-\frac{1}{2};k})$ (Step 12) and $G_{\ell-1}(J_{\ell-\frac{1}{2};k}, J_{\ell-\frac{1}{2};k})$ (Step 14). The computational cost for these steps at each level is $O(n_P(\ell)^3)$. Hence, the total cost for level ℓ is $O(n_B(\ell)n_P(\ell)^3)$ for $\ell > \frac{3}{2}$, since there are $n_B(\ell)$ blocks at Level ℓ. Similarly, the complexity for the extraction step is also $O(N)$.

Therefore, the total computational complexity is $O(N)$ by combining the construction step and extraction step if the assumption in [27] holds.

4 Numerical results

We show numerical results for the MPB equations in two dimensions to verify the performance of the proposed SelInvHIF. In particular, the scaling of the computational time of SelInvHIF is concerned. We set a uniform fugacity parameter $\Lambda = 0.2$ and a coupling parameter $\Xi = 1$. The error criteria of the PB solver and the self-consistent iteration are both 10^{-8}. The relative precision of the ID step is 10^{-8}. The initial values for the potentials in the iteration are always constant $\Phi^{(0)} = 0$ in our examples. Dirichlet boundary conditions are used for both the PB and the DH steps. The calculation is performed on a machine with Intel Xeon a 2.2GHz and 2TB memory. The statistics of calculation time are averaged over five times. We compute the relative L^2 to measure the accuracy of SelInvHIF:

$$\frac{\|\Phi - \Phi_{\text{ref}}\|_2}{\|\Phi_{\text{ref}}\|_2}$$

where Φ is the electric potential computed using SelInvHIF and Φ_{ref} is the electric potential computed with sufficiently large grid size. To measure the accuracy of the whole algorithm and the convergence with respect to the grid size, we also compute the absolute L^2 error $\frac{\|\Phi-\Phi_{\text{ref}}\|_2}{\sqrt{N}}$ using a reference solution Φ_{ref} computed with sufficiently large grid size.

4.1 Example 1: The Discrete Elliptic Differential Operator

We consider the diagonal part of the inverse of the discrete elliptic differential operator as the first example. Using the five-point stencil discretization, a 5-diagonal $N \times N$ sparse matrix D_5 is denoted as:

$$
D_5 = \begin{pmatrix} M & -I & & 0 \\ -I & \ddots & \ddots & \\ & \ddots & \ddots & -I \\ 0 & & -I & M \end{pmatrix}, \quad M = \begin{pmatrix} 4 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 4 \end{pmatrix}
$$

We then calculate the diagonal part of inverse of matrix D_5 by SelInvHIF and the exact method in [23], respectively. In Table 1, the absolute L^2 error between the numerical results and the reference solution obtained with corresponding matrix size by exact method are displayed. Table 1 also shows the relative L^2 errors, which verify the accuracy of SelInvHIF. Finally, Table 1 also shows the computational time of the algorithm and verifies the linear scaling of SelInvHIF.

Matrix size $\sqrt{N}$	SelInvHIF time	Absolute error	Relative error
256	$3.25E+1$	$4.78E-13$	$3.53E-8$
512	$1.54E+2$	$2.05E-13$	$5.39E-8$
1024	$6.15E+2$	$2.88E-13$	$2.73E-7$
2048	$2.43E+3$	$5.80E-13$	$2.00E-6$

Table 1: The CPU time, accuracy, and matrix size. The SelInvHIF time means the execution time spent for one step SelInvHIF.

