

An Efficient MILU Preconditioning for Solving the 2D Poisson equation with Neumann boundary condition

Yesom Park, Jeongho Kim, Jinwook Jung, Euntaek Lee, Chohong Min

January 27, 2018

Abstract

MILU preconditioning is known to be the optimal one among all the ILU-type preconditionings in solving the Poisson equation with Dirichlet boundary condition. It is optimal in the sense that it reduces the condition number from $O(h^{-2})$, which can be obtained from other ILU-type preconditioners, to $O(h^{-1})$. However, with Neumann boundary condition, the conventional MILU cannot be used since it is not invertible, and some MILU preconditionings achieved the order $O(h^{-1})$ only in rectangular domains.

In this article, we consider a standard finite volume method for solving the Poisson equation with Neumann boundary condition in general smooth domains, and introduce a new and efficient MILU preconditioning for the method in two dimensional general smooth domains. Our new MILU preconditioning achieved the order $O(h^{-1})$ in all our empirical tests. In addition, in a circular domain with a fine grid, the CG method preconditioned with the proposed MILU runs about two times faster than the CG with ILU.

1 Introduction

The Poisson equation is of primal importance in many areas of science and engineering, such as incompressible fluid flows [12, 14], electro-magnetic waves [9, 2], and image processing [16, 21]. Its solution is known to exist in a well-posed problem, but can not be explicitly formulated except for simplistic cases. Instead of providing an explicit formulation of the solution, many successful numerical methods have been developed to approximate the solution. To list a few among them, finite difference methods [20, 7], finite element methods [4, 17], and spectral methods [19, 13] are devised to solve the Poisson equation numerically.

The inverse Laplacian is a self-adjoint and compact operator [18], and has countably many eigenvalues that are real and have a subsequence converging to zero. Thus, the Laplace operator has a spectrum that ranges from a negative value to $-\infty$. So if we obtain an associated matrix of Laplace operator from any of the numerical methods, it has a large condition number that grows to ∞ as the step size of the grid decreases to zero. This is because the matrix is a discrete analogue of an unbounded operator. The large condition number not only delays the convergence of an iterative method for solving the associated linear system, but also invokes round-off errors to the loss of significant digits [8].

A basic example is the standard 5-point finite difference method on rectangular domains with Dirichlet boundary condition. The condition number of the matrix is given as $4h^{-2}/\pi^2$ on a unit square domain, and h is the uniform step size of grid. To obtain an accurate approximation, h needs to be small and then the condition number $4h^{-2}/\pi^2$ becomes very large. For an efficient calculation of an accurate approximation, one therefore needs to utilize a preconditioning to reduce the condition number such as Jacobi, Symmetric Gauss-Seidel (SGS), Incomplete LU (ILU), and Modified Incomplete LU (MILU) preconditioners.

In a seminal paper [6], Gustafsson showed that only the MILU preconditioning results in the condition number of size $O(h^{-1})$, while all the other ILU-type preconditionings result in $O(h^{-2})$. The same result was proved in [11] to hold true for the finite difference method by Gibou et al.

applied to the cases of general smooth domains. Also, the same result was observed in [22] for the finite difference method by Shortley and Weller [20].

While MILU preconditioning is the optimal choice among all ILU-type ones for Dirichlet boundary condition, the choice of optimal preconditioner is unclear for Neumann boundary condition. The conventional MILU cannot be defined in the case of Neumann boundary condition. From several papers [23, 24, 25], we could find out that a lot of MILU-type preconditioners can be devised from the general definition of MILU. Moreover, it is mentioned in the papers [24, 25] that those can be applied to solving Neumann boundary problems numerically. In addition, actual applications to rectangular domain case were given, resulting in $O(h^{-1})$. However, to the best of our knowledge, there is no appropriate preconditioner to get $O(h^{-1})$ on general smooth domains. The unpreconditioned matrix has a large condition number of size $O(h^{-2})$. Any of Jacobi, SGS, and ILU preconditioners does keep the order -2 , but just reduces the constant in $C \cdot h^{-2} \simeq O(h^{-2})$.

The Purvis-Burkhalter method [3] is a standard finite volume method for solving the Poisson equation with Neumann boundary condition. The method plays a central role for solving the incompressible Navier-Stokes' equations in general smooth domains [5] and the interaction between fluid and solid [10]. Most of the computational cost for solving incompressible fluid flows is occupied by the Poisson solver, and it is very required to find a good preconditioner for the method.

In order to obtain a preconditioner that results in $O(h^{-1})$ on general smooth domains, even for Poisson equation with Neumann boundary condition, we take a practical guide in [15] that suggests a mixture of more than 97% MILU and less than 3% ILU, while increasing the ratio of MILU as the step size decreases to 0. The mixture provides well-defined preconditioner and turns out to perform well. Due to the large weight $\geq 97\%$ of MILU, the mixture seems to behave like MILU. One important role of small portion of ILU makes the MILU matrix invertible. In the setting, one naturally becomes curious about the optimal ratio of MILU and ILU resulting in $O(h^{-1})$, which is exactly the theme of this article.

This article begins with some basic analysis on MILU preconditioning applied to Purvis-Burkhalter method. The analysis shows why MILU preconditioning breaks down and explains why MILU-ILU mixture is a well-defined preconditioner. Then we report intensive numerical tests that searches the optimal rates for various step size h . From the numerical results, we introduce a conjecture that the MILU-ILU preconditioning of a certain ratio reduces the condition number from $O(h^{-2})$ to $O(h^{-1})$. We provide empirical evidences that support the conjecture.

2 Basic analysis for MILU

In this section, we provide the basic setting for the domain and review the MILU preconditioning. More precisely, let Ω be a open subset in \mathbb{R}^2 . Then, we discretize the Poisson equation with pure Neumann boundary condition :

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ \frac{\partial u}{\partial n} = g, & \text{on } \partial\Omega. \end{cases} \quad (2.1)$$

2.1 Domain setting

In this subsection, we use Purvis-Burkhalter method [3, 5] to define the discrete domain Ω and Heaviside function which will be used in the later analysis. Let $h\mathbb{Z}^2$ denote the uniform grid in \mathbb{R}^2 with grid size h . Then, for each node $(x_i, y_j) \in h\mathbb{Z}^2$, we define the finite volume $C_{i,j}$ and its edges $E_{i\pm\frac{1}{2},j}$ and $E_{i,j\pm\frac{1}{2}}$ as follows:

$$\begin{aligned} C_{ij} &:= [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}], \\ E_{i\pm\frac{1}{2},j} &:= x_{i\pm\frac{1}{2}} \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}], \\ E_{i,j\pm\frac{1}{2}} &:= [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times y_{j\pm\frac{1}{2}}. \end{aligned} \quad (2.2)$$

Based on these grid points and edges, we define node and edge sets.

Definition 2.1 (Node and edge sets). *By $\Omega^h := \{(x_i, y_j) \in h\mathbb{Z}^2 \mid C_{ij} \cap \Omega \neq \emptyset\}$, we denote the set of all nodes whose control volumes intersect the domain. Moreover, we define the edge sets $E_x^h := \{(x_{i+\frac{1}{2}}, y_j) \mid E_{i+\frac{1}{2},j} \cap \Omega \neq \emptyset\}$ and $E_y^h := \{(x_i, y_{j+\frac{1}{2}}) \mid E_{i,j+\frac{1}{2}} \cap \Omega \neq \emptyset\}$ in the same way.*

We also define the total edge set $E^h := E_x^h \cup E_y^h$. Finally, we denote the total number of node $|\Omega^h|$ by K .

Now, we define the Heaviside function which measures how large portion of each edge belongs to the domain.

Definition 2.2 (Heaviside function). *For each edge $E_{i+\frac{1}{2},j}$ and $E_{i,j+\frac{1}{2}}$, we define the Heaviside function $H_{i+\frac{1}{2},j}$ and $H_{i,j+\frac{1}{2}}$ by*

$$H_{i+\frac{1}{2},j} = \frac{\text{length}(E_{i+\frac{1}{2},j} \cap \Omega)}{\text{length}(E_{i+\frac{1}{2},j})}, \quad H_{i,j+\frac{1}{2}} = \frac{\text{length}(E_{i,j+\frac{1}{2}} \cap \Omega)}{\text{length}(E_{i,j+\frac{1}{2}})},$$

respectively.

With these definitions, now we can discretize the Poisson equation (2.1). Let $u_{i,j}^h$ be the average value of u over C_{ij} . Then by divergence theorem, we have

$$\begin{aligned} \iint_{C_{ij} \cap \Omega} -\Delta u dx dy &= \iint_{C_{ij} \cap \Omega} f dx dy \\ &= \int_{\partial(C_{ij} \cap \Omega)} -\frac{\partial u}{\partial n} ds = - \int_{\partial C_{ij} \cap \Omega} \frac{\partial u}{\partial n} ds - \int_{C_{ij} \cap \partial \Omega} g ds =: \mathcal{I}_1 + \mathcal{I}_2. \end{aligned}$$

We approximate $\partial u / \partial x$ and $\partial u / \partial y$ at ∂C_{ij} by using central differences of $u_{i,j}^h$ s. As a result, \mathcal{I}_1 can be approximated by using the discrete Laplace operator as follows:

$$\begin{aligned} (L^h u^h)_{ij} &= H_{i+\frac{1}{2},j}(u_{i+1,j}^h - u_{ij}^h) - H_{i-\frac{1}{2},j}(u_{i,j}^h - u_{i-1,j}^h) \\ &\quad + H_{i,j+\frac{1}{2}}(u_{i,j+1}^h - u_{ij}^h) - H_{i,j-\frac{1}{2}}(u_{i,j}^h - u_{i,j-1}^h), \end{aligned} \quad (2.3)$$

where $u^h = (u_{i,j}^h) \in \mathbb{R}^K$, $L^h \in \mathbb{R}^{K \times K}$ is the discrete Laplace operator and $K = |\Omega_h|$. Let $b_{i,j}^h$ be the approximated value of $\mathcal{I}_2 + \iint_{C_{ij} \cap \Omega} f dx dy$. Then to solve the Poisson equation (2.1) discretely, it is sufficient to solve the following linear equation:

$$L^h u^h = b^h, \quad L^h \in \mathbb{R}^{K \times K}, \quad u^h, b^h \in \mathbb{R}^K. \quad (2.4)$$

Here, to vectorize the values on the grid point, we use the lexicographical ordering [1] starting from the left-lower part of the domain, in the same way as in [11]. That is, align nodes in Ω_h as follows: $\mathbf{x}_1 := (x_{i_1}, y_{j_1}) \leq \mathbf{x}_2 := (x_{i_2}, y_{j_2}) \leq \dots \leq \mathbf{x}_K := (x_{i_K}, y_{j_K})$, where

$$(x_i, y_j) \leq (x_{i'}, y_{j'}) \iff (j < j') \vee (j = j' \wedge i \leq i').$$

For simplicity, we use the matrix A instead of L^h and omit all h -superscripts. Note that A is symmetric positive semi-definite and $\mathcal{N}(A) = \text{span}\{1\}$, where 1 denotes the vector in \mathbb{R}^K whose all components are 1. From now on, we focus on the linear equation of the form

$$Au = b, \quad A \in \mathbb{R}^{K \times K}, \quad u, b \in \mathbb{R}^K. \quad (2.5)$$

where $A = (a_{rs})$ satisfies the following:

$$a_{rs} = \begin{cases} H_{i_r+\frac{1}{2},j_r} + H_{i_r-\frac{1}{2},j_r} + H_{i_r,j_r+\frac{1}{2}} + H_{i_r,j_r-\frac{1}{2}} & \text{if } s = r, \\ -H_{i_r \pm \frac{1}{2},j_r} & \text{if } j_s = j_r \text{ and } i_s = i_r \pm 1, \\ -H_{i_r,j_r \pm \frac{1}{2}} & \text{if } i_s = i_r \text{ and } j_s = j_r \pm 1, \\ 0 & \text{otherwise.} \end{cases} \quad (2.6)$$

Here, a_{rs} shows how the node \mathbf{x}_r and \mathbf{x}_s are connected.

2.2 Analysis on the preconditioner

In this subsection, we review the classical definition of Incomplete LU decomposition (ILU) and Modified Incomplete LU decomposition (MILU) and their analysis. To solve (2.5) by using iterative method, the condition number $\kappa(A)$ has significant influence on the convergence rate. Therefore, instead of solving (2.5) directly, one can solve the modified equation

$$M^{-1}Au = M^{-1}b, \quad A, M \in \mathbb{R}^{K \times K}, \quad u, b \in \mathbb{R}^K. \quad (2.7)$$

with $\kappa(M^{-1}A) < \kappa(A)$. There are two conditions for preconditioner M to minimize the condition number $\kappa(M^{-1}A)$:

- M should be similar to A in some sense to make $M^{-1}A$ closer to I , and hence make the condition number small.
- The inverse operation of M should be easily calculated.