4.2 Example 2: The Charge Density is a Continuous Function

The second example is a continuous charged distribution in the region $[0, L]^2$ with $L = 32$. The distribution of charge density is

$$\rho_f(x) = \frac{\sin(\frac{\pi x}{L})}{32}.$$

We then calculate the results of the MPB equations by SelInvHIF. Similarly, the left panel of Figure 3 visualizes the distribution of the convergent potential in this system with different matrix sizes $N = 256^2$, 512^2 and 1024^2. The right panel of Figure 3 displays the absolute error in different locations between the numerical results and the reference solution with a sufficiently large grid size $N = 2048^2$. The relative L^2 errors maintain approximate accuracy of first-order in Table 2 due to the discontinuous of the derivative of the potential at $x = L/2$. Table 2 also shows the accuracy of the whole algorithm to compute the potential Φ compared to a reference potential computed with a sufficiently large grid size $N = 2048^2$, which verifies the convergence of our algorithm. In addition, Table 2 also shows the computational time of the algorithm to verify the linear scaling of SelInvHIF. Finally, the scaling results of the SelInvHIF algorithm are shown in Figure 6.

Matrix size $\sqrt{N}$	Total time	SelInvHIF time	Absolute error	Relative error
256	$3.16E+1$	$3.10E+1$	$7.13E-3$	$5.06E-2$
512	$1.35E+2$	$1.32E+2$	$3.01E-3$	$2.16E-2$
1024	$5.81E+2$	$5.69E+2$	$1.02E-3$	$7.27E-3$
2048	$2.40E+3$	$2.35E+3$	-	-

Table 2: The CPU time, accuracy, and matrix size. The total time and the SelInvHIF time mean the execution time spent for the one step iteration in the whole program and the time for one step SelInvHIF, respectively.

4.3 Example 3: The Charge Density is a Delta Function

In last two examples, we consider discontinuous charged distribution in a region $[0, L]^2$ with $L = 32$. Let the charge density be

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

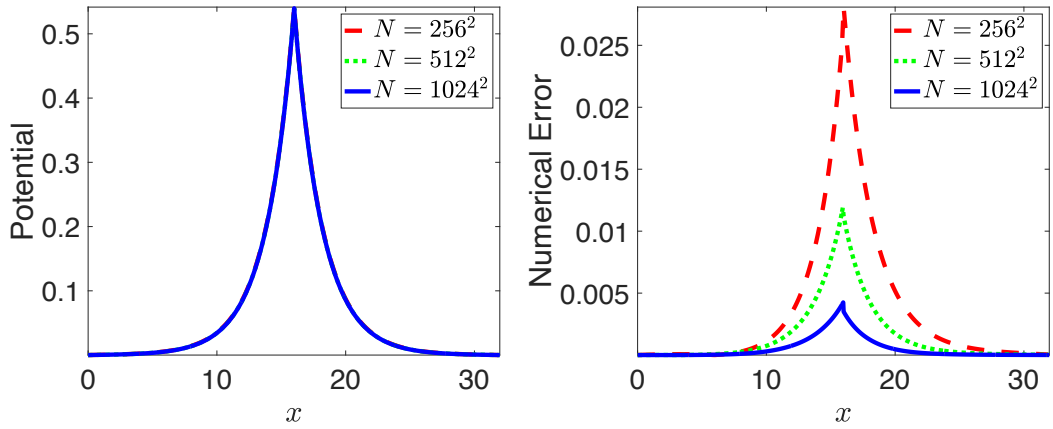

Figure 3: Numerical results about the continuous charge density. Left: potential distributions with different matrix size; Right: the absolute error between the numerical results and the reference solution with $N = 2048^2$.

We then calculate the results of the MPB equations by SelInvHIF. Similarly, the left panel of Figure 4 visualizes the distribution of the convergent potential in this system with different matrix sizes $N = 256^2$, 512^2 and 1024^2. The right panel of Figure 4 displays the absolute L^2 error between the numerical results and the reference solution obtained with a sufficiently large grid size $N = 2048^2$. The relative L^2 errors maintain approximate accuracy of first-order in Table 3 due to the discontinuous of the derivative of the potential at $x = 0.5L$. Table 3 also shows the accuracy of the whole algorithm to compute the potential Φ compared to a reference potential computed with a sufficiently large grid size $N = 2048^2$, which verifies the convergence of our algorithm. Furthermore, Table 3 also shows the computational time of the algorithm to verify the linear scaling of SelInvHIF. Finally, the scaling results of the SelInvHIF algorithm are shown in Figure 6.

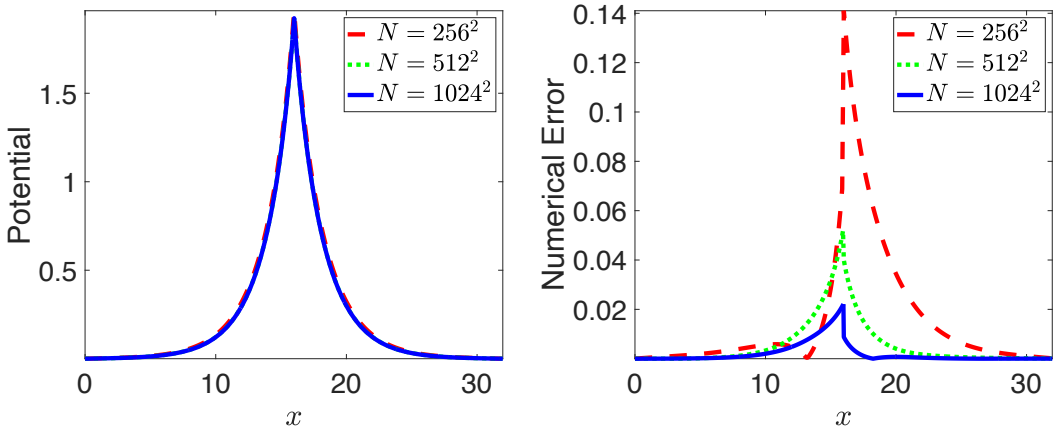

Figure 4: Numerical results about the charge density with a delta function. Left: potential distributions with different matrix size. Right: the absolute error between the numerical results and the reference solution with $N = 2048^2$.

Matrix size $\sqrt{N}$	Total time	SelInvHIF time	Absolute error	Relative error
256	$3.51E+1$	$3.34E+1$	$3.08E-2$	$6.25E-2$
512	$1.49E+2$	$1.41E+2$	$1.13E-2$	$2.30E-2$
1024	$6.23E+2$	$5.93E+2$	$4.54E-3$	$9.22E-3$
2048	$2.53E+3$	$2.42E+3$	-	-

Table 3: The CPU time, accuracy, and 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.

4.4 Example 4: The Charge Density is a Combined Delta Function

In the last example, we have two charged lines dividing a plane into three parts. The computational interval is $[0, L]$ with $L = 32$, where the region of $[0.4L, 0.6L]$ is inaccessible to ions. The fixed charge density is

$$\rho_f(x) = \delta(x - 0.4L) - \delta(x - 0.6L).$$

We solve the MPB equations using SelInvHIF. The left panel of Figure 5 visualizes the distribution of the convergent potential in this system with different matrix sizes $N = 256^2$, 512^2, and 1024^2. The right panel of Figure 5 displays the absolute L^2 error between the numerical results and the reference solution obtained with a sufficiently large grid size $N = 2048^2$. The relative L^2 errors maintain approximate accuracy of first-order in Table 4 due to the discontinuous of the derivative of the potential at $x = 0.4L$ and $x = 0.6L$. Table 4 also shows the accuracy of the whole algorithm to compute the potential Φ compared to a reference potential, which verifies the convergence of our algorithm. Furthermore, Table 4 also shows the computational time of the algorithm verify the linear scaling of SelInvHIF. Finally, the scaling results of the SelInvHIF algorithm are shown in Figure 6.