Note that the two extreme choices for M is $M = A$ and $M = I$. For the case $M = A$, M is same as A and hence $\kappa(M^{-1}A) = 1$. However, in this case, it is expensive to calculate the inverse operation of $M = A$. For the other extreme case, $M = I$, the inverse operation has no cost, but there is no effect on the condition number. Therefore, one has to choose the preconditioner M between these two extreme cases. Among many other preconditioners, we will focus on the two standard preconditioners, namely, ILU and conventional MILU.

2.2.1 Incomplete LU decomposition

The ILU preconditioner is defined by following procedure: Let $A = L + D + U$ where L, D and U denotes the (strictly) lower triangular, diagonal and (strictly) upper triangular matrix of A respectively. Then, define the preconditioner M as

$$M = (L + E)E^+(E + U),$$

for some appropriately chosen diagonal matrix E , and E^+ as the Penrose pseudoinverse of E . Note that M is the product of triangular matrices and diagonal matrix. Hence if E^{-1} exists, the inverse of M can be easily calculated and M can be used as a preconditioner. So let us temporarily assume E is invertible. Then we have

$$M = (L + E)E^{-1}(E + U) = LE^{-1}U + E + L + U.$$

To make M similar to A , ILU preconditioner M take E so that $LE^{-1}U + E$ has the same diagonal component with D . So the diagonal matrix E is defined by

$$(E + LE^{-1}U)_{ii} = D_{ii}, \quad i = 1, \dots, K.$$

For our case, let E_{i_k, j_k} be the diagonal element of E corresponding to the node point $\mathbf{x}_k = (x_{i_k}, y_{j_k})$ for $k = 1, \dots, K$. From above equation, we have recursive definition of ILU, which can be defined even when E is not invertible, as follows, :

$$E_{i_1, j_1} = a_{1,1}, \quad E_{i_k, j_k} = a_{k,k} - H_{i_k - \frac{1}{2}, j_k}^2 E_{i_k - 1, j_k}^+ - H_{i_k, j_k - \frac{1}{2}}^2 E_{i_k, j_k - 1}^+, \quad (2.8)$$

where a^+ is defined by

$$a^+ = \begin{cases} a^{-1} & \text{if } a \neq 0, \\ 0 & \text{if } a = 0, \end{cases}$$

and $E_{i_k - 1, j_k}, E_{i_k, j_k - 1}$ are diagonal elements of E corresponding to the node on the left/bottom of \mathbf{x}_k respectively.

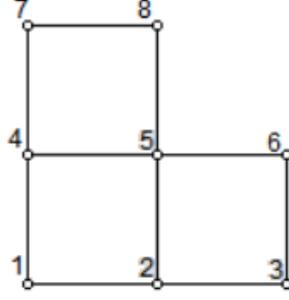


Figure 2.1: Example domain

2.2.2 Conventional Modified Incomplete LU decomposition

From the general definition of (Conventional) MILU preconditioner [23, 24, 25], there are a lot of choices for MILU preconditioners. Here we present the case which was used in the paper [11] as follows : Similar to ILU preconditioner, let $A = L + D + U$ where L, D and U denotes the (strictly) lower triangular, diagonal and (strictly) upper triangular matrix of A respectively. Then, define the preconditioner M as

$$M = (L + E)E^+(E + U),$$

for some diagonal matrix E . Again, we assume E is invertible temporarily. In this case, to make M similar to A , MILU preconditioner M takes E so that $LE^{-1}U + E$ has the same row sum with D . So the diagonal matrix E is defined by

$$\sum_{j=1}^K (E + LE^{-1}U)_{ij} = D_{ii}, \quad i = 1, \dots, K.$$

From the above equation, we have recursive definition of MILU, which can be defined even when E is not invertible, as follows:

$$\begin{aligned} E_{i_1, j_1} &= a_{1,1}, \\ E_{i_k, j_k} &= a_{k,k} - H_{i_k - \frac{1}{2}, j_k} E_{i_k - 1, j_k}^+ (H_{i_k - \frac{1}{2}, j_k} + l_{i_k - 1, j_k + \frac{1}{2}}) \\ &\quad - H_{i_k, j_k - \frac{1}{2}} E_{i_k, j_k - 1}^+ (H_{i_k, j_k - \frac{1}{2}} + l_{i_k + \frac{1}{2}, j_k - 1}). \end{aligned} \quad (2.9)$$

Here $l_{i_k - 1, j_k + \frac{1}{2}}$ and $l_{i_k + \frac{1}{2}, j_k - 1}$ are defined as

$$l_{i_k - 1, j_k + \frac{1}{2}} = \begin{cases} H_{i_p, j_p + \frac{1}{2}} & \text{if } \mathbf{x}_p := (x_{i_k} - h, y_{j_k}) \in \Omega_h, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$l_{i_k + \frac{1}{2}, j_k - 1} = \begin{cases} H_{i_q + \frac{1}{2}, j_q} & \text{if } \mathbf{x}_q := (x_{i_k}, y_{j_k} - h) \in \Omega_h, \\ 0, & \text{otherwise,} \end{cases}$$

and $E_{i_k - 1, j_k}, E_{i_k, j_k - 1}$ are diagonal elements of E corresponding to the node on the left/bottom of \mathbf{x}_k respectively. If the nodes are not in Ω_h , then they are ignored on the calculation.

Example 2.1. For the points in Figure 2.1, we give lexicographical ordering

Corollary 2.1. We say a node $(x_{i_k}, x_{j_k}) \in \Omega_h$ is at the (numerical) right-top corner if the right and upper edge of its control volume does not intersect with Ω , i.e.

$$H_{i_k + \frac{1}{2}, j_k} = H_{i_k, j_k + \frac{1}{2}} = 0.$$

Then we have

$$E_{i_k, j_k} = 0 \quad \text{iff} \quad (x_{i_k}, y_{j_k}) \text{ is at the numerical right-top corner of } \Omega_h.$$

Proof. It can be proved by using Proposition 2.1. □

Remark 2.1. To say a node is at the right-top corner in the real picture, there must be no nodes in Ω_h at its upper or right side. Hence the value of heavyside functions with respect to its upper or right side should be zero. However, note that the node at the numerical right-top corner may not be the node at the right-top corner in the real picture, i.e. for a node $(x_{i_k}, y_{j_k}) \in \Omega_h$ although we have $E_{i_k, j_k} = 0$, there might exists some node $(x^*, y^*) \in \Omega_h$ such that

$$(x^*, y^*) = (x_{i_k+1}, y_{j_k}) \quad \text{or} \quad (x^*, y^*) = (x_{i_k}, y_{j_k+1}).$$

This is possible when Ω is non-convex.