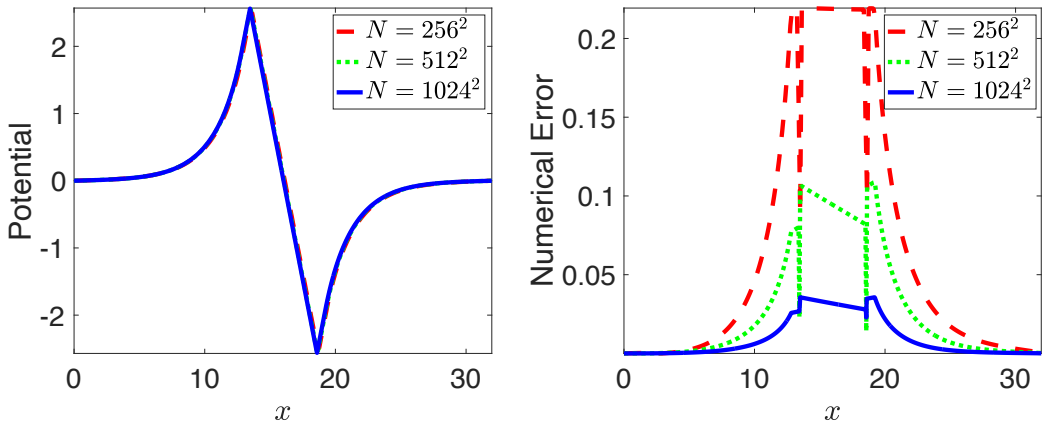

Figure 5: Numerical results about the charge density with a combined delta function. Left: potential distributions with different matrix size; Right: the absolute error between the numerical results and the reference solution with $N = 2048^2$.

Matrix size $\sqrt{N}$	Total time	SelInvHIF time	Absolute error	Relative error
256	$2.99E+1$	$2.90E+1$	$1.08E-1$	$1.22E-1$
512	$1.47E+2$	$1.42E+2$	$4.75E-1$	$5.34E-2$
1024	$6.06E+2$	$5.86E+2$	$1.59E-2$	$1.79E-2$
2048	$2.54E+3$	$2.45E+3$	-	-

Table 4: The CPU time, accuracy, and 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.

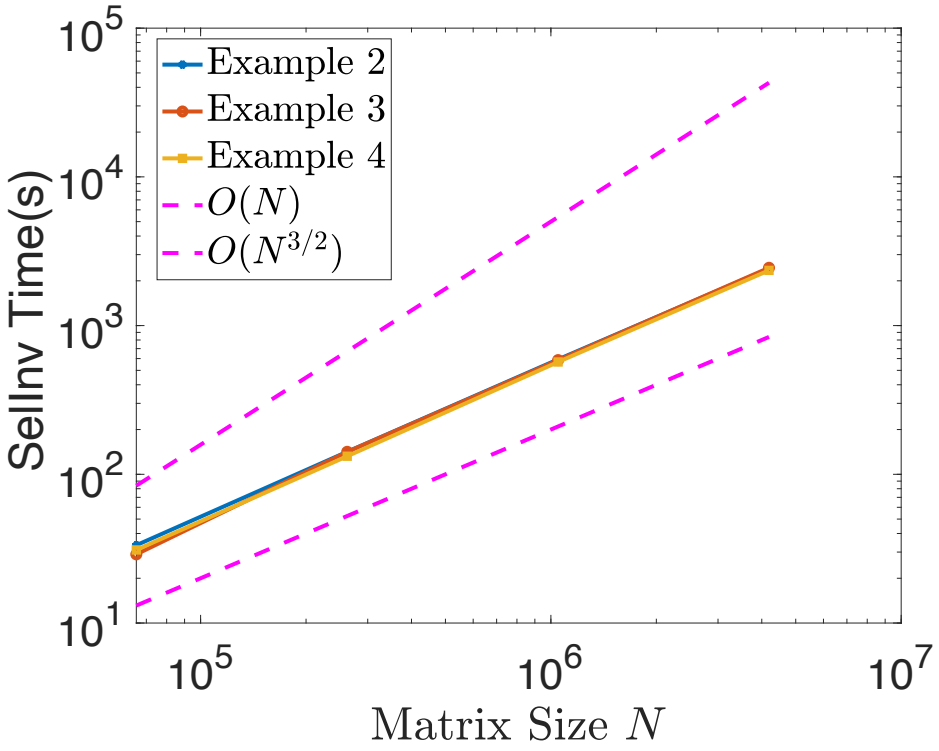

Figure 6: Scaling results for SelInvHIF time in solving MPB equations. The solid lines represent the computational time for one step SelInvHIF under the different charge distribution. The reference scalings (black dashed lines) of $O(N)$ and $O(N^{3/2})$.