Corollary 2.2. The inverse of MILU does not exist.

Proof. Consider

$$M = (L + E)E^+(U + E).$$

Since there is at least one node at the right-top corner in Ω_h , and by corollary 2.1, at least one of diagonal entry of E is zero. □

2.3 Mixture of MILU and ILU preconditioner

As in Example 2.1., we can see that the number of zero diagonal entries of E -matrix can be larger than 1, and Corollary 2.1 tells us it varies by shape of domain. So MILU preconditioner for A presented in Section 2.2 can not be used since it is not invertible, whereas ILU preconditioner can be used since it is invertible. Hence our idea is to mix ILU and MILU preconditioner with certain ratio, which is invertible.

Definition 2.3 (Mixture of MILU and ILU). For $r \in (0, 1)$, we define MILU-ILU(r) be the matrix $M = (L + \tilde{E})\tilde{E}^{-1}(\tilde{E} + U)$ where \tilde{E} is defined by the formula

$$\begin{aligned} \tilde{E}_{i_1, j_1} &= a_{1,1}, \\ \tilde{E}_{i_k, j_k} &= a_{k,k} - H_{i_k - \frac{1}{2}, j_k} \tilde{E}_{i_k - 1, j_k}^+ (H_{i_k - \frac{1}{2}, j_k} + (1-r)l_{i_k - 1, j_k + \frac{1}{2}}) \\ &\quad - H_{i_k, j_k - \frac{1}{2}} \tilde{E}_{i_k, j_k - 1}^+ (H_{i_k, j_k - \frac{1}{2}} + (1-r)l_{i_k + \frac{1}{2}, j_k - 1}). \end{aligned} \quad (2.10)$$

Proposition 2.2. For any $r \in (0, 1)$, MILU-ILU(r) mixture is invertible.

Proof. It suffices to show that any diagonal element of \tilde{E} is non-zero. Let \tilde{E} of MILU-ILU be $\text{diag}(\tilde{E}_{i_1, j_1}, \dots, \tilde{E}_{i_K, j_K})$, and $l := \inf_k (i_k + j_k)$. We will prove the following by induction on n .

$$i_k + j_k = n \Rightarrow \tilde{E}_{i_k, j_k} > \max\{(1-r)H_{i_k + \frac{1}{2}, j_k} + H_{i_k, j_k + \frac{1}{2}}, H_{i_k + \frac{1}{2}, j_k} + (1-r)H_{i_k, j_k + \frac{1}{2}}\} \quad (2.11)$$

For $i_k + j_k = l$, this node is at the left-bottom, i.e. $(x_{i_k-1}, y_{j_k}), (x_{i_k}, y_{j_k-1})$ are not in Ω_h . Thus

$$\tilde{E}_{i_k-1, j_k}^+ = 0, \quad \tilde{E}_{i_k, j_k-1}^+ = 0,$$

and for such nodes,

$$a_{k,k} \geq H_{i_k + \frac{1}{2}, j_k} + H_{i_k, j_k + \frac{1}{2}}.$$

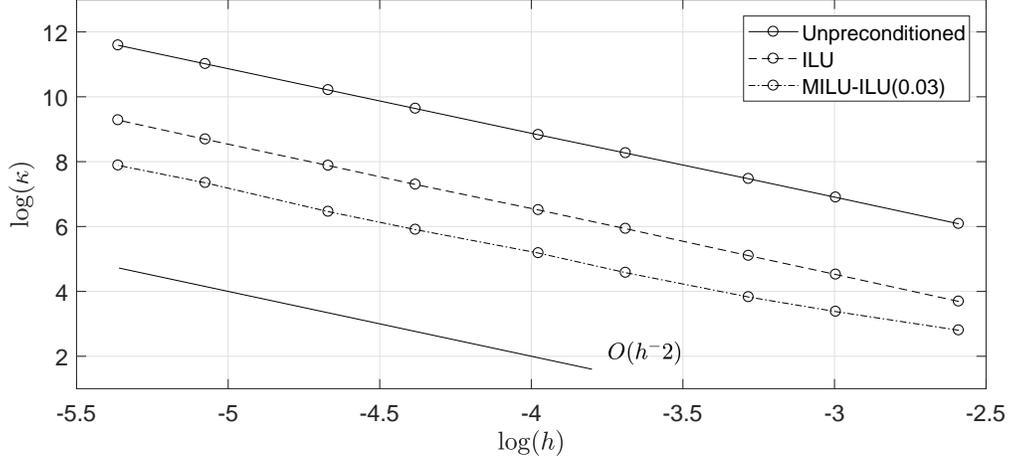


Figure 3.1: The performances of ILU and MILU-ILU (0.03) preconditioners in test #1: the graph of the condition number $\kappa = \kappa(M^{-1}A)$ with respect to the grid step size h in the log scales.

Thus

$$\begin{aligned} \tilde{E}_{i_k, j_k} &\geq H_{i_k + \frac{1}{2}, j_k} + H_{i_k, j_k + \frac{1}{2}} \\ &> \max\{(1-r)H_{i_k + \frac{1}{2}, j_k} + H_{i_k, j_k + \frac{1}{2}}, H_{i_k + \frac{1}{2}, j_k} + (1-r)H_{i_k, j_k + \frac{1}{2}}\}. \end{aligned}$$

So the initial step is proved. For induction step, assume that conclusion holds for $n = m$. Now choose any node satisfying $i_k + j_k = m + 1$. For the case that the node is at the left-bottom, it is the same as the initial step. If not,

$$\begin{aligned} \tilde{E}_{i_k, j_k} &= a_{k, k} - H_{i_k - \frac{1}{2}, j_k} \tilde{E}_{i_k - 1, j_k}^+ (H_{i_k - \frac{1}{2}, j_k} + (1-r)l_{i_k - 1, j_k + \frac{1}{2}}) \\ &\quad - H_{i_k, j_k - \frac{1}{2}} \tilde{E}_{i_k, j_k - 1}^+ (H_{i_k, j_k - \frac{1}{2}} + (1-r)l_{i_k + \frac{1}{2}, j_k - 1}) \\ &> a_{k, k} - H_{i_k - \frac{1}{2}, j_k} - H_{i_k, j_k - \frac{1}{2}} \\ &\geq \max\{(1-r)H_{i_k + \frac{1}{2}, j_k} + H_{i_k, j_k + \frac{1}{2}}, H_{i_k + \frac{1}{2}, j_k} + (1-r)H_{i_k, j_k + \frac{1}{2}}\}, \end{aligned} \tag{2.12}$$

which completes the proof. \square

3 Empirical tests on MILU-ILU preconditioning

In this section, we consider the mixture preconditioner of MILU and ILU. It was suggested in [15] to mix more of MILU from the default 97% as the grid step size decreases. This section is devoted to seeking the optimal ratio to mix and to finding out the performance of the MILU-ILU preconditioning with the optimal ratio.