5 Conclusions

A fast algorithm, SelInvHIF, is proposed to solve the MPB equations by combining the hierarchical interpolative factorization and the original selected inverse method. An $O(N)$ computational complexity in terms of the number of operations and memory is achieved to obtain the diagonal of the inverse of a sparse matrix discretized from an elliptic differential operator. We applied this algorithm to the two-dimensional MPB problems and attractive performance is obtained in terms of both accuracy and efficiency in solving the MPB equations. In the future, we will try to develop another fast algorithm with $O(N)$ complexity for three-dimensional problems based

on a similar construction.

Acknowledgment

Y. Tu and Z. Xu acknowledge the financial support from the National Natural Science Foundation of China (grant No. 12071288), Science and Technology Commission of Shanghai Municipality (grant No. 20JC1414100) and Strategic Priority Research Program of Chinese Academy of Sciences (grant No. XDA25010403). Q. Pang and H. Yang thank the support of the US National Science Foundation under award DMS-1945029.

References

[1] R. B. Schoch, J. Han, and P. Renaud. Transport phenomena in nanofluidics. *Rev. Mod. Phys.*, 80:839–883, 2008.

[2] H. Daiguji, P. Yang, and A. Majumdar. Ion transport in nanofluidic channels. *Nano Lett.*, 4(1):137–142, 2004.

[3] H. Boroudjerdi, Y.-W. Kim, A. Naji, R. R. Netz, X. Schlagberger, and A. Serr. Statics and dynamics of strongly charged soft matter. *Phys. Rep.*, 416:129–199, 2005.

[4] V. Liljeström, J. Seitsonen, and M. Kostiainen. Electrostatic self-assembly of soft matter nanoparticle cocrystals with tunable lattice parameters. *ACS Nano*, 9(11):11278–85, 2015.

[5] G. Gouy. Constitution of the electric charge at the surface of an electrolyte. *J. Phys.*, 9:457–468, 1910.

[6] D. L. Chapman. A contribution to the theory of electrocapillarity. *Phil. Mag.*, 25:475–481, 1913.

[7] M. Z. Bazant, K. Thornton, and A. Ajdari. Diffuse-charge dynamics in electrochemical systems. *Phys. Rev. E*, 70:021506, 2004.

[8] Z. Schuss, B. Nadler, and R. Eisenberg. Derivation of Poisson and Nernst-Planck equations in a bath and channel from a molecular model. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 64:036116, 2001.

[9] I. Borukhov, D. Andelman, and H. Orland. Steric effects in electrolytes: A modified Poisson-Boltzmann equation. *Phys. Rev. Lett.*, 79(3):435–438, 1998.

[10] M. Z. Bazant, B. D. Storey, and A. A. Kornyshev. Double layer in ionic liquids: overscreening versus crowding. *Phys. Rev. Lett.*, 106(4):046102, 2011.

[11] J.-L. Liu and R.S. Eisenberg. Molecular mean-field theory of ionic solutions: a Poisson-Nernst-Planck-Bikerman model. *arXiv:2004.10300*, 2020.

[12] Z. Xu, M. Ma, and P. Liu. Self-energy-modified Poisson-Nernst-Planck equations: WKB approximation and finite-difference approaches. *Phys. Rev. E*, 90(1):013307, 2014.

[13] H. Liu and Z. Wang. A free energy satisfying finite difference method for Poisson–Nernst–Planck equations. *J. Comput. Phys.*, 268(2):363–376, 2014.