3.1 Test #1 : MILU-ILU(0.03)

We first check the performance of the MILU-ILU preconditioning with $r = 0.03$. The Purvis-Burkhalter method is applied on the unit disc of center $(0, 0)$ with uniform grid step size h . The numerical results are reported in figure 3.1. In regard to the magnitude of condition number, MILU-ILU (0.03) achieves better results than ILU. The condition number grows as $O(h^{-2})$ in all cases: in regard to the growth order of condition number, MILU-ILU (0.03) and ILU are just as good as the unpreconditioned linear system.

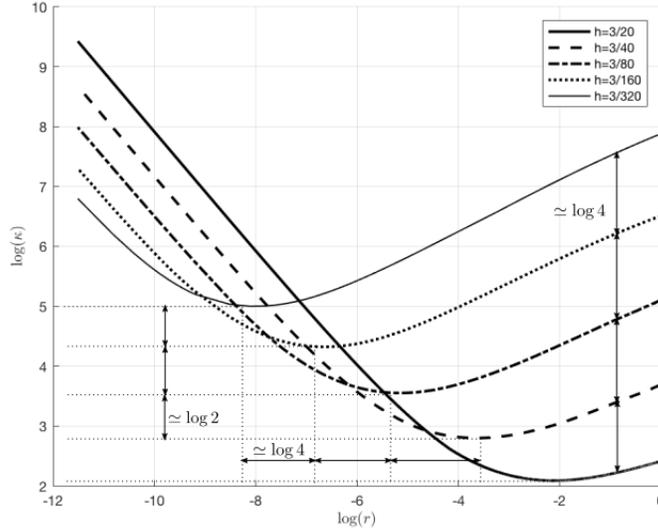


Figure 3.2: The graph of the condition number $\kappa = \kappa(M^{-1}A)$ with respect to the ratio of MILU-ILU for various h . It is remarkable that the graph just translates by the same amount $(-\log 4, \log 2)$ when h is reduced by half.

3.2 Test #2 : Optimal ratio to mix

It is known to be advantageous to take a smaller ratio r for a smaller grid step size h . A question in practice is how to give specific quantities to such qualitative statements. On the same setting as in test #1, a brute-force search is carried out to quantify the optimal ratio r with respect to h .

Figure 3.2 depicts the graph of the condition number κ with respect to the ratio r of MILU-ILU for each step size h . It is remarkable that the graph in the log scales just translates to the left by $\log 4$ and upward by $\log 2$ when h is reduced by half. In addition, in the figure, note that κ in the log scale increases by $\log 4$ when r is fixed, greater than 0.01, and h is reduced by half.

Hence the results in test #1 can be explained by the above paragraph. When the ratio $r = 3\%$ is fixed and h in the log scale is reduced by $\log 2$, then κ in the log scale increases by $\log 4$. We can deduce that $\log \kappa + 2 \log h$ is constant, and that κ grows as $O(h^{-2})$ as h becomes smaller.

Now, let us utilize the translation property to decide the ratio. When h is reduced by half (or $\log h \leftarrow \log h - \log 2$), let us take the ratio r to be a quarter of it ($\log r \leftarrow \log r - \log 4$) then $\log \kappa$ will increase by $\log 2$ ($\log \kappa \leftarrow \log \kappa + \log 2$) according to the translation property. This observation leads to the following conjecture that if $C = r \cdot h^{-2}$ is a fixed constant independent of h , then $\log \kappa + \log h$ is kept constant approximately, and κ grows as $O(h^{-1})$.

Conjecture : Let A be the matrix associated with the Purvis-Burkhalter method [3, 5] for solving the Poisson equation with Neumann boundary condition in a domain Ω , and let M be the MILU-ILU preconditioner with mixing ratio r . When $r = C \cdot h^2$ for some moderate constant $C > 0$ independent of h , then we have

$$\kappa(M^{-1}A) = O(h^{-1}),$$

for any smooth domain $\Omega \subset \mathbb{R}^2$. In practice, we may set $C = 1$.

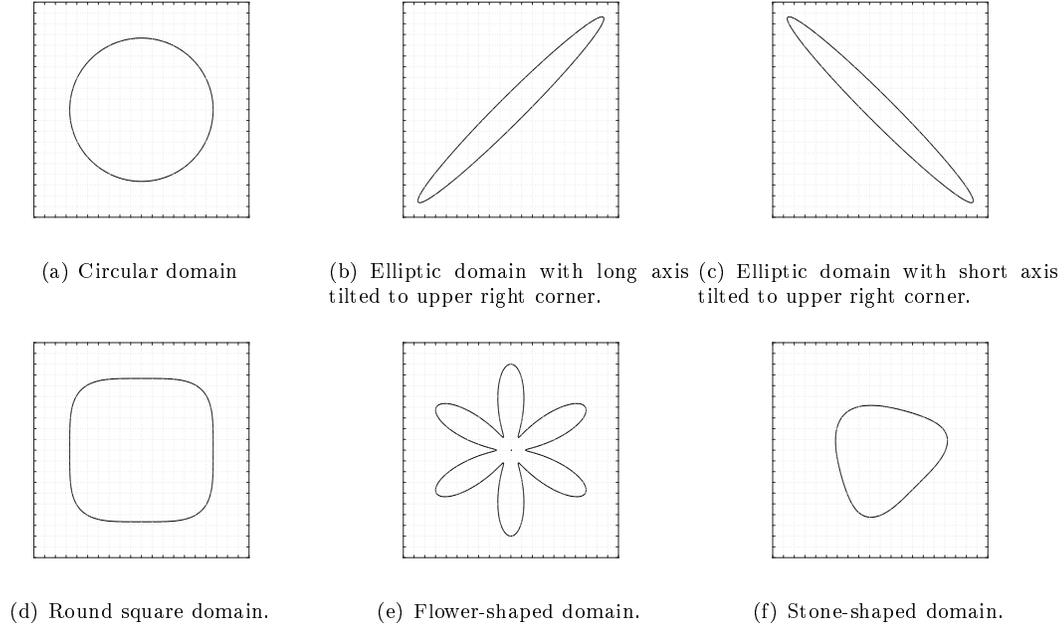


Figure 4.1: Various test domain with $N = 20$.

4 Numerical support of the conjecture

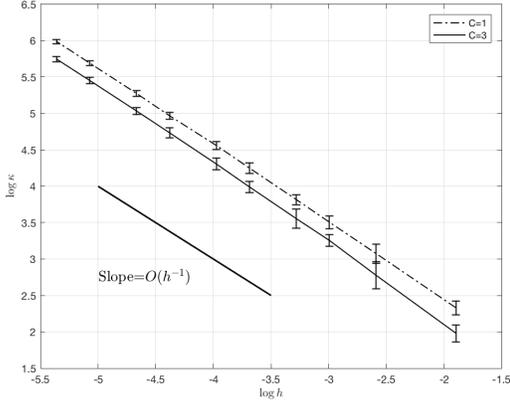
In this section, we provide pieces of numerical evidences to support the conjecture. Numerical tests are conducted on the various domains depicted in Figure 4.1.

Figure 4.2 shows how the condition number changes as h gets smaller in every domain depicted in Figure 4.1. Here, $r = h^2$ and $r = 3 \cdot h^2$ are used for the ratio r . In each domain Ω and constant $C = r/h^2$, the numerical results in Figure 4.2 indicate that the condition number grows as $O(h^{-1})$, which follows the conjecture. The conjecture was based upon the empirical observations on the disc. It states that the condition number $\kappa = \kappa(M^{-1}A)$ is of size $O(h^{-1})$ for any domain Ω and any constant $C = r/h^2 > 0$ independent of h , when M is the combination of MILU and ILU at the ratio of $1 - r$ to r .

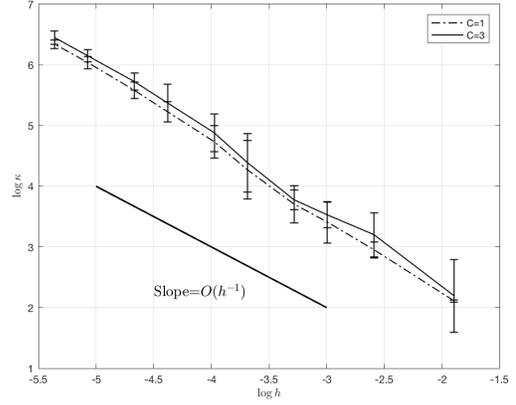
More specifically, we checked the plausibility of the conjecture for six different domains that are depicted in Figure 4.1 and two different values of $C = 1, 3$, making 12 sets of combination (Ω, C) . For each set (Ω, C) and each h , we perturb the domain by a random vector uniformly distributed in $(-h, h) \times (-h, h)$, and formed matrices A and M . Then, the maximum eigenvalue of $M^{-1}A$ is calculated by the power iteration, and the minimum eigenvalue by the inverse iteration, and then the condition number is taken to be their ratio. Due to the singularity of matrix A , the inverse iteration is run on the orthogonal complement, 1^\perp .

Figure 4.2 shows the graph of κ with respect to h in the log scales for each domain (a)-(f) in Figure 4.1 and for each constant $C = 1$ and $C = 3$. In each graph, κ is observed to behave as h^{-1} as step size h decreases, regardless of choice of the constant C . This supports the conjecture.

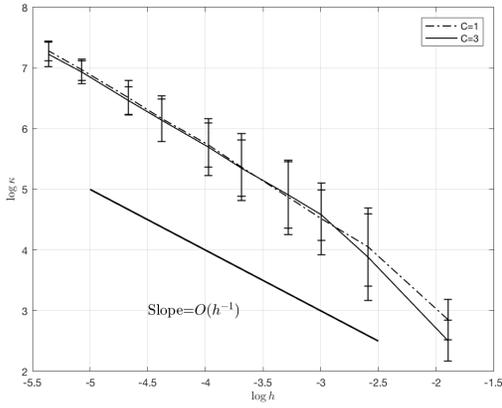
Now, we present an example that shows the practical importance of the conjecture. While the conventional ILU and MILU-ILU(3%) generates the condition number of size $O(h^{-2})$, the conjecture states that MILU-ILU(Ch^2) generates that of size $O(h^{-1})$. When h is small enough, $h^{-2} \gg h^{-1}$ and MILU-ILU(h^2) is expected to outperform the others by a large margin. One measure of the excellence is the number of iterations until convergence of the Preconditioned Conjugate Gradient. This number of iterations is proportional to the condition number and it is directly related to the computation time. We take an example with $h = 0.005$ and $\Omega = \{(x, y) \mid x^2 + y^2 < 1\}$. Figure 4.3 plots the residual norm with respect to iteration number for each of unpreconditioned, ILU, MILU-ILU(3%), and MILU-ILU(h^2). The computational time of MILU-



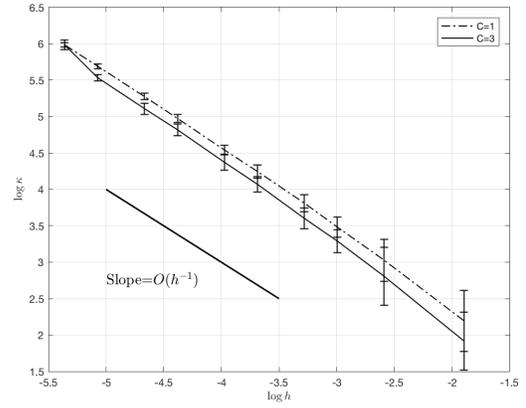
(a) Circular domain



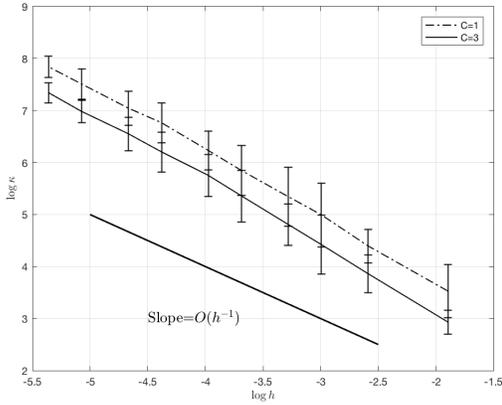
(b) Elliptic domain with long axis tilted to upper right corner



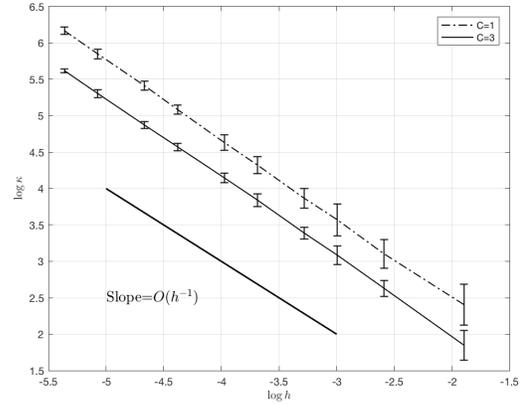
(c) Elliptic domain with short axis tilted to upper right corner