[14] C. Liu, C. Wang, S. Wise, X. Yue, and S. Zhou. A positivity-preserving, energy stable and convergent numerical scheme for the Poisson-Nernst-Plancksystem. *arXiv:2009.08076*, 2020.

[15] R. R. Netz and H. Orland. Beyond Poisson-Boltzmann: Fluctuation effects and correlation functions. *The European Physical Journal E*, 1(2):203–214, 2000.

[16] R. Podgornik. Electrostatic correlation forces between surfaces with surface specific ionic interactions. *Journal of Chemical Physics*, 91:5840–5849, 1989.

[17] R. R. Netz and H. Orland. Variational charge renormalization in charged systems. *European Physical Journal E*, 11(3):301–311, 2003.

[18] P. Liu, X. Ji, and Z. Xu. Modified Poisson-Nernst-Planck model with accurate coulomb correlation in variable media. *SIAM J. Appl. Math.*, 78:226–245, 2018.

[19] M. Ma, Z. Xu, and L. Zhang. Modified Poisson-Nernst-Planck model with coulomb and hard-sphere correlations. *arXiv: 2002.07489*, 2020.

[20] B. Corry, S. Kuyucak, and S. H. Chung. Dielectric self-energy in Poisson-Boltzmann and Poisson-Nernst-Planck models of ion channels. *Biophysical Journal*, 84(6):3594–3606, 2003.

[21] Z.-G. Wang. Fluctuation in electrolyte solutions: The self energy. *Phys. Rev. E*, 81:021501, 2010.

[22] M. Ma and Z. Xu. Self-consistent field model for strong electrostatic correlations and inhomogeneous dielectric media. *J. Chem. Phys.*, 141(24):244903, 2014.

[23] L. Lin, J. Lu, L. Ying, R. Car, and W. E. 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.

[24] L. Lin, C. Yang, J. Lu, L. Ying, and W. E. 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.

[25] L. Lin, C. Yang, Juan C. Meza, J. Lu, L. Y, and W. E. SelInv—An algorithm for selected inversion of a sparse symmetric matrix. *ACM Trans. Math. Softw.*, 37(4):40:1–40:19, 2011.

[26] J. Xia, Y. Xi, S. Cauley, and V. Balakrishnan. Fast sparse selected inversion. *SIAM Journal on Matrix Analysis and Applications*, 36(3):1283–1314, 2015.

[27] K. L. Ho and L. Ying. Hierarchical interpolative factorization for elliptic operators: Differential equations. *Comm. Pure and Appl. Math.*, 69(8):1415–1451, 2015.

[28] A. Gillman and P. G. Martinsson. 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.

[29] A. Gillman and P. G. Martinsson. An $O(N)$ algorithm for constructing the solution operator to 2D elliptic boundary value problems in the absence of body loads. *Advances in Computational Mathematics*, 40(4):773–796, 2014.

[30] L. Grasedyck, R. Kriemann, and S. L. Borne. Domain-decomposition based $\mathcal{H}$-LU preconditioners. *Numerische Mathematik*, 112(4):565–600, 2009.

[31] P. G. Schmitz and L. Ying. A fast direct solver for elliptic problems on general meshes in 2D. *J. Comput. Phys.*, 231(4):1314–1338, 2012.

[32] J. Xia, S. Chandrasekaran, M. Gu, and X. Li. Superfast multifrontal method for large structured linear systems of equations. *SIAM J. Matrix Anal. Appl.*, 31(3):1382–1411, 2009.

[33] Z. Xu and A.C. Maggs. Solving fluctuation-enhanced Poisson–Boltzmann equations. *J. Comput. Phys.*, 36(3):310–322, 2014.

[34] H. Cheng, Z. Gimbutas, P. Martinsson, and V. Rokhlin. On the compression of low rank matrices. *SIAM J. Sci. Comput.*, 26(4):1389–1404, 2005.

[35] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.