(d) Round square domain



(e) Flower-shaped domain



(f) Stone-shaped domain

Figure 4.2: The graph of the condition number κ with respect to the grid step size h for each domain (a)-(f) and each MILU-ILU ratio $r = h^2$ or $r = 3 \cdot h^2$. We conducted the same numerical test 20 times by translating the same domains by a vector $(e_{1,i}, e_{2,i}), i = 1, \dots, N$ where $e_{1,i}$ and $e_{2,i}$ are uniformly distributed random variables in $(-h, h)$. The graphs represent the average value of κ together with error bar. The rate $O(h^{-1})$ was observed not only from the average values but also from the error bar, which rules out the possibility of lucky observations and fortunate interfaces. The results support our conjecture.

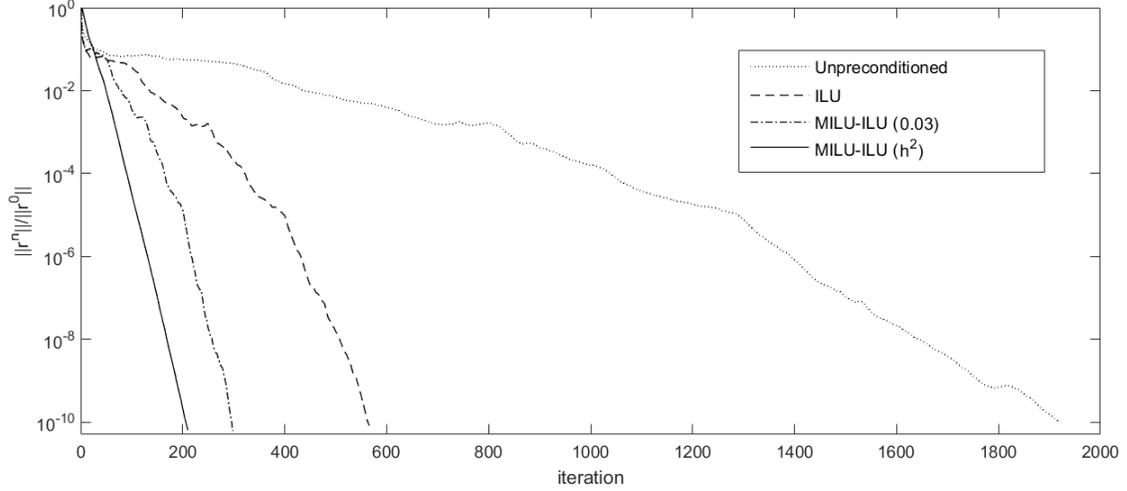


Figure 4.3: The graph for the residual norm with respect to the iteration of Preconditioned Conjugate Gradient on the linear system with $\Omega = \{(x, y) \mid x^2 + y^2 < 1\}$. In regard to the iteration number for convergence, the computational time of MILU-ILU(h^2) is merely about 67% and 51% of the computational times of MILU-ILU(3%) and ILU, respectively.

ILU(h^2) is merely about 67% of the computational time of MILU-ILU(3%), and about 51% of that of ILU.

Remark for the lexicographical ordering.

Although the tested domains are the same geometric domains, they show different performances. However, this difference is due to a lexicographical ordering of the cells, introduced in Section 2.1. More precisely, the reason that classical MILU preconditioning cannot be applied for pure Neumann boundary condition case is that the diagonal matrix E in the MILU preconditioner has a singularities (i.e., $E_{i,j} = 0$). On the other hand, Proposition 2.1 implies that the original MILU preconditioner has a singularities at the cells where both $H_{i+\frac{1}{2},j}$ and $H_{i,j+\frac{1}{2}}$ are 0. Those cells exactly correspond to the right-top corner cells of the domain. Since the MILU-ILU(r) preconditioning was intended to remedy those singularities, it is natural that the number of cells at the right-top corner impacts on the performance of MILU-ILU(r) preconditioning. Therefore, the bad performance in Figure 4.2 (c) compared to 4.2 (b) is probably due to the fact that those domains have the different number of cells at the right-top corner, and corresponding singularities. We note that there are four choices of lexicographical ordering for a 2D domain (eight choices for a 3D domain), and the performance of MILU-ILU(r) preconditioning can be changed according to this choice. For example, if we order the cells using the lexicographical ordering starting from the left-top corner, then the results in Figure 4.2 (b) and (c) will be reversed. When the domain is given and the direction of alignment of the domain is clear, it seems that the lexicographical ordering along the direction of the domain shows the best performance among other lexicographical orderings. When the direction of alignment of the domain is not clear, we have to find a way to detect the direction of alignment of domain, or to make the preconditioner ordering-independent.

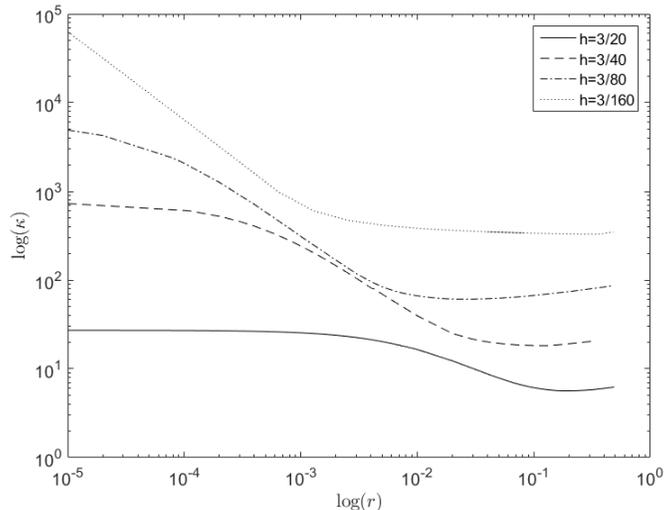


Figure 4.4: The graph of the condition number $\kappa = \kappa(M^{-1}A)$ with respect to the ratio of MILU-ILU for various h in the 3-dimensional domain. The graph is quite different from the two dimensional cases in figure 3.2. Therefore, the conjecture seems not valid for three dimensional domains.

Remarks for the three dimensional case.

We performed the numerical test on the unit sphere of center $(0, 0, 0)$ with uniform grid step size, to see whether numerical results still follow the conjecture in the case of three dimension. Figure 4.4 depicts the graph of the condition number κ with respect to the MILU-ILU with ratio r for each step size. As shown in Figure 4.4, the behavior in the three dimensional domain is quite different from the two dimensional case. Unlike the behavior for two dimensional domains, the graph in log scales is not convex and shifted in a more complex way than the two dimensional case. The graph translates to the left and upward when h decreases by half, but the differences between each graph are not uniform. Also, the position of the extreme values are irregular. However, it is hopeful that the tendency for the graph to be shifted as step size decreases, is similar to that of the two dimensional case. Moreover, when we track the optimal ratio of MILU-ILU in Figure 4.4, it still gives condition number of order $O(h^{-1})$ to some extent. Therefore, it seems possible to find the optimal ratio of MILU and ILU which gives condition number of order $O(h^{-1})$, even for the three dimensional case. Since we only conducted the numerical test in two dimensional domains, we cannot fully understand this result in this paper at this point and theoretically rigorous analysis is needed. After more careful analysis on two dimensional cases, we will be able to deal with the three dimensional cases, which is more complex.

5 Conclusion

In this paper, we presented the new mixing method between ILU and MILU to solve the Poisson equation with Neumann boundary condition. We varied the ratio of mixing ILU and MILU to find the optimal ratio, which gives the smallest condition number. We found that the optimal ratio r of ILU preconditioning should change as step size h changes. Moreover, we found that the exact optimal ratio is $r = C \cdot h^2$, where C can be any choice, as long as it does not depend on h . For this optimal ratio, the condition number behaves as h^{-1} , which is significantly enhanced compared to the previous performance, $O(h^{-2})$. There are several ways to develop this result. First, since we only conducted the numerical tests, the theoretical proof of the conjecture is still an open problem. Moreover, we performed the numerical test on the unit sphere of center $(0, 0, 0)$ with uniform grid step size to check to check the plausibility of the conjecture for three dimensional domains, and we

find that the behavior in the three dimensional domain is quite different from the two dimensional case. Since we only conducted the numerical tests in two dimensional domains and did not analyze them theoretically, it is not easy to understand the results for three dimensional cases until now. Furthermore, as discussed in Remark 4.1, when the direction of alignment of the domain is not clear, we need to detect the direction of alignment of the domain before we order the cells or make the preconditioner more versatile and ordering-independent. These interesting questions are left for the future works.

Acknowledgement

The research of C. Min and Y. Park was supported by Basic Science Research Program through the National Re-search Foundation of Korea (NRF) funded by the Ministry of Education (2017-006688) and the research of J. Kim, J. Jung and E. Lee was supported by the Samsung Science and Technology Foundation under Project (Number SSTF-BA1301-03).

References

- [1] I. S. Duff, G. A. Meurant. *The effect of ordering on preconditioned conjugate gradients*. *BIT*, 29(4):635–657, 1989.
- [2] M. Bonnet. Boundary integral equation methods for solids and fluids. *Meccanica*, 34(4):301–302, 1999.
- [3] J. W. Purvis, J. E. Burkhalter. Prediction of critical mach number for store configurations. *AIAA Journal*, 17(11):1170–1177, 1979.
- [4] P. G. Ciarlet. *The finite element method for elliptic problems*. SIAM, 2002.
- [5] Y. T. Ng, C. Min, F. Gibou. An efficient fluid–solid coupling algorithm for single-phase flows. *J. Comput. Phys.*, 228(23):8807–8829, 2009.
- [6] I. Gustafsson. A class of first order factorization methods. *BIT*, 18:142–156, 1978.
- [7] F. Gibou, R. Fedkiw, L.-T. Cheng, M. Kang. A second–order–accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys.*, 176:205–227, 2002.
- [8] E. Cheney, D. Kincaid. *Numerical mathematics and computing*. Nelson Education, 2012. page 97.
- [9] K. J. Marfurt. Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations. *Geophysics*, 49(5):533–549, 1984.
- [10] F. Gibou, C. Min. Efficient symmetric positive definite second-order accurate monolithic solver for fluid/solid interactions. *J. Comput. Phys.*, 231:3245–3263, 2012.
- [11] G. Yoon, C. Min. Analyses on the finite difference method by Gibou et al. for poisson equation. *J. Comput. Phys.*, 280:184–194, 2015.
- [12] D. Brown, R. Cortez, M. Minion. Accurate projection methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 168:464–499, 2001.
- [13] D. Gottlieb, S. Orszag. *Numerical analysis of spectral methods: theory and applications*. SIAM, 1977.
- [14] M. Sussman, P. Smereka, S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.
- [15] B. Robert. *Fluid simulation for computer graphics*. CRC Press, 2015.
- [16] E. Fatemi, S. Osher, L. I. Rudin. Nonlinear total variation based noise removal algorithms. *Physica D.*, 60(1-4):259–268, 1992.
- [17] B. Cockburn, George E, C. Shu. The development of discontinuous galerkin methods. In *Discontinuous Galerkin Methods*, pages 3–50. Springer, 2000.
- [18] M. S. Birman, M. Z. Solomjak. *Spectral theory of self-adjoint operators in Hilbert space*, volume 5. Springer Science & Business Media, 2012.

- [19] H. P. Pfeiffer, L. E. Kidder, M. A. Scheel, S. A. Teukolsky. A multidomain spectral method for solving elliptic equations. *Comput. Phys. Commun.*, 152(3):253–273, 2003.
- [20] G. H. Shortley, R. Weller. The numerical solution of Laplace's equation. *J. Appl. Phys.*, 9(5):334–348, 1938.
- [21] S. Osher, M. Burger, D. Goldfarb, J. Xu, W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale. Model. Sim.*, 4(2):460–489, 2005.
- [22] C. Min, G. Yoon. Comparison of eigenvalue ratios in artificial boundary perturbation and jacobi preconditioning for solving poisson equation. *J. Comput. Phys*, 349:1–10, 2017.
- [23] R. Beauwens. Modified incomplete factorization strategies. In *Preconditioned Conjugate Gradient Methods, Lecture Notes in Mathematics*, Springer:Berlin, 1457:1–16, 1990.
- [24] I. Gustafsson. A class of preconditioned conjugate gradient methods applied to the finite element equations. In *Preconditioned Conjugate Gradient Methods, Lecture Notes in Mathematics*, Springer:Berlin, 1457:44–57, 1990.
- [25] Y. Notay. Solving positive (semi) definite linear systems by preconditioned iterative methods. In *Preconditioned Conjugate Gradient Methods, Lecture Notes in Mathematics*, Springer:Berlin, 1457:105–125, 1990.